



Whitemarsh
Information Systems Corporation

Work Breakdown Structure

Whitemarsh Information Systems Corporation
2008 Althea Lane
Bowie, Maryland 20716
Tele: 301-249-1142
Email: Wiscorp@wiscorp.com
Web: www.wiscorp.com



Whitemarsh on Database

Work Breakdown Structure

November 2005



Whitemarsh
Information Systems Corporation

2008 Althea Lane
Bowie, Maryland 20716
(301) 249-1142

Work Breakdown Structure

Designations used by companies to distinguish their products are often claimed as trademarks. In all instances where Whitemarsh Press is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

©2005 by Whitemarsh Information Systems Corporation
All rights reserved

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If legal advice or other expert assistance is required, the services of competent professional person should be sought. FROM A DECLARATION OF PRINCIPLES JOINTLY ADOPTED BY A COMMITTEE OF THE AMERICAN BAR ASSOCIATION AND A COMMITTEE OF PUBLISHERS.

Reproduction or translation of any part of this work beyond that permitted by section 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright holder is unlawful. Requests for permission or further information should be addressed to Whitemarsh Press.

ISBN n-nnn-nnnnn-n

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1



Copyright 2005, Whitemarsh Information Systems Corporation
Proprietary Data, All Rights Reserved

Table of Contents

1	Preliminary Analysis Phase	1
1.01	Form project team, plan phase, & estimate project	1
1.01.01	Select manager for project phase	1
1.01.02	Identify administrative, clerical, and computer supports	1
1.01.03	Select project staff for phase	1
1.01.04	Secure commitments for staff availability	1
1.01.05	Create estimate for project	1
1.01.06	Create phase and task plans and schedules	1
1.01.07	Organize phase review committee	1
1.01.08	Conduct methods training as required	1
1.01.09	Set up phase documentation files	2
1.01.10	Create project plan and schedule estimates	2
1.01.11	Establish metadata repository	2
1.01.12	Interrogate and maintain metadata repository during phase	2
1.01.13	Establish effective computing environment, libraries, etc., for resource usage tracking, for storage of data, programs, and run streams for development and unit test, for system test and integration, and for production	2
1.01.14	Develop and/or use database standards for phase	2
1.01.15	Identify and allocate management and clerical overhead to phase	2
1.02	Create initial database project requirements description	2
1.02.01	Request and review documentation of prior related database efforts	2
1.02.02	Obtain initial scoping information for project	2
1.02.02.01	Conduct broad interviews with user management to determine business database objectives for database project	2
1.02.02.02	Identify, name and briefly describe the involved missions	2
1.02.02.03	Identify, name and briefly describe the business resources that are the foundation of the database objects	2
1.02.02.04	Identify, name and briefly describe the existing business information systems ..	2
1.02.02.05	Identify, name and briefly describe the critical business events that currently involve automation assists	3
1.02.02.06	Identify, name and briefly describe the business functions that must be performed	3
1.02.02.07	Identify, name and briefly describe the organizations involved in the enterprise database effort	3
1.02.03.01	Identify major suppliers & uses of database data	3
1.02.03.02	Identify, create, & analyze prototypical data and reports	3
1.02.03.03	Conduct preliminary data transfer and data conversion interview	3
1.02.04	Formulate list of business database objectives that are to be achieved	3
1.02.05	Formulate reasonable list of mechanisms to evaluate project success	3



Work Breakdown Structure

1.02.06	Reach consensus on database objectives and evaluation mechanisms	3
1.03	Create mission description model	3
1.04	Create business information systems plan	7
1.04.01	Create Ross resource life cycles	7
1.04.02	Create relationships between resource life cycle nodes and metabase component instances	8
1.04.03	Propose business information systems projects	8
1.04.04	Examine the state of automation for each associated business information system	8
1.04.04.01	Determine improvements required for logical database	8
1.04.04.02	Determine improvements required for physical database	8
1.04.04.03	Determine improvements required for interrogation	9
1.04.04.04	Determine improvements required for system control	9
1.04.04.05	Create system upgrade assessment report for information system	9
1.04.04.06	Estimate the resources required to upgrade the business information system . . .	11
1.04.04.07	Prepare consolidated business information system upgrade report for each business information systems associated with resource life cycle node	11
1.04.04.08	Create consolidated high-level estimate for all business information systems associated with each resource life cycle node	11
1.04.04.09	Create overall business information system upgrade presentation	12
1.04.04.10	Conduct phase review	12
1.04.05	Create and load resource life cycle node pert chart	12
1.04.06	Enter the resource life cycle node chart into a project management package as a PERT chart	12
1.04.07	Enter the PERT chart of the upgrade business information systems as a subproject for appropriate resource life cycle nodes	12
1.04.08	Resource load the resource life cycle subproject nodes	12
1.04.09	Project management scheduling	12
1.04.10	Publish and review the Information systems plan	13
1.05	Create preliminary analysis phase report	13
1.05.01	Select project from information systems plan to accomplish	13
1.05.02	Prepare database system scope narrative	13
1.05.03	Create plan & format for presenting mission descriptions, database domains, and all graphics	13
1.05.04	Create proposal for conceptual specification phase	13
1.05.05	Prepare final report for preliminary analysis phase	14
1.06	Create preliminary analysis presentation	14
1.07	Conduct phase review	14
2	Conceptual Specification Phase	15
2.01	Form project team, plan phase, and revise project plan	15
2.01.01	Select manager for project phase	15



2.01.02	Identify administrative, clerical, and computer supports	15
2.01.03	Select project staff for phase	15
2.01.04	Secure commitments for staff availability	15
2.01.05	Create phase and task plans and schedules	15
2.01.06	Organize phase review committee	15
2.01.07	Conduct methods training as required	15
2.01.08	Set up phase documentation files	15
2.01.09	Revise project plan and schedule estimates	15
2.01.10	Interrogate and maintain metadata repository	15
2.01.11	Develop and/or use database standards for phase	15
2.01.12	Review & revise preliminary analysis phase as necessary	15
2.02	Create logical database of business requirements	15
2.02.01	Perform detailed data analysis	15
2.02.02	Perform detailed data analysis for data transfer and conversion	16
2.02.03	Update project mission descriptions, mission description diagram, database domains & construct database domain diagrams	16
2.02.04	Define data integrity model	17
2.02.04.01	Finalize database domain diagram	17
2.02.04.02	Finalize database objects and database object diagram	17
2.02.04.03	Create generalized/abstracted/leveled database domain diagrams	17
2.02.04.04	Narrow database diagram to project scope	18
2.02.04.05	Update business and technical glossary	18
2.02.04.06	Specify data elements	18
2.02.04.07	Create database object table component	19
2.02.04.08	Specify database object table process component	21
2.02.04.09	Create database object information system component	24
2.02.04.10	Specify database object state component	25
2.02.05	Create or modify specified data model as appropriate	26
2.02.06.01	Perform data requirements analysis	26
2.02.06.02	Define retail data warehouse data integrity model	26
2.02.06.03	Identify data integrity requirements for data transfer and data conversion	26
2.02.06.04	Perform analysis to support data transfer and conversion from source databases to retail data warehouse databases	27
2.02.06.05	Verify retail data warehouse data integrity model	27
2.02.06.06	Create retail data warehouse logical database specification report	28
2.02.06.07	Create retail data warehouse logical database presentation	28
2.02.06.08	Conduct subphase review	28
2.02.06	Create logical database specification report	28
2.02.08	Create logical database presentation	29
2.02.09	Conduct subphase review	29
2.03	Specify physical database of business requirements	29



Work Breakdown Structure

2.03.01	Specify business information systems model	29
2.03.01.01	Identify business information systems appropriate for subordinate mission description	29
2.03.01.02	Create online data update system specification	29
2.03.01.03	Create batch data update system specification	31
2.03.01.04	Create data loading subsystem specification	33
2.03.01.05	Create specification for each data transfer and data conversion subsystem	35
2.03.01.06	Create enterprise-wide data collection system specification	35
2.03.02	Create business event model	38
2.03.03	Create detailed business function model	38
2.03.04	Create Organization Model	38
2.03.05	Estimate database size	39
2.03.06	Evaluate requirements for backup	39
2.03.07	Specify database integrity subsystem	39
2.03.08	Consolidate logical database changes to accommodate physical database	40
2.03.09	Prepare physical database requirements specification	40
2.03.10	Create physical database presentation	40
2.03.11	Conduct subphase review	40
2.04	Analyze interrogation requirements	40
2.04.01	Identify, name and then create information system process diagram for each specialized report subsystem	40
2.04.02	Analyze standard report requirements	41
2.04.03	Analyze ad-hoc report requirements	41
2.04.04	Prepare interrogation requirements report	41
2.04.05	Analyze data transfer and data conversion reporting requirements	41
2.04.06	Create interrogation presentation	42
2.04.07	Conduct subphase review	42
2.05	Specify system control requirements	42
2.05.01	Specify audit trail needs	42
2.05.02	Specify backup & recovery needs	42
2.05.03	Specify concurrent operations requirements	43
2.05.04	Specify security & privacy requirements	43
2.05.05	Specify reorganization requirements	43
2.05.06	Specify multiple database requirements	43
2.05.07	Prepare system control requirement report	44
2.05.08	Create system control presentation	44
2.05.09	Conduct subphase review	44
2.06	Validate cross product of data integrity and data transformation models	44
2.07	Create cross-product validation presentation	45
2.08	Conduct subphase review	45
2.09	Validate through prototyping	45



2.09.01	Select appropriate DBMS	45
2.09.02	Create logical database	45
2.09.03	Create test data	46
2.09.04	Create physical database	46
2.09.05	Create interrogation scenarios	46
2.09.06	Present design iteration	47
2.09.07	Create design iteration changes	47
2.09.08	Update database specification design	47
2.09.09	Walkthrough all components of the data transfer and data conversion	47
2.10	Consolidate logical database changes required to accommodate physical database interrogation, and system control additional requirements, and revise as necessary ...	48
2.11	Create conceptual specification phase report	48
2.11.01	Create introduction & scope	48
2.11.02	Create logical database requirements section	48
2.11.03	Create physical database requirements section	48
2.11.04	Conduct final review of functional completeness & acceptable interaction of data integrity model and data transformation model.	48
2.11.05	Create interrogation requirements section	48
2.11.06	Prepare system control requirements section	49
2.11.07	Prepare conceptual specification phase report	49
2.12	Create conceptual specification phase presentation	49
2.13	Conduct phase review	49
3	Binding Phase	51
3.01	Form project team, plan phase, and revise project plan	51
3.01.01	Select manager for project phase	51
3.01.02	Identify administrative, clerical, and computer supports	51
3.01.03	Select project staff for phase	51
3.01.04	Secure commitments for staff availability	51
3.01.05	Create phase plans and schedules	51
3.01.06	Organize phase review committee	51
3.01.07	Conduct methods training as required	51
3.01.08	Set up phase documentation files	51
3.01.09	Revise project plan and schedule estimates	51
3.01.10	Develop and/or use database standards for phase	51
3.01.11	Interrogate and maintain metadata repository during phase	51
3.02	Review conceptual specification phase for completeness and revise as necessary	51
3.03	Remedy any data availability and quality problems determined during conceptual specification phase	51
3.04	Determine Impact on Corporate Business Model	51
3.04.01	Evaluate tables that are boundary to outside systems and revise as necessary to maximize compatibility	51



Work Breakdown Structure

3.04.02	Determine impact on logical database	51
3.04.03	Determine impact on physical database specification	52
3.04.04	Determine impact on interrogation subsystems	53
3.04.05	Determine impact on system control facilities	53
3.04.06	Determine impact on ancillary supports	53
3.04.07	Determine impact on database system quality tests	53
3.04.08	Formulate corporate business model impact report	54
3.05	Evaluate, select and/or procure DBMS packages	54
3.05.01	Determine whether database application requires static relationship data structures	54
3.05.02	Create requirements document	54
3.05.03	Create DBMS requirements document presentation	54
3.05.04	Conduct subphase review	55
3.05.05	Issue RFP and select DBMS	55
3.05.06	Formulate DBMS selection report and presentation	55
3.05.07	Conduct subphase review	55
3.05.08	Install package	55
3.05.09	Conduct DBMS training	55
3.05.10	Determine DBMS performance characteristics	55
3.05.10.01	Develop prototypical implemented data model design	56
3.05.10.02	Develop prototypical implemented data model DDL and submit to DBMS	56
3.05.10.03	Develop prototypical data loading programs	56
3.05.10.04	Develop prototypical data update programs	56
3.05.10.05	Develop prototypical interrogation modules in HLI and NL	56
3.05.10.06	Develop statistically valid distributions of sample data	56
3.05.10.07	Assess DBMS system control capabilities	56
3.05.10.08	Run various tests at different loads, velocities,etc.	58
3.05.10.09	Develop graphs and tables to assist in database design and language selection	58
3.05.10.10	Create DBMS performance report	58
3.05.10.11	Create DBMS performance presentation	59
3.05.10.12	Conduct subphase review	59
3.05.11	Develop hints and tidbits section for documentation	59
3.05.12	Create a mapping from DBMS taxonomy to DBMS documentation	59
3.05.13	Develop generic examples for all database application functions	59
3.05.14	Adjust programming effort estimates for programming assists provided by natural languages	59
3.06	Evaluate, select and/or procure metadata repository packages	59
3.06.01	Create metadata repository requirements document	59
3.06.02	Create metadata repository requirements document presentation	60
3.06.03	Conduct subphase review	60



3.06.04	Issue RFP and select metadata repository	60
3.06.05	Formulate metadata repository selection report and presentation	60
3.06.06	Conduct subphase review	60
3.06.07	Install metadata repository package	61
3.06.08	Conduct metadata repository training	61
3.06.09	Accomplish metadata repository prototype implementation	61
3.06.10	Conduct metadata repository prototype demonstration	61
3.06.11	Create metadata repository prototype demonstration presentation	61
3.06.12	Conduct subphase review	61
3.07	Determine Implementation Strategy	62
3.07.01	Bind logical database to selected DBMS	62
3.07.02	Bind physical database to selected DBMS	63
3.07.03	Determine interrogation implementation strategy	64
3.07.04	Configure system control implementation strategy	64
3.07.05	Assess hardware availability	65
3.07.06	Determine implementation strategy	65
3.07.07	Prepare system proposal for implementation phase	66
3.07.08	Create implementation approach presentation	66
3.07.09	Conduct subphase review	66
3.08	Prepare binding phase report	66
3.09	Create binding phase presentation	67
3.10	Conduct phase review	67
4	Implementation Phase	69
4.01	Form project team, plan phase, and revise project plan	69
4.01.01	Select manager for project phase	69
4.01.02	Identify administrative, clerical, and computer supports	69
4.01.03	Select project staff for phase	69
4.01.04	Secure commitments for staff availability	69
4.01.05	Create phase plans and schedules	69
4.01.06	Organize phase review committee	69
4.01.07	Conduct methods training as required	69
4.01.08	Set up phase documentation files	69
4.01.09	Revise project plan and schedule estimates	69
4.01.10	Develop and/or use database standards for phase	69
4.01.11	Interrogate and maintain metadata repository during phase	69
4.02	Create or revise logical database	69
4.03	Create or revise physical database	70
4.03.01	Create or revise operational data model	70
4.03.02	Create or revise views	70
4.03.03	Create or revise each data update subsystem	70
4.03.04	Create or revise each data loading subsystem	72



Work Breakdown Structure

4.03.05	Create or revise each data conversion subsystem	73
4.03.06	Create or revise test data	74
4.03.07	Create or revise database backup procedures	74
4.03.08	Perform system development test on physical database	74
4.03.09	Create or revise physical database implementation report	74
4.03.10	Create or revise physical database implementation presentation	74
4.03.11	Conduct subphase review	75
4.04	Create or revise critical interrogation subsystems	75
4.04.01	Analyze implementation requirements & create schedule for specific interrogation subsystems	75
4.04.02	Analyze implementation requirements for specific interrogation subsystem ...	75
4.04.03	Create or revise specific interrogation unit	76
4.04.04	Create interrogation implementation report	77
4.04.05	Create or revise interrogation implementation presentation	77
4.04.06	Conduct subphase review	78
4.05	System control implementation	78
4.05.01	Implement or revise fully supported DBMS facilities	78
4.05.02	Implement or revise partially supported DBMS facilities	78
4.05.03	Acquire/implement and/or revise facilities not supported by DBMS	78
4.05.04	Create utilization scenarios & fully test all facilities	79
4.05.05	Create or revise operations documentation & guides	79
4.05.06	Determine, implement and maintain security and privacy for implementation phase activities	79
4.05.07	Create or revise system control implementation report	79
4.05.08	Create or revise system control implementation presentation	79
4.05.09	Conduct subphase review	80
4.06	Develop or revise ancillary supports	80
4.06.01	Develop or revise database system training	80
4.06.02	Develop or revise database hotline	80
4.06.03	Develop or revise database standards	81
4.06.04	Develop or revise final test data	81
4.06.05	Develop or revise database system documentation	81
4.06.05.01	Develop internal database documentation	81
4.06.05.02	Prepare user manuals	82
4.06.05.03	Prepare operations documentation	82
4.06.05.04	Review & finalize all documentation	82
4.06.06	Prepare ancillary supports implementation report	82
4.06.07	Create or revise ancillary supports presentation	83
4.06.09	Conduct subphase review	83
4.07	Plan and conduct system quality tests (system quality test)	83
4.07.01	Create or revise system operation scenarios	83



4.07.02	Create or revise tests to validate scenarios	83
4.07.03	Schedule database system quality test	84
4.07.04	Conduct system quality tests	84
4.07.05	Prepare system test report	84
4.07.06	Create system test presentation	84
4.07.07	Conduct subphase review	84
4.08	Create implementation phase report	85
4.09	Create implementation phase presentation	85
4.10	Conduct phase review	85
5	Conversion & Deployment Phase	86
5.01	Form project team, plan phase, and revise project plan	86
5.01.01	Select manager for project phase	86
5.01.02	Identify administrative, clerical, and computer supports	86
5.01.03	Select project staff for phase	86
5.01.04	Secure commitments for staff availability	86
5.01.05	Create phase plans and schedules	86
5.01.06	Organize phase review committee	86
5.01.07	Conduct methods training as required	86
5.01.08	Set up phase documentation files	86
5.01.09	Revise project plan and schedule estimates	86
5.01.10	Develop and/or use database standards for phase	86
5.01.11	Interrogate and maintain metadata repository during phase	86
5.02	Review prior phase for completeness and revise as necessary	86
5.03	Secure, install and test all equipment	86
5.04	Conduct all training	86
5.04.01	Conduct user training	86
5.04.02	Conduct system maintenance training	86
5.05	Convert all data	87
5.06	Create conversion and deployment report	87
5.07	Create conversion and deployment review presentation	87
5.08	Conduct phase review	87
5.09	Prepare estimate for production and administration phase	87
5.10	Prepare project production & administration phase funding	87
5.11	Present for funding review	87
6	Production and Administration Phase	89
6.01	Form project team, plan phase, and revise project plan	89
6.01.01	Select manager for project phase	89
6.01.02	Identify administrative, clerical, and computer supports	89
6.01.03	Select project staff for preliminary analysis phase	89
6.01.04	Secure commitments for staff availability	89
6.01.05	Create phase plans and schedules	89



Work Breakdown Structure

6.01.06	Organize phase review committee	89
6.01.07	Conduct methods training as required	89
6.01.08	Set up phase documentation files	89
6.01.09	Revise project plan and schedule estimates	89
6.01.10	Develop and/or use database standards for phase	89
6.01.11	Interrogate and maintain metadata repository during phase	89
6.02	Database system production	89
6.02.01	Commence database system	89
6.02.02	Conduct ongoing database system production	90
6.02.03	Develop and maintain views as necessary	90
6.03	Develop interrogation subsystems	90
6.03.02	Create views	90
6.03.03	Prototype through a natural language	90
6.03.04	Select cost efficient language	91
6.03.05	Implement interrogation subsystem	91
6.03.06	Perform system development test on interrogation subsystem	91
6.03.07	Assess production impact	91
6.03.08	Develop interrogation implementation report	91
6.03.09	Create interrogation implementation presentation	92
6.03.10	Conduct interrogation review	92
6.04	Application Optimization Assessment	92
6.04.01	Assess physical database performance	92
6.04.02	Assess interrogation performance	92
6.04.03	Assess system control performance	92
6.04.03.01	Assess audit trail performance	92
6.04.03.02	Assess performance of backup & recovery	93
6.04.03.03	Assess performance of message processing	93
6.04.03.04	Assess completeness and performance of security & privacy	93
6.04.03.05	Assess performance of concurrent operations	94
6.04.03.06	Assess sophistication of reorganization	94
6.04.03.07	Assess sophistication of multiple database object processing	94
6.04.03.08	Create system control assessment report	95
6.04.04	Create logical database change requirements to accommodate performance ...	95
6.04.05	Create consolidated application performance change requirements report	95
6.04.06	Create assessment to change ancillary support for application performance change requirements	95
6.04.07	Create consolidated application performance change requirements presentation	95
6.04.08	Conduct subphase review	96
6.05	Determine database system maintenance policy	96
6.05.01	Select maintenance approach, methods, and tools	96



6.05.01.01	Determine maintenance approach	96
6.05.01.02	Establish maintenance methods	96
6.05.01.03	Identify and select maintenance tools	96
6.05.01.04	Document selections and formalize findings	96
6.05.02	Determine procedures for standard and emergency maintenance	96
6.05.03	Publish maintenance policy	97
6.06	Perform standard system maintenance	97
6.06.01	Identify and assign maintenance problem	97
6.06.01.01	Identify maintenance need	97
6.06.01.02	Document maintenance need	97
6.06.01.03	Analyze the maintenance need	97
6.06.01.04	Identify proper person/organization to solve problem	97
6.06.02	Analyze problem and estimate maintenance effort	97
6.06.02.01	Analyze problem, determine solution and work around	98
6.06.02.02	Determine pervasiveness and issue alert as appropriate	98
6.06.02.03	Determine alternative of choice and obtain approval	98
6.06.02.04	Set priority of implementation	98
6.06.02.05	Notify proponent of schedule of change	98
6.06.03	Accomplish the maintenance change	98
6.06.03.01	Create conceptual specification of change	98
6.06.03.02	Create implementation specification of change	99
6.06.03.03	Implement the change	100
6.06.03.04	Perform system development test on change	101
6.06.03.05	Deliver the change	101
6.06.04	Plan and conduct system quality tests	101
6.06.05	Conduct required training	102
6.06.06	Install change into production	102
6.06.07	Monitor database application	102
6.07	Perform emergency maintenance	103
6.07.01	Document the problem	103
6.07.02	Verify the problem through independent methods	103
6.07.03	Determine solution to problems	103
6.07.04	Acquire approval for change	103
6.07.05	Change run-time modules	103
6.07.06	Modify source code and create database object for production library	103
6.07.07	Install change	103
6.07.08	Create standard maintenance change request	103
6.08	Accomplish database system revision	103
6.08.01	Propose database system revision	103
6.08.02	Prepare implementation strategy of revision	104
6.08.03	Implement revision	104



Work Breakdown Structure

6.08.03.01	Implement revised logical database	104
6.08.03.02	Implement revised physical database	104
6.08.03.03	Implement interrogation subsystem revisions	105
6.08.03.04	Accomplish system control changes	105
6.08.03.04.01	Revise fully supported DBMS facilities	105
6.08.03.04.02	Revise partially supported DBMS facilities	105
6.08.03.04.03	Revise facilities not supported by DBMS	105
6.08.03.04.04	Revise utilization scenarios & fully test all facilities	106
6.08.03.04.05	Revise operations documentation & guides	106
6.08.03.04.06	Create revised system control implementation report	106
6.08.03.04.07	Create revised system control implementation presentation	106
6.08.03.04.08	Conduct subphase review	106
6.08.04	Revise ancillary supports	106
6.08.05	Plan and conduct system quality tests	107
6.08.06	Prepare software release notice	107
6.08.07	Merge revision into production system	108
6.08.08	Perform enterprise model audit on evolution	108
6.08.09	Conduct revision review	108
6.09	Perform enterprise database audit	108
6.09.01	Audit metadata repository components and note discrepancies by identifying the items's unique identifier and suspected problem	108
6.09.02	Audit non-metadata repository data and computer programs	114
6.09.03	Audit ancillary support items	115
6.09.04	Formulate audit report	118
6.09.05	Monitor database system operation	118
6.09.06	Certify enterprise model audit pass/fail	118
INDEX	119







1

Preliminary Analysis Phase

1 Preliminary Analysis

1.01 Form project team, plan phase, & estimate project

1.01.01 Select manager for project phase

1.01.02 Identify administrative, clerical, and computer supports

1.01.03 Select project staff for phase

1.01.03.01 Review all job descriptions of staff to match requirements of DB project methodology

1.01.03.02 Interview and select database specialist

1.01.03.03 Interview and select functional users

1.01.04 Secure commitments for staff availability

1.01.04.01 Estimate time requirement for project manager

1.01.04.02 Estimate time requirement for administrative, clerical, and computer supports

1.01.05 Create estimate for project

1.01.05.01 Identify possible manager for project

1.01.05.02 Identify ideal project staff types and quantities for project

1.01.05.03 Identify appropriate WBS for project type

1.01.05.04 Review selected WBS to ensure that it has the least number of steps consonant with maximum quality

1.01.05.05 Configure steps into PERT chart

1.01.05.06 Identify appropriate metrics, work unit quantities, and work environment factors for each selected WBS step

1.01.05.07 Resource load each step by resource type

1.01.05.08 Ensure that project management package has appropriate calendar for project time period

1.01.05.09 Generate CPM and Gantt charts

1.01.05.10 Prepare project description for project and generally identify the missions, business functions, and database objects that are to be addressed

1.01.05.11 Review with key users and revise all project estimate components until complete and acceptable

1.01.05.12 Fix project base line for earned value reporting

1.01.06 Create phase and task plans and schedules

1.01.06.01 Create phase specific PERT charts

1.01.06.02 Create phase specific Gantt charts

1.01.06.03 Create phase CPM charts

1.01.06.04 Create phase manpower estimate charts

1.01.07 Organize phase review committee

1.01.08 Conduct methods training as required

Work Breakdown Structure

- 1.01.09 Set up phase documentation files
- 1.01.10 Create project plan and schedule estimates
 - 1.01.10.01 Create PERT charts
 - 1.01.10.02 Create Gantt charts
 - 1.01.10.03 Create CPM charts
 - 1.01.10.04 Create manpower estimate charts
- 1.01.11 Establish metadata repository¹
 - 1.01.11.01 Define tables and relationships
 - 1.01.11.02 Create schema and views
 - 1.01.11.03 Create update programs and reports
 - 1.01.11.04 Establish forms and procedures
 - 1.01.11.05 Accomplish initial metadata database loading
- 1.01.12 Interrogate and maintain metadata repository during phase
- 1.01.13 Establish effective computing environment, libraries, etc., for resource usage tracking, for storage of data, programs, and run streams for development and unit test, for system test and integration, and for production
- 1.01.14 Develop and/or use database standards for phase
- 1.01.15 Identify and allocate management and clerical overhead to phase
 - 1.01.15.01 Assess and allocate appropriate management overhead for reviews, supervision to the phase.
 - 1.01.15.02 Assess and allocate appropriate clerical support for various phase activities
- 1.02 Create initial database project requirements description
 - 1.02.01 Request and review documentation of prior related database efforts
 - 1.02.02 Obtain initial scoping information for project
 - 1.02.02.01 Conduct broad interviews with user management to determine business database objectives for database project
 - 1.02.02.02 Identify, name and briefly describe the involved missions
 - 1.02.02.03 Identify, name and briefly describe the business resources that are the foundation of the database objects
 - 1.02.02.04 Identify, name and briefly describe the existing business information systems

¹ The metadata repository is a repository product of Whitemarsh Information Systems. Metabase is available either as a standalone PC or PC/server software product or as a design specification for client development on their own computing environment.



- 1.02.02.05 Identify, name and briefly describe the critical business events that currently involve automation assists
- 1.02.02.06 Identify, name and briefly describe the business functions that must be performed
- 1.02.02.07 Identify, name and briefly describe the organizations involved in the enterprise database effort
- 1.02.03 Perform preliminary data analysis
 - 1.02.03.01 Identify major suppliers & uses of database data
 - 1.02.03.02 Identify, create, & analyze prototypical data and reports
 - 1.02.03.02.01 Identify prototypical data and reports
 - 1.02.03.02.02 Create matrix versions of data and reports
 - 1.02.03.02.03 Describe significant characteristics of data and reports
 - 1.02.03.02.04 Summarize data and reports fundamental orientation towards operational, control, or MIS
 - 1.02.03.02.05 Update metadata repository
 - 1.02.03.03 Conduct preliminary data transfer and data conversion interview
 - 1.02.03.03.01 Determine the number of systems to be converted
 - 1.02.03.03.02 Determine the type of automation for each system
 - 1.02.03.03.03 Determine the number of files and file access methods for each system
 - 1.02.03.03.04 Determine the file layouts for each file
 - 1.02.03.03.05 Update metadata repository with data transfer and data conversion findings
 - 1.02.03.03.06 Obtain user concurrence on interview findings
- 1.02.04 Formulate list of business database objectives that are to be achieved
 - 1.02.04.01 Formulate database objectives dealing with the business problem that is to be solved
 - 1.02.04.02 Formulate database objectives dealing with the system to be converted
 - 1.02.04.03 Formulate database objectives dealing with the design and development period
 - 1.02.04.04 Formulate database objectives dealing with the benefits to be achieved.
 - 1.02.04.05 Formulate list of business database objectives that are to be achieved during data conversion
- 1.02.05 Formulate reasonable list of mechanisms to evaluate project success
- 1.02.06 Reach consensus on database objectives and evaluation mechanisms
- 1.03 Create mission description model
 - 1.03.01 Create business mission submodel
 - 1.03.01.01 Succinctly create business mission in business policy terms
 - 1.03.01.01.01 Name mission



Work Breakdown Structure

- 1.03.01.01.02 Describe mission in business policy terms
- 1.03.01.01.03 Use prototypical source data and reports to assist in definition
- 1.03.01.01.04 Remove any organizational references from missions
- 1.03.01.01.05 Remove any techniques or details that indicate how the mission is accomplished
- 1.03.01.01.06 Review missions to ensure that only the end-result *what* is described, and that both *who* and *how* has been removed
- 1.03.01.01.07 Review mission description with key users and revise until complete and acceptable
- 1.03.01.01.08 Enter mission description into metadata repository

- 1.03.01.02 Partition business mission into sufficient stand alone subordinate missions
- 1.03.01.02.01 Identify stand alone subordinate missions
- 1.03.01.02.02 Describe subordinate missions in business terms
- 1.03.01.02.03 Use prototypical data and reports to assist in definition
- 1.03.01.02.04 Review subordinate mission with key users and revise until complete and acceptable
- 1.03.01.02.05 Repartition subordinate mission descriptions into lower level subordinate mission descriptions and repeat description until no longer necessary
- 1.03.01.02.06 Enter subordinate missions into metadata repository

- 1.03.01.03 Create mission description diagram

- 1.03.02 Create database domain submodel

- 1.03.02.01 Create database domains
- 1.03.02.01.01 Succinctly describe database domain from subordinate mission descriptions in business policy terms
- 1.03.02.01.01 Name database domain
- 1.03.02.01.02 Describe database domain in business policy terms
- 1.03.02.01.03 Use prototypical source data and reports to assist in definition
- 1.03.02.01.04 Validate database domain sufficiency by balancing the implied major database outputs against the major database contents
- 1.03.02.01.05 Review database domain with key users and revise until both non-redundant, complete, and acceptable
- 1.03.02.01.06 Relate database domains to missions and enter into metadata repository

- 1.03.02.02 Partition domains into sufficient stand alone subordinate database domains
- 1.03.02.02.01 Identify stand alone subordinate database subdomains
- 1.03.02.02.02 Describe subordinate database domains in business terms
- 1.03.02.02.03 Use prototypical data and reports to assist in definition



- 1.03.02.02.04 Review database subordinate database subdomains with key users and revise until complete and acceptable
- 1.03.02.02.05 Repartition subordinate database subdomains into lower level subordinate database subdomains and repeat description until no longer necessary
- 1.03.02.02.06 Enter subordinate database domains into metadata repository

- 1.03.02.03 Create initial database domain diagram for each database subdomain
 - 1.03.02.03.01 Identify and define entities in non-redundant fashion
 - 1.03.02.03.02 Identify and define obvious relationships among entities
 - 1.03.02.03.03 Construct database domain diagrams
 - 1.03.02.03.04 Review diagrams, entities, and relationships with users to assess relevance & revise as necessary

- 1.03.02.04 Merge subordinate database domain diagrams
 - 1.03.02.04.01 Select most complex subordinate database domain diagram
 - 1.03.02.04.02 Attempt merger of other database domain diagrams
 - 1.03.02.04.03 Adjust subordinate database domain entities & relationships until merger is possible
 - 1.03.02.04.04 Review combined database domain diagram for common understanding & agreement with key users
 - 1.03.02.04.05 Finalize database domain diagram

- 1.03.03 Create database object submodel
 - 1.03.03.01 Select entity from database domain diagram as a potential database object
 - 1.03.03.02 Create name for potential database object
 - 1.03.03.03 Create definition for potential database object
 - 1.03.03.04 Determine if multiple database object tables exist for potential database object
 - 1.03.03.05 Record as database objects those potential database objects that have multiple database object tables
 - 1.03.03.06 Record as database object tables those potential database objects that do not contain multiple database object tables
 - 1.03.03.07 Record as data elements those potential database objects that are not even database object tables; rather they describe single business fact
 - 1.03.03.08 Review discovered database objects, database object tables and data elements with users to ensure proper business names and business definitions
 - 1.03.03.09 Provide an example for each database object, property class, and data element
 - 1.03.03.10 Enter database objects, database object tables and data elements into metadata repository
 - 1.03.03.11 Present defined database objects and review as appropriate
 - 1.03.03.12 Create database object diagram



Work Breakdown Structure

- 1.03.03.13 Create relationships between database objects and database domains and then enter into metadata repository
- 1.03.04 Create high-level business function submodel
 - 1.03.04.01 Succinctly create business function in business policy terms
 - 1.03.04.01.01 Name business function
 - 1.03.04.01.02 Describe business function in business policy terms indicating appropriate business activities
 - 1.03.04.01.03 Use prototypical source data and reports to assist in definition
 - 1.03.04.01.04 Remove any organizational style indications from business functions
 - 1.03.04.01.05 Identify organizations that perform same high-level description of business function
 - 1.03.04.01.06 Remove any detailed techniques that indicate how the business function is accomplished
 - 1.03.04.01.07 Enter discovered business functions into metadata repository
 - 1.03.04.01.08 Enter discovered organizations into metadata repository
 - 1.03.04.01.09 Interrelate organizations and business functions, and enter interrelationships into metadata repository
 - 1.03.04.01.10 Interrelate business functions with missions and database objects, and enter into metadata repository
 - 1.03.04.01.11 Identify frequency of accomplishment by organization of business function
 - 1.03.04.01.12 Review business functions to ensure that how the business function is kept at a high-level
 - 1.03.04.01.13 Review business function description with key users and revise until complete and acceptable
 - 1.03.04.02 Partition business function into sufficient stand alone subordinate business functions
 - 1.03.04.02.01 Identify stand alone subordinate business functions
 - 1.03.04.02.02 Describe subordinate business functions in business terms
 - 1.03.04.02.03 Use prototypical data and reports to assist in definition
 - 1.03.04.02.04 Review subordinate business function with key users and revise until complete and acceptable
 - 1.03.04.02.05 Repartition subordinate business function descriptions into lower level subordinate business function descriptions and repeat description until no longer high-level
 - 1.03.04.03 Enter subordinate business functions into metadata repository
 - 1.03.04.04 Create high-level business function diagram



- 1.03.04.04.01 Identify high-level business functions & define its purpose, scope, etc., in terms of business activities
- 1.03.04.04.02 Identify the business environment & activities in which business function occurs
- 1.03.04.04.03 Identify organizational responsibility for business function
- 1.03.04.04.04 Identify frequency of business function
- 1.03.04.04.05 Provide an example of each business function flow diagram
- 1.03.04.04.06 If business function contains business functions then repeat only a few levels
- 1.03.04.04.07 Record business function data into metadata repository

- 1.03.05 Create business and technical terms glossary, and organize in learning order

- 1.03.06 Create mission model report
 - 1.03.06.01 Prepare mission scope and narrative
 - 1.03.06.02 Create plan & format for presenting mission descriptions, database domains, database objects, high level functions and all graphics
 - 1.03.06.03 Prepare report for mission description model

- 1.03.07 Create preliminary analysis presentation
 - 1.03.07.01 Review & revise preliminary analysis phase
 - 1.03.07.02 Construct features, advantages and benefits (FAB) of database system
 - 1.03.07.03 Use FABs to create use scenarios
 - 1.03.07.04 Graphically identify entities/processes that participate in scenario
 - 1.03.07.05 Create narrative for use scenario
 - 1.03.07.06 Identify key individual for use scenario
 - 1.03.07.07 Review use scenario & narrative with key individual and modify as appropriate
 - 1.03.07.08 Create A/V materials for use scenario presentation
 - 1.03.07.09 Review and finalize A/V materials prior to review

- 1.03.08 Conduct phase review
 - 1.03.08.01 Present mission description model
 - 1.03.08.02 Receive and respond to comments
 - 1.03.08.03 Modify model components as necessary
 - 1.03.08.04 Seek concurrence with phase

- 1.04 Create business information systems plan
 - 1.04.01 Create Ross² resource life cycles
 - 1.04.01.01 Identify critical corporate resources

²

Resource Life Cycles were created by Ron Ross. They are described in the book Resource Life Cycle Analysis



Work Breakdown Structure

- 1.04.01.02 Create life-cycle for each resource
- 1.04.01.03 Interconnect lifecycles via enablement-precedence vectors
- 1.04.01.04 Review with key users and revise until complete and acceptable

- 1.04.02 Create relationships between resource life cycle nodes and metabase component instances
 - 1.04.02.01 Create relationships between resource life cycle nodes and database objects, and enter into metadata repository
 - 1.04.02.02 Create relationships between resource life cycle nodes and subordinate mission descriptions, and enter into metadata repository
 - 1.04.02.03 Create relationships between resource life cycle nodes and subordinate business function descriptions, and enter into metadata repository
 - 1.04.02.04 Create relationships between resource life cycle nodes and existing systems, and enter into metadata repository
 - 1.04.02.05 Create relationships between resource life cycle nodes and existing files and databases, and enter into metadata repository
 - 1.04.02.06 Generate appropriate metadata repository reports to clearly display *as-is* and *to-be* versus resource life cycle nodes.
 - 1.04.02.07 Review with key users and revise until complete and acceptable

- 1.04.03 Propose business information systems projects
 - 1.04.03.01 Hypothesize as an operational subject database project all the database objects associated with a single resource life cycle
 - 1.04.03.02 Hypothesize as an operational information systems project all the resource life cycle nodes associated with a single subordinate mission description
 - 1.04.03.03 Reestablish the precedence-enablement vectors at the subordinate mission levels

- 1.04.04 Examine the state of automation for each associated business information system
 - 1.04.04.01 Determine improvements required for logical database
 - 1.04.04.01.01 Determine changes to database objects
 - 1.04.04.01.02 Determine changes to data elements
 - 1.04.04.01.03 Determine changes to tables and columns
 - 1.04.04.01.04 Determine changes to data integrity rules
 - 1.04.04.01.05 Prepare logical database impact report
 - 1.04.04.01.06 Quantify the logical database components required to be added, deleted or modified
 - 1.04.04.01.07 Create logical database impact presentation
 - 1.04.04.01.08 Conduct subphase review
 - 1.04.04.02 Determine improvements required for physical database



- 1.04.04.02.01 Determine changes in data distribution model
- 1.04.04.02.02 Determine changes in process distribution model
- 1.04.04.02.03 Determine changes in DBMS
- 1.04.04.02.04 Determine changes in end-user GUI
- 1.04.04.02.05 Quantify the physical database components required to be added, deleted or modified
- 1.04.04.02.06 Prepare physical database impact report
- 1.04.04.02.07 Create physical database impact presentation
- 1.04.04.02.08 Conduct subphase review

- 1.04.04.03 Determine improvements required for interrogation
 - 1.04.04.03.01 Determine improvements for ad hoc reports
 - 1.04.04.03.02 Determine improvements for standard reports
 - 1.04.04.03.03 Quantify the interrogation components required to be added, deleted or modified
 - 1.04.04.03.04 Prepare interrogation impact report
 - 1.04.04.03.05 Create interrogation impact presentation
 - 1.04.04.03.06 Conduct subphase review

- 1.04.04.04 Determine improvements required for system control
 - 1.04.04.04.01 Determine changes required for audit trails
 - 1.04.04.04.02 Determine impact on backup & recovery needs
 - 1.04.04.04.03 Determine impact on concurrent operations
 - 1.04.04.04.04 Determine impact on security & privacy
 - 1.04.04.04.05 Determine impact on reorganization
 - 1.04.04.04.06 Determine impact on multiple databases
 - 1.04.04.04.07 Quantify the system control components required to be added, deleted or modified
 - 1.04.04.04.08 Prepare system control impact report
 - 1.04.04.04.09 Create system control impact presentation
 - 1.04.04.04.10 Conduct subphase review

- 1.04.04.05 Create system upgrade assessment report for information system
 - 1.04.04.05.01 Create introduction & scope
 - 1.04.04.05.01.01 Include mission descriptions
 - 1.04.04.05.01.02 Include mission description diagrams
 - 1.04.04.05.01.03 Demarcate the intended users
 - 1.04.04.05.01.04 Identify the systems being upgraded
 - 1.04.04.05.01.05 Review list of business goals, database objectives, & criteria for success measurement for database system, and revise as necessary
 - 1.04.04.05.01.06 Provide a summary of pertinent statistics about database



Work Breakdown Structure

- 1.04.04.05.02 Create logical database requirements section
 - 1.04.04.05.02.01 Include database domains and database domain diagrams
 - 1.04.04.05.02.02 Include final data integrity model (database objects, tables, columns, and data integrity rules)
 - 1.04.04.05.02.03 Create final report of logical database
- 1.04.04.05.03 Create physical database requirements section
 - 1.04.04.05.03.01 Include data loading subsystem specifications
 - 1.04.04.05.03.02 Include data update subsystem specifications
 - 1.04.04.05.03.03 Include database size estimate changes
 - 1.04.04.05.03.04 Include data source quality reports
 - 1.04.04.05.03.05 Include backup requirements reports
 - 1.04.04.05.03.06 Include data integrity subsystem specifications
 - 1.04.04.05.03.07 Create final physical database report
- 1.04.04.04.04 Create business systems environment section
 - 1.04.04.05.04.01 Include business event specifications
 - 1.04.04.05.04.02 Include business function specifications
- 1.04.04.05.05 Conduct final review of functional completeness & acceptable interaction of logical, physical, and business system environment models
- 1.04.04.05.06 Create interrogation requirements section
 - 1.04.04.05.06.01 Include standard report requirements
 - 1.04.04.05.06.02 Include ad-hoc report requirements
 - 1.04.04.05.06.03 Prepare final consolidated interrogation requirements report
- 1.04.04.05.07 Prepare system control requirements section
 - 1.04.04.05.07.01 Include system control functional requirements
 - 1.04.04.05.07.02 Include system control resource requirements
 - 1.04.04.05.07.03 Prepare final consolidated system control specification
- 1.04.04.05.08 Prepare summary of business information system requirements report
 - 1.04.04.05.08.01 Prepare summary of logical database including statement of limitations, audience, etc.
 - 1.04.04.05.08.02 Prepare summary of physical database including statement of limitations, audience, etc.
 - 1.04.04.05.08.03 Prepare summary of interrogation including statement of limitations, audience, etc.
 - 1.04.04.05.08.04 Prepare summary of system control including statement of limitations, audience, etc.



- 1.04.04.05.08.05 Prepare a summary that specifies overall limitations, problem areas, etc.
- 1.04.04.06 Estimate the resources required to upgrade the business information system
 - 1.04.04.06.01 Identify possible manager for business information system upgrade project
 - 1.04.04.06.02 Identify ideal project staff types and quantities for business information system upgrade project
 - 1.04.04.06.03 Review selected WBS to ensure that it has the least number of steps consonant with maximum quality
 - 1.04.04.06.04 Configure steps into a PERT chart that is local to the business information system
 - 1.04.04.06.05 Identify appropriate metrics, work unit quantities, and work environment factors for each selected WBS step
 - 1.04.04.06.06 Resource load each step by resource type
 - 1.04.04.06.07 Ensure that business information system upgrade project management package has appropriate calendar for time period
 - 1.04.04.06.08 Generate CPM and Gantt charts
 - 1.04.04.06.09 Prepare business information system upgrade project description and identify the missions, business functions, and database objects that are to be addressed
 - 1.04.04.06.10 Review with key users and revise all business information system upgrade project estimate components until complete and acceptable
 - 1.04.04.06.11 Fix business information system upgrade project base line for earned value reporting
- 1.04.04.07 Prepare consolidated business information system upgrade report for each business information systems associated with resource life cycle node
 - 1.04.04.07.01 Prepare consolidated summary of logical database including statement of limitations, audience, etc.
 - 1.04.04.07.02 Prepare consolidated summary of physical database including statement of limitations, audience, etc.
 - 1.04.04.07.03 Prepare consolidated summary of interrogation including statement of limitations, audience, etc.
 - 1.04.04.07.04 Prepare consolidated summary of system control including statement of limitations, audience, etc.
 - 1.04.04.07.05 Prepare a consolidated summary that specifies overall limitations, problem areas, etc.
- 1.04.04.08 Create consolidated high-level estimate for all business information systems associated with each resource life cycle node
 - 1.04.04.08.01 Identify possible manager for consolidated project
 - 1.04.04.08.02 Adjust, if necessary, project staff types and quantities for consolidated project
 - 1.04.04.08.03 Identify overall WBS for consolidated set of projects



Work Breakdown Structure

- 1.04.04.08.04 Adjust, if necessary, metrics, quantities, and work environment factors for each selected WBS step
- 1.04.04.08.05 Create overall project PERT, Gantt, and CPM task plans and schedules
- 1.04.04.09 Create overall business information system upgrade presentation
 - 1.04.04.09.01 Modify features, advantages and benefits (FAB) of database system
 - 1.04.04.09.02 Include FABs to create database use scenarios
 - 1.04.04.09.03 Graphically identify tables or processes that participate in scenario
 - 1.04.04.09.04 Create narrative for database use scenario
 - 1.04.04.09.05 Identify key individual for database use scenario
 - 1.04.04.09.06 Review database use scenario & narrative with key individual and modify as appropriate
 - 1.04.04.09.07 Create A/V materials for database use scenario presentation
 - 1.04.04.09.08 Review and finalize A/V materials prior to review
- 1.04.04.10 Conduct phase review
 - 1.04.04.10.01 Present system upgrade specification
 - 1.04.04.10.02 Receive and respond to comments
 - 1.04.04.10.03 Modify as necessary
 - 1.04.04.10.04 Seek concurrence with system phase
- 1.04.05 Create and load resource life cycle node pert chart
- 1.04.06 Enter the resource life cycle node chart into a project management package as a PERT chart
- 1.04.07 Enter the PERT chart of the upgrade business information systems as a subproject for appropriate resource life cycle nodes
- 1.04.08 Resource load the resource life cycle subproject nodes
 - 1.04.08.01 Resource load each node by resource type
 - 1.04.08.02 Ensure that resource life cycle upgrade project management package has appropriate calendar for resource life cycle upgrade project time period
 - 1.04.08.03 Generate CPM and Gantt charts for individual subprojects within resource lifecycle node
 - 1.04.08.04 Review with key users and revise business information system plan estimate components until complete and acceptable
- 1.04.09 Project management scheduling
 - 1.04.09.01 Revise as necessary PERT charts of the overall set of resource life cycles
 - 1.04.09.02 Execute the project management algorithms for scheduling
 - 1.04.09.03 Create Gantt charts
 - 1.04.09.04 Create CPM charts



- 1.04.09.05 Create manpower estimate charts
- 1.04.10 Publish and review the Information systems plan
 - 1.04.10.01 Establish the Information systems plan review committee
 - 1.04.10.02 Identify proponents for each resource life cycle node and for each information system within each resource life cycle node
- 1.04.11 Create business information systems plan presentation
 - 1.04.11.01 Modify features, advantages and benefits (FAB) of business information systems plan
 - 1.04.11.02 Include FABs to create business information systems plan use scenarios
 - 1.04.11.03 Graphically major tables and/or business information systems that participate in scenario
 - 1.04.11.04 Create narrative for business information systems plan use scenario
 - 1.04.11.05 Identify key individual for business information systems plan use scenario
 - 1.04.11.06 Review business information systems plan use scenario & narrative with key individual and modify as appropriate
 - 1.04.11.07 Create A/V materials for business information systems plan use scenario presentation
 - 1.04.11.08 Review and finalize A/V materials prior to review
- 1.04.12 Conduct phase review
 - 1.04.12.01 Present resource life cycle node upgrade specification
 - 1.04.12.02 Receive and respond to comments
 - 1.04.12.03 Modify as necessary
 - 1.04.12.04 Seek concurrence for resource life cycle node upgrade plan
- 1.05 Create preliminary analysis phase report
 - 1.05.01 Select project from information systems plan to accomplish
 - 1.05.02 Prepare database system scope narrative
 - 1.05.03 Create plan & format for presenting mission descriptions, database domains, and all graphics
 - 1.05.04 Create proposal for conceptual specification phase
 - 1.05.04.01 Determine professional staffing requirements
 - 1.05.04.02 Determine support staffing requirements
 - 1.05.04.03 Determine facilities support requirements
 - 1.05.04.04 Determine automation support requirements
 - 1.05.04.05 Create Gantt chart for conceptual specification phase
 - 1.05.04.06 Create necessary proposal narrative



Work Breakdown Structure

- 1.05.05 Prepare final report for preliminary analysis phase
- 1.06 Create preliminary analysis presentation
 - 1.06.01 Review & revise preliminary analysis phase
 - 1.06.02 Construct features, advantages and benefits (FAB) of database system
 - 1.06.03 Include FABs to create use scenarios
 - 1.06.04 Graphically identify entities/processes that participate in scenario
 - 1.06.05 Create narrative for use scenario
 - 1.06.06 Identify key individual for use scenario
 - 1.06.07 Review use scenario & narrative with key individual and modify as appropriate
 - 1.06.08 Create A/V materials for use scenario presentation
 - 1.06.09 Review and finalize A/V materials prior to review
- 1.07 Conduct phase review
 - 1.07.01 Present preliminary analysis
 - 1.07.02 Receive and respond to comments
 - 1.07.03 Modify system as necessary
 - 1.07.04 Seek concurrence with system phase



2

Conceptual Specification Phase

- 2 Conceptual specification
 - 2.01 Form project team, plan phase, and revise project plan
 - 2.01.01 Select manager for project phase
 - 2.01.02 Identify administrative, clerical, and computer supports
 - 2.01.03 Select project staff for phase
 - 2.01.03.01 Interview and select database specialist
 - 2.01.03.02 Interview and select functional users
 - 2.01.04 Secure commitments for staff availability
 - 2.01.04.01 Estimate time requirement for project manager
 - 2.01.04.02 Estimate time requirement for administrative, clerical, and computer supports
 - 2.01.05 Create phase and task plans and schedules
 - 2.01.05.01 Create phase specific PERT charts
 - 2.01.05.02 Create phase specific Gantt charts
 - 2.01.05.03 Create phase CPM charts
 - 2.01.05.04 Create phase manpower estimate charts
 - 2.01.06 Organize phase review committee
 - 2.01.07 Conduct methods training as required
 - 2.01.08 Set up phase documentation files
 - 2.01.09 Revise project plan and schedule estimates
 - 2.01.09.01 Revise PERT charts
 - 2.01.09.02 Revise Gantt charts
 - 2.01.09.03 Revise CPM charts
 - 2.01.09.04 Revise manpower estimate charts
 - 2.01.10 Interrogate and maintain metadata repository
 - 2.01.11 Develop and/or use database standards for phase
 - 2.01.12 Review & revise preliminary analysis phase as necessary
 - 2.02 Create logical database of business requirements
 - 2.02.01 Perform detailed data analysis
 - 2.02.01.01 Identify all sources of data and all types of reports
 - 2.02.01.02 Collect descriptions of data from various forms, prior reports, and user interviews
 - 2.02.01.03 Analyze each data source and report
 - 2.02.01.03.01 Define general characteristics, i.e., structural, descriptive, or historical

Work Breakdown Structure

- 2.02.01.03.02 Determine data criticality to project mission description
- 2.02.01.03.03 Specify the general orientation of the data, i.e., operational, tactical, or strategic
- 2.02.01.03.04 Identify the typical report request mechanism
- 2.02.01.03.05 Determine the expected frequency of usage
- 2.02.01.03.06 Determine the granularity, i.e., operational, summary, projected
- 2.02.01.03.07 Determine data currency requirement, i.e., daily, monthly, quarterly, annual, etc.
- 2.02.01.03.08 Determine data availability, i.e., automated, manual, unknown
- 2.02.01.03.09 Determine data reliability
- 2.02.01.03.10 Enter characteristics into metabase's report, file or document meta entity types

- 2.02.01.04 Record report descriptions and characteristics onto report form and enter into metadata repository
- 2.02.01.05 Construct a data analysis report, and review & revise with users as necessary

- 2.02.02 Perform detailed data analysis for data transfer and conversion
- 2.02.02.01 Identify affected users, systems and files involved in the data transfer and/or data conversion process
- 2.02.02.02 Gather data transfer and data conversion file information
- 2.02.02.03 Identify all sources of data and all types of reports
- 2.02.02.04 Obtain a sample of the data and the format for each file
- 2.02.02.05 Collect descriptions of data from prior reports and interviews
- 2.02.02.06 Identify all file element data editing and validations
- 2.02.02.07 Load metadata repository with system file, file element, external view columns, database view columns, and views to reflect the inventory of files to be transferred or converted.
- 2.02.02.08 Construct data transfer and data conversion report, and review and revise with users as necessary
- 2.02.02.09 Determine the type of conversion best suited for the operating environment
- 2.02.02.10 Prepare a data analysis/data source report
- 2.02.02.11 Prepare detailed conversion plan
- 2.02.02.12 Present data analysis report and detail conversion plan to users

- 2.02.03 Update project mission descriptions, mission description diagram, database domains & construct database domain diagrams
- 2.02.03.01 Refine mission descriptions
- 2.02.03.02 Refine mission diagrams
- 2.02.03.03 Refine database domain descriptions
- 2.02.03.04 Refine database domain diagrams
- 2.02.03.05 Refine database objects and database object diagrams
- 2.02.03.06 Review and revise metadata repository data as appropriate



- 2.02.04 Define data integrity model
 - 2.02.04.01 Finalize database domain diagram
 - 2.02.04.01.01 Select most complex database domain diagram
 - 2.02.04.01.02 Attempt merger of other database domain diagrams
 - 2.02.04.01.03 Correct rejected diagrams
 - 2.02.04.01.04 Review database domain diagram with users for common understanding, & agreement, and revise as necessary to achieve sufficient simplicity
 - 2.02.04.01.05 Review database domain descriptions and diagrams against data analysis report to insure complete coverage
 - 2.02.04.01.06 Iteratively review database domain diagram with users, and revise as necessary until satisfactory
 - 2.02.04.01.07 Finalize database domain diagram section of conceptual specification
 - 2.02.04.02 Finalize database objects and database object diagram
 - 2.02.04.02.01 Select entity from database domain diagram as a potential database object
 - 2.02.04.02.02 Create name for potential database object
 - 2.02.04.02.03 Create definition for potential database object
 - 2.02.04.02.04 Determine if multiple database object tables exist for potential database object
 - 2.02.04.02.05 Record as database objects those potential database objects that have multiple database object tables
 - 2.02.04.02.06 Record as database object tables those potential database objects that do not contain multiple database object tables
 - 2.02.04.02.07 Record as data elements those potential database objects that are not even database object tables; rather they describe single business fact data element
 - 2.02.04.02.08 Review discovered database objects, database object tables and data elements with users to ensure proper business names and business definitions
 - 2.02.04.02.09 Provide an example for each database object, database object table, and data element
 - 2.02.04.02.10 Enter database objects, database object tables and data elements into metadata repository
 - 2.02.04.02.11 Present defined database objects and review as appropriate
 - 2.02.04.02.12 Create database object diagram
 - 2.02.04.02.13 Create relationships between database objects and database domains and then enter into metadata repository
- 2.02.04.03 Create generalized/abstracted/leveled database domain diagrams
 - 2.02.04.03.01 Create higher level database domain diagram if combined database domain diagram is too complex
 - 2.02.04.03.01.01 Identify and name higher level entity groups



Work Breakdown Structure

- 2.02.04.03.01.02 Identify higher level relationships between entity groups
- 2.02.04.03.01.03 Construct higher level database domain diagram
- 2.02.04.03.02 Assess level of simplicity achieved
- 2.02.04.03.03 Repeat higher level abstraction process to achieve yet higher levels as necessary
- 2.02.04.04 Narrow database diagram to project scope
- 2.02.04.04.01 Compare database domain diagram to project goals and identify entities clearly within project scope to project goals
- 2.02.04.04.02 Identify entities that are components of other automated systems and label as automated input data sources
- 2.02.04.04.03 Identify remaining entities as undetermined and seek resolution from users and management
- 2.02.04.04.04 Factor into project scope those entities that are determined by users and management to be included
- 2.02.04.04.05 Record excluded entities in "outside scope" section of project definition
- 2.02.04.04.06 Create revised database domain diagram that includes only those entities within project scope
- 2.02.04.04.07 Review and revise database domain diagrams to ensure consistency
- 2.02.04.04.08 Conduct subphase review
- 2.02.04.05 Update business and technical glossary
- 2.02.04.06 Specify data elements
- 2.02.04.06.01 Decompose each data source into data elements
- 2.02.04.06.02 Create decomposition algebra for subsequent recreation
- 2.02.04.06.03 Transform derived data into primitive elements
- 2.02.04.06.04 Transform data into best format and maintain transformed algebra
- 2.02.04.06.05 Use installation naming conventions to label data elements
- 2.02.04.06.06 Create appropriate data element value domains
- 2.02.04.06.07 Create definition, content, structure, algebra, etc.
- 2.02.04.06.08 Create data integrity statements for valid values, invalid values, etc.
- 2.02.04.06.09 Provide an example of the kind of data element that would be typified by the data element name
- 2.02.04.06.10 Define as data elements those entities that did not become database objects, if appropriate
- 2.02.04.06.11 Assess data element relevance to database domain diagram
- 2.02.04.06.12 Add relevant data elements to metadata repository
- 2.02.04.06.13 Add upper level data element metadata to metadata repository
- 2.02.04.06.13.01 Add data element concepts as necessary
- 2.02.04.06.13.02 Add value domains as appropriate



- 2.02.04.06.13.03 Add conceptual value domains as appropriate
- 2.02.04.06.13.04 Add concepts as appropriate
- 2.02.04.06.13 Present defined data elements and review as appropriate
- 2.02.04.07 Create database object table component
 - 2.02.04.07.01 Allocate data elements as columns to database objects
 - 2.02.04.07.01.01 Select database object for data element allocation
 - 2.02.04.07.01.02 Review database object's database object tables for suggestions of data elements
 - 2.02.04.07.01.03 Select data elements from metadata repository and assign to database objects as columns for database object tables
 - 2.02.04.07.01.04 Identify columns as single valued or multi valued
 - 2.02.04.07.01.05 Evaluate database object tables from database objects to determine suggested data element complete usage
 - 2.02.04.07.01.06 Search for additional data elements if database object tables are not exhausted
 - 2.02.04.07.01.07 Modify metadata repository with additionally discovered data elements
 - 2.02.04.07.01.08 Evaluate metadata repository to determine complete usage of all contained data elements
 - 2.02.04.07.01.09 Incorporate additional data elements into metadata repository if they are justified
 - 2.02.04.07.01.10 Adjust database object tables section of database object definition for addition of newly discovered data elements
 - 2.02.04.07.01.11 Assign newly discovered data elements to an database object
 - 2.02.04.07.01.12 Create localized column names where appropriate
 - 2.02.04.07.02 Create normalized database object table for each database object
 - 2.02.04.07.02.01 Examine columns for similar purpose
 - 2.02.04.07.02.02 Assign columns to database object tables of obvious name within database object
 - 2.02.04.07.02.03 Examine database object tables and split them into owner and member database object tables as required by single or multi-set status of subcollections of columns
 - 2.02.04.07.02.04 Identify the set of columns that make up a primary key for each database object table
 - 2.02.04.07.02.05 Examine database object tables for commonly employed columns
 - 2.02.04.07.02.06 Create a supertype database object table of the commonly employed columns



Work Breakdown Structure

- 2.02.04.07.02.07 Examine candidate database object tables for candidate columns that are used for only subsets of database object table rows
- 2.02.04.07.02.08 Create one or more subtype database object tables for the columns that are used for only subsets of database object table rows
- 2.02.04.07.02.09 Finalize database object table descriptions
- 2.02.04.07.02.10 Provide an example of the kind of database object table that would be typified by the database object table name
- 2.02.04.07.02.11 Create instances diagrams for each database object as a vehicle to illustrate data

- 2.02.04.07.03 Verify that database object tables are in third normal form
 - 2.02.04.07.03.01 Identify database object table primary key
 - 2.02.04.07.03.02 Check for multi-set columns, put into new or lower database object tables
 - 2.02.04.07.03.03 Check for partial primary-key dependencies, put into new database object tables
 - 2.02.04.07.03.04 Check for nonprimary key column dependencies into new database object tables
 - 2.02.04.07.03.05 Remove columns dependent on columns from other database object tables, and note the required relationship for later inclusion
 - 2.02.04.07.03.06 Reevaluate & correct stated name, function, etc. of database object as it relates to database object tables, if necessary
 - 2.02.04.07.03.07 Collect columns that were discarded for allocation to other database object tables of database object or to other database object's database object tables
 - 2.02.04.07.03.08 Revise database object table description
 - 2.02.04.07.03.09 Reassemble database object tables
 - 2.02.04.07.03.10 Create database object table graphic for each database object

- 2.02.04.07.04 Create global database object table graphic

- 2.02.04.07.05 Modify database object tables to allow for historical data
 - 2.02.04.07.05.01 Determine periodicity of data change
 - 2.02.04.07.05.02 Determine whether history is required
 - 2.02.04.07.05.03 Set forth policy if history is kept as data changes, or at specific intervals
 - 2.02.04.07.05.04 Specifically enumerate the intervals
 - 2.02.04.07.05.05 Specify rules for historical data retention if they change, i.e., daily for a month, then monthly for a year, then quarterly, etc.
 - 2.02.04.07.05.06 Specify algorithms for computing historical summaries, i.e., average, high, low, etc.
 - 2.02.04.07.05.07 Adjust segment descriptions and structures to account for historical data
 - 2.02.04.07.05.08 Revise database object table graphics



- 2.02.04.07.06 Update database object definition
 - 2.02.04.07.06.01 Check database object for proper name & purpose
 - 2.02.04.07.06.02 Revise example of the database object as necessary
 - 2.02.04.07.06.03 Review and revise metadata repository as appropriate
 - 2.02.04.07.06.04 Review and revise database object diagram as appropriate
- 2.02.04.07.07 Verify completeness of database object graphic
 - 2.02.04.07.07.01 Review complete database object table graphic against database objects to verify correctness
 - 2.02.04.07.07.02 Create global graphic of all database object tables within database objects
 - 2.02.04.07.07.03 Review global database object table graphic against database domain diagram to verify correctness
- 2.02.04.08 Specify database object table process component
 - 2.02.04.08.01 Identify data integrity requirements
 - 2.02.04.08.01.01 Specify as data integrity rules those inferred through relationship descriptions contained within and between database objects
 - 2.02.04.08.01.02 Specify as data integrity rules intra-database object table column dependencies
 - 2.02.04.08.01.03 Specify as data integrity rules the self contained row conditions that determine row acceptance into the database
 - 2.02.04.08.01.04 Specify as data integrity rules the inter database object table row conditions that determine whether a dependent database object table is accepted into the database
 - 2.02.04.08.01.05 Specify as data integrity rules inter-database object table element value conditional dependencies
 - 2.02.04.08.01.06 Specify as data integrity rules inter-database object table derived data dependencies
 - 2.02.04.08.02 Identify data integrity requirements for data transfer and data conversion
 - 2.02.04.08.02.01 Specify as data integrity rules changes in valid and invalid values for edit database object tables
 - 2.02.04.08.02.02 Specify as data integrity rules changes in range values for edit database object tables
 - 2.02.04.08.02.03 Specify as data integrity rules changes in database object table look-ups for edit database object tables
 - 2.02.04.08.02.04 Specify as data integrity rules all changes from derived external view columns to atomic database view columns
 - 2.02.04.08.02.05 Specify as data integrity rules all changes from atomic external view columns to derived database view columns



Work Breakdown Structure

- 2.02.04.08.02.06 Provide example of the data integrity rule
- 2.02.04.08.03 Specify all identified data integrity rules with a name & definition, function/role, owner & member database object table, and relational algebra
- 2.02.04.08.04 Provide an example of the data integrity rule
- 2.02.04.08.05 Record data integrity rule in metadata repository
- 2.02.04.08.06. Specify all database object table processes
 - 2.02.04.08.06.01 Create an add, delete, and modify database object table process for each database object table
 - 2.02.04.08.06.02 Create selection logic to obtain each database object table row
 - 2.02.04.08.06.03 Create appropriate messages for different types of database object table process actions
 - 2.02.04.08.06.04 Provide an example of database object table process
 - 2.02.04.08.06.05 Record database object table process data into metadata repository
- 2.02.04.08.07. Specify intra database object referential action database object table processes
 - 2.02.04.08.07.01 Identify database object table within a database object
 - 2.02.04.08.07.02 Create appropriate referential action add, delete, and modify database object table process for all database object tables within the database object
 - 2.02.04.08.07.03 Create selection logic to obtain each database object table row
 - 2.02.04.08.07.04 Create appropriate messages for different types of database object table process actions
 - 2.02.04.08.07.05 Provide an example of referential action database object table process
 - 2.02.04.08.07.06 Record referential action database object table process data into metadata repository
- 2.02.04.08.08 Specify inter database object referential action database object table processes
 - 2.02.04.08.08.01 Create database object tables that intersect database objects
 - 2.02.04.08.08.02 Identify the preeminent database object owner for the intersecting data structure segments
 - 2.02.04.08.08.02 Create appropriate referential action add, delete, and modify database object table process for all database object intersection database object tables
 - 2.02.04.08.08.03 Create selection logic to obtain each database object intersection database object table row
 - 2.02.04.08.08.04 Create appropriate messages for different types of database object table process actions
 - 2.02.04.08.08.05 Provide an example of referential action database object table process



- 2.02.04.08.08.06 Record referential action database object table process data into metadata repository
- 2.02.04.08.09 Specify all data integrity rule-based database object table processes
 - 2.02.04.08.09.01 Identify appropriate database object table for data integrity rule-based database object table processes
 - 2.02.04.08.09.02 Create appropriate add, delete, and modify database object table processes to accomplish data integrity rule
 - 2.02.04.08.09.03 Create selection logic to obtain each database object table row
 - 2.02.04.08.09.04 Create appropriate messages for different types of database object table process actions
 - 2.02.04.08.09.05 Provide an example of data integrity rule-based database object table process
 - 2.02.04.08.09.06 Record data integrity rule-based database object table process data into metadata repository
- 2.02.04.08.10 Create specified views for database object table processes
 - 2.02.04.08.10.01 Name and describe the specified view
 - 2.02.04.08.10.02 Identify appropriate specified view columns from database object table
 - 2.02.04.08.10.03 Identify appropriate specified view navigation logic to navigate database objects
 - 2.02.04.08.10.04 Identify appropriate specified view select clauses to obtain database object table rows
 - 2.02.04.08.10.05 Create appropriate messages for different types of view actions
 - 2.02.04.08.10.06 Record specified view definition into metadata repository
- 2.02.04.08.11 Create additional required database object table process model data integrity rules
 - 2.02.04.08.11.01 Specify any additional data integrity rules that are inferred through relationship descriptions contained in database object tables
 - 2.02.04.08.11.02 Specify any additional data integrity rules that are intra-database object table column dependencies
 - 2.02.04.08.11.03 Specify any additional data integrity rules that are self contained record conditions that determine database object table row acceptance into the database
 - 2.02.04.08.11.04 Specify any additional data integrity rules that are inter database object table row conditions that determine whether a dependent database object table row is accepted into the database
 - 2.02.04.08.11.05 Specify any additional data integrity rules that are inter-database object table element value conditional dependencies
 - 2.02.04.08.11.06 Specify any additional data integrity rules that are inter-database object table derived data dependencies



Work Breakdown Structure

- 2.02.04.08.11.07 Specify all identified data integrity rules with a name & definition, function/role, owner & member database object table , and relational algebra
- 2.02.04.08.11.08 Provide an example of the data integrity rule
- 2.02.04.08.11.09 Record data integrity rule in metadata repository
- 2.02.04.08.11.10 Identify appropriate database object table for data integrity rule-based primitive database object table processes
- 2.02.04.08.11.11 Create appropriate add, delete, and modify primitive database object table processes to accomplish data integrity rule
- 2.02.04.08.11.12 Create selection logic to obtain each database object table row
- 2.02.04.08.11.13 Create appropriate messages for different types of database object table process actions
- 2.02.04.08.11.14 Provide an example of data integrity rule-based primitive database object table process
- 2.02.04.08.11.15 Record data integrity rule-based primitive database object table process data into metadata repository

- 2.02.04.08.12 Validate database object table process model
- 2.02.04.08.12.01 Evaluate each database object table process for clear, unambiguous meaning
- 2.02.04.08.12.02 Evaluate specified views for appropriate construction
- 2.02.04.08.12.03 Evaluate database object table processes for conflicting updates
- 2.02.04.08.12.04 Evaluate all data integrity rules by information system process to ensure that no rules are in conflict.

- 2.02.04.08.13 Create database object table process model deliverable
- 2.02.04.08.13.01 Create database object table process model narrative
- 2.02.04.08.13.02 Finalize all database object table process model information system process flow diagrams

- 2.02.04.08.14 Conduct subphase review

- 2.02.04.09 Create database object information system component

- 2.02.04.09.01 Create detailed database object information system and diagrams
- 2.02.04.09.01.01 Identify each database object information system& define its purpose, scope, etc., in terms of database object transformation
- 2.02.04.09.01.02 Record each database object information system descriptively and place appropriately into a database object information system diagram
- 2.02.04.09.01.03 Identify specific subordinate database object information systems that are included in database object information system



- 2.02.04.09.01.04 Specify rules that governs database object information system success within database object information system process flow diagram
- 2.02.04.09.01.05 Specify required reactions as a consequence of database object information system failure
- 2.02.04.09.01.06 Provide an example of each database object information system flow diagram
- 2.02.04.09.01.07 If database object information system contains subordinate database object information systems then repeat until unnecessary
- 2.02.04.09.02 Estimate implementation phase efforts for each database object information system
- 2.02.04.09.03 Create documentation for each database object information system
 - 2.02.04.09.03.01 Create database object information system narrative
 - 2.02.04.09.03.02 Create summary of design requirements
- 2.02.04.09.04 Conduct subphase review
- 2.02.04.10 Specify database object state component
 - 2.02.04.10.01 Identify appropriate database lifecycle resources
 - 2.02.04.10.01.01 For each database life cycle resource
 - 2.02.04.10.01.02 Identify the life cycle nodes
 - 2.02.04.10.01.03 For each life cycle node
 - 2.02.04.10.01.03.01 Identify the database objects that are modified by the life cycle node
 - 2.02.04.10.01.03.02 For each identified database object
 - 2.02.04.10.01.03.02.01 Identify the database object tables that must be added/deleted/or modified to accomplish the business policy of the node
 - 2.02.04.10.01.03.02.02 Name the end result of the database object instance transformation as the database object state
 - 2.02.04.10.01.03.02.03 Identify the database object information systems that must be executed to accomplish the database object transformation
 - 2.02.04.10.01.03.02.03 Store the database object state and the algebra for accomplishing the database object state transformation through database object information systems into the metabase
- 2.02.04.10.02 Identify the database life resources
- 2.02.05 Create or modify specified data model as appropriate
 - 2.02.05.01 Examine every database object and set of database object tables for generalization
 - 2.02.05.02 As necessary, create generalized subject or subordinate subject within specified data model
 - 2.02.05.03 Name, describe, and enter subject within specified data model



Work Breakdown Structure

- 2.02.05.04 Determine if database object tables already exist as specified data model entities.
- 2.02.05.05 If existing, create relationships between database object table columns and specified data model entity attributes
- 2.02.05.06 If not existing, create generalized entities from database object tables
- 2.02.05.07 Create attributes within newly created entities
- 2.02.05.08 Create relationship between newly created attributes and database object table columns
- 2.02.05.09 Create relationship between newly created attributes and data elements
- 2.02.05.010 Create meta category value assignments as appropriate
- 2.02.05.011 Create value domain assignments as appropriate

- 2.02.06 Create retail data warehouse database design
 - 2.02.06.01 Perform data requirements analysis
 - 2.02.06.01.01 Identify types of reports that are to be produced by retail data warehouse
 - 2.02.06.01.02 Collect descriptions of data from various forms, prior reports, and user interviews
 - 2.02.06.01.03 Analyze each data source and report to uncover the source of data and its required derivations
 - 2.02.06.01.04 Record report descriptions and characteristics onto report form and enter into metadata repository
 - 2.02.06.01.05 Construct a data analysis report, and review & revise with users as necessary
 - 2.02.06.02 Define retail data warehouse data integrity model
 - 2.02.06.02.01 Identify database objects from database object diagram that are appropriate for participation in the retail data warehouse database
 - 2.02.06.02.02 Identify/create normalized table structure from each database object appropriate for retail data warehouse
 - 2.02.06.02.03 Finalize retail data warehouse table descriptions
 - 2.02.06.02.04 Provide an example of the kind of retail data warehouse table that would be typified by the table name
 - 2.02.06.02.05 Verify that retail data warehouse tables support sufficient history to satisfy reports
 - 2.02.06.02.06 Verify that retail data warehouse tables contain sufficient derived data to support summary level reports
 - 2.02.06.03 Identify data integrity requirements for data transfer and data conversion
 - 2.02.06.03.01 Specify as data integrity rules changes in valid and invalid values for edit tables
 - 2.02.06.03.02 Specify as data integrity rules changes in range values for edit tables
 - 2.02.06.03.03 Specify as data integrity rules changes in table look-ups for edit tables
 - 2.02.06.03.04 Specify as data integrity rules all changes from derived external view columns to atomic database view columns



- 2.02.06.03.05 Specify as data integrity rules all changes from atomic external view columns to derived database view columns
- 2.02.06.03.06 Provide example of the data integrity rule
- 2.02.06.03.07 Specify all identified data integrity rules with a name & definition, function/role, owner & member table, and relational algebra
- 2.02.06.03.08 Provide an example of the data integrity rule
- 2.02.06.03.09 Record data integrity rule in metadata repository

- 2.02.06.04 Perform analysis to support data transfer and conversion from source databases to retail data warehouse databases
 - 2.02.06.04.01 Gather data transfer and data conversion information such as necessary data value transformations and derivations
 - 2.02.06.04.02 Construct data transfer and data conversion report, and review and revise with users as necessary
 - 2.02.06.04.03 Determine the type of conversion best suited for the operating environment
 - 2.02.06.04.04 Prepare a data analysis/data source report
 - 2.02.06.04.05 Prepare detailed conversion plan
 - 2.02.06.04.06 Present data analysis report and detail conversion plan to users

- 2.02.06.05 Verify retail data warehouse data integrity model
 - 2.02.06.05.01 Review retail data warehouse tables for well defined quality
 - 2.02.06.05.01.01 Examine retail data warehouse table for well defined function
 - 2.02.06.05.01.02 Examine row for unique occurrences
 - 2.02.06.05.01.03 Evaluate appropriateness of assigned columns
 - 2.02.06.05.01.04 Evaluate retail data warehouse table uniqueness
 - 2.02.06.05.01.05 Evaluate reliability of element values over all retail data warehouse table instances
 - 2.02.06.05.01.06 Evaluate adequacy of historical and derived data requirements
 - 2.02.06.05.01.07 Evaluate overall retail data warehouse table quality
 - 2.02.06.05.02 Review data transfer and conversion components for well defined quality
 - 2.02.06.05.02.01 Review system, file, file element, external view column, database view column, and view for clear, unambiguous definition and name
 - 2.02.06.05.02.02 Review external view column and database view column for correct data editing and validation clause
 - 2.02.06.05.03 Review data integrity rules for well defined quality
 - 2.02.06.05.03.01 Evaluate data integrity rules for clear, unambiguous meaning & function
 - 2.02.06.05.03.02 Evaluate data integrity rule relational algebra for unique owner & distinct member selection



Work Breakdown Structure

- 2.02.06.05.03.03 Evaluate intersection tables for sufficiency & adequacy
- 2.02.06.05.04 Review data integrity model for well defined quality
 - 2.02.06.05.04.01 Review & revise database domain diagram and relationship annotations
 - 2.02.06.05.04.02 Review & revise tables versus elements for clarity and consistency
 - 2.02.06.05.04.03 Review & revise tables versus data integrity rules for clarity & consistency
 - 2.02.06.05.04.04 Review & revise elements versus data integrity rules for clarity & consistency
 - 2.02.06.05.04.05 Review and revise table graphic for clarity & consistency
- 2.02.06.05.05 Review data integrity model against business database objectives and measurements of project success to determine if revisions are necessary
- 2.02.06.06 Create retail data warehouse logical database specification report
 - 2.02.06.06.01 Compile database object section
 - 2.02.06.06.02 Compile table section
 - 2.02.06.06.03 Compile data integrity rule section
 - 2.02.06.06.04 Review & revise retail data warehouse logical database section of specifications
- 2.02.06.07 Create retail data warehouse logical database presentation
 - 2.02.06.07.01 Modify features, advantages and benefits (FAB) of database system
 - 2.02.06.07.02 Include FABs to create logical database use scenarios
 - 2.02.06.07.03 Graphically identify database objects/processes that participate in scenario
 - 2.02.06.07.04 Create narrative for logical database use scenario
 - 2.02.06.07.05 Identify key individual for logical database use scenario
 - 2.02.06.07.06 Review logical database use scenario & narrative with key individual and modify as appropriate
 - 2.02.06.07.07 Create A/V materials for logical database use scenario presentation
 - 2.02.06.07.08 Review and finalize A/V materials prior to review
- 2.02.06.08 Conduct subphase review
 - 2.02.06.08.01 Present logical database
 - 2.02.06.08.02 Receive and respond to comments
 - 2.02.06.08.03 Modify system as necessary
 - 2.02.06.08.04 Seek concurrence with system subphase
- 2.02.06 Create logical database specification report
 - 2.02.07.01 Compile mission description section
 - 2.02.07.02 Compile database mission description diagram section
 - 2.02.07.03 Compile database domain section



- 2.02.07.04 Compile database domain diagram section
- 2.02.07.04 Compile data integrity model
- 2.02.07.04.01 Compile database object section
- 2.02.07.04.02 Compile data element section
- 2.02.07.04.03 Compile data integrity rule section
- 2.02.07.05 Review & revise logical database section of conceptual specifications
- 2.02.08 Create logical database presentation
- 2.02.08.01 Modify features, advantages and benefits (FAB) of database system
- 2.02.08.02 Include FABs to create logical database use scenarios
- 2.02.08.03 Graphically identify database objects/processes that participate in scenario
- 2.02.08.04 Create narrative for logical database use scenario
- 2.02.08.05 Identify key individual for logical database use scenario
- 2.02.08.06 Review logical database use scenario & narrative with key individual and modify as appropriate
- 2.02.08.07 Create A/V materials for logical database use scenario presentation
- 2.02.08.08 Review and finalize A/V materials prior to review
- 2.02.09 Conduct subphase review
- 2.02.09.01 Present logical database
- 2.02.09.02 Receive and respond to comments
- 2.02.09.03 Modify system as necessary
- 2.02.09.04 Seek concurrence with system subphase
- 2.03 Specify physical database of business requirements
- 2.03.01 Specify business information systems model
- 2.03.01.01 Identify business information systems appropriate for subordinate mission description
- 2.03.01.02 Create online data update system specification
- 2.03.01.02.01 Create detailed information system processes and diagrams for online data update system
- 2.03.01.02.01.01 Identify each user-required information system process & define its purpose, scope, etc., in terms of business activities
- 2.03.01.02.01.02 Record each information system process descriptively and place appropriately into a information system process diagram
- 2.03.01.02.01.03 Review the business environment & activities in which information system process occurs to ensure the information system process's sufficiency



Work Breakdown Structure

- 2.03.01.02.01.04 Identify and record organizational responsibility for information system process
- 2.03.01.02.01.05 Identify frequency of information system process
- 2.03.01.02.01.06 Identify specific subordinate information system processes that are included in information system process
- 2.03.01.02.01.07 Specify rules that governs information system process success within information system process flow diagram
- 2.03.01.02.01.08 Specify required reactions as a consequence of information system process failure
- 2.03.01.02.01.09 Provide an example of each information system process flow diagram
- 2.03.01.02.01.10 If information system process contains subordinate information system processes then repeat until unnecessary

- 2.03.01.02.02 Create screen specification
- 2.03.01.02.02.01 Name, describe, and sketch the screen, and identify the information system process that invokes the screen
- 2.03.01.02.02.02 Review the screen for simplicity and single-purpose. If complex, then divide the screen into subordinate single-purpose screens.
- 2.03.01.02.02.03 Identify appropriate screen elements
- 2.03.01.02.02.04 Use where ever possible the list of columns, but if necessary add new columns element and its new definition to the list of columns
- 2.03.01.02.02.05 Specify the help message that is to be present for each screen element
- 2.03.01.02.02.06 Specify the valid and invalid value data integrity rules for each screen element
- 2.03.01.02.02.07 Specify the legal and illegal range value data integrity rules for each screen element
- 2.03.01.02.02.08 Specify the database object table look up data integrity rules for each screen element
- 2.03.01.02.02.09 Specify which screen elements must be valued for legal entry
- 2.03.01.02.02.10 Identify which screen elements should be protected from update
- 2.03.01.02.02.11 Identify any internal processes that must be accomplished for each screen, and/or screen element
- 2.03.01.02.02.12 Specify the required function keys and the actions that are to be taken for each
- 2.03.01.02.02.13 Identify appropriate specified navigation logic among screens if different than that implied by the information system process diagram
- 2.03.01.02.02.14 Connect all input screens to appropriate external views with information system control logic units
- 2.03.01.02.02.15 Connect all input screen specified views to appropriate information system control logic units



- 2.03.01.02.03 Design file operations
 - 2.03.01.02.03.01 Identify file that must be read or written
 - 2.03.01.02.03.01 Identify appropriate external view with information system control logic unit
 - 2.03.01.02.03.03 Identify any internal processes that must be accomplished for each file cell or file record
- 2.03.01.02.04 Create internal processes
 - 2.03.01.02.04.01 Identify and name internal process
 - 2.03.01.02.04.02 Describe internal process
 - 2.03.01.02.04.03 Create external view with information system control logic unit, or with screen, or with screen element, or with file operation, or with file cell
 - 2.03.01.02.04.04 Create data transformation algorithm
 - 2.03.01.02.04.05 Record internal process into metadata repository along with relationship metadata
- 2.03.01.02.05 Estimate implementation phase efforts for each online data update subsystem
- 2.03.01.02.06 Review and revise human support subsystems affected by new on-line data update subsystems
 - 2.03.01.02.06.01 Identify each human support subsystem
 - 2.03.01.02.06.02 Create process diagram to determine all manual processes
 - 2.03.01.02.06.03 Optimize manual processes to take advantage of new tools
 - 2.03.01.02.06.04 Identify new training requirements
 - 2.03.01.02.06.05 Identify any necessary job description revisions
 - 2.03.01.02.06.06 Formulate plan to implement changes to human support subsystems
- 2.03.01.02.07 Create documentation for each online data update subsystem
 - 2.03.01.02.07.01 Create online data update subsystem narrative
 - 2.03.01.02.07.02 Create summary of design requirements
- 2.03.01.02.08 Conduct subphase review
- 2.03.01.03 Create batch data update system specification
 - 2.03.01.03.01 Create detailed information system processes and diagrams for batch data update system
 - 2.03.01.03.01.01 Identify each user-required information system process & define its purpose, scope, etc., in terms of business activities
 - 2.03.01.03.01.02 Record each information system process descriptively and place appropriately into a information system process diagram



Work Breakdown Structure

- 2.03.01.03.01.03 Review the business environment & activities in which information system process occurs to ensure the information system process's sufficiency
- 2.03.01.03.01.04 Identify and record organizational responsibility for information system process
- 2.03.01.03.01.05 Identify frequency of information system process
- 2.03.01.03.01.06 Identify specific subordinate information system processes that are included in information system process
- 2.03.01.03.01.07 Specify rules that governs information system process success within information system process flow diagram
- 2.03.01.03.01.08 Specify required reactions as a consequence of information system process failure
- 2.03.01.03.01.09 Provide an example of each information system process flow diagram
- 2.03.01.03.01.10 If information system process contains subordinate information system processes then repeat until unnecessary
- 2.03.01.03.01.11 Connect all information system processes to appropriate specified views
- 2.03.01.03.01.12 Connect all information system processes specified views to appropriate database object table processes
- 2.03.01.03.02 Design file operations
 - 2.03.01.03.02.01 Identify file that must be read or written
 - 2.03.01.03.02.02 Identify appropriate external view with information system control logic unit
 - 2.03.01.03.02.03 Identify any internal processes that must be accomplished for each file cell or file record
- 2.03.01.03.03 Create internal processes
 - 2.03.01.03.03.01 Identify and name internal process
 - 2.03.01.03.03.02 Describe internal process
 - 2.03.01.03.03.03 Create external view with information system control logic unit, or with screen, or with screen element, or with file operation, or with file cell
 - 2.03.01.03.03.04 Create data transformation algorithm
 - 2.03.01.03.03.05 Record internal process into metadata repository along with relationship metadata
- 2.03.01.03.04 Estimate implementation phase efforts for each batch data update subsystem
- 2.03.01.03.05 Review and revise human support subsystems affected by new batch data update subsystems
 - 2.03.01.03.05.01 Identify each human support subsystem
 - 2.03.01.03.05.02 Create process diagram to determine all manual processes



- 2.03.01.03.05.03 Optimize manual processes to take advantage of new tools
- 2.03.01.03.05.04 Identify new training requirements
- 2.03.01.03.05.05 Identify any necessary job description revisions
- 2.03.01.03.05.06 Formulate plan to implement changes to human support subsystems

- 2.03.01.03.06 Create documentation for each batch data update subsystem
 - 2.03.01.03.06.01 Create batch data update subsystem narrative
 - 2.03.01.03.06.02 Create summary of batch data update subsystem design requirements

- 2.03.01.03.07 Conduct subphase review

- 2.03.01.04 Create data loading subsystem specification
 - 2.03.01.04.01 Identify, name, and then create information system process diagram for each data load subsystem that illustrates any load sequence requirements

 - 2.03.01.04.02 Evaluate initial data source quality for each data load subsystem
 - 2.03.01.04.02.01 Identify data source for each element
 - 2.03.01.04.02.02 Identify source data format
 - 2.03.01.04.02.03 Identify difference between database & source data formats, editing, etc.
 - 2.03.01.04.02.04 Sample source data to evaluate quality
 - 2.03.01.04.02.05 Evaluate differences among multiple sources of same data
 - 2.03.01.04.02.06 Devise strategy to resolve data source differences
 - 2.03.01.04.02.07 Estimate resources required to bring source data into conformance with database requirements
 - 2.03.01.04.02.08 Prepare source data evaluation report

 - 2.03.01.04.03 Review and revise human support subsystems affected by new data load subsystems
 - 2.03.01.04.03.01 Identify each human support subsystem
 - 2.03.01.04.03.02 Create information system process diagram to determine all data processes
 - 2.03.01.04.03.03 Optimize processes to take advantage of new tools
 - 2.03.01.04.03.05 Identify new training requirements
 - 2.03.01.04.03.05 Identify any necessary job description revisions
 - 2.03.01.04.03.06 Formulate plan to implement changes to human support subsystems

 - 2.03.01.04.04 Create specifications for each load subsystem
 - 2.03.01.04.04.01 Identify size & number of records for each database object table
 - 2.03.01.04.04.02 Create a logical record-type loading sequence chart
 - 2.03.01.04.04.03 Specify editing & validation for each database object table
 - 2.03.01.04.04.04 Specify inter-record editing & validation
 - 2.03.01.04.04.05 Determine error alternatives for all editing and validation



Work Breakdown Structure

- 2.03.01.04.04.06 Connect all data loading file definitions to appropriate specified views
- 2.03.01.04.04.07 Connect all data loading file specified views to appropriate database object table processes
- 2.03.01.04.04.08 Estimate implementation phase efforts for each data loading subsystem
- 2.03.01.04.04.09 Determine required backup during database loading

- 2.03.01.04.05 Create screens
 - 2.03.01.04.05.01 Name, describe, and sketch the screen, and identify the information system process that invokes the screen
 - 2.03.01.04.05.02 Review the screen for simplicity and single-purpose. If complex, then divide the screen into subordinate single-purpose screens.
 - 2.03.01.04.05.03 Identify appropriate screen elements
 - 2.03.01.04.05.04 Use where ever possible the list of columns, but if necessary add new columns and its new definition to the list of columns
 - 2.03.01.04.05.05 Specify the help message that is to be present for each screen element
 - 2.03.01.04.05.06 Specify the valid and invalid value Data integrity rules for each screen element
 - 2.03.01.04.05.07 Specify the legal and illegal range value Data integrity rules for each screen element
 - 2.03.01.04.05.08 Specify the database object table look up Data integrity rules for each screen element
 - 2.03.01.04.05.09 Specify which screen elements must be valued for legal entry
 - 2.03.01.04.05.10 Identify which screen elements should be protected from update
 - 2.03.01.04.05.11 Identify any internal processes that must be accomplished for each screen, and/or screen element
 - 2.03.01.04.05.12 Specify the required function keys and the actions that are to be taken for each
 - 2.03.01.04.05.13 Identify appropriate specified navigation logic among screens if different than that implied by the information system process diagram
 - 2.03.01.04.05.14 Connect all input screens to appropriate external views with information system control logic units
 - 2.03.01.04.05.15 Connect all input screen specified views to appropriate information system control logic units

- 2.03.01.04.06 Design file operations
 - 2.03.01.04.06.01 Identify file that must be read or written
 - 2.03.01.04.06.02 Identify appropriate external view with information system control logic unit
 - 2.03.01.04.06.03 Identify any internal processes that must be accomplished for each file cell or file record



- 2.03.01.04.07 Create internal processes
 - 2.03.01.04.07.01 Identify and name internal process
 - 2.03.01.04.07.02 Describe internal process
 - 2.03.01.04.07.03 Create external view with information system control logic unit, or with screen, or with screen element, or with file operation, or with file cell
 - 2.03.01.04.07.04 Create data transformation algorithm
 - 2.03.01.04.07.05 Record internal process into metadata repository along with relationship metadata
- 2.03.01.04.08 Estimate implementation phase efforts for overall load subsystem
- 2.03.01.04.09 Prepare documentation for each load subsystem
- 2.03.01.04.10 Conduct subphase review
- 2.03.01.05 Create specification for each data transfer and data conversion subsystem
 - 2.03.01.05.01 Review and revise information system process diagram for each data conversion subsystem that illustrates any data conversion sequence requirements
 - 2.03.01.05.02 Optimize processes to take advantage of new tools
 - 2.03.01.05.03 Identify size and number of records for each database object table
 - 2.03.01.05.04 Create a logical record-type data conversion sequence chart
 - 2.03.01.05.05 Specify editing & validation for each database object table
 - 2.03.01.05.06 Specify data integrity rules
 - 2.03.01.05.07 Determine error alternatives for all editing and validation
 - 2.03.01.05.08 Estimate implementation phase efforts for overall data conversion subsystem
 - 2.03.01.05.09 Determine required back-up during database data conversion
 - 2.03.01.05.10 Prepare documentation for each data conversion subsystem
 - 2.03.01.05.11 Conduct subphase review
- 2.03.01.06 Create enterprise-wide data collection system specification
 - 2.03.01.06.01 Identify corporate forms for which data collection systems are to be created
 - 2.03.01.06.02 Analyze each corporate form and design specific information system to collect data
 - 2.03.01.06.02.01 Identify each user-required information system process & define its purpose, scope, etc., in terms of business activities
 - 2.03.01.06.02.02 Record each information system process descriptively and place appropriately into a information system process diagram
 - 2.03.01.06.02.03 Review the business environment & activities in which information system process occurs to ensure the information system process's sufficiency



Work Breakdown Structure

- 2.03.01.06.02.04 Identify and record organizational responsibility for information system process
- 2.03.01.06.02.05 Identify frequency of information system process
- 2.03.01.06.02.06 Identify specific subordinate information system processes that are included in information system process
- 2.03.01.06.02.07 Specify rules that governs information system process success within information system process flow diagram
- 2.03.01.06.02.08 Specify required reactions as a consequence of information system process failure
- 2.03.01.06.02.09 Provide an example of each information system process flow diagram
- 2.03.01.06.02.10 If information system process contains subordinate information system processes then repeat until unnecessary

- 2.03.01.06.03 Create data collection screen specification
- 2.03.01.06.03.01 Name, describe, and sketch the screen, and identify the information system process that invokes the screen
- 2.03.01.06.03.02 Review the screen for simplicity and single-purpose. If complex, then divide the screen into subordinate single-purpose screens.
- 2.03.01.06.03.03 Identify appropriate screen elements
- 2.03.01.06.03.04 Use where ever possible the list of columns, but if necessary add new columns and its new definition to the list of columns
- 2.03.01.06.03.05 Specify the help message that is to be present for each screen element
- 2.03.01.06.03.06 Specify the valid and invalid value Data integrity rules for each screen element
- 2.03.01.06.03.07 Specify the legal and illegal range value Data integrity rules for each screen element
- 2.03.01.06.03.08 Specify the database object table look up Data integrity rules for each screen element
- 2.03.01.06.03.09 Specify which screen elements must be valued for legal entry
- 2.03.01.06.03.10 Identify which screen elements should be protected from update
- 2.03.01.06.03.11 Identify any internal processes that must be accomplished for each screen, and/or screen element
- 2.03.01.06.03.12 Specify the required function keys and the actions that are to be taken for each
- 2.03.01.06.03.13 Identify appropriate specified navigation logic among screens if different than that implied by the information system process diagram
- 2.03.01.06.03.14 Connect all input screens to appropriate external views with information system control logic units
- 2.03.01.06.03.15 Connect all input screen specified views to appropriate information system control logic units



- 2.03.01.06.04 Create internal processes
 - 2.03.01.06.04.01 Identify and name internal process
 - 2.03.01.06.04.02 Describe internal process
 - 2.03.01.06.04.03 Create external view with information system control logic unit, or with screen, or with screen element, or with file operation, or with file cell
 - 2.03.01.06.04.04 Create data transformation algorithm
 - 2.03.01.06.04.05 Record internal process into metadata repository along with relationship metadata
- 2.03.01.06.05 Create normalized database object table for each data collection form
 - 2.03.01.06.05.01 Examine candidate columns for similar purpose
 - 2.03.01.06.05.02 Assign candidate columns to candidate database object tables of obvious name within data collection form
 - 2.03.01.06.05.03 Examine candidate database object tables and split into owner and member database object tables as required by single or multi-set status of subcollections of candidate columns
 - 2.03.01.06.05.04 Identify the set of columns that make up a primary key for each database object table
 - 2.03.01.06.05.05 Examine candidate database object tables for commonly employed candidate columns
 - 2.03.01.06.05.06 Create a supertype database object table of the commonly employed candidate columns
 - 2.03.01.06.05.07 Examine candidate database object tables for candidate columns that are used for only subsets of database object table rows
 - 2.03.01.06.05.08 Create one or more subtype database object tables for the candidate columns that are used for only subsets of database object table rows
 - 2.03.01.06.05.09 Finalize database object table descriptions
 - 2.03.01.06.05.10 Provide an example of the kind of database object table that would be typified by the database object table name
- 2.03.01.06.06 Estimate implementation phase efforts for each data collection system
- 2.03.01.06.07 Review and revise human support subsystems affected by new data collection systems
 - 2.03.01.06.07.01 Identify each human support subsystem
 - 2.03.01.06.07.02 Create process diagram to determine all manual processes
 - 2.03.01.06.07.03 Optimize manual processes to take advantage of new data collection system
 - 2.03.01.06.07.04 Identify new training requirements
 - 2.03.01.06.07.05 Identify any necessary job description revisions
 - 2.03.01.06.07.06 Formulate plan to implement changes to human support subsystems



Work Breakdown Structure

- 2.03.01.06.08 Create documentation for each data collection subsystem
 - 2.03.01.06.08.01 Create online data collection subsystem narrative
 - 2.03.01.06.08.02 Create summary of design requirements
- 2.03.01.06.09 Conduct subphase review
- 2.03.02 Create business event model
 - 2.03.02.01 Identify and describe business events that are business information system triggers
 - 2.03.02.02 Identify the business information system that are to be performed to adequately respond to the business event
 - 2.03.02.03 Provide an example of each business event
 - 2.03.02.04 Record business event and interconnect with business information systems into metadata repository
- 2.03.03 Create detailed business function model
 - 2.03.03.01 Select high-level business function for detailing
 - 2.03.03.02 Select organization(s) relevant for business function detailing
 - 2.03.03.03 Identify appropriate next levels of business functions that are to be accomplished by the specific organization(s)
 - 2.03.03.04 Identify business forms involved with business function
 - 2.03.03.05 Identify document involved with business function
 - 2.03.03.06 Record business form and/or document details and inter-relationships with business function into metadata repository
 - 2.03.03.07 If business function contains business functions then repeat until business event boundary
 - 2.03.03.08 Identify the business events required to trigger information systems
 - 2.03.03.09 Provide an example of each business function
 - 2.03.03.10 Record business function hierarchy and interconnect with business events into metadata repository
- 2.03.04 Create Organization Model
 - 2.03.04.01 Identify organizations involved in various missions
 - 2.03.04.02 Identify subordinate organizations as necessary and enter all organization units into metadata repository
 - 2.03.04.03 Identify the business functions that are performed by the organization
 - 2.03.04.04 Identify the role the organization plays with respect to the business function
 - 2.03.04.05 Record business function role relationship between organizational units and business functions
 - 2.03.04.06 Identify the quantity of staff hours and quantity of staff required to perform the identified function



- 2.03.05 Estimate database size
 - 2.03.05.01 Estimate number of rows per table
 - 2.03.05.02 Estimate overhead per table
 - 2.03.05.03 Estimate overall database size
 - 2.03.05.04 Record database size in conceptual specifications
- 2.03.06 Evaluate requirements for backup
 - 2.03.06.01 Determine media for backup
 - 2.03.06.02 Determine method for backup
 - 2.03.06.03 Estimate resources to perform backup
 - 2.03.06.04 Scope procedures for accomplishing backup
 - 2.03.06.05 Estimate elapsed time for backup
 - 2.03.06.06 Prepare backup conceptual specification documentation
- 2.03.07 Specify database integrity subsystem
 - 2.03.07.01 Create a information system process diagram that measures database structural integrity
 - 2.03.07.02 Name and define each information system process
 - 2.03.07.03 Determine the actions that are to be taken when an integrity test is passed
 - 2.03.07.04 Determine the actions that are to be taken when an integrity test is failed
 - 2.03.07.05 Determine how to compute a measure of database integrity
 - 2.03.07.06 Determine what and how to report database integrity failures
 - 2.03.07.07 Identify changes required in the logical database to accommodate integrity subsystem
 - 2.03.07.08 Identify changes required in the physical database to accommodate integrity subsystem
 - 2.03.07.09 Create data integrity subsystem views as appropriate
 - 2.03.07.10 Connect data integrity subsystem views to appropriate information system processes, database object table processes, and Data integrity rules
 - 2.03.07.11 Create a minispecification for each information system process
 - 2.03.07.12 Estimate implementation phase efforts
 - 2.03.07.13 Create data integrity subsystem narrative
 - 2.03.07.14 Create summary of implementation phase requirements
 - 2.03.07.15 Create data integrity conceptual specification documentation
 - 2.03.07.16 Conduct subphase review
- 2.03.08 Consolidate logical database changes to accommodate physical database
- 2.03.09 Prepare physical database requirements specification
 - 2.03.09.01 Review & revise database data transformation model
 - 2.03.09.02 Review & revise each database update subsystem specification



Work Breakdown Structure

- 2.03.09.03 Review & revise each database loading subsystem specification
- 2.03.09.04 Review & revise database size estimate
- 2.03.09.05 Review & revise source data quality report
- 2.03.09.06 Review & revise backup requirements
- 2.03.09.07 Review & revise data integrity subsystem specification
- 2.03.09.08 Review physical database requirements section against business database objectives and measurements of project success to determine if revisions are necessary
- 2.03.09.09 Compile physical database conceptual specification report

- 2.03.10 Create physical database presentation
 - 2.03.10.01 Modify features, advantages and benefits (FAB) of database system
 - 2.03.10.02 Include FABs to create physical database subsystem scenarios
 - 2.03.10.03 Graphically identify tables or processes that participate in scenario
 - 2.03.10.04 Create narrative for scenario
 - 2.03.10.05 Identify key individual for scenario
 - 2.03.10.06 Review scenario & narrative with key individual and modify as appropriate
 - 2.03.10.07 Create A/V materials for scenario presentation
 - 2.03.10.08 Review and finalize A/V materials prior to review

- 2.03.11 Conduct subphase review
 - 2.03.11.01 Present physical database
 - 2.03.11.02 Receive and respond to comments
 - 2.03.11.03 Modify system as necessary
 - 2.03.11.04 Seek concurrence with system subphase

- 2.04 Analyze interrogation requirements
 - 2.04.01 Identify, name and then create information system process diagram for each specialized report subsystem

 - 2.04.02 Analyze standard report requirements
 - 2.04.02.01 Identify generic report formats
 - 2.04.02.02 Identify tables, elements & relationships
 - 2.04.02.03 Identify table selection & sorting
 - 2.04.02.04 Identify database design changes that are necessary to accommodate report
 - 2.04.02.05 Determine computations & transformation requirements
 - 2.04.02.06 Use useful decomposition algebra previously specified
 - 2.04.02.07 Estimate total number of reports
 - 2.04.02.08 Estimate total implementation phase efforts
 - 2.04.02.09 Estimate total processing
 - 2.04.02.10 Allocate processing over monthly cycles



- 2.04.02.11 Validate against data availability matrix
- 2.04.02.12 Create standard report requirements section

- 2.04.03 Analyze ad-hoc report requirements
 - 2.04.03.01 Identify generic report formats
 - 2.04.03.02 Identify database tables, elements, & relationships
 - 2.04.03.03 Identify selection & sorting
 - 2.04.03.04 Identify database design changes that are necessary to accommodate report
 - 2.04.03.05 Determine computations & transformation requirements
 - 2.04.03.06 Use useful decomposition algebra previously specified
 - 2.04.03.07 Estimate life span of ad-hoc reports
 - 2.04.03.08 estimate quantity of ad-hoc reports
 - 2.04.03.09 Estimate frequency of report execution
 - 2.04.03.10 Estimate total processing
 - 2.04.03.11 Allocate processing over monthly cycles
 - 2.04.03.12 Validate against data availability matrix
 - 2.04.03.13 Create ad-hoc report requirements section

- 2.04.04 Prepare interrogation requirements report
 - 2.04.04.01 Review & revise standard reports requirements
 - 2.04.04.02 Review & revise ad-hoc report requirements
 - 2.04.04.03 Consolidate database design change requirements to accommodate interrogation
 - 2.04.04.04 Prepare consolidated interrogation requirements report

- 2.04.05 Analyze data transfer and data conversion reporting requirements
 - 2.04.05.01 Identify generic report formats
 - 2.04.05.02 Identify tables, elements and relationships
 - 2.04.05.03 Determine computations and transformation requirements
 - 2.04.05.04 Estimate total number of reports
 - 2.04.05.05 Estimate total implementation phase efforts
 - 2.04.05.06 Estimate total processing

- 2.04.06 Create interrogation presentation
 - 2.04.06.01 Modify features, advantages and benefits (FAB) of database system
 - 2.04.06.02 Include FABs to create interrogation use scenarios
 - 2.04.06.03 Graphically identify tables or processes that participate in scenario
 - 2.04.06.04 Create narrative for interrogation use scenario
 - 2.04.06.05 Identify key individual for interrogation use scenario
 - 2.04.06.06 Review interrogation use scenario & narrative with key individual and modify as appropriate
 - 2.04.06.07 Create A/V materials for interrogation use scenario presentation



Work Breakdown Structure

- 2.04.06.08 Review and finalize A/V materials prior to review
- 2.04.07 Conduct subphase review
 - 2.04.07.01 Present interrogation
 - 2.04.07.02 Receive and respond to comments
 - 2.04.07.03 Modify system as necessary
 - 2.04.07.04 Seek concurrence with system subphase
- 2.05 Specify system control requirements
 - 2.05.01 Specify audit trail needs
 - 2.05.01.01 Identify audit trail needs for change control
 - 2.05.01.02 Identify audit trail needs for security tracking
 - 2.05.01.03 Identify audit trail needs to satisfy legal requirements
 - 2.05.01.04 Identify audit trail content
 - 2.05.01.05 Estimate number of audit trail transactions per cycle
 - 2.05.01.06 Estimate processing costs per audit transaction
 - 2.05.01.07 Identify volume of audit trail transactions
 - 2.05.01.08 Determine required audit trail life span
 - 2.05.01.09 Identify logical database changes required to support audit trails
 - 2.05.02 Specify backup & recovery needs
 - 2.05.02.01 Classify report & update operation required currency
 - 2.05.02.02 Identify impact of database crash
 - 2.05.02.03 Determine resources for recovery
 - 2.05.02.04 Prepare backup & recovery requirements
 - 2.05.02.05 Determine extra resources for transaction processing
 - 2.05.02.06 Determine resources required for recovery
 - 2.05.02.07 Prepare backup & recovery requirements report
 - 2.05.03 Specify concurrent operations requirements
 - 2.05.03.01 Categorize update transactions by cycle
 - 2.05.03.02 Determine resource availability diminished by lockout processing
 - 2.05.03.03 Categorize report transactions by cycle
 - 2.05.03.04 Determine resource availability diminished by report processing
 - 2.05.03.05 Reevaluate method of update transaction processing to reduce lockout requirements
 - 2.05.03.06 Prepare concurrent operations report
 - 2.05.04 Specify security & privacy requirements
 - 2.05.04.01 Identify legal security requirements



- 2.05.04.02 Identify "company" confidential requirements
- 2.05.04.03 Identify personnel data security requirements
- 2.05.04.04 Identify columns and tables required to be held secure
- 2.05.04.05 Create strategy for user access profiles
- 2.05.04.06 Identify requirements for capturing violations
- 2.05.04.07 Instigate policy study for security violations
- 2.05.04.08 Prepare security & privacy requirements report

- 2.05.05 Specify reorganization requirements
- 2.05.05.01 Determine expected row volume volatility per table
- 2.05.05.02 Identify expected frequency & quantity of element type changes
- 2.05.05.03 Identify expected frequency & quantity of table changes
- 2.05.05.04 Identify expected frequency & quantity of relationship type changes
- 2.05.05.05 Develop scenarios to accomplish database reorganization
- 2.05.05.06 Estimate processing resources & frequency of database reorganization over annual period
- 2.05.05.07 Prepare reorganization requirements report

- 2.05.06 Specify multiple database requirements
- 2.05.06.01 Identify interrelated databases
- 2.05.06.02 Identify data interconnection
- 2.05.06.03 Identify method of interconnection
- 2.05.06.04 Identify requirement for coordinated update
- 2.05.06.05 Determine method to validate coordinated update
- 2.05.06.06 Identify requirement for multiple database system control facilities
- 2.05.06.07 Prepare multiple database requirement report

- 2.05.07 Specify data transfer and data conversion backup and recovery needs
- 2.05.07.01 Identify impact of database crash
- 2.05.07.02 Determine resources for recovery

- 2.05.07 Prepare system control requirement report

- 2.05.07.01 Consolidate system control functional requirements
- 2.05.07.01.01 Review & revise audit trail requirements
- 2.05.07.01.02 Review & revise backup & recovery needs
- 2.05.07.01.03 Review & revise concurrent operations requirements
- 2.05.07.01.04 Review & revise security & privacy requirements
- 2.05.07.01.05 Review & revise reorganization requirements
- 2.05.07.01.06 Review & revise multiple database report
- 2.05.07.01.07 Prepare consolidated system control requirements report



Work Breakdown Structure

- 2.05.07.02 Review system control specification against business database objectives and measurements of project success to determine if revisions are necessary
- 2.05.07.03 Consolidate database changes requirements to accommodate system control
- 2.05.07.04 Prepare system control report
 - 2.05.07.04.01 Review & revise system control requirements report
 - 2.05.07.04.02 Review & revise system control resource requirements report
 - 2.05.07.04.03 Prepare consolidate system control report for conceptual specification
- 2.05.08 Create system control presentation
 - 2.05.08.01 Modify features, advantages and benefits (FAB) of database system
 - 2.05.08.02 Include FABs to create system control use scenarios
 - 2.05.08.03 Graphically identify tables or processes that participate in scenario
 - 2.05.08.04 Create narrative for system control use scenario
 - 2.05.08.05 Identify key individual for system control use scenario
 - 2.05.08.06 Review system control use scenario & narrative with key individual and modify as appropriate
 - 2.05.08.07 Create A/V materials for system control use scenario presentation
 - 2.05.08.08 Review and finalize A/V materials prior to review
- 2.05.09 Conduct subphase review
 - 2.05.09.01 Present database system components
 - 2.05.09.02 Receive and respond to comments
 - 2.05.09.03 Modify system as necessary
 - 2.05.09.04 Seek concurrence with system subphase
- 2.06 Validate cross product of data integrity and data transformation models
 - 2.06.01 Use data integrity model to functionally validate the completeness of the data transformation model
 - 2.06.02 Use the data transformation model to functionally validate the completeness of the data integrity model
 - 2.06.03 Resolve any functional imbalance between the data integrity model and the data transformation model
- 2.07 Create cross-product validation presentation
 - 2.07.01 Modify features, advantages and benefits (FAB) of database system
 - 2.07.02 Include FABs to create use scenarios
 - 2.07.03 Graphically identify tables or processes that participate in scenario
 - 2.07.04 Create narrative for use scenario
 - 2.07.05 Identify key individual for use scenario
 - 2.07.06 Review use scenario & narrative with key individual and modify as appropriate



- 2.07.07 Create A/V materials for use scenario presentation
- 2.07.08 Review and finalize A/V materials prior to review

- 2.08 Conduct subphase review
 - 2.08.01 Present cross-product validation
 - 2.08.02 Receive and respond to comments
 - 2.08.03 Modify system as necessary
 - 2.08.04 Seek concurrence with system subphase

- 2.09 Validate through prototyping
 - 2.09.01 Select appropriate DBMS
 - 2.09.01.01 Determine most functional & flexible schema-DDL creation & submission language
 - 2.09.01.02 Determine most functional & flexible of sub-schema DDL submission language
 - 2.09.01.03 Determine most functional & flexible table data loading utility
 - 2.09.01.04 Determine most flexible relationship specification & interrogation language
 - 2.09.01.05 Determine most functional & flexible data update transaction utility
 - 2.09.01.06 Determine most functional & flexible backup, recovery & logical database reorganization language
 - 2.09.01.07 Select the DBMS which maximizes required features
 - 2.09.02 Create logical database
 - 2.09.02.01 Divide database into appropriate partitions
 - 2.09.02.01.01 Review database for 6-10 table partitions
 - 2.09.02.01.02 Create database graphic for each partition
 - 2.09.02.01.03 Identify value-based relationships for tables within each partition
 - 2.09.02.01.04 Select most significant elements for each table
 - 2.09.02.01.05 Select table primary key & appropriate secondary keys
 - 2.09.02.01.06 Determine necessary columns, data types & editing clauses
 - 2.09.02.01.07 Create prototype partitions report
 - 2.09.02.02 Create implemented data model DDL for each prototype
 - 2.09.02.02.01 Create implemented data model schema level DDL
 - 2.09.02.02.02 Create implemented data model table DDL
 - 2.09.02.02.03 Create implemented data model table column DDL
 - 2.09.02.02.04 Create implemented data model relationship DDL
 - 2.09.02.02.05 Create implemented data model editing & validation clauses
 - 2.09.02.02.06 Create implemented data model view DDL for data editing
 - 2.09.02.02.07 Create implemented data model view for update transactions



Work Breakdown Structure

- 2.09.02.02.08 Create implemented data model view for report transactions
- 2.09.02.03 Submit implemented data model DDL to DBMS to create logical database
 - 2.09.02.03.01 Code implemented data model schema DDL into file
 - 2.09.02.03.02 Code implemented data model view DDL into file
 - 2.09.02.03.03 Submit to DBMS
 - 2.09.02.03.04 Correct as necessary
- 2.09.03 Create test data
 - 2.09.03.01 Create table data coding form
 - 2.09.03.02 Identify strategy to create realistic test data
 - 2.09.03.03 Create statistically significant test data for each table
 - 2.09.03.04 Create test data collections appropriate for each prototype
 - 2.09.03.05 Enter test data onto files
 - 2.09.03.06 Prepare test data report
- 2.09.04 Create physical database
 - 2.09.04.01 Create data loading program
 - 2.09.04.02 Allocate computer space for data
 - 2.09.04.03 Execute load program to load test data
 - 2.09.04.04 Validate volume & correctness of loaded data
- 2.09.05 Create interrogation scenarios
 - 2.09.05.01 Identify prototype demonstration audiences
 - 2.09.05.02 For each audience determine prototype demonstration needs
 - 2.09.05.03 Create typical reports in end user language
 - 2.09.05.04 Create update transactions in end-user language
 - 2.09.05.05 Test user demonstrations to build complete scripts
 - 2.09.05.06 Create user documentation & demonstration handouts
- 2.09.06 Present design iteration
 - 2.09.06.01 Establish schedule for prototype demonstrations
 - 2.09.06.02 Issue preparatory material to participants
 - 2.09.06.03 Initiate demonstration through presentation of goals & database objectives of database system
 - 2.09.06.04 Conduct formal component of demonstration
 - 2.09.06.05 Conduct ad hoc component of demonstration
 - 2.09.06.06 Solicit applicability to user needs
 - 2.09.06.06.01 Document required tables
 - 2.09.06.06.02 Document required columns



- 2.09.06.06.03 Document required relationships
- 2.09.06.07 Prepare demonstration report
- 2.09.07 Create design iteration changes
 - 2.09.07.01 Evaluate change requirements
 - 2.09.07.02 Determine effect on fundamental logical database model
 - 2.09.07.03 Determine whether extent of changes are unique or common
 - 2.09.07.04 Determine effect on other users if changes are made
 - 2.09.07.05 Reevaluate whether participant database domains have enough commonality for single database
 - 2.09.07.06 Formulate design changes
 - 2.09.07.07 Formulate report to justify database redesign
 - 2.09.07.08 Prepare a report indicating incorrect database domain identification
- 2.09.08 Update database specification design
 - 2.09.08.01 Review & revise logical database specification
 - 2.09.08.02 Review & revise physical database specification
 - 2.09.08.03 Review & revise interrogation specification
 - 2.09.08.04 Prepare database design change report
- 2.09.09 Walkthrough all components of the data transfer and data conversion
 - 2.09.09.01 Create test data
 - 2.09.09.02 Create physical database
 - 2.09.09.03 Create interrogation scenarios
 - 2.09.09.04 Present design iteration
 - 2.09.09.05 Prepare demonstration report
- 2.10 Consolidate logical database changes required to accommodate physical database interrogation, and system control additional requirements, and revise as necessary
- 2.11 Create conceptual specification phase report
 - 2.11.01 Create introduction & scope
 - 2.11.01.01 Include mission descriptions
 - 2.11.01.02 Include mission description diagrams
 - 2.11.01.03 Demarcate the intended users
 - 2.11.01.04 Identify the systems being replaced
 - 2.11.01.05 Review list of business goals, database objectives, & criteria for success measurement for database system, and revise as necessary
 - 2.11.01.06 Provide a summary of pertinent statistics about database



Work Breakdown Structure

- 2.11.02 Create logical database requirements section
 - 2.11.02.01 Include database domains and database domain diagrams
 - 2.11.02.02 Include final data integrity model
 - 2.11.02.03 Create final report of logical database
- 2.11.03 Create physical database requirements section
 - 2.11.03.01 Include data transformation model
 - 2.11.03.02 Include data loading subsystem specifications
 - 2.11.03.03 Include data update subsystem specifications
 - 2.11.03.04 Include business event specifications
 - 2.11.03.05 Include business function specifications
 - 2.11.03.06 Include database size estimate
 - 2.11.03.07 Include data source quality reports
 - 2.11.03.08 Include backup requirements reports
 - 2.11.03.09 Include data integrity subsystem specifications
 - 2.11.03.10 Create final physical database report
- 2.11.04 Conduct final review of functional completeness & acceptable interaction of data integrity model and data transformation model.
- 2.11.05 Create interrogation requirements section
 - 2.11.05.01 Include standard report requirements
 - 2.11.05.02 Include ad-hoc report requirements
 - 2.11.05.03 Prepare final consolidated interrogation requirements report
- 2.11.06 Prepare system control requirements section
 - 2.11.06.01 Include system control functional requirements
 - 2.11.06.02 Include system control resource requirements
 - 2.11.06.03 Prepare final consolidated system control specification
- 2.11.07 Prepare conceptual specification phase report
 - 2.11.07.01 Prepare summary of logical database including statement of limitations, audience, etc.
 - 2.11.07.02 Prepare summary of physical database including statement of limitations, audience, etc.
 - 2.11.07.03 Prepare summary of interrogation including statement of limitations, audience, etc.
 - 2.11.07.04 Prepare summary of system control including statement of limitations, audience, etc.
 - 2.11.07.05 Prepare a summary that specifies overall limitations, problem areas, etc.



- 2.12 Create conceptual specification phase presentation
 - 2.12.01 Modify features, advantages and benefits (FAB) of database system
 - 2.12.02 Include FABs to create database use scenarios
 - 2.12.03 Graphically identify tables or processes that participate in scenario
 - 2.12.04 Create narrative for database use scenario
 - 2.12.05 Identify key individual for database use scenario
 - 2.12.06 Review database use scenario & narrative with key individual and modify as appropriate
 - 2.12.07 Create A/V materials for database use scenario presentation
 - 2.12.08 Review and finalize A/V materials prior to review
- 2.13 Conduct phase review
 - 2.13.01 Present conceptual specification
 - 2.13.02 Receive and respond to comments
 - 2.13.03 Modify system as necessary
 - 2.13.04 Seek concurrence with system phase





3

Binding Phase

3 Binding

- 3.01 Form project team, plan phase, and revise project plan
 - 3.01.01 Select manager for project phase
 - 3.01.02 Identify administrative, clerical, and computer supports
 - 3.01.03 Select project staff for phase
 - 3.01.03.01 Interview and select database specialist
 - 3.01.03.02 Interview and select functional users
 - 3.01.04 Secure commitments for staff availability
 - 3.01.04.01 Estimate time requirement for project manager
 - 3.01.04.02 Estimate time requirement for administrative, clerical, and computer supports
 - 3.01.05 Create phase plans and schedules
 - 3.01.05.01 Create phase specific PERT charts
 - 3.01.05.02 Create phase specific Gantt charts
 - 3.01.05.03 Create phase CPM charts
 - 3.01.05.04 Create phase manpower estimate charts
 - 3.01.06 Organize phase review committee
 - 3.01.07 Conduct methods training as required
 - 3.01.08 Set up phase documentation files
 - 3.01.09 Revise project plan and schedule estimates
 - 3.01.09.01 Revise PERT charts
 - 3.01.09.02 Revise Gantt charts
 - 3.01.09.03 Revise CPM charts
 - 3.01.09.04 Revise manpower estimate charts
 - 3.01.10 Develop and/or use database standards for phase
 - 3.01.11 Interrogate and maintain metadata repository during phase
- 3.02 Review conceptual specification phase for completeness and revise as necessary
- 3.03 Remedy any data availability and quality problems determined during conceptual specification phase
- 3.04 Determine Impact on Corporate Business Model
 - 3.04.01 Evaluate tables that are boundary to outside systems and revise as necessary to maximize compatibility
 - 3.04.02 Determine impact on logical database
 - 3.04.02.01 Determine impact on mission descriptions

Work Breakdown Structure

- 3.04.02.02 Determine impact on mission description diagrams
- 3.04.02.03 Determine impact on database domains
- 3.04.02.04 Determine impact on database domain diagram
- 3.04.02.05 Determine impact on business and technical glossary

- 3.04.02.06 Determine impact on data integrity model
 - 3.04.02.06.01 Determine impact on database objects
 - 3.04.02.06.02 Determine impact on data elements
 - 3.04.02.06.03 Determine impact on data integrity rules

- 3.04.02.07 Create logical database impact report
- 3.04.02.08 Create logical database presentation
- 3.04.02.09 Conduct subphase review

- 3.04.03 Determine impact on physical database specification
 - 3.04.03.01 Determine impact on data transformation model
 - 3.04.03.01.01 Determine impact on primitive data transformations
 - 3.04.03.01.02 Determine impact on information system process diagram
 - 3.04.03.01.03 Create data transformation model impact report
 - 3.04.03.01.04 Conduct subphase review

 - 3.04.03.02 Determine impact on existing data update subsystems
 - 3.04.03.02.01 Analyze impacted data update subsystems
 - 3.04.03.02.02 Analyze impacted data input formats
 - 3.04.03.02.03 Determine impact on human support subsystems
 - 3.04.03.02.04 Determine impact on documentation
 - 3.04.03.02.05 Create data update subsystem impact report
 - 3.04.03.02.06 Conduct subphase review

 - 3.04.03.03 Determine impact on existing data loading subsystems
 - 3.04.03.03.01 Analyze impacted data load subsystems
 - 3.04.03.03.02 Analyze impacted data input formats
 - 3.04.03.03.03 Determine impact on human support subsystems
 - 3.04.03.03.04 Determine impact on documentation
 - 3.04.03.03.05 Create data update subsystem impact report
 - 3.04.03.03.06 Conduct subphase review

 - 3.04.03.04 Determine impact on database size
 - 3.04.03.05 Determine impact on requirements for backup
 - 3.04.03.06 Determine impact on database integrity subsystem



- 3.04.03.07 Prepare physical database requirement report
- 3.04.03.08 Create physical database impact presentation
- 3.04.03.09 Conduct subphase review

- 3.04.04 Determine impact on interrogation subsystems
 - 3.04.04.01 Analyze impact on standard report requirements
 - 3.04.04.02 Analyze impact on ad-hoc report requirements
 - 3.04.04.03 Prepare interrogation impact report
 - 3.04.04.04 Create interrogation impact presentation
 - 3.04.04.05 Conduct subphase review

- 3.04.05 Determine impact on system control facilities
 - 3.04.05.01 Determine impact on audit trails
 - 3.04.05.02 Determine impact on backup & recovery needs
 - 3.04.05.03 Determine impact on concurrent operations
 - 3.04.05.04 Determine impact on security & privacy
 - 3.04.05.05 Determine impact on reorganization
 - 3.04.05.06 Determine impact on multiple databases
 - 3.04.05.07 Prepare system control impact report
 - 3.04.05.08 Create system control impact presentation
 - 3.04.05.09 Conduct subphase review

- 3.04.06 Determine impact on ancillary supports
 - 3.04.06.01 Determine impact on database training
 - 3.04.06.02 Determine impact on database hotline
 - 3.04.06.03 Determine impact on database standards
 - 3.04.06.04 Determine impact on test data
 - 3.04.06.05 Determine impact on documentation
 - 3.04.06.06 Prepare ancillary supports impact report
 - 3.04.06.07 Create ancillary supports impact presentation
 - 3.04.06.08 Conduct subphase review

- 3.04.07 Determine impact on database system quality tests
 - 3.04.07.01 Analyze appropriate system quality tests
 - 3.04.07.02 Determine impact on testing scenarios
 - 3.04.07.03 Determine impact on backup and recovery procedures
 - 3.04.07.04 Determine impact on system performance and test success criteria
 - 3.04.07.05 Determine impact on required operations, procedures, and guides that are to be evaluated
 - 3.04.07.06 Determine impact on existing system quality test plan
 - 3.04.07.07 Prepare system quality test impact report



Work Breakdown Structure

- 3.04.07.08 Create system quality test impact presentation
- 3.04.07.09 Conduct subphase review

- 3.04.08 Formulate corporate business model impact report
 - 3.04.08.01 Create logical database impact section
 - 3.04.08.02 Create physical database impact section
 - 3.04.08.03 Create interrogation impact section
 - 3.04.08.04 Create system control section
 - 3.04.08.05 Create ancillary supports impact section
 - 3.04.08.06 Create system quality test impact section

- 3.05 Evaluate, select and/or procure DBMS packages
 - 3.05.01 Determine whether database application requires static relationship data structures
 - 3.05.01.01 Estimate expected frequency of prototype revisions throughout life of database
 - 3.05.01.02 Determine whether relationship changes are isolated to few parts of the database or are wide spread
 - 3.05.01.03 If wide spread, categorize database as requiring only dynamic relationship data structures
 - 3.05.01.04 If well defined and few, categorize database as best suited to multiple static relationship data structures
 - 3.05.01.05 If few or not at all, categorize database as best suited to only static relationship data structures
 - 3.05.01.06 Choose DBMS that minimizes discrepancies between conceptual specification requirements & DBMS functions available
 - 3.05.02 Create requirements document
 - 3.05.02.01 Define logical database requirements as specified in conceptual specification
 - 3.05.02.02 Define physical database requirements as specified in conceptual specification
 - 3.05.02.03 Define interrogation requirements as specified in conceptual specification
 - 3.05.02.04 Define system control requirements as specified in conceptual specification
 - 3.05.03 Create DBMS requirements document presentation
 - 3.05.03.01 Modify features, advantages and benefits (FAB) of database system
 - 3.05.03.02 Include FABs to illustrate DBMS requirements
 - 3.05.03.03 Graphically identify database objects/processes that participate in illustrated requirement
 - 3.05.03.04 Create narrative for DBMS requirements
 - 3.05.03.05 Identify key individual for review of DBMS requirements



- 3.05.03.06 Review DBMS requirements & narrative with key individual and modify as appropriate
- 3.05.03.07 Create A/V materials for DBMS requirements presentation
- 3.05.03.08 Review and finalize A/V materials prior to review

- 3.05.04 Conduct subphase review
 - 3.05.04.01 Present DBMS requirements
 - 3.05.04.02 Receive and respond to comments
 - 3.05.04.03 Modify system as necessary
 - 3.05.04.04 Seek concurrence with system subphase

- 3.05.05 Issue RFP and select DBMS
 - 3.05.05.01 Create a complete example for vendor illustrated responses
 - 3.05.05.02 Issue requirements specification and receive responses
 - 3.05.05.03 Conduct oral presentations as necessary
 - 3.05.05.04 Secure written commitments for all undocumented and upcoming features
 - 3.05.05.05 Evaluate candidates against standard
 - 3.05.05.06 Evaluate candidates against each other
 - 3.05.05.07 Evaluate cost differential between systems in terms of cost avoidance in application development
 - 3.05.05.08 Benchmark selected system

- 3.05.06 Formulate DBMS selection report and presentation
 - 3.05.06.01 Modify features, advantages and benefits (FAB) of database system
 - 3.05.06.02 Include FABs to support DBMS selection
 - 3.05.06.03 Create narrative for DBMS selection
 - 3.05.06.04 Identify key individual for review of DBMS selection
 - 3.05.06.05 Review DBMS selection narrative with key individual and modify as appropriate
 - 3.05.06.06 Create A/V materials for DBMS selection presentation
 - 3.05.06.07 Review and finalize A/V materials prior to review

- 3.05.07 Conduct subphase review
 - 3.05.07.01 Present DBMS selection decision
 - 3.05.07.02 Receive and respond to comments
 - 3.05.07.03 Modify selection as necessary
 - 3.05.07.04 Seek concurrence with DBMS selection subphase

- 3.05.08 Install package
- 3.05.09 Conduct DBMS training

- 3.05.10 Determine DBMS performance characteristics



Work Breakdown Structure

- 3.05.10.01 Develop prototypical implemented data model design
- 3.05.10.02 Develop prototypical implemented data model DDL and submit to DBMS
- 3.05.10.03 Develop prototypical data loading programs
- 3.05.10.04 Develop prototypical data update programs
- 3.05.10.05 Develop prototypical interrogation modules in HLI and NL
- 3.05.10.06 Develop statistically valid distributions of sample data

- 3.05.10.07 Assess DBMS system control capabilities
 - 3.05.10.07.01 Assess audit trail capabilities
 - 3.05.10.07.01.01 Assess acceptability at audit trail record content
 - 3.05.10.07.01.02 Assess acceptability of capture capability from all update sources & modes
 - 3.05.10.07.01.03 Assess acceptability of audit trail reports
 - 3.05.10.07.01.04 Assess acceptability of audit trail logical transactions within a multiple user environment
 - 3.05.10.07.01.05 Assess acceptability of audit trail transactions within a multiple database environment
 - 3.05.10.07.01.06 Assess audit trail backup & recovery
 - 3.05.10.07.01.07 Assess audit trail capability to reorganize its record content
 - 3.05.10.07.01.08 Assess impact on data loading transactions
 - 3.05.10.07.01.09 Assess impact on data update transactions
 - 3.05.10.07.01.10 Assess impact on interrogation transactions
 - 3.05.10.07.01.11 Prepare audit trail acceptability report
 - 3.05.10.07.02 Assess backup & recovery capability
 - 3.05.10.07.02.01 Assess the acceptability of basic design
 - 3.05.10.07.02.02 Assess the adequacy of backup & recovery for batch type jobs
 - 3.05.10.07.02.03 Assess the adequacy of backup & recovery for on-line update
 - 3.05.10.07.02.04 Assess the adequacy of backup & recovery for logical transactions on multiple databases
 - 3.05.10.07.02.05 Assess the adequacy of backup & recovery for high volume multi-user update in batch & on-line
 - 3.05.10.07.02.06 Assess the adequacy of backup & recovery for the backup & recovery capabilities
 - 3.05.10.07.02.07 Assess the impact on data loading transactions
 - 3.05.10.07.02.08 Assess the impact on data update transactions
 - 3.05.10.07.02.09 Prepare backup & recovery acceptability report
 - 3.05.10.07.03 Assess message processing capabilities
 - 3.05.10.07.03.01 Examine messages to determine their understandability



- 3.05.10.07.03.02 Determine whether message types can automatically trigger DBA created messages
- 3.05.10.07.03.03 Determine whether source & user identity of certain messages can be logged
- 3.05.10.07.03.04 Determine whether message content can be changed or augmented
- 3.05.10.07.03.05 Determine whether error processing can be enabled & disabled for certain run units
- 3.05.10.07.03.06 Determine impact on data update transactions
- 3.05.10.07.03.07 Determine impact on data loading transactions
- 3.05.10.07.03.08 Determine impact on interrogation transactions
- 3.05.10.07.03.09 Prepare messages acceptability report

- 3.05.10.07.04 Assess security & privacy capabilities
- 3.05.10.07.04.01 Examine DBMS mechanism to establish & maintain security & privacy to determine ease-of-use
- 3.05.10.07.04.02 Determine the DBMS ability to protect security & privacy from O/S utility tampering
- 3.05.10.07.04.03 Determine security & privacy ability to protect row, column & relationship
- 3.05.10.07.04.04 Determine security & privacy ability to protection the basis of values, conditions, etc.
- 3.05.10.07.04.05 Determine security & privacy ability to meet specific needs stated in the conceptual specifications
- 3.05.10.07.04.06 Determine impact on data update transactions
- 3.05.10.07.04.07 Determine impact on data loading transactions
- 3.05.10.07.04.08 Determine impact on interrogation transactions
- 3.05.10.07.04.09 Prepare security & privacy acceptability report

- 3.05.10.07.05 Assess concurrent operations capabilities
- 3.05.10.07.05.01 Assess DBMS capabilities to perform multiple update commands within all languages
- 3.05.10.07.05.02 Assess DBMS capabilities to perform multiple updates to the same database
- 3.05.10.07.05.03 Assess DBMS capability to perform multiple updates to the same row
- 3.05.10.07.05.04 Assess DBMS resources that are consumed to effect various concurrent access controls
- 3.05.10.07.05.05 Determine impact on production operations
- 3.05.10.07.05.06 Prepare concurrent operations acceptability report

- 3.05.10.07.06 Assess reorganization capabilities



Work Breakdown Structure

3.05.10.07.06.01	Assess DBMS capabilities to logically restructure a table to delete, modify or add elements
3.05.10.07.06.02	Assess DBMS capabilities to physically resequence rows within a table
3.05.10.07.06.03	Assess DBMS capability to resize storage structure components without database reloading
3.05.10.07.06.04	Assess DBMS capability to add, change, or modify inter-table relationships
3.05.10.07.06.05	Assess DBMS reorganization impact on production applications
3.05.10.07.06.06	Prepare reorganization capability report
3.05.10.07.07	Assess multiple database object processing capabilities
3.05.10.07.07.01	Assess capability to operate multiple databases within one DBMS copy
3.05.10.07.07.02	Assess capability to operate multiple databases from various DBMS languages
3.05.10.07.07.03	Assess capability to operate database editions from DBMS languages
3.05.10.07.07.04	Assess capability for multiple database update transaction audit trail logs
3.05.10.07.07.05	Assess capability for multiple database backup & recovery & other system control facilities
3.05.10.07.07.06	Assess impact on production operations
3.05.10.07.07.07	Prepare reorganization capabilities report
3.05.10.07.08	Create DBMS system control capability assessment presentation
3.05.10.07.08.01	Modify features, advantages and benefits (FAB) of database system
3.05.10.07.08.02	Include FABs to create system control capability impact scenarios
3.05.10.07.08.03	Create narrative for system control capability impact scenario
3.05.10.07.08.04	Identify key individual for review of system control capability impact scenario
3.05.10.07.08.05	Review system control capability impact scenario & narrative with key individual and modify as appropriate
3.05.10.07.08.06	Create A/V materials for system control capability impact scenario presentation
3.05.10.07.08.07	Review and finalize A/V materials prior to review
3.05.10.07.09	Conduct subphase review
3.05.10.07.09.01	Present system control capability assessment
3.05.10.07.09.02	Receive and respond to comments
3.05.10.07.09.03	Modify system as necessary
3.05.10.07.09.04	Seek concurrence with system subphase
3.05.10.08	Run various tests at different loads, velocities, etc.
3.05.10.09	Develop graphs and tables to assist in database design and language selection



- 3.05.10.10 Create DBMS performance report
- 3.05.10.10.01 Construct prototypical database designs section
- 3.05.10.10.02 Construct prototypical implemented data model DDL section
- 3.05.10.10.03 Construct prototypical data loading program section
- 3.05.10.10.04 Construct prototypical data update program section
- 3.05.10.10.05 Construct prototypical interrogation section
- 3.05.10.10.06 Construct system control assessment section
- 3.05.10.10.07 Construct performance statistics section

- 3.05.10.11 Create DBMS performance presentation
- 3.05.10.11.01 Modify features, advantages and benefits (FAB) of database system
- 3.05.10.11.02 Include FABs to create DBMS use scenarios
- 3.05.10.11.03 Graphically identify tables or processes that participate in scenario
- 3.05.10.11.04 Create narrative for use scenario
- 3.05.10.11.05 Identify key individual for use scenario
- 3.05.10.11.06 Review use scenario & narrative with key individual and modify as appropriate
- 3.05.10.11.07 Create A/V materials for use scenario presentation
- 3.05.10.11.08 Review and finalize A/V materials prior to review

- 3.05.10.12 Conduct subphase review
- 3.05.10.12.01 Present DBMS performance assessment results
- 3.05.10.12.02 Receive and respond to comments
- 3.05.10.12.03 Modify as necessary
- 3.05.10.12.04 Seek concurrence with subphase

- 3.05.11 Develop hints and tidbits section for documentation
- 3.05.12 Create a mapping from DBMS taxonomy to DBMS documentation
- 3.05.13 Develop generic examples for all database application functions
- 3.05.14 Adjust programming effort estimates for programming assists provided by natural languages

- 3.06 Evaluate, select and/or procure metadata repository packages

- 3.06.01 Create metadata repository requirements document
- 3.06.01.01 Define logical database requirements for graphics and analytics
- 3.06.01.02 Define physical database requirements to support data loading, update, and maintenance
- 3.06.01.03 Define interrogation requirements to support needed methodology product presentations



Work Breakdown Structure

- 3.06.01.04 Define system control requirements to support the needs of any sophisticated database application
- 3.06.02 Create metadata repository requirements document presentation
 - 3.06.02.01 Modify features, advantages and benefits (FAB) of database system
 - 3.06.02.02 Include FABs to illustrate metadata repository requirements
 - 3.06.02.03 Graphically identify database objects/processes that participate in illustrated requirement
 - 3.06.02.04 Create narrative for metadata repository requirements
 - 3.06.02.05 Identify key individual for review of metadata repository requirements
 - 3.06.02.06 Review metadata repository requirements & narrative with key individual and modify as appropriate
 - 3.06.02.07 Create A/V materials for metadata repository requirements presentation
 - 3.06.02.08 Review and finalize A/V materials prior to review
- 3.06.03 Conduct subphase review
 - 3.06.03.01 Present metadata repository requirements
 - 3.06.03.02 Receive and respond to comments
 - 3.06.03.03 Modify system as necessary
 - 3.06.03.04 Seek concurrence with system subphase
- 3.06.04 Issue RFP and select metadata repository
 - 3.06.04.01 Create a complete example for vendor illustrated responses
 - 3.06.04.02 Issue requirements specification and receive responses
 - 3.06.04.03 Conduct oral presentations as necessary
 - 3.06.04.04 Secure written commitments for all undocumented and upcoming features
 - 3.06.04.05 Evaluate candidates against requirements
 - 3.06.04.06 Evaluate candidates against each other
 - 3.06.04.07 Evaluate cost differential between metadata repository systems in terms of cost avoidance in application development
- 3.06.05 Formulate metadata repository selection report and presentation
 - 3.06.05.01 Modify features, advantages and benefits (FAB) of database system
 - 3.06.05.02 Include FABs to support metadata repository selection
 - 3.06.05.03 Create narrative for metadata repository selection
 - 3.06.05.04 Identify key individual for review of metadata repository selection
 - 3.06.05.05 Review metadata repository selection narrative with key individual and modify as appropriate
 - 3.06.05.06 Create A/V materials for metadata repository selection presentation
 - 3.06.05.07 Review and finalize A/V materials prior to review



- 3.06.06 Conduct subphase review
- 3.06.06.01 Present metadata repository selection decision
- 3.06.06.02 Receive and respond to comments
- 3.06.06.03 Modify selection as necessary
- 3.06.06.04 Seek concurrence with metadata repository selection subphase

- 3.06.07 Install metadata repository package
- 3.06.08 Conduct metadata repository training

- 3.06.09 Accomplish metadata repository prototype implementation
- 3.06.09.01 Accomplish logical database development or modifications to accomodate metadata repository requirements
- 3.06.09.02 Accomplish physical database development or modifications to accomodate metadata repository requirements
- 3.06.09.03 Accomplish interrogation development or modifications to accomodate metadata repository requirements
- 3.06.09.04 Accomplish system control development or modifications to accomodate metadata repository requirements

- 3.06.10 Conduct metadata repository prototype demonstion
- 3.06.10.01 Conduct and assess demonstration of metadata repository capabilities versus logical database requirements
- 3.06.10.02 Conduct and assess demonstration of metadata repository capabilities versus physical database requirements
- 3.06.10.03 Conduct and assess demonstration of metadata repository capabilities versus interrogation requirements
- 3.06.10.04 Conduct and assess demonstration of metadata repository capabilities versus system control requirements

- 3.06.11 Create metadata repository prototype demonstration presentation
- 3.06.11.01 Modify features, advantages and benefits (FAB)of database system
- 3.06.11.02 Include FABs to create capability scenarios
- 3.06.11.03 Create narrative for capability scenario
- 3.06.11.04 Identify key individual for review of capability scenario
- 3.06.11.05 Review capability scenario & narrative with key individual and modify as appropriate
- 3.06.11.06 Create A/V materials for capability scenario presentation
- 3.06.11.07 Review and finalize A/V materials prior to review

- 3.06.12 Conduct subphase review
- 3.06.12.01 Present metadata repository selection conclusion



Work Breakdown Structure

- 3.06.12.02 Receive and respond to comments
- 3.06.12.03 Modify system as necessary
- 3.06.12.04 Seek concurrence with metadata repository selection subphase

- 3.07 Determine Implementation Strategy
 - 3.07.01 Bind logical database to selected DBMS
 - 3.07.01.01 Transform and or revise dynamic database
 - 3.07.01.01.01 Create physical Database diagram from table graphic
 - 3.07.01.01.02 Determine whether relational or ILF
 - 3.07.01.01.03 For ILF, create or revise tables with owner & single-level members
 - 3.07.01.01.04 For relational, create or revise single level tables
 - 3.07.01.01.05 Define value-based relationships between tables and update various conceptual specification components
 - 3.07.01.01.06 Identify fields for intra-table relationships as secondary keys
 - 3.07.01.02 Transform and/or revise static database
 - 3.07.01.02.01 Create physical database diagram from table graphic
 - 3.07.01.02.02 Determine whether network or hierarchical
 - 3.07.01.02.03 Create or revise network structures
 - 3.07.01.02.03.01 Identify as tables those with no inter-table related repeating groups
 - 3.07.01.02.03.02 Define relationships that structurally connect these tables
 - 3.07.01.02.03.03 Identify other inter-table relationships
 - 3.07.01.02.04 Create or revise hierarchical structures
 - 3.07.01.02.04.01 Identify hierarchies of most stable relationships
 - 3.07.01.02.04.02 Identify value-based relationships between hierarchies
 - 3.07.01.02.04.03 Identify fields for intra-table relationships as secondary keys
 - 3.07.01.03 Record transformation or revision details in specification
 - 3.07.01.03.01 Identify additional dependency definitions, conditions, constraints, etc.
 - 3.07.01.03.02 Create additional tables as needed
 - 3.07.01.03.03 Create additional elements as needed
 - 3.07.01.03.04 Create additional data integrity rules as needed
 - 3.07.01.04 Create logical database transformation or revision report
 - 3.07.01.04.01 Review & adjust logical database narratives from conceptual specification
 - 3.07.01.04.02 Review & revise all transformation tables
 - 3.07.01.04.03 Review & revise all transformation elements



- 3.07.01.04.04 Review & revise all transformation data integrity rules
- 3.07.01.05 Create logical database transformation or revision presentation
 - 3.07.01.05.01 Modify features, advantages and benefits (FAB) of database system
 - 3.07.01.05.02 Include FABs to create use scenarios
 - 3.07.01.05.03 Graphically identify tables processes that participate in scenario
 - 3.07.01.05.04 Create narrative for use scenario
 - 3.07.01.05.05 Identify key individual for use scenario
 - 3.07.01.05.06 Review use scenario & narrative with key individual and modify as appropriate
 - 3.07.01.05.07 Create A/V materials for use scenario presentation
 - 3.07.01.05.08 Review and finalize A/V materials prior to review
- 3.07.01.06 Conduct subphase review
 - 3.07.01.06.01 Present logical database transformation or revision
 - 3.07.01.06.02 Receive and respond to comments
 - 3.07.01.06.03 Modify system as necessary
 - 3.07.01.06.04 Seek concurrence with system subphase
- 3.07.02 Bind physical database to selected DBMS
 - 3.07.02.01 Create or revise storage structure (static only)
 - 3.07.02.01.01 Determine most frequent table entry point within static structure
 - 3.07.02.01.02 Identify tables most frequently accessed subsequent to entry point
 - 3.07.02.01.03 Create database diagrams of access structures
 - 3.07.02.01.04 Identify mechanism of relationship between structures
 - 3.07.02.01.05 Update transformation specification as appropriate
 - 3.07.02.02 Select access strategy
 - 3.07.02.02.01 Select or revise access strategy for entry table
 - 3.07.02.02.01.01 Determine predominate access relationship between siblings for access structure entry table
 - 3.07.02.02.01.02 Identify field for hash/calc access
 - 3.07.02.02.01.03 Identify whether sibling relationship is entry-order or by specific field order
 - 3.07.02.02.01.04 Identify fields by each table is to be ordered
 - 3.07.02.02.01.05 Indicate that access is relative record access (data integrity ruleect)
 - 3.07.02.02.02 Select or revise access strategy for subordinate tables
 - 3.07.02.02.02.01 Identify subordinate table to be connected by relationship
 - 3.07.02.02.02.02 Identify conditions or constraints to bind owners & members



Work Breakdown Structure

- 3.07.02.02.02.03 Determine whether sibling relationship is entry-order or by specific field order
- 3.07.02.02.02.04 Indicate that access is relative record access (data integrity rulelect)
- 3.07.02.02.02.05 Identify field by which table is to be ordered
- 3.07.02.02.02.06 Determine whether occurrences are to be "placed-near"

- 3.07.02.02.03 Validate selected or revised access strategy
 - 3.07.02.02.03.01 Combine all table access into single access strategy graphic
 - 3.07.02.02.03.02 Identify most significant update scenarios & report sequences
 - 3.07.02.02.03.03 Conduct "walk-throughs" to determine processing sequences
 - 3.07.02.02.03.04 Evaluate overall processing efficiency

- 3.07.03 Determine interrogation implementation strategy
 - 3.07.03.01 Review processing requirements and determine if reports are efficient database users
 - 3.07.03.02 Identify database structure additions to efficiently support significant interrogations
 - 3.07.03.03 Assess cost benefit of database change in terms of additional data loading and data update

 - 3.07.03.04 Modify database design
 - 3.07.03.04.01 Modify or add elements
 - 3.07.03.04.02 Modify or add tables
 - 3.07.03.04.03 Modify or add data integrity rules
 - 3.07.03.04.04 Modify data integrity subsystem design and estimates
 - 3.07.03.04.05 Modify data update subsystem design and estimates
 - 3.07.03.04.06 Modify data loading subsystem design and estimates
 - 3.07.03.04.07 Modify database size
 - 3.07.03.04.08 Modify all supporting documentation

- 3.07.04 Configure system control implementation strategy
 - 3.07.04.01 Determine method of system control implementation
 - 3.07.04.01.01 Identify those requirements that are fully supported by DBMS facilities
 - 3.07.04.01.02 Identify those requirements that are partially DBMS supported
 - 3.07.04.01.03 Identify those requirements that are without DBMS support
 - 3.07.04.01.04 Determine those facilities which require additional system control facilities to operate
 - 3.07.04.01.05 Determine extent of lockout required by each facility
 - 3.07.04.01.06 Create a DBMS support availability matrix



- 3.07.04.02 Create system control implementation estimate & report
 - 3.07.04.02.01 Estimate the cost & time to implement fully supported DBMS facilities
 - 3.07.04.02.02 Estimate the cost & time to implement the partially DBMS supported facilities
 - 3.07.04.02.03 Estimate the cost & time to design, implement & test DBMS facilities not at all supported
 - 3.07.04.02.04 Assign priorities to each item & rank order by importance to application
 - 3.07.04.02.05 Create a system control report detailing availability, cost & importance
- 3.07.04.03 Include impact on production applications
 - 3.07.04.03.01 Factor in impact due to audit trails
 - 3.07.04.03.02 Factor in impact due to backup and recovery
 - 3.07.04.03.03 Factor in impact due to security and privacy
 - 3.07.04.03.04 Factor in impact due to concurrent operations
 - 3.07.04.03.05 Factor in impact due to multiple database object processing
- 3.07.04.04 Create system control implementation estimate presentation
 - 3.07.04.04.01 Modify features, advantages and benefits (FAB) of database system
 - 3.07.04.04.02 Include FABs to illustrate reasons for system control implementation estimate
 - 3.07.04.04.03 Create narrative for system control implementation estimate
 - 3.07.04.04.04 Identify key individual for system control implementation estimate
 - 3.07.04.04.05 Review system control implementation estimate & narrative with key individual and modify as appropriate
 - 3.07.04.04.06 Create A/V materials for system control implementation estimate scenario presentation
 - 3.07.04.04.07 Review and finalize A/V materials prior to review
- 3.07.04.05 Conduct subphase review
 - 3.07.04.05.01 Present system control implementation estimate
 - 3.07.04.05.02 Receive and respond to comments
 - 3.07.04.05.03 Modify system as necessary
 - 3.07.04.05.04 Seek concurrence with system subphase
- 3.07.05 Assess hardware availability
 - 3.07.05.01 Determine effect of proposed database system on CPU available capacity
 - 3.07.05.02 Determine effect of proposed database system on available mass storage
 - 3.07.05.03 Determine effect of proposed database system on telecommunications facilities
 - 3.07.05.04 Formulate hardware requirements estimates
 - 3.07.05.05 Create hardware enhancement proposals that include cost, procurement alternatives, and delivery schedules
- 3.07.06 Determine implementation strategy



Work Breakdown Structure

- 3.07.06.01 Identify implementation alternatives
 - 3.07.06.01.01 Define each alternative in sufficient detail to evaluate advantages and drawbacks
 - 3.07.06.01.02 Determine the logical database effect on the existing conceptual specification design resulting from alternative
 - 3.07.06.01.03 Determine the physical database effect on the existing conceptual specification design resulting from alternative
 - 3.07.06.01.04 Determine the interrogation effect on the existing conceptual specification design resulting from alternative
 - 3.07.06.01.05 Determine the system control effect on the existing conceptual specification design resulting from alternative
 - 3.07.06.01.06 Define interface boundaries requirement to outside systems
 - 3.07.06.01.07 Determine approximate implementation cost for each alternative
 - 3.07.06.01.08 Determine cost versus benefit ratio for each alternative
- 3.07.06.02 Determine Gantt/PERT/CPM charts for each alternative
- 3.07.06.03 Present alternative for review and selection
- 3.07.06.04 Consolidate Gantt, PERT, and CPM charts for selected alternative
- 3.07.07 Prepare system proposal for implementation phase
 - 3.07.07.01 Adjust subsystems or specifications per special study results
 - 3.07.07.02 Create estimate for implementation phase
 - 3.07.07.03 Obtain approval for implementation
- 3.07.08 Create implementation approach presentation
 - 3.07.08.01 Modify features, advantages and benefits (FAB) of database system
 - 3.07.08.02 Include FABs to create system implementation strategy
 - 3.07.08.03 Create narrative for system control implementation estimate scenario
 - 3.07.08.04 Identify key individual for system implementation strategy
 - 3.07.08.05 Review system implementation strategy & narrative with key individual and modify as appropriate
 - 3.07.08.06 Create A/V materials for system implementation strategy presentation
 - 3.07.08.07 Review and finalize A/V materials prior to review
- 3.07.09 Conduct subphase review
 - 3.07.09.01 Present database system implementation strategy
 - 3.07.09.02 Receive and respond to comments
 - 3.07.09.03 Modify strategy as necessary
 - 3.07.09.04 Seek concurrence with implementation strategy
- 3.08 Prepare binding phase report
 - 3.08.01 Prepare section on data availability



- 3.08.02 Prepare section on corporate business model impact
- 3.08.03 Prepare section on DBMS selection and evaluation
- 3.08.04 Prepare section on implementation strategy

- 3.09 Create binding phase presentation
 - 3.09.01 Modify features, advantages and benefits (FAB) of database system
 - 3.09.02 Include FABs to create system implementation strategy
 - 3.09.03 Create narrative for system control implementation estimate scenario
 - 3.09.04 Identify key individual for system implementation strategy
 - 3.09.05 Review system implementation strategy & narrative with key individual and modify as appropriate
 - 3.09.06 Create A/V materials for system implementation strategy presentation
 - 3.09.07 Review and finalize A/V materials prior to review

- 3.10 Conduct phase review
 - 3.10.01 Present binding report
 - 3.10.02 Receive and respond to comments
 - 3.10.03 Modify report as necessary
 - 3.10.04 Seek concurrence with implementation phase strategy





4

Implementation Phase

4 Implementation

- 4.01 Form project team, plan phase, and revise project plan
 - 4.01.01 Select manager for project phase
 - 4.01.02 Identify administrative, clerical, and computer supports
 - 4.01.03 Select project staff for phase
 - 4.01.03.01 Interview and select database specialist
 - 4.01.03.02 Interview and select functional users
 - 4.01.04 Secure commitments for staff availability
 - 4.01.04.01 Estimate time requirement for project manager
 - 4.01.04.02 Estimate time requirement for administrative, clerical, and computer supports
 - 4.01.05 Create phase plans and schedules
 - 4.01.05.01 Create phase specific PERT charts
 - 4.01.05.02 Create phase specific Gantt charts
 - 4.01.05.03 Create phase CPM charts
 - 4.01.05.04 Create phase manpower estimate charts
 - 4.01.06 Organize phase review committee
 - 4.01.07 Conduct methods training as required
 - 4.01.08 Set up phase documentation files
 - 4.01.09 Revise project plan and schedule estimates
 - 4.01.09.01 Revise PERT charts
 - 4.01.09.02 Revise Gantt charts
 - 4.01.09.03 Revise CPM charts
 - 4.01.09.04 Revise manpower estimate charts
 - 4.01.10 Develop and/or use database standards for phase
 - 4.01.11 Interrogate and maintain metadata repository during phase
- 4.02 Create or revise logical database
 - 4.02.01 Define implemented data model schema clauses
 - 4.02.02 Define implemented data model table clauses
 - 4.02.03 Define implemented data model table condition clauses
 - 4.02.04 Define implemented data model inter-table relationship clauses
 - 4.02.05 Define implemented data model intra & inter-table data integrity clauses
 - 4.02.06 Define implemented data model relationship clauses
 - 4.02.07 Submit implemented data model DDL & make corrections

4.03 Create or revise physical database

- 4.03.01 Create or revise operational data model
 - 4.03.01.01 Create operational data model for main database file allocation
 - 4.03.01.02 Create operational data model for database support files allocation
 - 4.03.01.03 Create operational data model for database configured for data loading
 - 4.03.01.04 Create operational data model for database configured for data update
 - 4.03.01.05 Create operational data model for backup files
 - 4.03.01.06 Create documentation to support use of each operational data model
- 4.03.02 Create or revise views
 - 4.03.02.01 Identify view requirements for each update subsystem as appropriate
 - 4.03.02.02 Identify view requirements for each load program
 - 4.03.02.03 Identify view requirements for each interrogation program
 - 4.03.02.04 Identify data interfaces needed for system control functions, if needed
 - 4.03.02.05 Create appropriate views
 - 4.03.02.06 Create documentation for each view
- 4.03.03 Create or revise each data update subsystem
 - 4.03.03.01 Divide batch & on-line transactions
 - 4.03.03.02 Create or revise, then implement & test each batch update subsystem
 - 4.03.03.02.01 Create or revise batch subsystems
 - 4.03.03.02.01.01 Create general design for a transaction driven program
 - 4.03.03.02.01.02 Identify & engineer standard approach to implement batch programs so as to avoid repetitive analysis, specification, coding testing & documentation of common or functionally similar logic
 - 4.03.03.02.01.03 Create design of update transaction
 - 4.03.03.02.01.04 Create design of update transaction error processing
 - 4.03.03.02.01.05 Review & adjust transactions to conform to design of transformed database
 - 4.03.03.02.01.06 Create programming specifications for each update transaction
 - 4.03.03.02.01.07 Create programming specification for transaction rollback
 - 4.03.03.02.01.08 Identify appropriate views for subsystem
 - 4.03.03.02.01.09 Review each transaction for completeness & consistency with respect to the system as a whole
 - 4.03.03.02.02 Implement batch update subsystems
 - 4.03.03.02.02.01 Program each transaction
 - 4.03.03.02.02.02 Program each error message transaction



- 4.03.03.02.02.03 Program transaction validation
- 4.03.03.02.02.04 Walk through program code of each transaction
- 4.03.03.02.02.05 Conduct walk through of DBMS DML logic for efficiency and integrity
- 4.03.03.02.02.06 Evaluate placement of commits with respect to efficiency and integrity
- 4.03.03.02.02.07 Validate appropriate interaction with database through views
- 4.03.03.02.02.08 Conduct walk through of prototype stream of batch updates to validate overall interaction

- 4.03.03.02.03 Perform system development test on each batch update subsystem
- 4.03.03.02.03.01 Generate an appropriate set of test data to validate transaction operation
- 4.03.03.02.03.02 Create scenarios for subsystem test
- 4.03.03.02.03.03 Test each transaction individually & then in collections
- 4.03.03.02.03.04 Conduct operation validation against test database
- 4.03.03.02.03.05 Validate all error messages & transaction validation operations
- 4.03.03.02.03.06 Validate all other system control features
- 4.03.03.02.03.07 Construct resource consumption data & validate against requirements

- 4.03.03.03 Create or revise, then implement & test each on-line update subsystem

- 4.03.03.03.01 Create or revise on-line data update subsystems
- 4.03.03.03.01.01 Create general design for a transaction driven program
- 4.03.03.03.01.02 Identify & engineer standard approach to implement on-line programs so as to avoid repetitive analysis, specification, coding testing & documentation of common or functionally similar logic

- 4.03.03.03.01.03 Create design for update transaction
- 4.03.03.03.01.04 Create design of update transaction error processing
- 4.03.03.03.01.05 Review & adjust transactions to conform to design of transformed database

- 4.03.03.03.01.06 Create programming specifications for each transaction
- 4.03.03.03.01.07 Create programming specification for transaction rollback
- 4.03.03.03.01.08 Identify appropriate view for each transaction

- 4.03.03.03.02 Implement on-line data update subsystems
- 4.03.03.03.02.01 Program each transaction
- 4.03.03.03.02.02 Program each message & validation subordinate transactions
- 4.03.03.03.02.03 Program if necessary transaction rollback
- 4.03.03.03.02.04 Program whatever links are required to chain transactions
- 4.03.03.03.02.05 Program if necessary multiple transaction rollback
- 4.03.03.03.02.06 Conduct walk through of program code for transactions
- 4.03.03.03.02.07 Conduct walk through of DBMS DML logic for efficiency and integrity
- 4.03.03.03.02.08 Evaluate placement of commits with respect to efficiency and integrity



Work Breakdown Structure

- 4.03.03.03.02.09 Validate appropriate interaction with database through views
- 4.03.03.03.03 Perform system development test on each on-line data update subsystems
 - 4.03.03.03.03.01 Generate an appropriate set of test data to validate transaction operation
 - 4.03.03.03.03.02 Create scenario to test single & concurrent update transactions
 - 4.03.03.03.03.03 Test each transaction individually & then concurrently
 - 4.03.03.03.03.04 Validate correct transaction operation
 - 4.03.03.03.03.05 Validate all error messages & transaction validation operations
 - 4.03.03.03.03.06 Validate acceptable database reaction to conflicting concurrent operations
 - 4.03.03.03.03.07 Validate all other system control features
 - 4.03.03.03.03.08 Construct resource consumption data & validate against requirements
- 4.03.04 Create or revise each data loading subsystem
 - 4.03.04.01 Update data loading subsystem conceptual specification report
 - 4.03.04.01.01 Revise logical table loading sequence chart to conform to transformed database
 - 4.03.04.01.02 Revise editing & validation for each table
 - 4.03.04.01.03 Revise inter-table editing & validation
 - 4.03.04.01.04 Revise error processing procedures
 - 4.03.04.01.05 Create a data loading subsystem implementation phase specification
 - 4.03.04.02 Design or revise static data loading subsystems
 - 4.03.04.02.01 Identify entry point tables
 - 4.03.04.02.02 Create program minispecification to read data & store row
 - 4.03.04.02.03 Create program minispecification to read member table, store it in the database & update relationships for which row is a member
 - 4.03.04.02.04 Create program minispecification to effect editing & validation
 - 4.03.04.02.05 Determine appropriate action for entry table and/or member table load failure
 - 4.03.04.02.06 Create program minispecification for error processing
 - 4.03.04.03 Design or revise dynamic data loading subsystems
 - 4.03.04.03.01 Identify major "owner" tables
 - 4.03.04.03.02 Create program minispecification to read data & store row
 - 4.03.04.03.03 Create program minispecification to load "member" row & to value for owner-member relationship
 - 4.03.04.03.04 Create program minispecification to effect editing & validation
 - 4.03.04.03.05 Determine whether "member" is to be loaded if "owner" fails editing and/or validation test
 - 4.03.04.03.06 Create program minispecification for error processing
 - 4.03.04.04 Implement data loading subsystems (static)



- 4.03.04.04.01 Program each minispecification
- 4.03.04.04.02 Program editing & validation
- 4.03.04.04.03 Walk through program code of load subsystem
- 4.03.04.04.04 Conduct walk through of DBMS DML logic for efficiency and integrity
- 4.03.04.04.05 Evaluate placement of commits with respect to efficiency and integrity
- 4.03.04.04.06 Validate appropriate interaction with database through views
- 4.03.04.04.07 Conduct walk through prototype stream of data load subsystem

- 4.03.04.05 Perform system development test on data loading subsystems
- 4.03.04.05.01 Generate an appropriate set of test data to validate load subsystem
- 4.03.04.05.02 Create strategy to validate correctness of data loading
- 4.03.04.05.03 Perform simple data load to validate correct operation
- 4.03.04.05.04 Validate all error messages & error processing
- 4.03.04.05.05 Validate acceptable database reaction to database load errors
- 4.03.04.05.06 Construct resource consumption data on test database loads

- 4.03.05 Create or revise each data conversion subsystem
- 4.03.05.01 Update data conversion subsystem conceptual specification report
- 4.03.05.01.01 Revise logical table data conversion sequence chart to conform to transformed database
- 4.03.05.01.02 Revise editing & validation for each table
- 4.03.05.01.03 Revise inter-table editing & validation
- 4.03.05.01.04 Revise error processing procedures
- 4.03.05.01.05 Create a data conversion subsystems implementation phase specification

- 4.03.05.02 Design or revise static data conversion subsystems
- 4.03.05.02.01 Identify entry report point tables
- 4.03.05.02.02 Create program mini-specification to read data & store data
- 4.03.05.02.03 Create program mini-specification to effect editing & validation
- 4.03.05.02.04 Determine appropriate action for entry table failure
- 4.03.05.02.05 Create program mini-specification for error processing

- 4.03.05.03 Design or revise dynamic data conversion subsystems
- 4.03.05.03.01 Identify tables
- 4.03.05.03.02 Create program mini-specification to read, convert, and store row
- 4.03.05.03.03 Create program mini-specification to effect editing & validation
- 4.03.05.03.04 Create program mini-specification for error processing

- 4.03.05.04 Implement data conversion subsystems (static)
- 4.03.05.04.01 Program the mini-specification
- 4.03.05.04.02 Program the editing & validation



Work Breakdown Structure

- 4.03.05.04.03 Walk through program code of data conversion subsystem
- 4.03.05.04.04 Conduct walk through of DBMS DML logic for efficiency and integrity
- 4.03.05.04.05 Evaluate placement of commits with respect to efficiency and integrity
- 4.03.05.04.06 Validate appropriate interaction with database through subschema
- 4.03.05.04.07 Conduct walk through prototype stream of data conversion subsystem

- 4.03.06 Create or revise test data
 - 4.03.06.01 Review test data from conceptual specification prototype for applicability
 - 4.03.06.02 Modify test data to conform to transformed database
 - 4.03.06.03 Create test data for each data loading & data update subsystem
 - 4.03.06.04 Validate correctness of test data
 - 4.03.06.05 Organize test data in permanent form for many database operations

- 4.03.07 Create or revise database backup procedures
 - 4.03.07.01 Identify DBMS mechanism for database backup
 - 4.03.07.02 Establish procedure to involve database backup
 - 4.03.07.03 Test backup procedure on loaded test database
 - 4.03.07.04 Validate correct operation
 - 4.03.07.05 Construct resource consumption data on test database backups

- 4.03.08 Perform system development test on physical database
 - 4.03.08.01 Create testing scenarios for data loading, data update, data conversion & backup subsystems
 - 4.03.08.02 Determine method to validate tests against predetermined results
 - 4.03.08.03 Conduct sufficient tests to certify valid, reliable & consistent operation
 - 4.03.08.04 Review & validate all physical database implementation documentation

- 4.03.09 Create or revise physical database implementation report
 - 4.03.09.01 Review & adjust physical database narratives from conceptual specification
 - 4.03.09.02 Create complete documentation for storage structure
 - 4.03.09.03 Create complete documentation for access strategy
 - 4.03.09.04 Create complete documentation for data loading subsystem
 - 4.03.09.05 Create complete documentation for data update subsystem
 - 4.03.09.06 Create complete documentation for test data
 - 4.03.09.07 Create complete documentation for database backup procedure

- 4.03.10 Create or revise physical database implementation presentation
 - 4.03.10.01 Modify features, advantages and benefits (FAB) of database system
 - 4.03.10.02 Include FABs to illustrate physical database implementation
 - 4.03.10.03 Create narrative for physical database implementation
 - 4.03.10.04 Identify key individual for physical database implementation



- 4.03.10.05 Review physical database implementation & narrative with key individual and modify as appropriate
- 4.03.10.06 Create A/V materials for physical database implementation presentation
- 4.03.10.07 Review and finalize A/V materials prior to review

- 4.03.11 Conduct subphase review
- 4.03.11.01 Present physical database system implementation
- 4.03.11.02 Receive and respond to comments
- 4.03.11.03 Modify system as necessary
- 4.03.11.04 Seek concurrence with system subphase

- 4.04 Create or revise critical interrogation subsystems
- 4.04.01 Analyze implementation requirements & create schedule for specific interrogation subsystems

- 4.04.02 Analyze implementation requirements for specific interrogation subsystem

- 4.04.02.01 Identify specific implementation
- 4.04.02.01.01 Adjust generic interrogation requirements to transformed database
- 4.04.02.01.02 Adjust generic tables & elements to requirements of transformed database
- 4.04.02.01.03 Adjust generic data integrity rules to transformed database
- 4.04.02.01.04 Identify specific reports required for implementation
- 4.04.02.01.05 Create preliminary reports requirements statement for each identified report

- 4.04.02.02 Analyze interrogation subsystem complexity
- 4.04.02.02.01 Identify reports by number of required databases
- 4.04.02.02.02 Identify reports by outside data file requirements
- 4.04.02.02.03 Identify reports by database update in addition to reporting
- 4.04.02.02.04 Identify reports by sophistication of data calculation & processing
- 4.04.02.02.05 Identify reports by output format complexity
- 4.04.02.02.06 Create a report complexity matrix

- 4.04.02.03 Estimate resources required to create or revise subsystem reports
- 4.04.02.03.01 Identify & engineer standard approach to implement on-line programs so as to avoid repetitive analysis, specification, coding testing & documentation of common or functionally similar logic
- 4.04.02.03.02 Estimate resources required to create most simple reports
- 4.04.02.03.03 Estimate resources required to generate moderate reports
- 4.04.02.03.04 Estimate resources required to generate complex reports
- 4.04.02.03.05 Identify priority or importance for report
- 4.04.02.03.06 Create a priority & resource requirements matrix



Work Breakdown Structure

- 4.04.02.04 Determine implementation order for generating specific subsystems
 - 4.04.02.04.01 Present report priority & estimated resource requirements
 - 4.04.02.04.02 Determine implementation order
 - 4.04.02.04.03 Establish implementation schedule
- 4.04.03 Create or revise specific interrogation unit
 - 4.04.03.01 Create or revise unit minispecification
 - 4.04.03.01.01 Create an introductory section
 - 4.04.03.01.02 Include the required tables
 - 4.04.03.01.03 Include the required elements
 - 4.04.03.01.04 Include the data integrity rules that are required for inter-table navigation
 - 4.04.03.01.05 Create pseudocode or acceptable alternative that reflects transformations and processing steps
 - 4.04.03.01.06 Indicate inputs & formats
 - 4.04.03.01.07 Indicate outputs & formats
 - 4.04.03.01.08 Identify appropriate views
 - 4.04.03.02 Prototype unit components through a natural language
 - 4.04.03.02.01 Identify DBMS language that least inhibits report generation
 - 4.04.03.02.02 Translate minispecification into DBMS language
 - 4.04.03.02.03 Submit to DBMS & debug
 - 4.04.03.02.04 Demonstrate to users to validate design
 - 4.04.03.02.05 Project current resource expenditure to production volume
 - 4.04.03.02.06 Determine frequency of use & number of users for report
 - 4.04.03.02.07 Create validated design report
 - 4.04.03.03 Select final interrogation language for each unit component
 - 4.04.03.03.01 If interrogation module's life span exceeds that of DBMS choose ANSI standard language for module implementation
 - 4.04.03.03.02 If interrogation module's life span is within life span of DBMS, choose the most cost-effective DBMS natural language
 - 4.04.03.03.03 Create estimate for generations report
 - 4.04.03.04 Implement interrogation unit
 - 4.04.03.04.01 Program each minispecification
 - 4.04.03.04.02 Program data input file editing, processing, etc.
 - 4.04.03.04.03 Program any system control facilities needed for any database updates
 - 4.04.03.04.04 Walk through program code
 - 4.04.03.04.05 Validate appropriate interaction with database through views
 - 4.04.03.04.06 Conduct walk through of DBMS DML logic for efficiency and integrity



- 4.04.03.04.07 Create appropriate program documentation
- 4.04.03.05 Perform system development test on each interrogation unit
 - 4.04.03.05.01 Generate an appropriate set of test data to validate report
 - 4.04.03.05.02 Perform basic testing to validate report operation
 - 4.04.03.05.03 Validate all error processing & messages
 - 4.04.03.05.04 Construct resource consumption data on test database
 - 4.04.03.05.05 Extrapolate consumption to production database
 - 4.04.03.05.06 Determine whether estimates are within acceptable limits
- 4.04.03.06 Assess production impact
 - 4.04.03.06.01 Summarize interrogation resource consumption
 - 4.04.03.06.02 Identify schedule for execution
 - 4.04.03.06.03 Compute cyclical requirements estimates
 - 4.04.03.06.04 Factor in data update subsystem resource requirements
 - 4.04.03.06.05 Determine whether combined resources estimates are acceptable
 - 4.04.03.06.06 Determine which interrogation module consumes excessive resources and revise as appropriate
- 4.04.03.07 Develop interrogation unit implementation report
 - 4.04.03.07.01 Create standard documentation for each interrogation module/unit
 - 4.04.03.07.02 Create appropriate run/JCL instructions
 - 4.04.03.07.03 Summarize resource consumption requirements
 - 4.04.03.07.04 Summarize cyclical report requirements
 - 4.04.03.07.05 Coordinate interrogation module production requirements with update transaction data availability
- 4.04.04 Create interrogation implementation report
- 4.04.05 Create or revise interrogation implementation presentation
 - 4.04.05.01 Modify features, advantages and benefits (FAB) of database system
 - 4.04.05.02 Include FABs to illustrate interrogation implementation
 - 4.04.05.03 Graphically identify tables or processes that participate in each interrogation implementation
 - 4.04.05.04 Create narrative for each interrogation implementation
 - 4.04.05.05 Identify key individual for review of interrogation implementation
 - 4.04.05.06 Review interrogation implementation & narrative with key individual and modify as appropriate
 - 4.04.05.07 Create A/V materials for use interrogation implementation presentation
 - 4.04.05.08 Review and finalize A/V materials prior to review



Work Breakdown Structure

- 4.04.06 Conduct subphase review
- 4.04.06.01 Present interrogation implementation
- 4.04.06.02 Receive and respond to comments
- 4.04.06.03 Modify system as necessary

- 4.05 System control implementation
 - 4.05.01 Implement or revise fully supported DBMS facilities
 - 4.05.01.01 Determine mechanism to invoke facility
 - 4.05.01.02 Thoroughly validate facility with test database
 - 4.05.01.03 Estimate impact on production operations
 - 4.05.02 Implement or revise partially supported DBMS facilities
 - 4.05.02.01 Create design & specification for each program/module that must be implemented
 - 4.05.02.02 Determine impact on logical, physical & interrogation database software modules
 - 4.05.02.03 Estimate resources required for implementation
 - 4.05.02.04 Review design & resource estimate & schedule implementation
 - 4.05.02.05 Implement & test facilities against test database
 - 4.05.02.06 Estimate impact on production operations
 - 4.05.03 Acquire/implement and/or revise facilities not supported by DBMS
 - 4.05.03.01 Determine whether facility is available from other sources
 - 4.05.03.02 Create or revise, then implement & test facilities
 - 4.05.03.02.01 Create or revise design & specification for each program/module that must be implemented
 - 4.05.03.02.02 Determine impact on logical, physical & interrogation database software modules
 - 4.05.03.02.03 Estimate resources required for implementation
 - 4.05.03.02.04 Review design & resource estimate & schedule implementation
 - 4.05.03.02.05 Implement & test facilities against test database part of DBMS
 - 4.05.03.03 Check with DBMS vendor user group for facility availability
 - 4.05.03.04 Determine whether facility can be available on a timely basis
 - 4.05.03.05 Acquire & rigorously test facility
 - 4.05.03.06 Determine impact on production operations
 - 4.05.04 Create utilization scenarios & fully test all facilities
 - 4.05.04.01 Determine conditions that necessitate facility utilization
 - 4.05.04.02 Determine method or procedures for facility invocation
 - 4.05.04.03 For recurring facility usage establish schedule of utilization
 - 4.05.04.04 Create tests of facilities against various sizes & configurations of the test database



- 4.05.04.05 Review production impact & adjust as necessary
- 4.05.05 Create or revise operations documentation & guides
 - 4.05.05.01 Create complete operational documentation for each facility
 - 4.05.05.02 Create complete schedules for periodic facilities
 - 4.05.05.03 Adjust system production costs by resource requirements for recurring facilities
 - 4.05.05.04 Predict the number of occurrences of system control facilities that occur only reactively
 - 4.05.05.05 Create an overall production impact estimate, facility use procedures & schedule for use
- 4.05.06 Determine, implement and maintain security and privacy for implementation phase activities
 - 4.05.06.01 Determine security and privacy profiles for implementation phase users
 - 4.05.06.01.01 Identify various securable facilities, such as databases, run-units, views, etc.
 - 4.05.06.01.02 Create user profile for each users as needed
 - 4.05.06.01.03 Review security profile with security officer
 - 4.05.06.02 Implement user profiles to DBMS/security facility
 - 4.05.06.03 Maintain user profiles as needed.
- 4.05.07 Create or revise system control implementation report
 - 4.05.07.01 Review & adjust system control narratives from conceptual specification
 - 4.05.07.02 Create complete documentation for each system control component
 - 4.05.07.03 Finalize user documentation for facility invocation
 - 4.05.07.04 Update all resource consumption estimates for production operations
 - 4.05.07.05 Finalize format & documentation of the test database for system control testing
- 4.05.08 Create or revise system control implementation presentation
 - 4.05.08.01 Modify features, advantages and benefits (FAB) of database system
 - 4.05.08.02 Include FABs to illustrate system control implementation
 - 4.05.08.03 Create narrative for system control implementation
 - 4.05.08.04 Identify key individual for system control implementation review
 - 4.05.08.05 Review system control implementation & narrative with key individual and modify as appropriate
 - 4.05.08.06 Create A/V materials for system control implementation presentation
 - 4.05.08.07 Review and finalize A/V materials prior to review
- 4.05.09 Conduct subphase review
 - 4.05.09.01 Present system control implementation components



Work Breakdown Structure

- 4.05.09.02 Receive and respond to comments
- 4.05.09.03 Modify system as necessary
- 4.05.09.04 Seek concurrence with system subphase

- 4.06 Develop or revise ancillary supports
 - 4.06.01 Develop or revise database system training
 - 4.06.01.01 Identify source of database technology training
 - 4.06.01.02 Identify source of DBMS training

 - 4.06.01.03 Identify training required for various subsystem users
 - 4.06.01.03.01 Determine training needs for data loading subsystem users
 - 4.06.01.03.02 Determine training needs for data update subsystem users
 - 4.06.01.03.03 Determine training needs for data conversion subsystem users
 - 4.06.01.03.04 Determine training needs for system control facilities users
 - 4.06.01.03.05 Determine training needs for systems maintenance personnel
 - 4.06.01.03.06 Determine training needs for interrogation subsystem users
 - 4.06.01.03.06.01 Determine training needs for standard report users
 - 4.06.01.03.06.02 Determine training needs for ad hoc report users
 - 4.06.01.03.06.03 Determine training needs for DBMS natural language users
 - 4.06.01.04 Identify training required for general aspects of database project
 - 4.06.01.05 Create detailed training outline for each training module
 - 4.06.01.06 Develop training materials
 - 4.06.01.07 Test & revise training
 - 4.06.01.08 Create training schedule
 - 4.06.01.09 Write new user job descriptions
 - 4.06.01.10 Assist in planning hiring/transferring of user personnel
 - 4.06.01.11 Estimate user implementation manpower costs
 - 4.06.02 Develop or revise database hotline
 - 4.06.02.01 Identify the various groups that require technical assistance
 - 4.06.02.02 Identify the specific knowledge needed by each group
 - 4.06.02.03 Determine the availability of training to initially assist user group
 - 4.06.02.04 Determine the materials necessary to support the hotline groups
 - 4.06.02.05 Develop the training material needed for the hotline group
 - 4.06.02.06 Develop hotline group procedures
 - 4.06.02.07 Test & revise hotline personnel & procedures
 - 4.06.02.08 Create hotline operational plan
 - 4.06.03 Develop or revise database standards



- 4.06.03.01 Define rationale & benefits for standards
- 4.06.03.02 Propose standards for each database component
- 4.06.03.03 Review & revise standards as necessary to achieve complete integrated set
- 4.06.03.04 Review & revise standards to conform to organizational requirements
- 4.06.03.05 Devise method to constantly review standards for usability & applicability
- 4.06.03.06 Determine effect of standards on production operations
- 4.06.03.07 Create a final standards manual for database project

- 4.06.04 Develop or revise final test data
 - 4.06.04.01 Review & finalize test data for data update subsystem
 - 4.06.04.02 Review & finalize test data for data loading subsystems
 - 4.06.04.03 Review & finalize test data for backup
 - 4.06.04.04 Review & revise test data for major interrogation subsystems
 - 4.06.04.05 Review & revise test data for each system control component
 - 4.06.04.06 Review & revise test data for system prototype demonstration
 - 4.06.04.07 Review & revise test data for system prototype demonstration
 - 4.06.04.08 Create test data report & use procedures

- 4.06.05 Develop or revise database system documentation
 - 4.06.05.01 Develop internal database documentation
 - 4.06.05.01.01 Review & revise all conceptual documentation as affected by implementation
 - 4.06.05.01.01.01 metadata repository updates: mission descriptions, database domains, database objects, reports, business terms, policies, table, columns, data integrity rules, business information systems
 - 4.06.05.01.02 Review & revise all implementation documentation for clarity, correctness and uniformity
 - 4.06.05.01.02.01 metadata repository tables: schema, views, DBMS record, DBMS element, subsystem, program, module, etc.
 - 4.06.05.01.02.02 Inter-module logic control charts
 - 4.06.05.01.02.03 Module logic narrative and diagrams
 - 4.06.05.01.02.04 Program logic narrative & diagrams
 - 4.06.05.01.02.05 Subsystem logic narrative and diagrams
 - 4.06.05.01.02.06 Annotated computer software listings
 - 4.06.05.01.02.07 Messages and meanings
 - 4.06.05.01.02.08 Test data and testing procedures
 - 4.06.05.01.03 Review & revise all documentation relating to ancillary supports, especially hotline



Work Breakdown Structure

- 4.06.05.01.04 Develop strategy for documentation change control
- 4.06.05.01.05 Determine mechanism for documentation, duplication and change dissemination
- 4.06.05.01.06 Create documentation report

- 4.06.05.02 Prepare user manuals
 - 4.06.05.02.01 Prepare logical database user manuals
 - 4.06.05.02.02 Prepare data update operations manuals
 - 4.06.05.02.03 Prepare interrogation subsystem user manuals
 - 4.06.05.02.04 Prepare system control user manuals

- 4.06.05.03 Prepare operations documentation
 - 4.06.05.03.01 Create final procedures or JCL
 - 4.06.05.03.02 Create job or run narratives for data update, data loading, and interrogation subsystems
 - 4.06.05.03.03 Create job or run narratives for system control functions such as backup and recovery, security and privacy, reorganization, and the like
 - 4.06.05.03.04 Create job schedule requirements
 - 4.06.05.03.05 Create run setup instructions
 - 4.06.05.03.06 Create job cataloguing instructions
 - 4.06.05.03.07 Create off-line storage library instructions
 - 4.06.05.03.08 Assemble operations documentation package
 - 4.06.05.03.09 Review operations documents with operations manager and change documentation as required

- 4.06.05.04 Review & finalize all documentation
 - 4.06.05.04.01 Review all ancillary support products, that is, documentation, procedures, etc for correctness and the like
 - 4.06.05.04.02 Revise as necessary
 - 4.06.05.04.03 Prepare final products
 - 4.06.05.04.04 Prepare procedure for duplication & distribution
 - 4.06.05.04.05 Review procedure for change, control & products revision distribution

- 4.06.06 Prepare ancillary supports implementation report
 - 4.06.06.01 Create final statement on production operations impact
 - 4.06.06.02 Create overall schedule for ancillary supports for maintenance & administrative phase

- 4.06.07 Create or revise ancillary supports presentation
 - 4.06.07.01 Modify features, advantages and benefits (FAB) of database system
 - 4.06.07.02 Include FABs to illustrates ancillary supports utilization
 - 4.06.07.03 Develop appropriate documentation to illustrate use of ancillary supports



- 4.06.07.04 Identify key individual for review of ancillary supports
- 4.06.07.05 Review ancillary supports & narrative with key individual and modify as appropriate
- 4.06.07.06 Create A/V materials for use scenario presentation
- 4.06.07.07 Review and finalize A/V materials prior to review

- 4.06.09 Conduct subphase review
- 4.06.09.01 Present ancillary support components
- 4.06.09.02 Receive and respond to comments
- 4.06.09.03 Modify system as necessary
- 4.06.09.04 Seek concurrence with system subphase

- 4.07 Plan and conduct system quality tests (system quality test)

- 4.07.01 Create or revise system operation scenarios
- 4.07.01.01 Create scenario & schedule for initial data acquisition & database loading
- 4.07.01.02 Create scenario & schedule for a complete set of data updates
- 4.07.01.03 Create scenario & schedule for database interrogation subsystems
- 4.07.01.04 Create scenario & schedule for various database system control facilities
- 4.07.01.05 Create scenario & schedule for utilization of ancillary supports
- 4.07.01.06 Combine all scenarios & schedules into one overall proposed operations plan

- 4.07.02 Create or revise tests to validate scenarios
- 4.07.02.01 Identify artificial and production tests appropriate to validate correct production environment.
- 4.07.02.02 Identify method to acquire production environment or to use a subset without change
- 4.07.02.03 Create testing scenarios that simulate production environment
- 4.07.02.04 Identify backup and recovery procedures for system operation
- 4.07.02.05 Identify all ancillary supports and establish criteria for acceptance
- 4.07.02.06 Establish system performance and test success criteria
- 4.07.02.07 Identify all required operations, procedures, and guides that are to be evaluated
- 4.07.02.08 Formulate complete system quality test plan

- 4.07.03 Schedule database system quality test
- 4.07.03.01 Review all schedules, scenarios & tests, and build in delays due to process failures
- 4.07.03.02 Review proposed schedule to determine acceptability
- 4.07.03.03 Optimize schedule as necessary
- 4.07.03.04 Determine beginning date & time for system test



Work Breakdown Structure

- 4.07.04 Conduct system quality tests
 - 4.07.04.01 Acquire system quality test environment
 - 4.07.04.02 Accomplish database schema changes
 - 4.07.04.02.01 Create modified database
 - 4.07.04.02.02 Reorganize database as needed
 - 4.07.04.03 Install software into test environment
 - 4.07.04.04 Execute artificial tests
 - 4.07.04.05 Execute production tests
 - 4.07.04.06 If either tests fail, construct system quality test failure report
 - 4.07.04.07 If both tests pass, construct system quality test passes report
 - 4.07.04.08 If system quality test fails, restore pretest environment
 - 4.07.04.09 If system quality test passes, proceed with production environment modification
- 4.07.05 Prepare system test report
 - 4.07.05.01 Review all documentation & adjust as necessary
 - 4.07.05.02 Review all proposed operations, schedules & revise as necessary
 - 4.07.05.03 Review all proposal resource consumption estimates & revise as necessary
 - 4.07.05.04 Combine all documentation into one and revise as necessary
- 4.07.06 Create system test presentation
 - 4.07.06.01 Modify features, advantages and benefits (FAB) of database system
 - 4.07.06.02 Include FABs to illustrate system test
 - 4.07.06.03 Create narrative for system test
 - 4.07.06.04 Identify key individual for review of system test
 - 4.07.06.05 Review system test narrative with key individual and modify as appropriate
 - 4.07.06.06 Create A/V materials for system test presentation
 - 4.07.06.07 Review and finalize A/V materials prior to review
- 4.07.07 Conduct subphase review
 - 4.07.07.01 Present system test components
 - 4.07.07.02 Receive and respond to comments
 - 4.07.07.03 Modify system as necessary
 - 4.07.07.04 Seek concurrence with system subphase
- 4.08 Create implementation phase report
 - 4.08.01 Include the physical database implementation report
 - 4.08.02 Include the interrogation implementation report
 - 4.08.03 Include the system control implementation report
 - 4.08.04 Include the ancillary supports implementation report



- 4.08.05 Include the system test report

- 4.09 Create implementation phase presentation
 - 4.09.01 Modify features, advantages and benefits (FAB) of database system
 - 4.09.02 Include FABs to illustrate database system implementation
 - 4.09.03 Create narrative for database system implementation
 - 4.09.04 Identify key individual for review of database system implementation
 - 4.09.05 Review database implementation & narrative with key individual and modify as appropriate
 - 4.09.06 Create A/V materials for database system implementation presentation
 - 4.09.07 Review and finalize A/V materials prior to review

- 4.10 Conduct phase review
 - 4.10.01 Present implementation phase components
 - 4.10.02 Receive and respond to comments
 - 4.10.03 Modify system as necessary
 - 4.10.04 Seek concurrence with system subphase



5

Conversion & Deployment Phase

5 Conversion & Deployment

5.01 Form project team, plan phase, and revise project plan

5.01.01 Select manager for project phase

5.01.02 Identify administrative, clerical, and computer supports

5.01.03 Select project staff for phase

5.01.03.01 Interview and select database specialist

5.01.03.02 Interview and select functional users

5.01.04 Secure commitments for staff availability

5.01.04.01 Estimate time requirement for project manager

5.01.04.02 Estimate time requirement for administrative, clerical, and computer supports

5.01.05 Create phase plans and schedules

5.01.05.01 Create phase specific PERT charts

5.01.05.02 Create phase specific Gantt charts

5.01.05.03 Create phase CPM charts

5.01.05.04 Create phase manpower estimate charts

5.01.06 Organize phase review committee

5.01.07 Conduct methods training as required

5.01.08 Set up phase documentation files

5.01.09 Revise project plan and schedule estimates

5.01.09.01 Revise PERT charts

5.01.09.02 Revise Gantt charts

5.01.09.03 Revise CPM charts

5.01.09.04 Revise manpower estimate charts

5.01.10 Develop and/or use database standards for phase

5.01.11 Interrogate and maintain metadata repository during phase

5.02 Review prior phase for completeness and revise as necessary

5.03 Secure, install and test all equipment

5.04 Conduct all training

5.04.01 Conduct user training

5.04.02 Conduct system maintenance training

5.04.03 Conduct data transfer and data conversion user training

5.05 Convert all data

- 5.06 Create conversion and deployment report
- 5.07 Create conversion and deployment review presentation
 - 5.07.01 Modify features, advantages and benefits (FAB) of database system
 - 5.07.02 Include FABs to illustrate database system implementation
 - 5.07.03 Create narrative for database system implementation review
 - 5.07.04 Identify key individual for review of conversion and deployment review
 - 5.07.05 Review database conversion and deployment review & narrative with key individual and modify as appropriate
 - 5.07.06 Create A/V materials for conversion and deployment presentation
 - 5.07.07 Review and finalize A/V materials prior to review
- 5.08 Conduct phase review
 - 5.08.01 Present conversion and deployment review
 - 5.08.02 Receive and respond to comments
 - 5.08.03 Modify system as necessary
 - 5.08.04 Seek concurrence with system phase
- 5.09 Prepare estimate for production and administration phase
- 5.10 Prepare project production & administration phase funding
- 5.11 Present for funding review





6

Production and Administration Phase

6 Production & Administration

6.01 Form project team, plan phase, and revise project plan

6.01.01 Select manager for project phase

6.01.02 Identify administrative, clerical, and computer supports

6.01.03 Select project staff for preliminary analysis phase

6.01.03.01 Interview and select database specialist

6.01.03.02 Interview and select functional users

6.01.04 Secure commitments for staff availability

6.01.04.01 Estimate time requirement for project manager

6.01.04.02 Estimate time requirement for administrative, clerical, and computer supports

6.01.05 Create phase plans and schedules

6.01.05.01 Create phase specific PERT charts

6.01.05.02 Create phase specific Gantt charts

6.01.05.03 Create phase CPM charts

6.01.05.04 Create phase manpower estimate charts

6.01.06 Organize phase review committee

6.01.07 Conduct methods training as required

6.01.08 Set up phase documentation files

6.01.09 Revise project plan and schedule estimates

6.01.09.01 Revise PERT charts

6.01.09.02 Revise Gantt charts

6.01.09.03 Revise CPM charts

6.01.09.04 Revise manpower estimate charts

6.01.10 Develop and/or use database standards for phase

6.01.11 Interrogate and maintain metadata repository during phase

6.02 Database system production

6.02.01 Commence database system

6.02.01.01 Begin operation of all ancillary supports

6.02.01.02 Create database structure

6.02.01.03 Perform initial database load

6.02.01.04 Begin operation of data update subsystems

6.02.01.05 Begin operation of interrogation subsystems

6.02.01.06 Begin operation of various system control supports

Work Breakdown Structure

- 6.02.02 Conduct ongoing database system production
 - 6.02.02.01 Perform cyclical updates
 - 6.02.02.02 Perform ad hoc updates
 - 6.02.02.03 Perform cyclical reports
 - 6.02.02.04 Perform ad hoc reports
 - 6.02.02.05 Conduct ongoing training
 - 6.02.02.06 Provide hotline services
 - 6.02.02.07 Perform system control supports
 - 6.02.02.08 Operate database system
- 6.02.03 Develop and maintain views as necessary
- 6.02.04 Develop and maintain security and privacy facilities as needed
- 6.02.05 Conduct parallel operations as appropriate
- 6.02.06 Create production commencement report
- 6.02.07 Create production commencement presentation
- 6.02.08 Perform production commencement review
- 6.03 Develop interrogation subsystems
 - 6.03.01 Create requirements specifications
 - 6.03.01.01 Create an introductory section detailing problem addressed, scope, etc.
 - 6.03.01.02 Include the required tables
 - 6.03.01.03 Include the required elements
 - 6.03.01.04 Include the data integrity rules that are required for inter-table navigation
 - 6.03.01.05 Include a statement of transformation & processing steps
 - 6.03.01.06 Indicate inputs & formats
 - 6.03.01.07 Indicate outputs & formats
 - 6.03.02 Create views
 - 6.03.02.01 Identify view requirements for each interrogation requirement and program as appropriate
 - 6.03.02.02 Create documentation for each view
 - 6.03.02.03 Create relationships among views, database object information systems, database object table processes and data integrity rules as appropriate
 - 6.03.03 Prototype through a natural language
 - 6.03.03.01 Identify DBMS language that least inhibits report generation
 - 6.03.03.02 Translate mini-specification into DBMS language
 - 6.03.03.03 Submit to DBMS & debug
 - 6.03.03.04 Demonstrate to users to validate design
 - 6.03.03.05 Project current resource expenditure to production volume
 - 6.03.03.06 Determine frequency of use & number of users for report



- 6.03.03.07 Create validated design report
- 6.03.04 Select cost efficient language
 - 6.03.04.01 If interrogation module's life span exceeds that of DBMS choose ANSI standard language for module implementation
 - 6.03.04.02 If interrogation module's life span is within life span of DBMS, choose the most cost-effective DBMS natural language
 - 6.03.04.03 Create estimate for generations report
- 6.03.05 Implement interrogation subsystem
 - 6.03.05.01 Program each mini-specification
 - 6.03.05.02 Program data input file editing, processing, etc.
 - 6.03.05.03 Program any system control facilities needed for any database updates
 - 6.03.05.04 Walk through program code
 - 6.03.05.05 Validate appropriate interaction with database through views
 - 6.03.05.06 Conduct walk through of DBMS DML logic for efficiency and integrity
 - 6.03.05.07 Create appropriate program documentation
- 6.03.06 Perform system development test on interrogation subsystem
 - 6.03.06.01 Generate an appropriate set of test data to validate report
 - 6.03.06.02 Perform basic testing to validate report operation
 - 6.03.06.03 Validate all error processing & messages
 - 6.03.06.04 Construct resource consumption data on test database
 - 6.03.06.05 Extrapolate consumption to production database
 - 6.03.06.06 Determine whether estimates are within acceptable limits
- 6.03.07 Assess production impact
 - 6.03.07.01 Summarize interrogation resource consumption
 - 6.03.07.02 Identify schedule for execution
 - 6.03.07.03 Compute cyclical requirements estimates
 - 6.03.07.04 Factor in data update subsystem resource requirements
 - 6.03.07.05 Determine whether combined resources estimates are acceptable
 - 6.03.07.06 Determine which interrogation module consumes excessive resources
- 6.03.08 Develop interrogation implementation report
 - 6.03.08.01 Create standard documentation for interrogation module/subsystem
 - 6.03.08.02 Create appropriate run/JCL instructions
 - 6.03.08.03 Summarize resource consumption requirements
 - 6.03.08.04 Summarize cyclical report requirements
 - 6.03.08.05 Coordinate interrogation module production requirements with update transaction data availability



Work Breakdown Structure

- 6.03.09 Create interrogation implementation presentation
 - 6.03.09.01 Modify features, advantages and benefits (FAB) of database system
 - 6.03.09.02 Include FABs to illustrate interrogation implementation
 - 6.03.09.03 Graphically identify tables or processes that participate in each interrogation implementation
 - 6.03.09.04 Create narrative for each interrogation implementation
 - 6.03.09.05 Identify key individual for review of interrogation implementation
 - 6.03.09.06 Review interrogation implementation & narrative with key individual and modify as appropriate
 - 6.03.09.07 Create A/V materials for use interrogation implementation presentation
 - 6.03.09.08 Review and finalize A/V materials prior to review
- 6.03.10 Conduct interrogation review
 - 6.03.10.01 Present interrogation implementation
 - 6.03.10.02 Receive and respond to comments
 - 6.03.10.03 Modify system as necessary
- 6.04 Application Optimization Assessment
 - 6.04.01 Assess physical database performance
 - 6.04.01.01 Assess data online and batch update for ease of use and acceptable performance
 - 6.04.01.02 Assess data batch data loading for ease of use and acceptable performance
 - 6.04.01.03 Formulate strategy for improving performance
 - 6.04.01.04 Determine acceptability of eliminating indexes
 - 6.04.03.05 Create physical database performance assessment report
 - 6.04.02 Assess interrogation performance
 - 6.04.02.01 Assess the ease of use of interrogation
 - 6.04.02.02 Assess the performance of interrogation
 - 6.04.02.03 Determine requirement to add indexes
 - 6.04.02.04 Formulate strategy for improving ease of use and performance
 - 6.04.03.05 Create interrogation performance assessment report
 - 6.04.03 Assess system control performance
 - 6.04.03.01 Assess audit trail performance
 - 6.04.03.01.01 Assess acceptability at audit trail record content
 - 6.04.03.01.02 Assess acceptability of capture capability from all update sources & modes
 - 6.04.03.01.03 Assess acceptability of audit trail reports
 - 6.04.03.01.04 Assess acceptability of audit trail logical transactions within a multiple user environment



- 6.04.03.01.05 Assess acceptability of audit trail transactions within a multiple database environment
- 6.04.03.01.06 Assess audit trail backup & recovery
- 6.04.03.01.07 Assess audit trail capability to reorganize its record content
- 6.04.03.01.08 Assess impact on data loading transactions
- 6.04.03.01.09 Assess impact on data update transactions
- 6.04.03.01.10 Assess impact on interrogation transactions
- 6.04.03.01.11 Prepare audit trail acceptability report

- 6.04.03.02 Assess performance of backup & recovery
 - 6.04.03.02.01 Assess the acceptability of basic design
 - 6.04.03.02.02 Assess the adequacy of backup & recovery for batch type jobs
 - 6.04.03.02.03 Assess the adequacy of backup & recovery for on-line update
 - 6.04.03.02.04 Assess the adequacy of backup & recovery for logical transactions on multiple databases
 - 6.04.03.02.05 Assess the adequacy of backup & recovery for high volume multi-user update in batch & on-line
 - 6.04.03.02.06 Assess the adequacy of backup & recovery for the backup & recovery capabilities
 - 6.04.03.02.07 Assess the impact on data loading transactions
 - 6.04.03.02.08 Assess the impact on data update transactions
 - 6.04.03.02.09 Prepare backup & recovery acceptability report

- 6.04.03.03 Assess performance of message processing
 - 6.04.03.03.01 Examine messages to determine their understandability
 - 6.04.03.03.02 Determine whether message types can automatically trigger DBA created messages
 - 6.04.03.03.03 Determine whether source & user identity of certain messages can be logged
 - 6.04.03.03.04 Determine whether message content can be changed or augmented
 - 6.04.03.03.05 Determine whether error processing can be enabled & disabled for certain run units
 - 6.04.03.03.06 Determine impact on data update transactions
 - 6.04.03.03.07 Determine impact on data loading transactions
 - 6.04.03.03.08 Determine impact on interrogation transactions
 - 6.04.03.03.09 Prepare messages acceptability report

- 6.04.03.04 Assess completeness and performance of security & privacy
 - 6.04.03.04.01 Assess acceptability of DBMS mechanism to establish & maintain security & privacy to determine ease of use
 - 6.04.03.04.02 Reevaluate the DBMS ability to protect security & privacy from O/S utility tampering
 - 6.04.03.04.03 Reevaluate the security & privacy ability to protect row, column & relationship



Work Breakdown Structure

- 6.04.03.04.04 Reevaluate the security & privacy ability to protection the basis of values, conditions, etc.
- 6.04.03.04.05 Reevaluate the security & privacy ability to meet specific needs stated in the conceptual specifications
- 6.04.03.04.06 Assess impact on data update transactions
- 6.04.03.04.07 Assess impact on data loading transactions
- 6.04.03.04.08 Assess impact on interrogation transactions
- 6.04.03.04.09 Prepare security & privacy acceptability report

- 6.04.03.05 Assess performance of concurrent operations
- 6.04.03.05.01 Assess DBMS capabilities to perform multiple update commands within all languages
- 6.04.03.05.02 Assess DBMS capabilities to perform multiple updates to the same database
- 6.04.03.05.03 Assess DBMS capability to perform multiple updates to the same row
- 6.04.03.05.04 Assess DBMS resources that are consumed to effect various concurrent access controls
- 6.04.03.05.05 Determine impact on production operations
- 6.04.03.05.06 Prepare concurrent operations acceptability report

- 6.04.03.06 Assess sophistication of reorganization
- 6.04.03.06.01 Assess DBMS capabilities to logically restructure a table to delete, modify or add elements
- 6.04.03.06.02 Assess DBMS capabilities to physically resequence rows within a table
- 6.04.03.06.03 Assess DBMS capability to resize storage structure components without database reloading
- 6.04.03.06.04 Assess DBMS capability to add, change, or modify inter-table relationships
- 6.04.03.06.05 Assess DBMS reorganization impact on production applications
- 6.04.03.06.06 Prepare reorganization capability report

- 6.04.03.07 Assess sophistication of multiple database object processing
- 6.04.03.07.01 Assess capability to operate multiple databases within one DBMS copy
- 6.04.03.07.02 Assess capability to operate multiple databases from various DBMS languages
- 6.04.03.07.03 Assess capability to operate database editions from DBMS languages
- 6.04.03.07.04 Assess capability for multiple database update transaction audit trail logs
- 6.04.03.07.05 Assess capability for multiple database backup & recovery & other system control facilities
- 6.04.03.07.06 Assess impact on production operations
- 6.04.03.07.07 Prepare multiple database object processing capabilities report

- 6.04.03.08 Create system control assessment report
- 6.04.03.08.01 Incorporate audit trail section



- 6.04.03.08.02 Incorporate message processing section
- 6.04.03.08.03 Incorporate security & privacy section
- 6.04.03.08.04 Incorporate database recovery section
- 6.04.03.08.05 Incorporate concurrent operations section
- 6.04.03.08.06 Incorporate multiple database section
- 6.04.03.08.07 Incorporate reorganization section

- 6.04.04 Create logical database change requirements to accommodate performance
 - 6.04.04.01 Identify requirement to denormalize database structures
 - 6.04.04.01 Identify requirements for additional derived data
 - 6.04.04.01 Identify requirements for additional data integrity rules

- 6.04.05 Create consolidated application performance change requirements report
 - 6.04.05.01 Create logical database change requirements section
 - 6.04.05.02 Create physical database change requirements section
 - 6.04.05.03 Create interrogation change requirements section
 - 6.04.05.04 Create system control change requirements section

- 6.04.06 Create assessment to change ancillary support for application performance change requirements
 - 6.04.06.01 Assess requirements to revise database training
 - 6.04.06.02 Assess requirements to revise hotline
 - 6.04.06.03 Assess requirements to revise database standards
 - 6.04.06.04 Assess requirements to revise test data

- 6.04.07 Create consolidated application performance change requirements presentation
 - 6.04.07.01 Modify features, advantages and benefits (FAB) of database system
 - 6.04.07.02 Include FABs to create application performance change requirements impact scenarios
 - 6.04.07.03 Create narrative for application performance change requirements scenario
 - 6.04.07.04 Identify key individual for review of application performance change requirements scenario
 - 6.04.07.05 Review application performance change requirements scenario & narrative with key individual and modify as appropriate
 - 6.04.07.06 Create A/V materials for application performance change requirements scenario presentation
 - 6.04.07.07 Review and finalize A/V materials prior to review

- 6.04.08 Conduct subphase review
 - 6.04.08.01 Present application performance change requirements assessment
 - 6.04.08.02 Receive and respond to comments



Work Breakdown Structure

- 6.04.08.03 Modify application performance change requirements as necessary
- 6.04.08.04 Seek concurrence with application performance change requirements

- 6.05 Determine database system maintenance policy
 - 6.05.01 Select maintenance approach, methods, and tools
 - 6.05.01.01 Determine maintenance approach
 - 6.05.01.01.01 Assign maintenance to development team
 - 6.05.01.01.02 Assign maintenance to special system maintenance team
 - 6.05.01.01.03 Assign maintenance to maintenance pool
 - 6.05.01.02 Establish maintenance methods
 - 6.05.01.02.01 Make ad hoc changes
 - 6.05.01.02.02 Make batches of changes
 - 6.05.01.02.03 Make homogeneous change batches
 - 6.05.01.02.04 Make changes on determined time cycles
 - 6.05.01.03 Identify and select maintenance tools
 - 6.05.01.03.01 Use formal application change procedures
 - 6.05.01.03.02 Use super zap type tools
 - 6.05.01.04 Document selections and formalize findings
 - 6.05.02 Determine procedures for standard and emergency maintenance
 - 6.05.02.01 Review and revise existing procedures or develop new ones
 - 6.05.02.02 Develop cases or scenarios to validate procedures
 - 6.05.02.03 Establish control and feedback mechanisms
 - 6.05.02.03.01 Identify metadata repository reports for component changes
 - 6.05.02.03.02 Identify appropriate forms and reports for maintenance control and reporting
 - 6.05.02.03.03 Review and revise call lists for emergency and standard maintenance
 - 6.05.02.03.04 Establish maintenance review committees
 - 6.05.02.03.05 Review, revise, and publish control and reporting mechanisms
 - 6.05.02.04 Publish standard and emergency maintenance procedures
 - 6.05.03 Publish maintenance policy
- 6.06 Perform standard system maintenance



- 6.06.01 Identify and assign maintenance problem
 - 6.06.01.01 Identify maintenance need
 - 6.06.01.01.01 Review error reports
 - 6.06.01.01.02 Solicit new requirements
 - 6.06.01.01.03 Propose enhancements
 - 6.06.01.01.04 Respond to government regulations
 - 6.06.01.02 Document maintenance need
 - 6.06.01.02.01 Document conditions surrounding need
 - 6.06.01.02.02 Specify precisely need database objectives
 - 6.06.01.02.03 Indicate affected or related projects
 - 6.06.01.02.04 Identify metadata repository components affected
 - 6.06.01.02.05 Create maintenance need document
 - 6.06.01.03 Analyze the maintenance need
 - 6.06.01.03.01 Obtain need evidence
 - 6.06.01.03.02 Organize evidence into cases or scenarios
 - 6.06.01.03.03 Classify as familiar or unfamiliar
 - 6.06.01.03.04 Investigate unfamiliar to uncover possible causes
 - 6.06.01.03.05 Compare known against other known problem areas
 - 6.06.01.03.06 Develop cause or solution hypothesis
 - 6.06.01.03.07 Test hypothesis
 - 6.06.01.03.08 Document the cause
 - 6.06.01.04 Identify proper person/organization to solve problem
 - 6.06.01.04.01 Assign to general management if policy problem
 - 6.06.01.04.02 Assign to user of data entry or quality problem
 - 6.06.01.04.03 Assign to applications maintenance if application problem
 - 6.06.01.04.04 Assign to systems software if operating system type problem
 - 6.06.01.04.05 Assign to vendor if problem is package related
- 6.06.02 Analyze problem and estimate maintenance effort
 - 6.06.02.01 Analyze problem, determine solution and work around
 - 6.06.02.01.01 Use metadata repository to identify components to change
 - 6.06.02.01.02 Review operations guides to identify needed changes
 - 6.06.02.01.03 Review training manuals or guides to identify needed changes
 - 6.06.02.01.04 Review computer support manuals to identify needed changes
 - 6.06.02.01.05 Review standards, policy, or procedures to identify needed changes
 - 6.06.02.01.06 Develop complete change material and review/revise as appropriate



Work Breakdown Structure

- 6.06.02.01.07 Develop problem work around
- 6.06.02.02 Determine pervasiveness and issue alert as appropriate
 - 6.06.02.02.01 Determine problem impact on other systems
 - 6.06.02.02.02 Develop system impact statement
 - 6.06.02.02.03 Perform risk analysis to assess special fix or general solution
 - 6.06.02.02.04 For general solutions, develop plan
 - 6.06.02.02.05 For special fix, develop plan
 - 6.06.02.02.06 Develop notice for affected users
- 6.06.02.03 Determine alternative of choice and obtain approval
 - 6.06.02.03.01 Identify alternative
 - 6.06.02.03.02 Evaluate cost-benefit of alternative
 - 6.06.02.03.02.01 Determine cost to implement
 - 6.06.02.03.02.02 Determine cost to operate
 - 6.06.02.03.02.03 Determine cost to convert
 - 6.06.02.03.02.03.01 Determine data conversion costs
 - 6.06.02.03.02.03.02 Determine program conversion costs
 - 6.06.02.03.02.03.03 Determine ancillary supports conversion costs
 - 6.06.02.03.02.04 Determine annual benefits
 - 6.06.02.03.02.05 Determine rate of return or cash flow rate of return
- 6.06.02.03.03 Accept or reject alternative
- 6.06.02.04 Set priority of implementation
- 6.06.02.05 Notify proponent of schedule of change
- 6.06.03 Accomplish the maintenance change
 - 6.06.03.01 Create conceptual specification of change
 - 6.06.03.01.01 Estimate effort for conceptual specification of change
 - 6.06.03.01.02 Identify entity instances that need to be changed
 - 6.06.03.01.02.01 Select and review mission description for change
 - 6.06.03.01.02.02 Select and review database domain for change
 - 6.06.03.01.02.03 Select and review business term for change
 - 6.06.03.01.02.04 Select and review database objects for change



- 6.06.03.01.02.05 Select and review columns for change
- 6.06.03.01.02.06 Select and review tables for change
- 6.06.03.01.02.07 Select and review tables for change
- 6.06.03.01.02.08 Select and review data integrity rules for change
- 6.06.03.01.02.09 Select and review information system processes for change
- 6.06.03.01.02.10 Select and review primitive data transformations for change

- 6.06.03.01.03 Develop specific change transaction
- 6.06.03.01.04 Evaluate entity instance change with respect to other applications use of instance
- 6.06.03.01.05 Specify additional entity instances to accomplish change
- 6.06.03.01.06 Formulate change transaction

- 6.06.03.01.07 Identify ancillary support changes
 - 6.06.03.01.07.01 Specify changes to training
 - 6.06.03.01.07.02 Specify changes to test data
 - 6.06.03.01.07.03 Specify changes to operational procedures
 - 6.06.03.01.07.04 Specify changes to external documentation
 - 6.06.03.01.07.05 Specify changes to hotline
 - 6.06.03.01.07.06 Create consolidated supports changes specification

- 6.06.03.01.08 Develop consolidated change document

- 6.06.03.02 Create implementation specification of change
 - 6.06.03.02.01 Estimate effort for implementation specification

 - 6.06.03.02.02 Identify metadata that need to be changed
 - 6.06.03.02.02.01 Select and review schema DDL for change
 - 6.06.03.02.02.02 Select and review schema tables for change
 - 6.06.03.02.02.03 Select and review schema columns for change
 - 6.06.03.02.02.04 Select and review subsystems for change
 - 6.06.03.02.02.05 Select and review programs for change
 - 6.06.03.02.02.06 Select and review modules for change

 - 6.06.03.02.03 Identify corresponding implemented system component to change
 - 6.06.03.02.03.01 Identify and review schemas for change
 - 6.06.03.02.03.02 Identify and review views for change
 - 6.06.03.02.03.03 Identify and review programs for change
 - 6.06.03.02.03.04 Identify and review modules for change
 - 6.06.03.02.03.05 Identify and review job control languages for change
 - 6.06.03.02.03.06 Identify and review non database files for change
 - 6.06.03.02.03.07 Identify and review test data for change



Work Breakdown Structure

- 6.06.03.02.03.08 Identify and review internal documentation for change
- 6.06.03.02.03.09 Identify and review system development tests for change
- 6.06.03.02.04 Develop specific change proposals
- 6.06.03.02.05 Evaluate change with respect to other application users
- 6.06.03.02.06 If compatible change, formulate changes
- 6.06.03.02.07 If not compatible change, formulate additional components to accomplish change
- 6.06.03.02.08 Specify changes to ancillary supports
 - 6.06.03.02.08.01 Specify changes to training materials and procedures
 - 6.06.03.02.08.02 Specify changes to test data
 - 6.06.03.02.08.03 Specify changes to operational procedures
 - 6.06.03.02.08.04 Specify changes to external documentation
 - 6.06.03.02.08.05 Specify changes to hotline methods and practices
 - 6.06.03.02.08.06 Compile ancillary support change specifications
- 6.06.03.02.09 Develop consolidated change document
- 6.06.03.03 Implement the change
 - 6.06.03.03.01 Plan the change process
 - 6.06.03.03.02 Operate current system until change occurs
 - 6.06.03.03.03 Obtain needed access to software and data for change
 - 6.06.03.03.04 Acquire software change environment
 - 6.06.03.03.05 Develop system development test requirements
 - 6.06.03.03.05.01 Specify requirements for functional testing
 - 6.06.03.03.05.01.01 Develop checklist for conformance of metadata repository changes
 - 6.06.03.03.05.01.02 Develop checklist for conformance of system component changes
 - 6.06.03.03.05.01.03 Develop checklist for conformance of ancillary support changes
 - 6.06.03.03.05.01.04 Consolidate functional check lists
 - 6.06.03.03.05.02 Develop success criteria for all tests
 - 6.06.03.03.05.03 Specify plan for regression testing
 - 6.06.03.03.05.04 Specify plan to investigate economy, efficiency, and effectiveness
 - 6.06.03.03.05.05 Specify plan for stress testing
 - 6.06.03.03.05.06 Create a consolidate SDT plan and document
- 6.06.03.03.06 Accomplish the change
 - 6.06.03.03.06.01 Accomplish metadata repository changes
 - 6.06.03.03.06.02 Accomplish system component changes



- 6.06.03.03.06.03 Accomplish ancillary support changes
- 6.06.03.03.06.04 Create system development test testing package
- 6.06.03.04 Perform system development test on change
- 6.06.03.04.01 Validate conformance to functional requirements
- 6.06.03.04.02 Perform unit, integration, & system testing
 - 6.06.03.04.02.01 Create environment for computer testing
 - 6.06.03.04.02.02 Create test data sets for tests
 - 6.06.03.04.02.03 Validate that unit performs function as specified
 - 6.06.03.04.02.04 Accomplish integration testing
 - 6.06.03.04.02.05 Accomplish system tests
- 6.06.03.04.03 Perform regression testing
- 6.06.03.04.04 Perform stress testing
- 6.06.03.04.05 Perform economy, efficiency, and effectiveness testing
- 6.06.03.04.06 Prepare system development test report
- 6.06.03.05 Deliver the change
 - 6.06.03.05.01 Create consolidated change document
 - 6.06.03.05.02 Prepare and or package all deliverables prior to system development test
 - 6.06.03.05.03 Create notification of change completion
- 6.06.04 Plan and conduct system quality tests
 - 6.06.04.01 Plan the system quality tests
 - 6.06.04.01.01 Identify artificial and production tests appropriate to validate correct production environment.
 - 6.06.04.01.02 Identify method to acquire production environment or to use a subset without change
 - 6.06.04.01.03 Create testing scenarios that simulate production environment
 - 6.06.04.01.04 Identify backup and recovery procedures for system operation
 - 6.06.04.01.05 Identify all ancillary supports and establish criteria for acceptance
 - 6.06.04.01.06 Establish system performance and test success criteria
 - 6.06.04.01.07 Identify all required operations, procedures, and guides that are to be evaluated
 - 6.06.04.01.08 Formulate complete system quality test plan
 - 6.06.04.02 Conduct system quality tests
 - 6.06.04.02.01 Acquire system quality test environment
 - 6.06.04.02.02 Accomplish database schema changes



Work Breakdown Structure

- 6.06.04.02.02.01 Create modified database
- 6.06.04.02.02.02 Reorganize database as needed
- 6.06.04.02.03 Install software into test environment
- 6.06.04.02.04 Execute artificial tests
- 6.06.04.02.05 Execute production tests
- 6.06.04.02.06 If either tests fail, construct system quality test failure report
- 6.06.04.02.07 If both tests pass, construct system quality test passes report
- 6.06.04.03 If system quality test fails, restore pretest environment
- 6.06.04.04 If system quality test passes, proceed with production environment modification
- 6.06.05 Conduct required training
 - 6.06.05.01 Establish schedule for classes
 - 6.06.05.02 Determine needed facilities and equipment
 - 6.06.05.03 Accomplish training
 - 6.06.05.04 Obtain reviews of training
- 6.06.06 Install change into production
 - 6.06.06.01 Determine materials to change
 - 6.06.06.02 Create change package
 - 6.06.06.03 Distribute release materials
 - 6.06.06.04 Install change into production library
 - 6.06.06.05 Delete unnecessary facilities
 - 6.06.06.06 Finalize changed system documentation
 - 6.06.06.06.01 Finalize metadata repository changes
 - 6.06.06.06.02 Finalize training materials changes
 - 6.06.06.06.03 Finalize external documentation changes
 - 6.06.06.06.04 Finalize internal documentation changes
- 6.06.07 Monitor database application
 - 6.06.07.01 Set or revise monitoring goals and database objectives
 - 6.06.07.02 Set or revise standards of performance
 - 6.06.07.02.01 Set or revise standards for hotline
 - 6.06.07.02.02 Set or revise standards for training
 - 6.06.07.02.03 Set or revise standards for documentation
 - 6.06.07.02.04 Set or revise standards for system operation
 - 6.06.07.03 Establish mechanisms of measurement and feedback



- 6.06.07.04 Monitor system operation
- 6.06.07.05 Analyze reported problems

6.07 Perform emergency maintenance

- 6.07.01 Document the problem
- 6.07.02 Verify the problem through independent methods
- 6.07.03 Determine solution to problems
- 6.07.04 Acquire approval for change
- 6.07.05 Change run-time modules
- 6.07.06 Modify source code and create database object for production library
- 6.07.07 Install change
- 6.07.08 Create standard maintenance change request

6.08 Accomplish database system revision

- 6.08.01 Propose database system revision
 - 6.08.01.01 Define new/modified functions by subsystem
 - 6.08.01.02 Define scope of existing subsystem modifications
 - 6.08.01.03 Generally define the extent of logical database changes
 - 6.08.01.04 Generally define the extent of physical database changes
 - 6.08.01.05 Generally define the extent of system control changes
 - 6.08.01.06 Generally define the extent of documentation and training changes
 - 6.08.01.07 Estimate the revision release schedule
 - 6.08.01.07.01 Logical database changes
 - 6.08.01.07.02 Physical database changes
 - 6.08.01.07.03 System control releases
 - 6.08.01.07.04 Ancillary supports
 - 6.08.01.08 Create database system revision proposal presentation
 - 6.08.01.08.01 Identify features, advantages and benefits (FAB) of database revision
 - 6.08.01.08.02 Include FABs to justify database system revision
 - 6.08.01.08.03 Create narrative for database system revision review
 - 6.08.01.08.04 Identify key individual for review
 - 6.08.01.08.05 Review database revision with key individual and modify as appropriate
 - 6.08.01.08.06 Create A/V materials for database system revision presentation
 - 6.08.01.08.07 Review and finalize A/V materials
 - 6.08.01.09 Conduct revision proposal review
 - 6.08.01.09.01 Present database system revision
 - 6.08.01.09.02 Receive and respond to comments



Work Breakdown Structure

- 6.08.01.09.03 Modify revision as necessary
- 6.08.01.09.04 Seek concurrence for revision work

- 6.08.01.10 Prepare estimate for revision
- 6.08.01.11 Prepare revision project funding document
- 6.08.01.12 Present for funding review

- 6.08.02 Prepare implementation strategy of revision
 - 6.08.02.01 Identify implementation alternatives
 - 6.08.02.02 Determine Gantt/PERT/CPM charts for each alternative
 - 6.08.02.03 Present alternatives for review and selection
 - 6.08.02.04 Consolidate Gantt, PERT, and CPM charts for selected alternatives
 - 6.08.02.05 Prepare system proposal for implementation phase
 - 6.08.02.06 Create implementation approach presentation
 - 6.08.02.07 Conduct subphase review

- 6.08.03 Implement revision
 - 6.08.03.01 Implement revised logical database
 - 6.08.03.01.01 Transform revised database
 - 6.08.03.01.02 Record details in transformation specification
 - 6.08.03.01.03 Create logical database implementation report
 - 6.08.03.01.04 Create logical database implementation presentation
 - 6.08.03.01.05 Conduct subphase review
 - 6.08.03.02 Implement revised physical database
 - 6.08.03.02.01 Determine appropriate storage structure
 - 6.08.03.02.02 Determine access strategy
 - 6.08.03.02.03 Revise DDL, storage structure DDL & views
 - 6.08.03.02.04 Revise or design, then implement & test data update subsystem
 - 6.08.03.02.05 Revise or design, then implement & test data loading subsystem
 - 6.08.03.02.06 Revise data integrity subsystem
 - 6.08.03.02.07 Revise test data
 - 6.08.03.02.08 Revise database backup procedures
 - 6.08.03.02.09 Perform system development test on physical database
 - 6.08.03.02.10 Create physical database implementation report
 - 6.08.03.02.11 Create physical database implementation presentation
 - 6.08.03.02.12 Conduct subphase review
 - 6.08.03.03 Implement interrogation subsystem revisions
 - 6.08.03.03.01 Create views
 - 6.08.03.03.02 Prototype through a natural language



- 6.08.03.03.03 Select cost efficient language
- 6.08.03.03.04 Implement interrogation subsystem
- 6.08.03.03.05 Perform system development test on interrogation subsystem
- 6.08.03.03.06 Assess production impact
- 6.08.03.03.07 Develop interrogation implementation report
- 6.08.03.03.08 Create interrogation implementation presentation
- 6.08.03.03.09 Conduct interrogation review

- 6.08.03.04 Accomplish system control changes
 - 6.08.03.04.01 Revise fully supported DBMS facilities
 - 6.08.03.04.01.01 Determine mechanism to invoke facility
 - 6.08.03.04.01.02 Thoroughly validate facility with test database
 - 6.08.03.04.01.03 Estimate impact on production operations
 - 6.08.03.04.02 Revise partially supported DBMS facilities
 - 6.08.03.04.02.01 Create design & specification for each program/module that must be implemented
 - 6.08.03.04.02.02 Determine impact on logical, physical & interrogation database software modules
 - 6.08.03.04.02.03 Estimate resources required for implementation
 - 6.08.03.04.02.04 Review design & resource estimate & schedule implementation
 - 6.08.03.04.02.05 Implement & test facilities against test database
 - 6.08.03.04.02.06 Estimate impact on production operations
 - 6.08.03.04.03 Revise facilities not supported by DBMS
 - 6.08.03.04.03.01 Determine whether facility is available from other sources
 - 6.08.03.04.03.02 Revise, then implement & test facilities
 - 6.08.03.04.03.02.01 Create or revise design & specification for each program/module that must be implemented
 - 6.08.03.04.03.02.02 Determine impact on logical, physical & interrogation database software modules
 - 6.08.03.04.03.02.03 Estimate resources required for implementation
 - 6.08.03.04.03.02.04 Review design & resource estimate & schedule implementation
 - 6.08.03.04.03.02.05 Implement & test facilities against test database part of DBMS
 - 6.08.03.04.04 Revise utilization scenarios & fully test all facilities
 - 6.08.03.04.04.01 Determine conditions that necessitate facility utilization
 - 6.08.03.04.04.02 Determine method or procedures for facility invocation
 - 6.08.03.04.04.03 For recurring facility usage establish schedule of utilization



Work Breakdown Structure

- 6.08.03.04.04.04 Create tests of facilities against various sizes & configurations of the test database
- 6.08.03.04.04.05 Review production impact & adjust as necessary
- 6.08.03.04.05 Revise operations documentation & guides
 - 6.08.03.04.05.01 Revise operational documentation for each facility
 - 6.08.03.04.05.02 Revise schedules for periodic facilities
 - 6.08.03.04.05.03 Adjust system production costs by resource requirements for recurring facilities
 - 6.08.03.04.05.04 Predict the number of occurrences of system control facilities that occur only reactively
 - 6.08.03.04.05.05 Revise an overall production impact estimate, facility use procedures & schedule for use
- 6.08.03.04.06 Create revised system control implementation report
- 6.08.03.04.07 Create revised system control implementation presentation
- 6.08.03.04.08 Conduct subphase review
- 6.08.04 Revise ancillary supports
 - 6.08.04.01 Revise database training
 - 6.08.04.02 Revise database hotline
 - 6.08.04.03 Revise database standards
 - 6.08.04.04 Revise test data
 - 6.08.04.05 Develop database system documentation
 - 6.08.04.05.01 Develop internal database documentation
 - 6.08.04.05.02 Prepare user manuals
 - 6.08.04.06 Prepare operations documentation
 - 6.08.04.07 Review & finalize all materials
 - 6.08.04.08 Prepare ancillary supports implementation reports
 - 6.08.04.09 Create ancillary supports presentation
 - 6.08.04.10 Conduct subphase review
- 6.08.05 Plan and conduct system quality tests
 - 6.08.05.01 Plan the system quality tests
 - 6.08.05.01.01 Develop artificial and production tests appropriate to validate correct operation within production environment
 - 6.08.05.01.02 Identify method to acquire production environment or to use a subset without change



- 6.08.05.01.03 Create testing scenarios that simulate production environment
- 6.08.05.01.04 Identify backup and recovery procedures for system operation
- 6.08.05.01.05 Identify all ancillary supports and establish criteria for acceptance
- 6.08.05.01.06 Establish system performance and test success criteria
- 6.08.05.01.07 Identify all required operations, procedures, and guides that are to be evaluated
- 6.08.05.01.08 Formulate complete system quality test plan

- 6.08.05.02 Conduct system quality tests
 - 6.08.05.02.01 Acquire system quality test environment

 - 6.08.05.02.02 Accomplish database schema changes
 - 6.08.05.02.02.01 Create modified database
 - 6.08.05.02.02.02 Reorganize database as needed

 - 6.08.05.02.03 Install software into production environment
 - 6.08.05.02.04 Execute artificial tests
 - 6.08.05.02.05 Execute production tests
 - 6.08.05.02.06 If either tests fail, construct system quality test failure report
 - 6.08.05.02.07 If both tests pass, construct system quality test passes report

- 6.08.05.03 If system quality test fails, restore pretest environment
- 6.08.05.04 If system quality test passes, proceed with production environment modification

- 6.08.06 Prepare software release notice
 - 6.08.06.01 Create release numeric identifier
 - 6.08.06.02 Create release name if appropriate
 - 6.08.06.03 Determine exact release date
 - 6.08.06.04 Create summary of release
 - 6.08.06.05 Describe each new function or revision
 - 6.08.06.06 Describe each modification
 - 6.08.06.07 Describe the DDL changes
 - 6.08.06.08 Set down the release schedule for DDL, software, and data availability
 - 6.08.06.09 Enumerate the modifications or re-releases of documentation for user guides, external and internal documents

- 6.08.07 Merge revision into production system
 - 6.08.07.01 Merge software into production environment
 - 6.08.07.02 Perform database DDL modifications as necessary
 - 6.08.07.03 Perform database reloads as necessary
 - 6.08.07.04 Perform data conversions as necessary



Work Breakdown Structure

- 6.08.08 Perform enterprise model audit on evolution
- 6.08.09 Conduct revision review

- 6.09 Perform enterprise database audit
 - 6.09.01 Audit metadata repository components and note discrepancies by identifying the items's unique identifier and suspected problem
 - 6.09.01.01 Audit tasks
 - 6.09.01.01.01 Review each task to determine conformance to the Whitemarsh methodology
 - 6.09.01.01.02 Review task descriptions, deliverable names and descriptions for correctness
 - 6.09.01.02 Audit task assignments
 - 6.09.01.02.01 Review assignments for completion dates, deliverable produced, and realistic estimates
 - 6.09.01.02.02 Review task assignment estimates against actual charges to determine unacceptable deviations
 - 6.09.01.03 Audit charges
 - 6.09.01.03.01 Review charges in conjunction with task assignments, and tasks to determine whether charge descriptions match tasks and assignments
 - 6.09.01.03.02 Make sure that persons submitting charges have submitted backup time sheets, and are valid employees
 - 6.09.01.03.03 If the charges are for deliverables, require the identification of the deliverables to validate the appropriateness of the charges
 - 6.09.01.04 Audit persons
 - 6.09.01.04.01 Review all person records for correct data, spelling, non duplicates, and validity for project work
 - 6.09.01.05 Audit information system process
 - 6.09.01.05.01 Review each information system process for correct definition, naming, etc. for relevance to actual business activities that occur
 - 6.09.01.05.02 Review all subordinate events, and the tree structure of nested information system processes to ensure that each information system process is complete, and well defined
 - 6.09.01.05.03 Follow all information system processes to their terminal information system process and verify correct utilization of database object processes and views.
 - 6.09.01.06 Audit business terms
 - 6.09.01.06.01 Review each term for correct spelling, definition, abbreviation, and the like



- 6.09.01.06.02 Spot check the documentation, DBMS DDL, software source code for correct utilization, especially for abbreviations
- 6.09.01.07 Audit data integrity rules
 - 6.09.01.07.01 Review data integrity rules for correct definition, naming, and description
 - 6.09.01.07.02 Review columns and tables referenced by the Data integrity rule for correct reference
 - 6.09.01.07.03 Review computer modules--in data integrity rule correctly referenced for correct language specification
 - 6.09.01.07.04 Create data that violates a Data integrity rule and run an appropriate program to test the effect. if the transaction is rejected, then the Data integrity rule is working
- 6.09.01.08 Audit mission descriptions and mission description diagrams
 - 6.09.01.08.01 Review the mission descriptions for correct name and definition
 - 6.09.01.08.02 Review the mission description diagrams for correct names, symbols, and symbol interactions
- 6.09.01.09 Audit database domains
 - 6.09.01.09.01 Review the database domains for correct name and definition
- 6.09.01.10 Audit data elements
 - 6.09.01.10.01 Review for correct naming, definition, and example especially as it relates to the mission description
 - 6.09.01.10.02 Identify all data elements not related to database table columns and justify inclusion
 - 6.09.01.10.03 Verify that all element integrity rules are properly reflected in schema DDL or data update modules
 - 6.09.01.10.04 Make sure that all data elements are allocated to at least one database domain
 - 6.09.01.10.05 Verify that appropriate need category is identified
 - 6.09.01.10.07 Verify that if there is a related data element value domain that it is correctly defined and its values are correct
- 6.09.01.11 Audit database objects
 - 6.09.01.11.01 Review for correct definition, naming, etc. review for correct utilization within the scope of a database domain
 - 6.09.01.11.02 Make sure all database objects are allocated to at least one business database domain
 - 6.09.01.11.03 Audit database object tables
 - 6.09.01.11.03.01 Identify all database object tables relate to the database object for correct allocation



Work Breakdown Structure

- 6.09.01.11.03.02 Verify that example is appropriate
- 6.09.01.11.03.03 Verify that properties are correctly and appropriately defined
- 6.09.01.11.03.04 Verify that appropriate need category is identified

- 6.09.01.11.04 Audit database object table processes
 - 6.09.01.11.04.01 Ensure that all database processes for database structure segment existence tests are acceptable
 - 6.09.01.11.04.01 Ensure that all database processes for individual database structure column tests are acceptable
 - 6.09.01.11.04.01 Ensure that all database processes for database structure segment referential integrity tests are acceptable

- 6.09.01.11.05 Audit database object databases information systems
 - 6.09.01.11.05.01 Ensure that database object information systems retain database integrity or rollback
 - 6.09.01.11.05.02 Ensure that all database object table processes are properly included in database object business informations systems

- 6.09.01.11.06 Audit database object states
 - 6.09.01.11.06.01 Ensure that all business resource life cycle states are identified
 - 6.09.01.11.06.02 Ensure that all database object information systems are properly allocated

- 6.09.01.12 Audit tables
 - 6.09.01.12.01 Review tables for correct naming, definition, and the like
 - 6.09.01.12.02 Verify that each table corresponds to an appropriate database object and database domain
 - 6.09.01.12.03 Verify that the elements allocated to the table cause it to be in third normal form. if not, then justify nested structures. review load, update, and delete modules to verify that noun acceptable modifications occur
 - 6.09.01.12.04 Verify that all tables related to DBMS tables
 - 6.09.01.12.05 Verify that table example is appropriate

- 6.09.01.13 Audit columns
 - 6.09.01.13.01 Review columns for correct naming, definition, etc. verify that correct elements, tables, and tables are referenced
 - 6.09.01.13.02 Verify that one or more elements are correctly identified as a primary key or is part of a primary key
 - 6.09.01.13.03 Make sure that each element's definition is a subordinate definition of a table, and is included in a related database object's properties
 - 6.09.01.13.04 Review related Data integrity rule's for correct inclusion
 - 6.09.01.13.05 Review the element's function group for appropriate allocation



- 6.09.01.14 Audit database domain diagrams
 - 6.09.01.14.01 Review identified entities and relationships proper naming, and the like
 - 6.09.01.14.02 Ensure that each database domain diagrams maps to an appropriate database domain and in turn to an appropriate mission descriptions
- 6.09.01.15 Audit inward and/or outward file specifications
 - 6.09.01.15.01 Review each file's file elements to ensure that the file elements accurately reflect the file generation and/or reading programs used for generation/access
 - 6.09.01.15.02 Review each file element to ensure that the file element is correctly identified and named
 - 6.09.01.15.03 Check that the file elements are correctly mapped to the appropriate external view columns, database view columns, views, column, and table.
 - 6.09.01.15.04 Check that the file elements are utilizing the proper edit and validation checks
 - 6.09.01.15.05 Check that the inward or outward data conversions for file elements are correctly specified and are programmed correctly
- 6.09.01.16 Audit edit and validation tables
 - 6.09.01.16.01 Review and validate each table for correct values and meanings
 - 6.09.01.16.02 Identify the element that uses the table and make sure that the values are appropriate
 - 6.09.01.16.03 Review all the tables to find duplicates and recommend deletions where appropriate
- 6.09.01.17 Audit user identifiers
 - 6.09.01.17.01 Review user identifiers for appropriate description
 - 6.09.01.17.02 Review library of files for each user identifier and clear out unnecessary ones if possible
 - 6.09.01.17.03 Drop user identifiers that are no longer being used
- 6.09.01.18 Audit reports
 - 6.09.01.18.01 Review each report's report elements to ensure that the report elements accurately reflect the report generation
 - 6.09.01.18.02 Review each report element to ensure that the report element is correctly identified and named
 - 6.09.01.18.03 Check that the report elements are correctly mapped to the appropriate external view columns, database view columns, views, column, and table.
 - 6.09.01.18.04 Check that the report elements are utilizing the proper edit and validation checks
 - 6.09.01.18.05 Check that the outward data conversions for report elements are correctly specified and are programmed correctly



Work Breakdown Structure

- 6.09.01.18.01 Review the report's name, description, headers, footers, control breaks and all calculated report elements that defines the report to ensure that the data produced accurately reflects the headings
- 6.09.01.19 Audit policies
 - 6.09.01.19.01 Review the policy name, description, and proper citation
 - 6.09.01.19.02 Review the database objects, mission descriptions, database domains, and elements to ensure appropriate references
- 6.09.01.19 Audit screens
 - 6.09.01.20.01 Review each screen's screen elements to ensure that they accurately reflect the screen programs used for data entry
 - 6.09.01.20.02 Review each screen element to ensure that the screen element is correctly identified and named, and if applicable the appropriate help is displayed when invoked
 - 6.09.01.20.03 Check that the screen elements are correctly mapped to the appropriate external view columns, database view columns, views, column, and table.
 - 6.09.01.20.04 Check that the screen elements are utilizing the proper edit and validation checks
 - 6.09.01.20.05 Check that the inward or outward data conversions for screen elements are correctly specified and are programmed correctly
- 6.09.01.21 Audit schemas
 - 6.09.01.21.01 Review the name, description, DBMS name, and mission description referenced by the schema for accuracy and clarity
 - 6.09.01.21.02 Examine all identified DBMS tables for correct relationships
- 6.09.01.22 Audit DBMS tables
 - 6.09.01.22.01 Review the name, DBMS name, and schema references
 - 6.09.01.22.02 Examine the identified tables of correct relationships to specification records
 - 6.09.01.22.03 Verify that the columns allocated to the table cause it to be in third normal form. if not, then justify nested structures. review load, update, and delete modules to verify that unacceptable modifications occur
- 6.09.01.23 Audit DBMS columns
 - 6.09.01.23.01 Review all DBMS columns inclusions to verify correct naming, data type, picture, and the like
 - 6.09.01.23.02 Verify that the picture is compatible with the DBMS column picture
 - 6.09.01.23.03 Access the related DBMS table, and specification DBMS columns to verify appropriate references
- 6.09.01.24 Audit subsystems



- 6.09.01.24.01 Review the name and description of the subsystem to assess correctness
- 6.09.01.24.02 Access related programs to determine if their overall purpose matches that of the subsystem
- 6.09.01.24.03 Access related mission description to determine appropriateness

- 6.09.01.25 Audit modules
- 6.09.01.25.01 Review the name and description of the module to assess correctness
- 6.09.01.25.02 Access the related programs to determine whether the module is actually present in the source code
- 6.09.01.25.03 Access the DBMS tables to determine if the module deals with the referenced tables
- 6.09.01.25.04 Access the primitive data transformations to determine whether the module's purpose corresponds to that of the database object process's
- 6.09.01.25.05 Access the actual module source code to review the database object process requirements to the module's source language
- 6.09.01.25.06 If the module contains a Data integrity rule, access the Data integrity rule through column to determine the module is correctly identified as containing the Data integrity rule

- 6.09.01.26 Audit programs
- 6.09.01.26.01 Review the name, description, and language of the program to assess correctness
- 6.09.01.26.02 Access related modules to determine whether the description of the program is supported by the module descriptions
- 6.09.01.26.03 Access the subsystem description to assess whether the program's purpose corresponds to that of the subsystem
- 6.09.01.26.04 Access the program's code to determine whether all the modules are actually present

- 6.09.02 Audit non-metadata repository data and computer programs

- 6.09.02.01 Audit program listings
- 6.09.02.01.01 Determine that each program is identified by the metadata repository
- 6.09.02.01.02 Review each program for adequate flow documentation and normalized top-down structure
- 6.09.02.01.03 Evaluate program for adequate cohesion, coupling, and the like
- 6.09.02.01.04 Make sure each program-local variable is defined within the program
- 6.09.02.01.05 Review the program's organization for ANSI standard language, and overall clear and concise documentation
- 6.09.02.01.06 If job control files are required to operate the program, make sure that these JCL files are properly referenced from within the program, and the general description of the purpose of the JCL is included



- 6.09.02.01.07 Whenever a program uses a non-database table and/or file, its name, purpose, structure, and general content should be clearly stated
- 6.09.02.01.08 Whenever a program uses a test data set its name, purpose, structure, and general content should be clearly stated
- 6.09.02.02 Audit JCL listings
 - 6.09.02.02.01 Review each listing for clarity, comments, and clear statement of purpose, use, and logic
 - 6.09.02.02.02 Ensure that each JCL listing contains references to the programs and/or data files that are to be used
 - 6.09.02.02.03 Whenever a JCL procedure uses a test data set its name, purpose, structure, and general content should be clearly stated
- 6.09.02.03 Audit database DDL
 - 6.09.02.03.01 Cross check each database DDL against that which is contained in the metadata repository
 - 6.09.02.03.02 Cross check any schema-based integrity clauses for column and row integrity with element and data integrity rules
- 6.09.02.04 Audit views
 - 6.09.02.04.01 Cross check each view that is contained in the metadata repository
- 6.09.02.05 Audit non database tables and files
 - 6.09.02.05.01 Review each identified (from programs and JCL) for correct name, purpose, content, and structure that is implied by the program or JCL file
 - 6.09.02.05.02 Identify all other tables and files and determine why it has not been properly identified as belonging to a program or a JCL file
 - 6.09.02.05.03 Access the metadata repository to determine that all tables and files are properly defined within the metadata repository
- 6.09.02.06 Audit test data sets
 - 6.09.02.06.01 Review each identified (from programs and JCL) for correct name, purpose, content, and structure that is implied by the program or JCL file
 - 6.09.02.06.02 Identify all other test data sets and determine why it has not been properly identified as belonging to a program or a JCL file
 - 6.09.02.06.03 Access the metadata repository to determine that all test data sets are properly defined within the metadata repository
- 6.09.02.07 Audit source library and version generation system
 - 6.09.02.07.01 The structure, content, and linguistic organization of the source library should be thoroughly evaluated for clarity, and orthogonality



- 6.09.02.07.02 Create tests of software source with already known results to determine adequacy of the procedures for testing the addition, deletion, and modification of data integrity rules, modules, programs, nondatabase data files, etc
- 6.09.03 Audit ancillary support items
 - 6.09.03.01 Audit database documentation
 - 6.09.03.01.01 Database documentation titles, descriptions, dates and versions as they exist in the metadata repository should be reviewed to verify conformance to the actual publications
 - 6.09.03.01.02 The complete collection of database documentation should be evaluated to determine its adequacy
 - 6.09.03.01.03 The procedure by which database documentation is gathered should be examined to determine whether the most current documents are collected and whether the collection process is comprehensive and complete
 - 6.09.03.01.04 Once the collection is produced, it should be evaluated for overall consistency, format, and clarity
 - 6.09.03.02 Audit user manuals
 - 6.09.03.02.01 User documentation titles, descriptions, dates and versions as they exist in the metadata repository should be reviewed to verify conformance to the actual publications
 - 6.09.03.02.02 The procedure by which user documentation is generated should be examined to determine whether the most current information is included and whether the material is comprehensive and complete
 - 6.09.03.03 Audit user help documentation
 - 6.09.03.03.01 The procedure by which user help documentation is generated should be examined to determine whether the most current information is included and whether the material is comprehensive and complete
 - 6.09.03.03.02 Once the collection is produced, it should be evaluated for overall consistency, format, and clarity
 - 6.09.03.04 Audit operations documentation
 - 6.09.03.04.01 Operations documentation titles, descriptions, dates and versions as they exist in the metadata repository should be reviewed to verify conformance to actual publications
 - 6.09.03.04.02 The complete collection of operations documentation should be evaluated to determine its adequacy



- 6.09.03.04.03 The procedure by which operations documentation is gathered should be examined to determine whether the most current documents are collected and whether the collection process is comprehensive and complete
- 6.09.03.04.04 Once the collection is produced, it should be evaluated for overall consistency, format, and clarity
- 6.09.03.04.05 For all procedures that accomplish normal system operation, that is, data entry, update, and reporting should be followed--exactly--to determine accuracy and operations efficiency
- 6.09.03.04.06 For all procedures that accomplish abnormal system operations: security maintenance, backup and recovery, reorganization, database crash recovery, etc. should be followed--exactly--to determine accuracy and operations efficiency

- 6.09.03.05 Audit internal system and subsystem design documentation
- 6.09.03.05.01 Internal system and subsystem documentation titles, descriptions, dates and versions as they exist in the metadata repository should be reviewed to verify conformance to the actual publications
- 6.09.03.05.02 The complete collection of system and subsystem documentation should be evaluated to determine its adequacy
- 6.09.03.05.03 The documentation should indicate flow, interconnections, database and nondatabase data requirements, physical device requirements, printer outputs, and the like

- 6.09.03.06 Audit hotline documentation, policies and procedures
- 6.09.03.06.01 Hotline documentation policies and procedures titles, descriptions, dates and versions as they exist in the metadata repository should be reviewed to verify conformance to the actual publications
- 6.09.03.06.02 The complete collection of hotline documentation should be evaluated to determine its adequacy
- 6.09.03.06.03 The procedure by which hotline documentation is gathered should be examined to determine whether the most current documents are collected and whether the collection process is comprehensive and complete
- 6.09.03.06.04 Once the collection is produced, it should be evaluated for overall consistency, format, and clarity
- 6.09.03.06.05 All hotline procedures responding to normal operation questions: data entry, update, and reporting should be reviewed through typical scenarios to ensure adequate hotline personnel training, access to appropriate materials, etc
- 6.09.03.06.06 The procedures that log, classify, and account for hotline calls should be reviewed to determine whether the hotline reporting forms provide information regarding database system adequacy, training sufficiency, performance, etc

- 6.09.03.07 Audit training materials for database, DBMS, and system use:



- 6.09.03.07.01 Training materials, titles, descriptions, dates and versions as they exist in the metadata repository should be reviewed to verify conformance to the actual publications
- 6.09.03.07.02 Student evaluations should be reviewed to identify training problem areas, for example, lab assignments, and lectures
- 6.09.03.07.03 Hotline call sheets should be reviewed to discover problem areas that might be addressed by training

- 6.09.03.08 Audit system specification procedures
- 6.09.03.08.01 The system specification, implementation, and operation procedures should be reviewed to determine adequacy, completeness, clarity, and efficiency
- 6.09.03.08.02 Evaluated should be the process of analyzing requirements, data entry, operations, report production, user training, and the like

- 6.09.03.09 Audit system development tests
- 6.09.03.09.01 Each system development test for each type of software, procedure, test data, training materials documentation, etc., should be reviewed to ensure that the testing procedures are valid, discriminating, and reliable
- 6.09.03.09.02 system quality test results should be reviewed to determine whether pass/fail performance reflects on the adequacy of the SDTs.

- 6.09.03.10 Audit system quality tests
- 6.09.03.10.01 Each system quality test for each type of software, procedure, test data, training materials documentation, etc., should be reviewed to ensure that the testing procedures are valid, discriminating, and reliable
- 6.09.03.10.02 The process of notification of the testers should be reviewed to determine sufficiency of notice and a full awareness of the items being tested.
- 6.09.03.10.03 Enterprise model audit results should be reviewed to determine whether pass/fail performance reflects on the adequacy of the sdts.

- 6.09.03.11 Audit system audits
- 6.09.03.11.01 The review the system audit procedure, and based on the results, modifications to the system specification, implementation, and operation procedures accomplished to lessen the quantity of enterprise model audit failures

- 6.09.04 Formulate audit report
- 6.09.04.01 For specifications or documentation, indicate the type of revision required
- 6.09.04.02 For data, test files, and tables, formulate change requirement
- 6.09.04.03 For procedures, formulate scope of necessary change to procedure and test plan
- 6.09.04.04 For computer program changes, formulate types of changes required, including specification of test data, and test plan



Work Breakdown Structure

- 6.09.04.05 Create combined change document for submittal as standard system maintenance tasks
- 6.09.05 Monitor database system operation
 - 6.09.05.01 Set or revise monitoring goals and database objectives
 - 6.09.05.02 Set or revise standards of performance
 - 6.09.05.02.01 Set or revise standards for hotline
 - 6.09.05.02.02 Set or revise standards for training
 - 6.09.05.02.03 Set or revise standards for documentation
 - 6.09.05.02.04 Set or revise standards for system operation
 - 6.09.05.03 Establish mechanisms of measurement and feedback
 - 6.09.05.04 Monitor system operation
 - 6.09.05.05 Generate system use statistics
 - 6.09.05.06 Analyze reported problems
- 6.09.06 Certify enterprise model audit pass/fail
 - 6.09.06.01 Review discrepancy reports from audit
 - 6.09.06.02 Review reported problems from system operation
 - 6.09.06.03 If no problems, then test passes
 - 6.09.06.04 If problems, then determine action plan for remediation and submit problems to standard system maintenance procedure



INDEX

Access Strategy	63, 64, 74, 104
ANSI	76, 91, 114
Application optimization	92
Audit trail	42, 44, 56, 58, 92-95
Audit Trails	9, 42, 53, 65
Backup and Recovery	43, 53, 65, 82, 83, 101, 107, 116
Benchmark	55
Binding	51, 66, 67
Binding Phase	51, 66, 67
Business Event	10, 38, 48
Business Event Model	38
Business Function	6-8, 10, 38, 39, 48
Business Function Model	38
Business Information Systems	2, 7, 8, 11-13, 29, 38, 81
Business Mission	3, 4
Business Policy	3, 4, 6, 25
Business Requirements	15, 29
Business term	99
Business Terms	4, 6, 81, 109
Calc	63
Chain	71
Column	19-21, 23, 27, 46, 57, 94, 110-114
Conceptual Specification Phase	13, 15, 48, 49, 51
Concurrent Operations	9, 43, 44, 53, 57, 65, 72, 94, 95
Conversion and Deployment	87
Corporate Business Model	51, 54, 66
Data Conversion	3, 16, 21, 26, 27, 35, 41, 43, 47, 73, 74, 80, 86, 98
Data element	5, 17-19, 29, 110
Data element name	18
Data Elements	5, 8, 17-19, 26, 52, 109
Data file	75
Data integrity	8, 10, 17, 18, 21-24, 26-30, 34-36, 39, 40, 44, 45, 48, 52, 62-64, 69, 75, 76, 81, 90, 95, 99, 104, 109, 111, 113-115
Data integrity model	10, 17, 26-29, 45, 48, 52
Data integrity rule	22-24, 27-29, 109, 113
Data Loading	10, 33, 34, 45, 46, 48, 52, 56, 57, 59, 64, 70, 72-74, 80-82, 92-94, 104
Data security	43
Data Structure	22
Data type	113
Data Update Subsystem	10, 31, 33, 48, 52, 64, 70, 74, 77, 80, 81, 91, 104
Database domain	4, 5, 10, 16-18, 21, 28, 29, 47, 48, 52, 99, 109-111
Database Domain Submodel	4



Database object . . .	5, 16, 17, 19-26, 28-30, 32, 34-37, 39, 58, 65, 90, 94, 95, 103, 109, 110, 113
Database object information system	24, 25
Database Object Information Systems	25, 90, 110
Database object process	113
Database object state	25
Database Object States	110
Database Project	2, 8, 80, 81
Database view	16, 22, 27, 111, 112
DBMS	9, 45, 46, 54-59, 62-66, 71-74, 76-81, 90, 91, 94, 105, 106, 109, 111-113, 117
DBMS column	113
DBMS table	113
Derived data	18, 21, 24, 26, 27, 95
Document	16, 38, 47, 54, 59, 60, 96, 97, 99-101, 103, 104, 118
Domain	4, 5, 10, 16-18, 21, 26, 28, 29, 47, 48, 52, 99, 109-111
Enterprise	3, 35, 108, 118
Extent	47, 64, 103
External view	16, 22, 27, 31, 32, 35, 37, 111, 112
File	3, 16, 27, 31, 32, 34, 35, 37, 46, 70, 75, 77, 91, 111, 114, 115
File cell	31, 32, 35, 37
Form	1, 15, 16, 20, 26, 35, 37, 38, 46, 51, 69, 74, 86, 89, 110, 113
Glossary	7, 18, 52
Hierarchy	38
Hotline	53, 80-82, 90, 95, 99, 100, 103, 106, 116-118
Implementation Phase	25, 31, 33-35, 37, 40-42, 66, 67, 69, 72, 73, 79, 85, 104
Implementation Strategy	62, 64-67, 104
Implemented data model	46, 56, 59, 69, 70
Information system	8-13, 24, 25, 29-40, 52, 99, 109
Information system control logic	31, 32, 34, 35, 37
Information Systems Plan	7, 13
Internal process	31, 32, 35, 37
Interrogation	9-11, 40-42, 45-49, 53, 54, 56, 57, 59, 61, 64, 66, 70, 75-78, 80-83, 85, 89-95, 105, 106
Job	1, 31, 33, 38, 80, 82, 100, 114
Life Cycle	7, 8, 11-13, 25, 110
Logical Database . .	8, 10, 11, 15, 28, 29, 39, 40, 42, 45-49, 51, 52, 54, 59, 61-63, 66, 69, 82, 95, 103, 104
Logical Database Reorganization	45
Message Processing	56, 93, 95
Metabase	2, 8, 25
Metadata . . .	2-8, 15-19, 21-24, 26, 27, 31, 32, 35, 37-39, 51, 59-61, 69, 81, 86, 89, 96-102, 108, 114-117



Metadata Repository	2-8, 15-19, 21-24, 26, 27, 31, 32, 35, 37-39, 51, 59-61, 69, 81, 86, 89, 96-98, 100-102, 108, 114-117
Mission	3, 4, 7-9, 13, 16, 29, 48, 52, 81, 99, 109, 111-113
Mission description	3, 4, 7-9, 16, 29, 48, 52, 99, 109, 112, 113
Mission description model	3, 7
Object	5, 16, 17, 19-26, 28-30, 32, 34-37, 39, 58, 65, 90, 94, 95, 103, 109, 110, 113
Operational data model	70
Ordered	63
Organization	6, 38, 39, 97, 114, 115
Owner-member relationship	72
Physical Database	8-11, 29, 39, 40, 46-49, 52-54, 59, 61-63, 66, 70, 74, 75, 85, 92, 95, 103-105
Preliminary Analysis	1, 7, 13-15, 89
Primary key	19, 20, 37, 46, 111
Primitive data transformations	52, 99, 113
Project	1-3, 8, 11-13, 15, 16, 18, 28, 40, 44, 51, 69, 76, 80, 81, 86, 87, 89, 90, 104, 108
Project Management	1, 11, 12
Property class	5
Prototype	46, 47, 54, 61, 71, 73, 74, 76, 81, 90, 105
Referential integrity	110
Relational algebra	22, 24, 27, 28
Reorganization	9, 43-45, 53, 57, 58, 82, 94, 95, 116
Report	7-11, 13-17, 26-28, 33, 39-44, 46-49, 52-58, 60, 62, 64-67, 72-77, 79-82, 84, 85, 87, 90-95, 101, 102, 104-107, 112, 117, 118
Repository	2-8, 15-19, 21-24, 26, 27, 31, 32, 35, 37-39, 51, 59-61, 69, 81, 86, 89, 96-98, 100-102, 108, 114-117
Resource	1, 2, 7, 8, 10-13, 25, 43, 44, 49, 71-74, 76-79, 84, 90, 91, 105, 106, 110
Restore	84, 102, 107
Retention	20
Role	22, 24, 27, 39
Rollback	70, 71, 110
Row	21-24, 27, 43, 57, 72, 73, 94, 114
Schema	2, 45, 46, 69, 81, 84, 99, 102, 107, 109, 112, 114
Screen	30-32, 34-37, 112
Screen element	30-32, 34-37, 112
Security and Privacy	65, 79, 82, 90
Segment	21, 110
Specified data model	26
Staff	1, 11, 15, 39, 51, 69, 86, 89
Subordinate business function	6, 8
Subordinate database domain	5
Subordinate mission description	8, 29



System Quality Tests	53, 83, 84, 101, 102, 107, 117
Table . . . v, 17, 19-24, 26-28, 30, 32, 34-37, 39, 41, 43, 45, 46, 57, 58, 62-64, 69, 72, 73, 76, 81, 90, 94, 109-114	
Table look up	30, 34, 36
Technical Terms	7
Test Data	46, 47, 53, 71-74, 77, 81, 82, 91, 95, 99-101, 104-106, 114, 115, 117, 118
Third normal form	20, 110, 113
Tree structure	109
Trigger	38, 57, 93
User . . . 2, 3, 9, 15, 26, 29, 32, 36, 43, 46, 47, 56, 57, 78-80, 82, 86, 93, 97, 106, 108, 111, 112, 115-117	
User Training	86, 117
Value-based	46, 62
Value-based relationships	46, 62
Warehouse Databases	27

