



Whitemarsh  
Information Systems Corporation

## Database Objects, The Foundation Stones of Enterprise Database

Whitemarsh Information Systems Corporation  
2008 Althea Lane  
Bowie, Maryland 20716  
Tele: 301-249-1142  
Email: [mmgorman@wiscorp.com](mailto:mmgorman@wiscorp.com)  
Web: [www.wiscorp.com](http://www.wiscorp.com)

## Table of Contents

Class Topic Outline .....	i
1 Why Database Objects .....	1
2 Database Management Systems .....	8
2.1 The National Defense Data Management Systems .....	8
2.2 The Commercial Database Management Systems .....	9
2.3 Independent Logical File DBMS .....	10
2.4 Relational Database Management Systems .....	11
2.5 Business Data Relationships .....	12
2.6 Data Models .....	21
2.7 ANSI Database Management System Specifications .....	24
2.7.1 The Network Data Language (NDL) Project .....	24
2.7.2 The SQL Projects .....	25
2.8 The Effect of ANSI Database Specifications on Database Objects .....	26
3 Modern Computing Environments .....	28
3.1 Computer Language Evolution .....	29
3.2 Enterprise Information Systems .....	30
3.3 Hardware and Operating Systems .....	31
3.4 Peopleware .....	32
3.5 Outsourcing: Scourge or Salvation .....	33
3.6 World-wide Enterprises .....	35
3.7 Enterprise Data Architectures .....	36
3.8 Data Architecture Example .....	39
3.9 Requirements for Modern Enterprise Information Systems .....	42
4 Database Objects .....	43
4.1 Database Object Environment .....	46
4.2 Why ANSI/SQL for Database Objects .....	49
4.3 Transportation Public Safety Database Objects .....	50
4.2.1 Data Structure .....	52
4.2.2 Database Object Processes .....	60
4.2.3 Database Object Information Systems .....	64
4.2.4 Database Object States .....	66
4.5 Business Information Systems .....	70
4.6 Business Events .....	74
4.7 Business Functions .....	78
4.8 Business Organizations .....	80
4.9 Database Object Summary .....	81



5	Methodology Support .....	82
6	Repository Requirements .....	85
7	ANSI/SQL Support .....	93
7.1	Data Structure .....	93
7.2	Database Processes .....	95
7.3	Database Object Information Systems .....	97
7.4	Database Object States .....	99
8	Summary .....	100



## Class Topic Outline

<b>Session 1:</b>  Chapter 1. Why Database Objects Chapter 2. Database Management Systems Chapter 3. Successful Computing Environments	<b>Session 3:</b>  Chapter 7. Sales & Marketing Database Objects Chapter 8. Business Information System, Event, Function, and Organization Chapter 9. Database Object Summary
<b>Session 2:</b>  Chapter 4. Database Objects Chapter 5. Transportation Database Objects Chapter 6. Courts Database Objects	<b>Session 4:</b>  Chapter 10. Repository Support Chapter 11. Methodology Support Chapter 12. ANSI/SQL Support Chapter 13. Summary



## 1 Why Database Objects

Two managers were trying to produce a three year marketing plan. One manager stated that the sales in the East were up. The other said they were flat. The first showed numbers to prove the point. The second showed an equally impressive set of numbers that proved the counter point.

Finally, it was discovered that one manager was using “sales” based on sales organizations credited for specific sales, and the other was using “sales” based on addresses of product deliveries. In exasperation, they both exclaimed: “How can we plan when we’re not working off the same *sheet of music!*”



- What should be on the *sheet of music*?
- The notes for the oboe's part, the violins, or the orchestra director?
- The orchestra director's score not only contains a unified set of notes for all parts, but also the rhythm (cadence, meter, pulse), tempo (momentum and speed), articulation (clearness, distinctiveness), and expression (phraseology and style).
- The marketing plan certainly required much more than just notes.
- To be effective, accurate, and able to respond to unforeseen emergencies (first violinist's broken string), it requires both the static (sales numbers) and the dynamics (all the environmentals).
- With both, agreements (quality music) can be reached. Plans can be executed, tracked, and adjusted, just like a good symphony.



- Policies and procedures, that is, database objects bring order, consistency, and predictability. The larger the enterprise, the greater the dependence on policies and procedures.
- Data is the evidence of policy execution.
- Procedures are the techniques, methods, or processes by which policies are carried out.
- If an enterprise has the policy is to be profitable, then its balance statement, produced by processing all the general and subsidiary journals is the measure of adherence to the policy.
- If policy is met the enterprise must be profitable.



Database objects are the Foundation Stones for Enterprise Database.

Enterprise database is an organizational operating condition

- defined policy coherence and integrity
- consistency in policy transformations throughout the enterprise
- irrespective of functional and organizational style
- irrespective of policy transformation technology





## Data Distribution Effects

Questions regarding data distribution effects	Semantic Control			
	Centralized		Decentralized	
	Data Storage Control			
	Centralized	Decentralized	Centralized	Decentralized
Is data able to be shared among sites?	yes	yes	no	no
Is concurrent processing of the same data possible?	yes	maybe	no	no
Are common or corporate reports possible?	yes	yes	no	no
Can there be an overbearing "big brother" feeling?	yes	maybe	no	no
Is there local control and ownership?	no	maybe	yes	yes
Does there need to be common data standards & policies?	yes	yes	no	no
Can local data requirements be satisfied?	maybe	yes	maybe	yes



## Program and System Distribution Effects

Questions regarding program/system distribution effects	Development Control			
	Centralized		Decentralized	
	Execution Location			
	Central- ized	Decentral- ized	Centralized	Decentral- ized
Is the same program able to be shared among sites?	yes	yes	no	no
Is concurrent processing of the same data possible?	yes	maybe	no	no
Are common or corporate reports guaranteed?	yes	maybe	no	no
Can there be an overbearing "big brother" feeling?	yes	maybe	no	no
Is there local control and ownership?	no	maybe	yes	yes
Does there need to be common processing standards & practices?	yes	yes	no	no
Can local processing requirements be satisfied?	maybe	yes	maybe	yes



## Database Object Distribution Effects

Questions regarding database object distribution effects	Semantic Control			
	Centralized		Decentralized	
	Development Control			
	Centralized	Decentral ized	Centralized	Decentral ized
Are database objects able to be shared among sites?	yes	yes	no	no
Is concurrent processing of the same database object instance possible?	yes	maybe	no	no
Are common or corporate reports possible?	yes	yes	no	no
Can there be an overbearing "big brother" feeling?	yes	maybe	no	no
Is there local control and ownership?	no	maybe	yes	yes
Does there need to be common data standards & policies?	yes	yes	no	no
Can local data requirements be satisfied?	maybe	yes	maybe	yes



## **2 Database Management Systems**

### **2.1 The National Defense Data Management Systems**

- Creators: MITRE, Lincoln Labs and System Development Corporation
- Basis: Treat Identification, Analysis and Response
- Complex hierarchical data structures
- Indexed field and separated relationships
- Column and table constraints, stored procedures, assertions, triggers, and referential integrity
- Abstract data types
- Distributed database and two phase commit



## 2.2 The Commercial Database Management Systems

- Creators: IBM (IMS), BF Goodrich (IDMS), Cincom (TOTAL), Univac (DMS-1100).
- Data Structures: Network
- Record access and record access through relationships
- Database design based squarely on policy
- Database designs actually represented database object data structures



## 2.3 Independent Logical File DBMS

Creators: Software Ag (ADABAS), Information Builders (FOCUS), CCA (Model 204)

- Data Structures: Network
- Record access through indexes, relationship access through shared index values
- Database design based on existing file structures
- Database object data structures spread across multiple file structures.



## 2.4 Relational Database Management Systems

Creators: IBM (System R), Oracle (Oracle),

- Data Structures: Two dimension structures (columns(fields) and rows (records))
- Row access via indexes as sets, and inter-record access through shared values
- Database design based on third normal form constructions
- Database object data structures were NO MORE!



## 2.5 Business Data Relationships

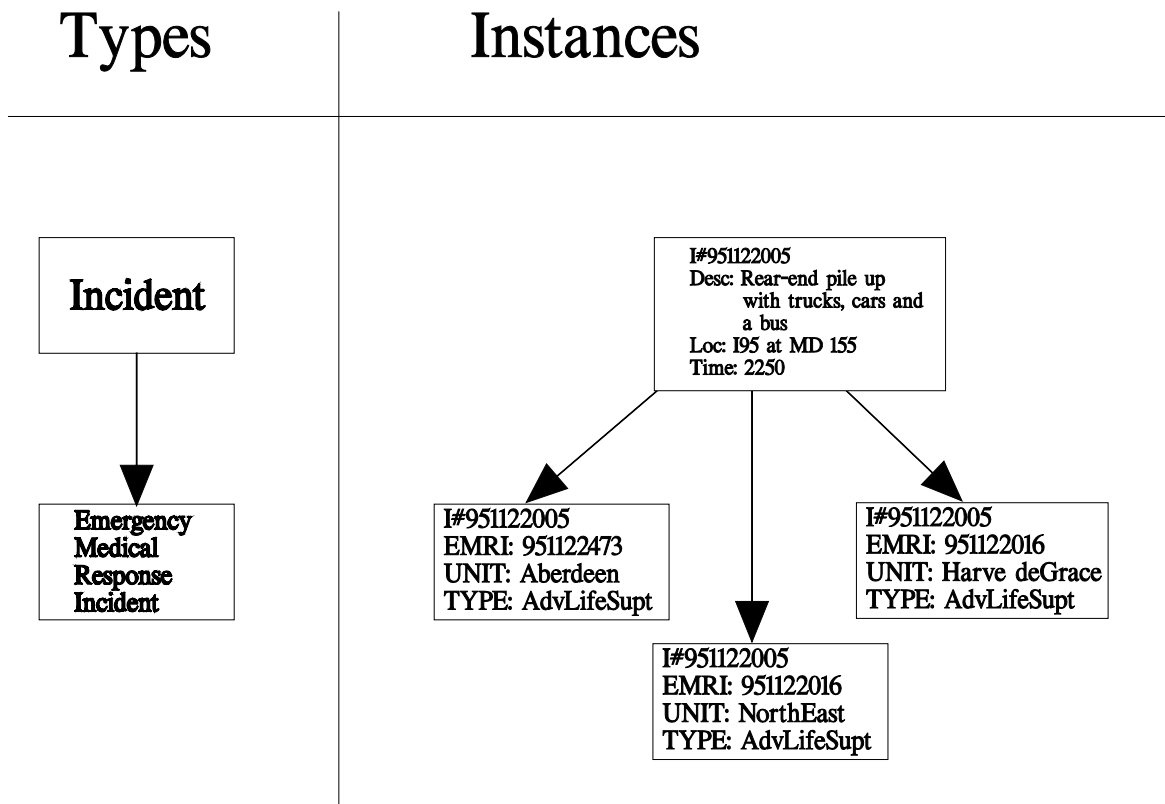
The eight types of relationships among business data are:

- One-to-many
- Owner-multiple-member
- Singular-one-member
- Singular-multiple-member
- Recursive
- Many-to-many
- One-to-one
- Inferential

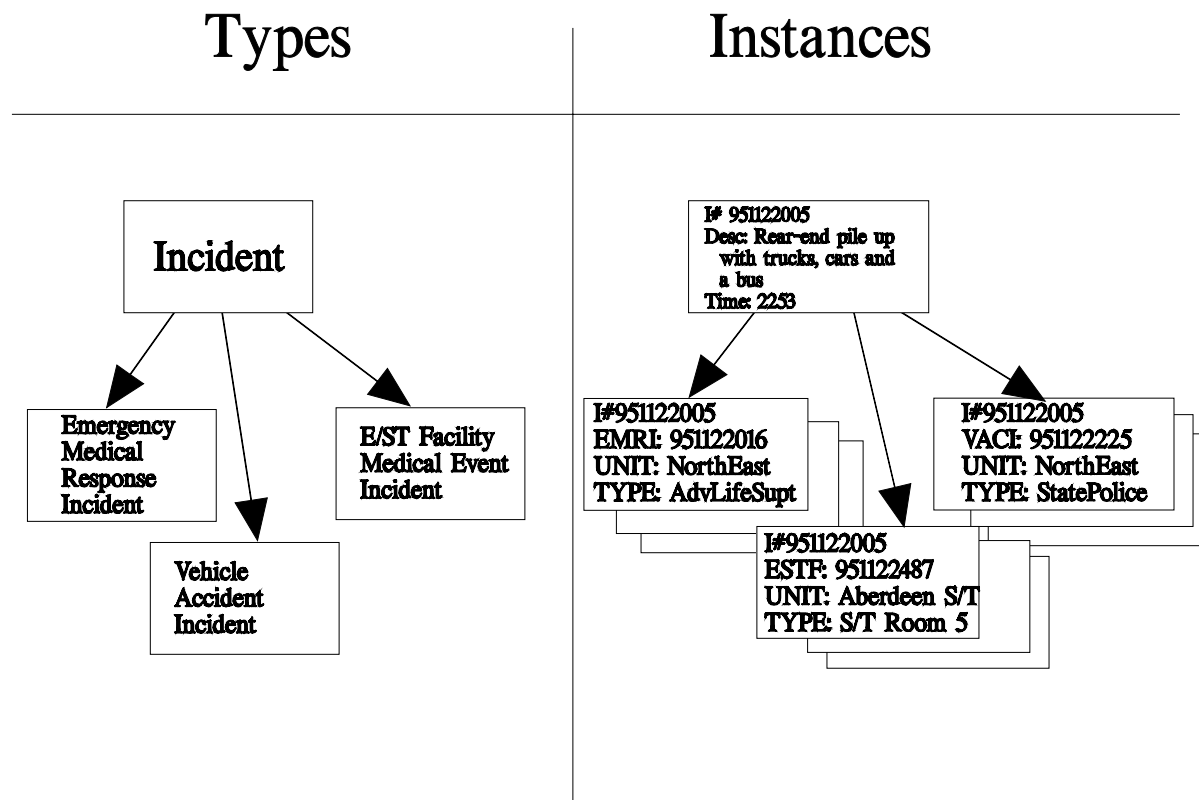




## One-to-many



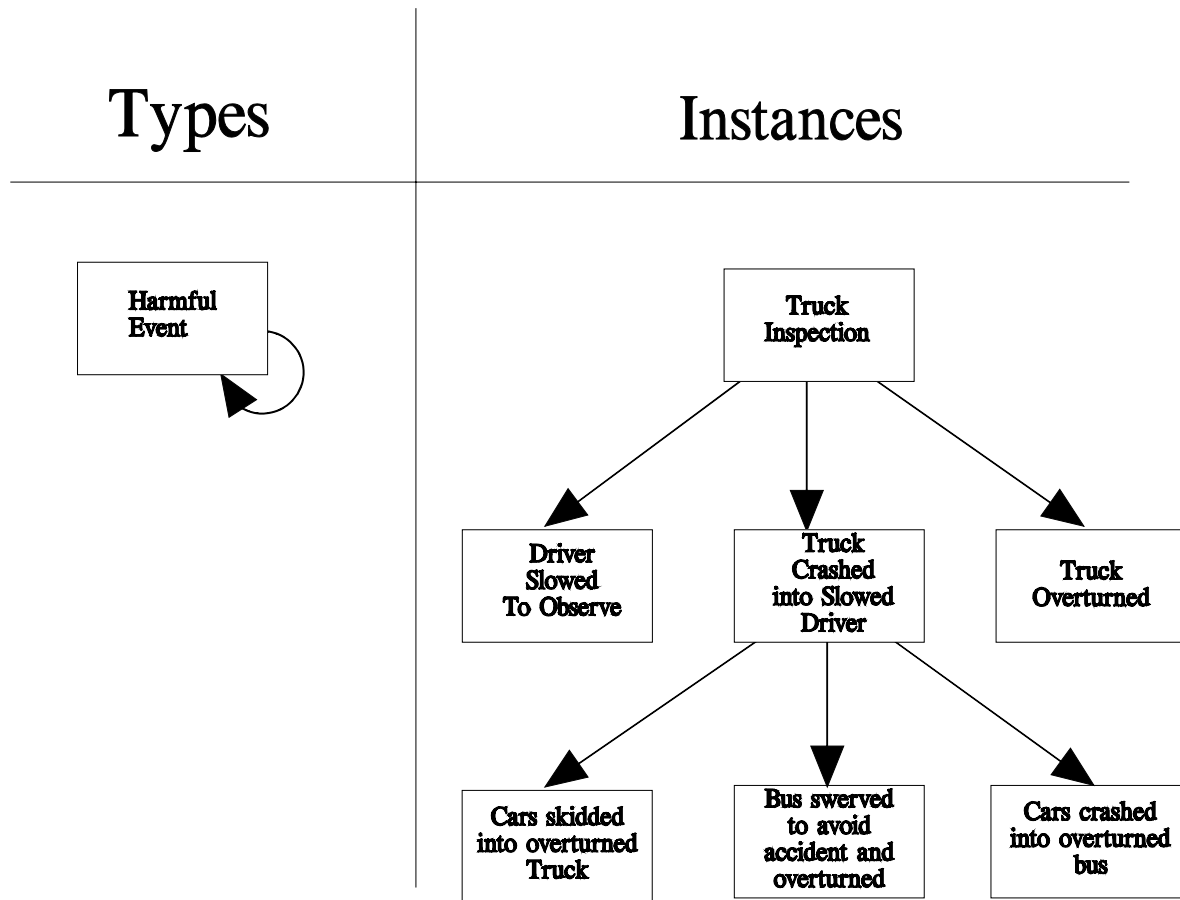
## Owner-multiple-member



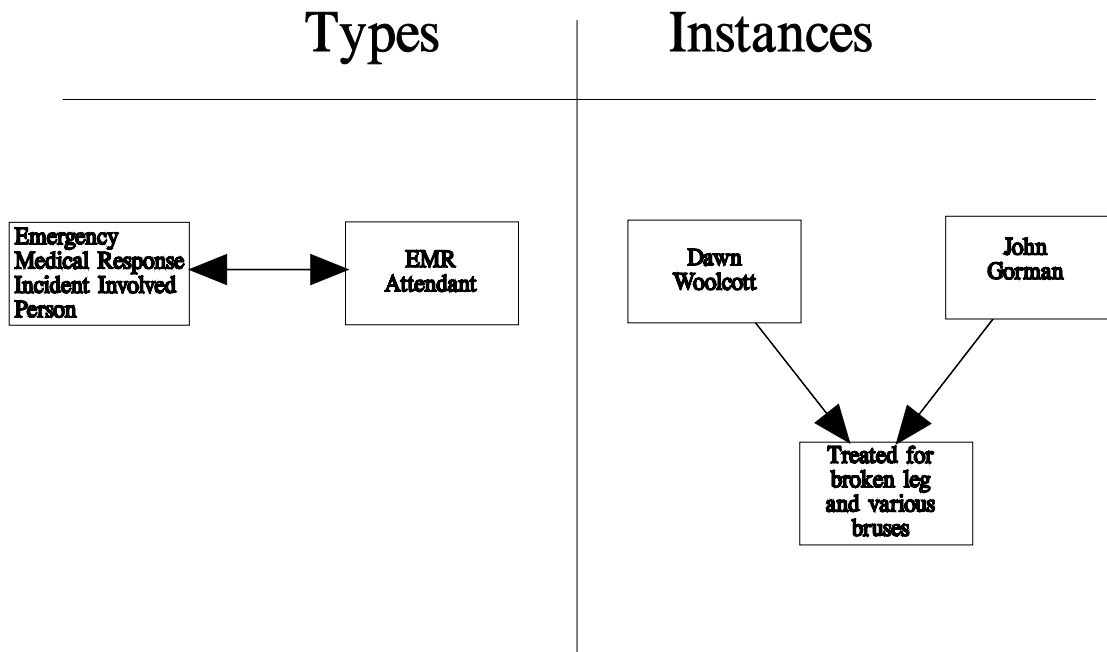




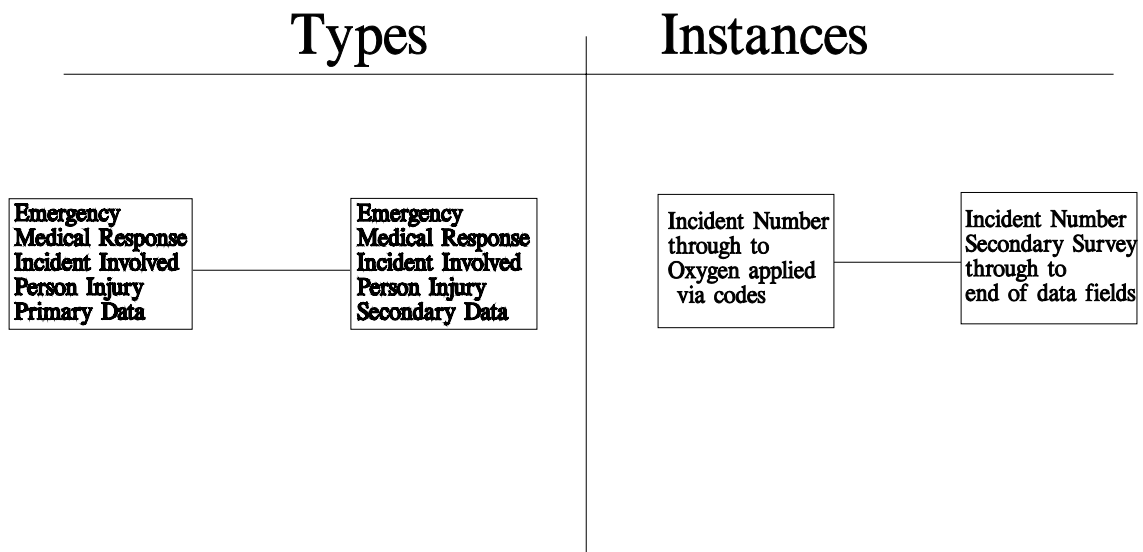
## Recursive



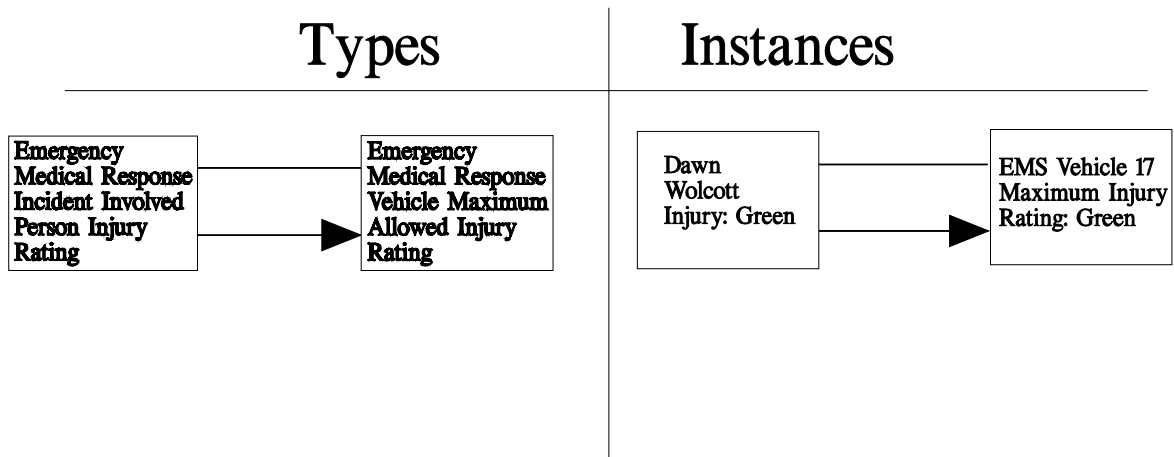
## Many-to-many



## One-to-one



## Inferential





## 2.6 Data Models

Relationship Type	DATA MODEL			
	Data Definition Language Declaration		Data Manipulation Language Simulation	
	ANSI Network	Hierarchy	Independent Logical File	Relational
Owner, 1 member	Direct & static	Direct & static	Direct & dynamic	Direct & dynamic
Owner, multiple members	Direct & static	Indirect & static	Indirect & dynamic	Indirect & dynamic
No owner and 1 member	Direct & static	No	Direct & dynamic	Direct & dynamic
No owner, multi-members	Direct & static	No	No	No
Recursive	Direct & static	Indirect & dynamic	Direct & dynamic	Indirect & dynamic
Many to many	Indirect & static	No	Direct & dynamic	Indirect & dynamic
One to one	Direct & static	Indirect & dynamic	Direct & dynamic	Direct & dynamic
Inferential	Indirect & static	No	Direct & dynamic	Direct & dynamic



One Little Parentheses Makes all the Difference in the World

Data Model	$\left\{ \begin{array}{c} \text{Table} \\ \text{Structure} \end{array} \right\} \left\{ \begin{array}{c} \text{Intertable} \\ \text{Relationships} \end{array} \right\} \left\{ \begin{array}{c} \text{Operations} \end{array} \right\}$
Data Model	$\left\{ \begin{array}{c} \text{Data Definition} \\ \text{Language} \end{array} \right\} \left\{ \begin{array}{c} \text{Data Manipulation} \\ \text{Language} \end{array} \right\}$
Dynamic Data Model	$\left\{ \text{DDL} \left\{ \begin{array}{c} \text{Table} \\ \text{Structure} \end{array} \right\} \right\} \left\{ \text{DML} \left\{ \begin{array}{c} \text{Intertable} \\ \text{Relationships} \end{array} \right\} \text{Operations} \right\} \right\}$
Static Data Model	$\left\{ \text{DDL} \left\{ \begin{array}{cc} \text{Table} & \text{Intertable} \\ \text{Structure} & \text{Relationships} \end{array} \right\} \right\} \left\{ \text{DML} \left\{ \begin{array}{c} \text{Operations} \end{array} \right\} \right\} \right\}$



Characteristics		Data Models			
		Static		Dynamic	
		Network	Hierarchical	Independent Logical File	Relational
Column Types in Table	Single valued	yes	yes	yes	yes
	Multi valued	yes	no	yes	no
	Multiple dimensions	yes	no	no	no
	Group	yes	no	no	no
	Repeating Group	yes	no	yes	no
Relationships		Table structures with predefined owners and members that create networks and hierarchies	Two dimensional table structures with predefined member table structures that create hierarchies	Multiple independent table structures. Relationships discovered through DML	Multiple, independent two dimensional tables. Relationships discovered through DML
Operations	Row	Add, Delete, Find, Modify	Add, Delete, Find, Modify	Add, Delete, Find, Modify	Add, Delete, Find, Modify
	Relationship	Connect, Disconnect, Get owner Get member Get next	Get owner Get member Get next	Join, Intersect, Division, Union, Difference	Join, Intersect, Division, Union, Project
Data Definition Language		Table Structure Relationships	Table Structure Relationships	Table Structure	Table Structure
Data Manipulation Language		Operations	Operations	Relationships Operations	Relationships Operations



## **2.7 ANSI Database Management System Specifications**

### **2.7.1 The Network Data Language (NDL) Project**

- Based on CODASYL 1969, 1971, and 1973 Journals of Development
- Network data structures
- Record access through primary key and, interrecord access through relationships
- Column and table constraints, and stored procedures.
- Standardization started in 1978
  - DDL completed in 1982
  - DML completed in 1984
- Vendors: Oracle (Vax DBMS) and Computer Associates (IDMS)



## 2.7.2 The SQL Projects

- Based on IBM's System R and ANSI/SPARC DBSSG's Study Group
- Two dimensional data structures of columns and rows
- Row access through primary key, access through sets and, interrecord access through shared values
- Column and table constraints, and stored procedures.
- Standardization started in 1982
  - DDL completed in 1984
  - DML completed in 1986
- Vendors: Oracle, Sybase, Informics, et al.



## 2.8 The Effect of ANSI Database Specifications on Database Objects

- ANSI NDL enable databases object data structure definitions
- ANSI SQL/86 trashed database object data structures, some column constraints
- ANSI SQL/89 brought back referential integrity
- ANSI SQL/92 brought table constraints, assertions, and many other goodies
- ANSI SQL/99 has abstract data types, extended operators, etc. The SQL/99 data model is NOT RELATIONAL It is Independent Logical File.



## SQL/99 Foundation Features

- Tables that have been enhanced to support new built-in data types (boolean, enumerated, extensions to character sets, translations, and collations)
- BLOB and CLOB data types
- Abstract Data Types (user defined data type with behavior, an encapsulated internal structure, and access characteristics of public, protected, or private)
- Array
- Row Types (table person (SSN, name(first, middle, last), address(street, city, state, zip(four, five))))))
- User Defined Functions
- Predicate extensions (for all, for some, similar to, cursor extensions, null values, assertions, view updatability, joins)
- Triggers
- Roles (enhancements to security), & Savepoints
- Recursion



### 3 Modern Computing Environments

One in which you can:

- Develop and maintain sophisticated and complex systems that access data and perform appropriate processes
- Accomplish information system development and maintenance *within* the life span of a modern business problem
- Develop a greatly increased quantity of information system solutions to business problems at a lower per unit cost
- Create applications that are both relevant to the individual and applicable to the enterprise as a whole





### 3.1 Computer Language Evolution

#### Goals of Language Evolution

- Making languages independent of the host computer and operating systems
- Accomplishing more computing activities with less and less human effort.

#### Generations:

- Machine Language
- Assembler
- 3GLs (Fortran, Cobol, Ada, PL/1, etc.)
- 4GLs (Focus, Power Builder, Clarion for Windows, etc.)



## 3.2 Enterprise Information Systems

Information systems seem to have gone full cycle.

- 60's From one per application
- 70's & 80's few for the whole organization
- 90's back to one for each application  
PC's institutional anarchy

Client/server comes in four flavors:

- IQ = 0 clients (formerly known as time-sharing)
- Uploading then server processing
- Uploading data and interactive server based processing, and down loading with client processing
- Cooperative processing between clients and servers



### 3.3 Hardware and Operating Systems

1980-1992	PCs and mini-computers	Require DBMS, data sharing and cooperative client server
1970-1992	Mainframes	Require DBMS, data sharing and cooperative client server

Common characteristics:

- Database management systems for data definition, sharing, storage, updating and reporting all using the ANSI standard database language, SQL for their definition, updating, and reporting operations.
- Intelligent terminals, usually PCs that are self-contained, Windows/GUI (graphic user interface) displays, that contain central processing units and multiple gigabyte (billion character (*character* is like the letter *A* or the number *4*)) storage for data entry, editing, and local data staging/storage.
- Data access and storage platforms that house DBMSs and can process shared data requests in small configurations (25 simultaneous users, one 100 MIPS (million instructions per second) processor, 100 megabytes (million) memory, and 6-10 gigabytes storage) to large configurations (1000s of simultaneous users, 16-64 100 MIPS processors, 500-1,000 megabytes of shared memory, and 60-100 gigabytes storage)
- Local area networks serving 10-50 users to combinations of wide-area networks (WANs) and LANS to serve thousands of users.



### 3.4 Peopleware

Tools:

- I-CASE, repository and code generators cost \$125,000
- Supports 10-15 staff
- Five year life span

Cost per billable staff hour \$4.00, or about 4% of each staff hour cost.

Productivity effect 40%. Therefore:

- 40% less staff, or
- 40% more projects



### 3.5 Outsourcing: Scourge or Salvation

#### **BASIS:**

#### **Employee:**

- If average salary \$60,000
- G&A and overhead multiplier is 2.5
- Per year staff cost is \$150,000
- From 22 to 55 the total cost is \$4,950,000
- Retirement 1,200,000
- The total cost: \$6,150,000

#### **Contractor:**

- Outsourcer at \$100/hour
- Annual cost is \$184,000.
- Over 30 year cost \$5,520,000.

#### **Choice:**

- Contractor cheaper than employee by \$600,000 per person
- Business' can trade every year for better and newer models.



### **Outsourcing Success Characteristics:**

- A well-defined database oriented methodology that clearly set down procedures, specified products, and quality reviews.
- A repository that enabled the continued use and reuse of metadata to accelerate development and maintenance.

### **If you outsource then you must:**

- Preserve corporate memory
- Have ICASE, repository and code generators on YOUR side of the fence
- Operate at a high level of SEI (quality methodology and repeatability)



## 3.6 World-wide Enterprises

### Characteristics:

- Heterogeneous hardware
- Heterogeneous operating systems
- DBMS from multiple vendors
- Disparate application packages

### Five common application classes for business:

- Human resource management
- Sales and marketing
- Manufacturing
- Logistics
- Customer management

### Corporations perform all five functions:

- Locally
- Country-wide
- Regionally
- Internationally



## 3.7 Enterprise Data Architectures

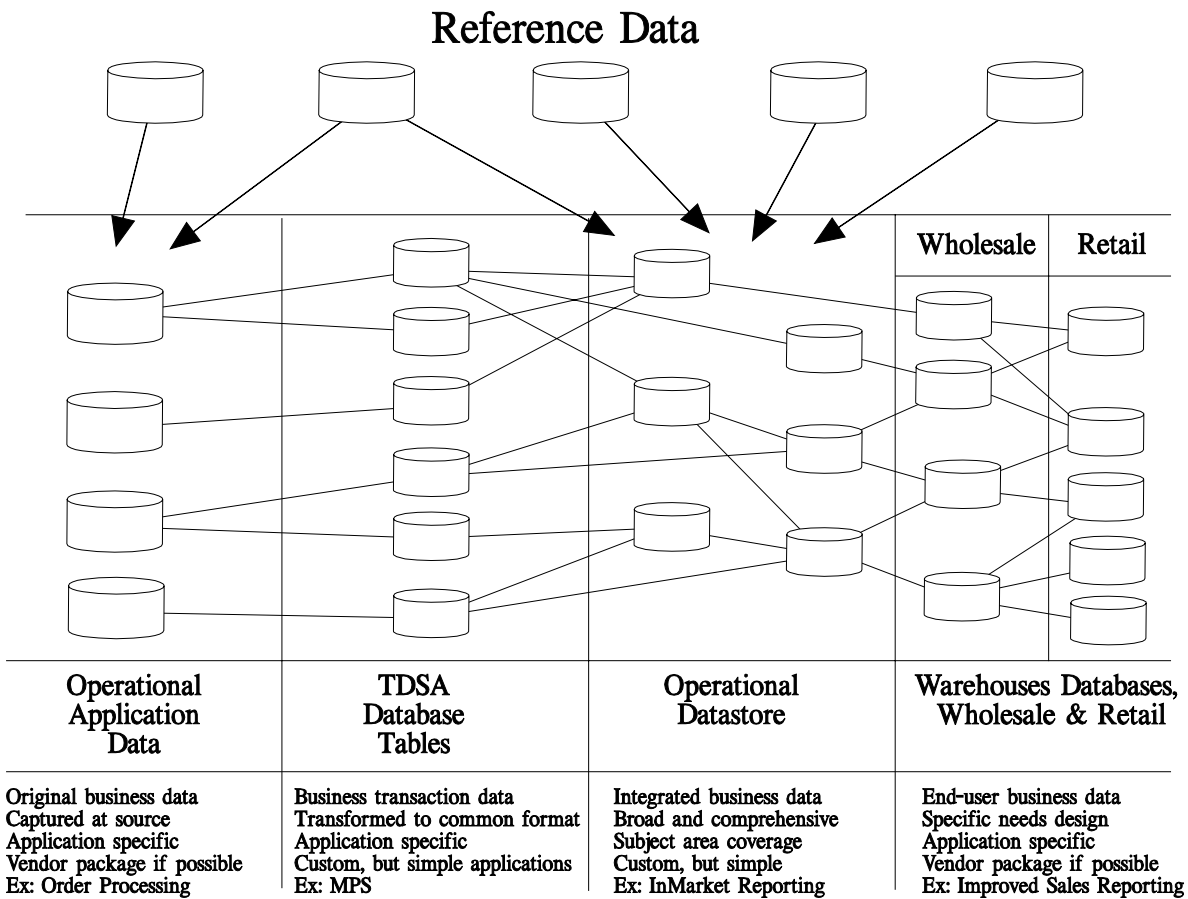
### Five Database Classes

- Reference data databases
- Original data collection databases
- Transaction data staging area databases
- Integrated subject area databases
- Warehouse databases

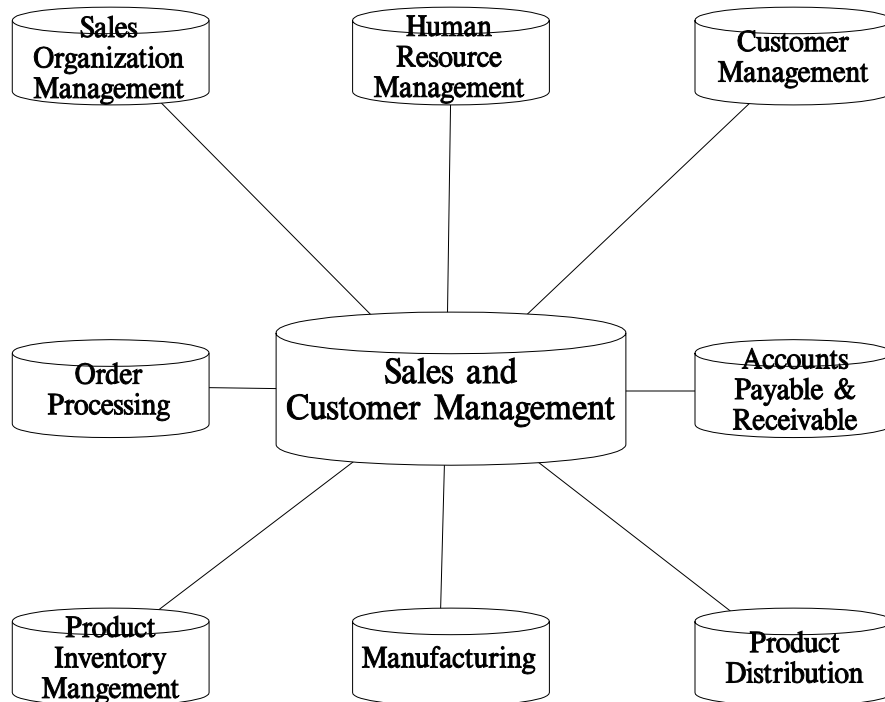




## Database Classes Component of Data Architectures



**Original Data Capture Database Applications  
feeding an Single Subject Area Database**

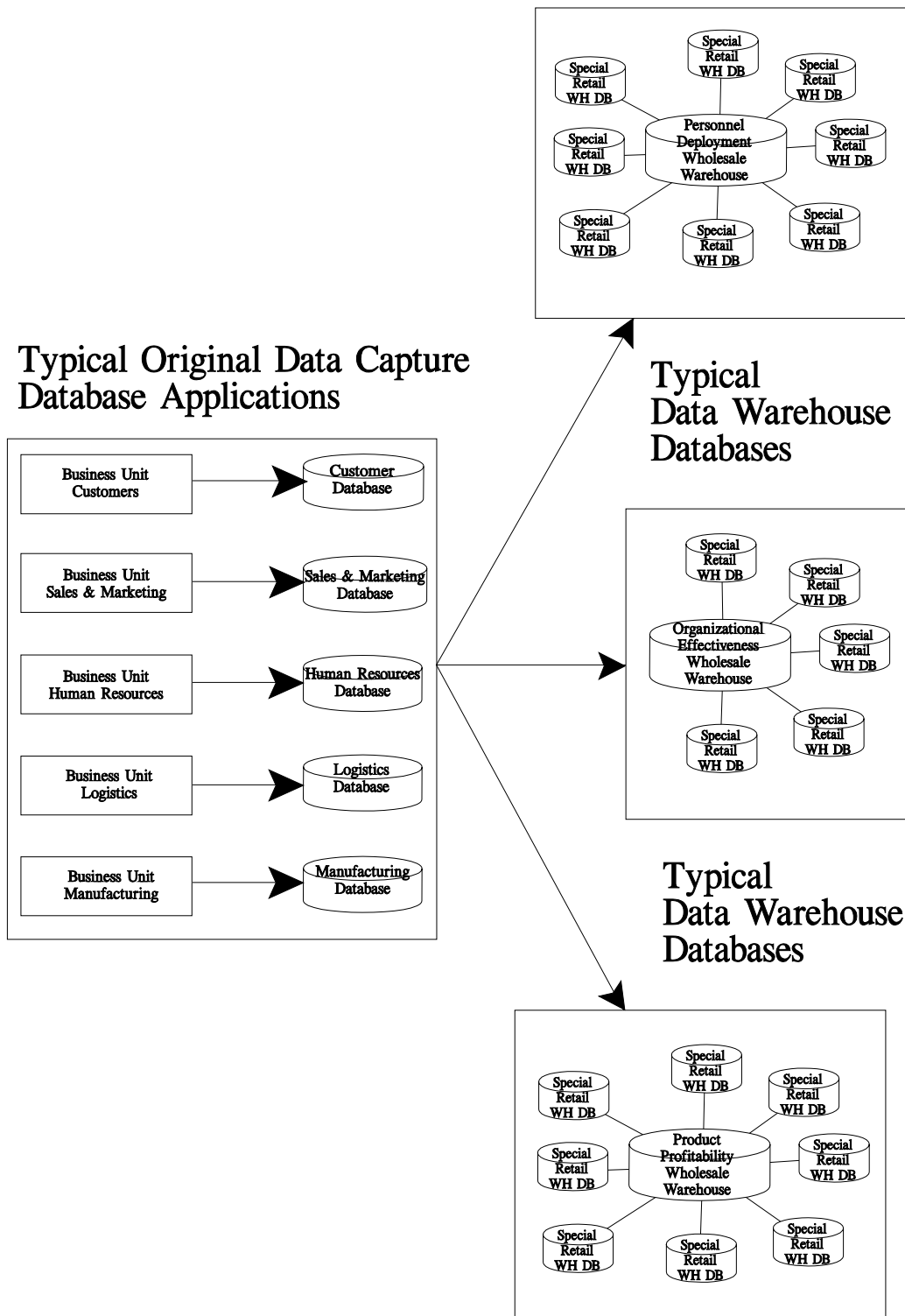


Many Enterprises have 5-10 Subject Area Databases, for example:

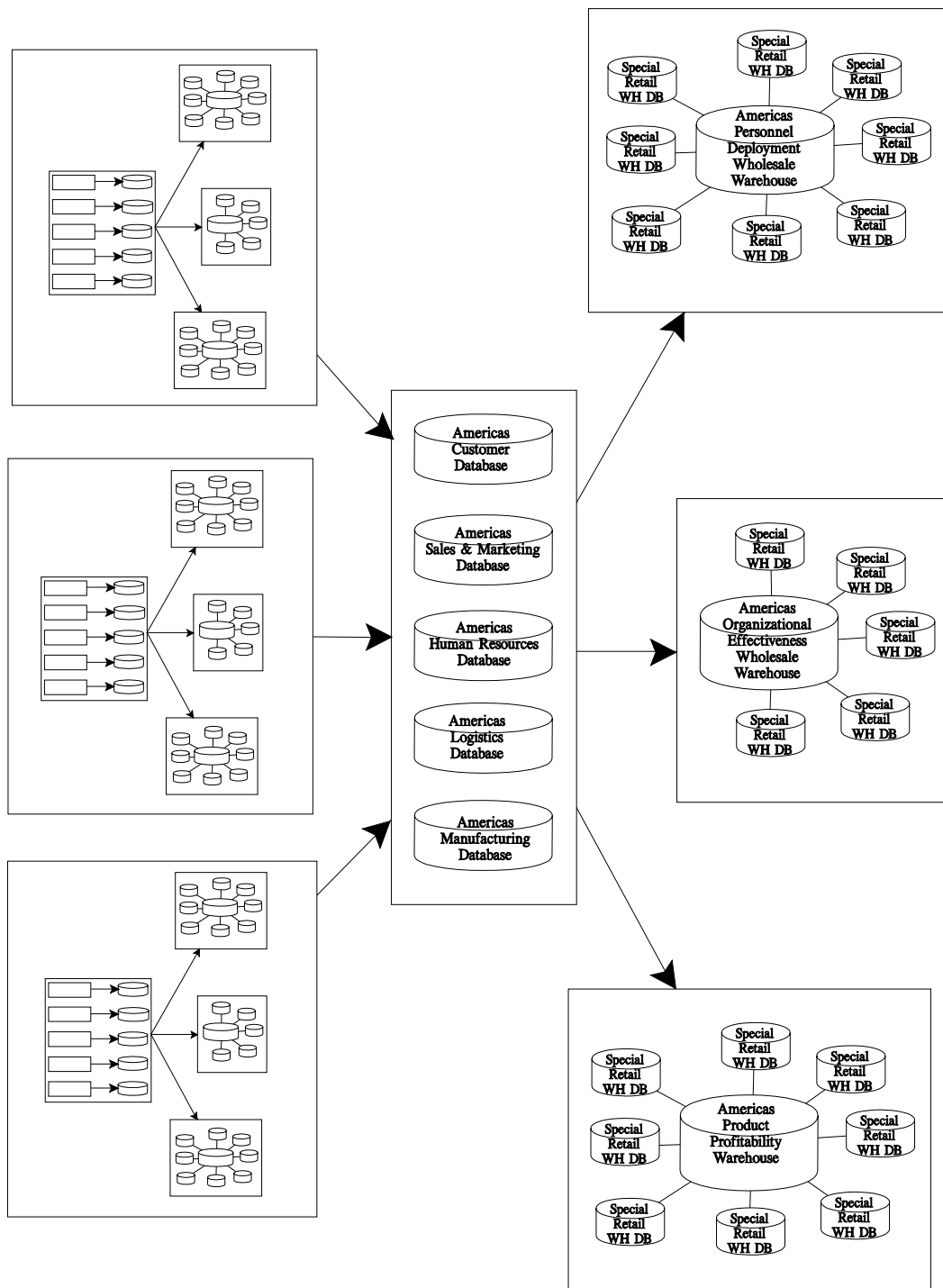
- Sales and Customer Management
- Human Resources
- Logistics
- Finance
- Manufacturing
- Inventory
- Distribution



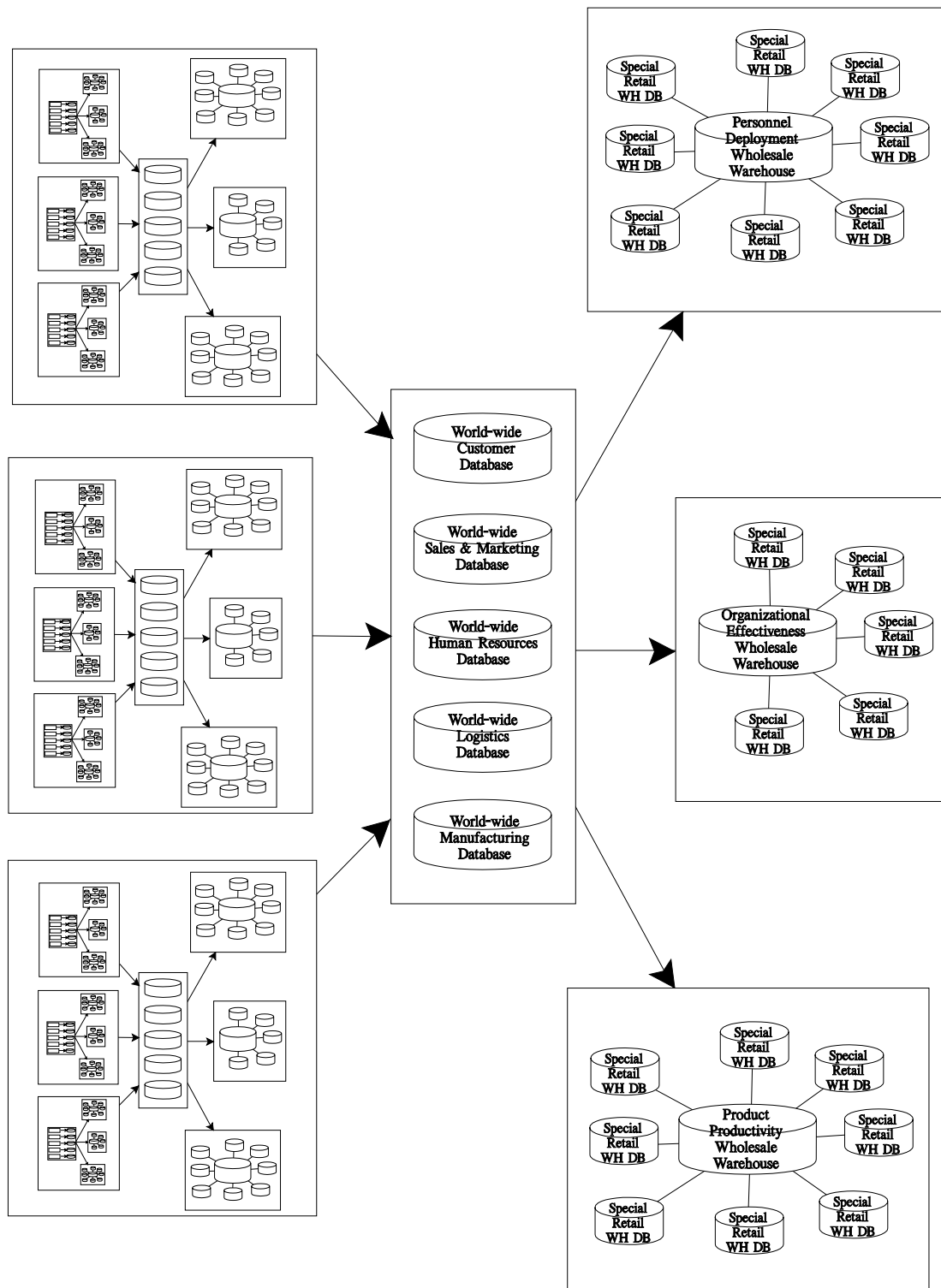
### 3.8 Data Architecture Example–Business Unit



## Data Architecture for Multiple Business Units Across a Large Geographic Area



## Data Architecture for Multiple Business Units World Wide



### 3.9 Requirements for Modern Enterprise Information Systems

The minimum requirements for modern enterprises' information systems include:

- The need to respond to increased information system demands in less time
- The need to create information systems that are significantly more complex and involve both distributed database and distributed processing
- The need to deploy information systems that serve world-wide user communities
- The need to perform continuous information systems changes to respond to unfolding and evolving needs

These information system needs are best accomplished through:

- CASE, repositories and code generators
- Standard environments of ANSI DBMS and computer languages
- The creation, deployment, and evolution of database objects as standard templates for application development through CASE, repositories, and code generators that at the foundation of all business information systems



## 4 Database Objects

- Much has been written about database objects. Data-biased persons see the world inside-out from those who are process-biased.
- Process-extremist: database object is a combination of process with a data structure component.
- Data-fanatic: a database object is a data structure with a process component.
- Can they be reconciled? Aren't they just inside-out's of each other?
- No. It's the same kind of difference as exists between data flow diagrams and entity-relationship diagrams. They are NOT inside-out's of each other. They have a fundamentally different orientation.



Database was always intended to be representations of coherent structures of data that matched well defined areas of policy and the supporting behavior to match well known business event states.

*A database object is an instance of a data structure that proceeds through predefined states according to embedded process transformations.*

**Database Object Components:**

- Database object data structures
- Database object processes
- Database object information systems
- Database object states



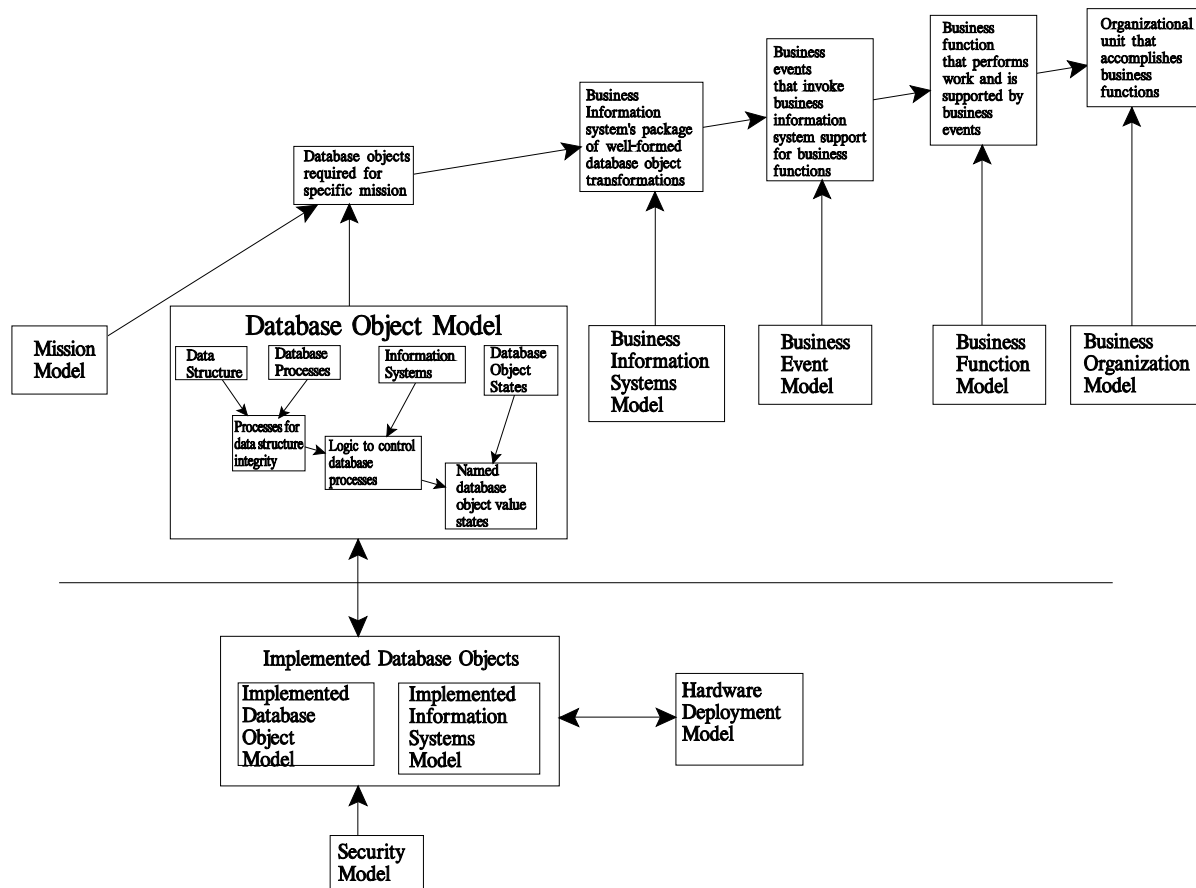


- **Data Structure:** the set of data structures that map onto the different value sets for real world database objects such as an auto accident, vehicle and emergency medicine incident.
- **Database Object Process:** the set of database object processes that:
  - ◆ Enforce the integrity of data structure fields,
  - ◆ References between database objects and actions among contained:
    - ◆ data structure segments,
    - ◆ the proper computer-based rules governing data structure segment
      - ▶ insertion,
      - ▶ modification, and
      - ▶ deletion.
- **Database Object Information System:** the set of specifications that
  - control,
  - sequence, and
  - iterate

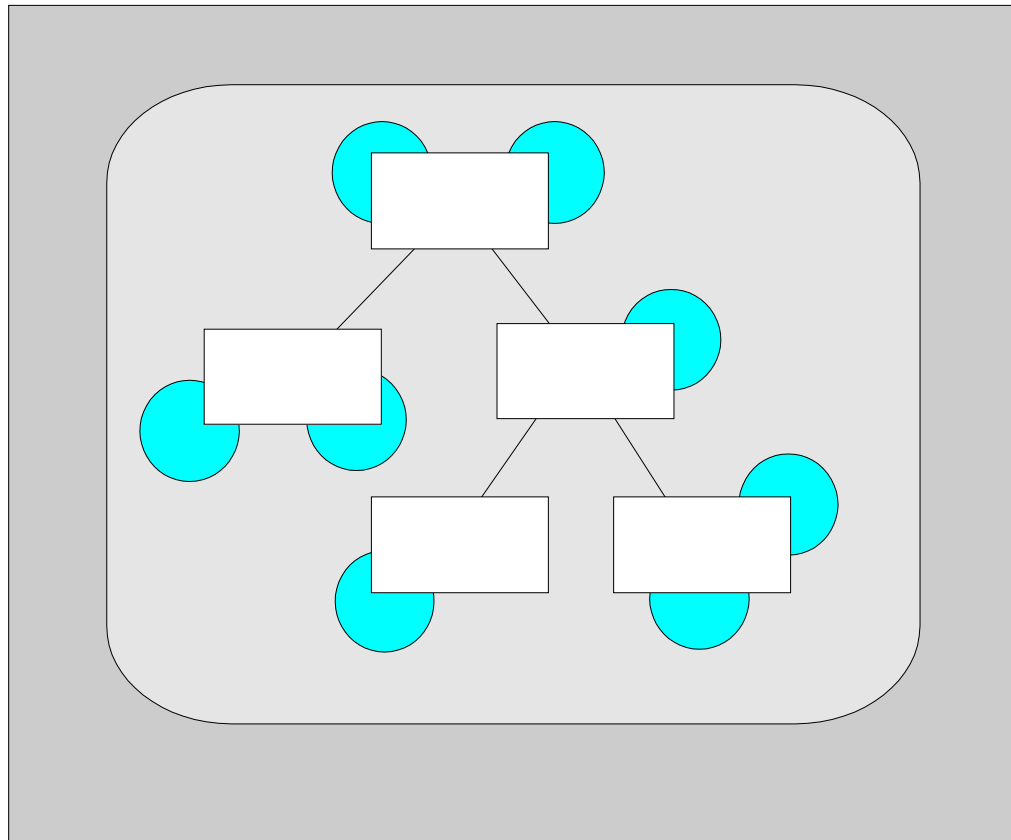
the execution of various database object processes that cause changes in database object states to achieve specific value-based states in conformance to the requirements of business policies.
- **Database object State:** The value states of a database object that represent the after-state of the successful accomplishment of one or more recognizable business events.



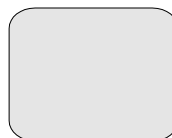
## 4.1 Database Object Environment



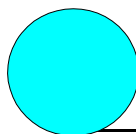
# Database Object Composition Paradigm



Data structure segment



Database object  
information system



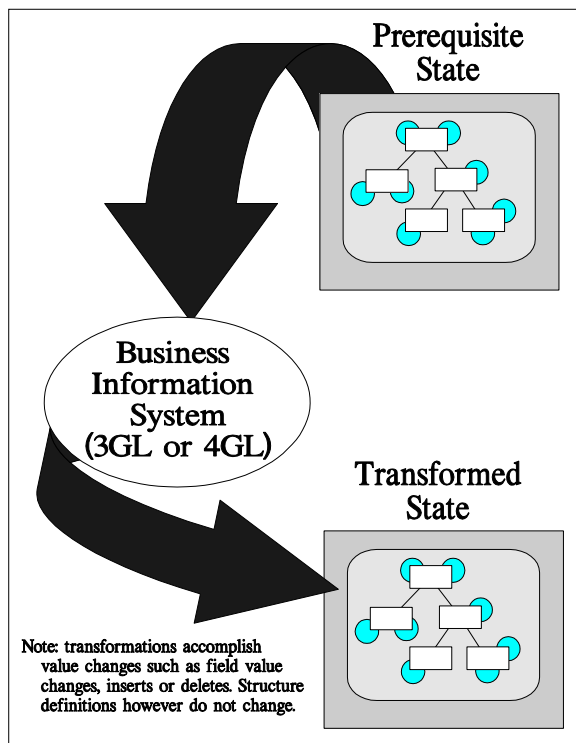
Database object process



Database object state



# Execution Paradigm

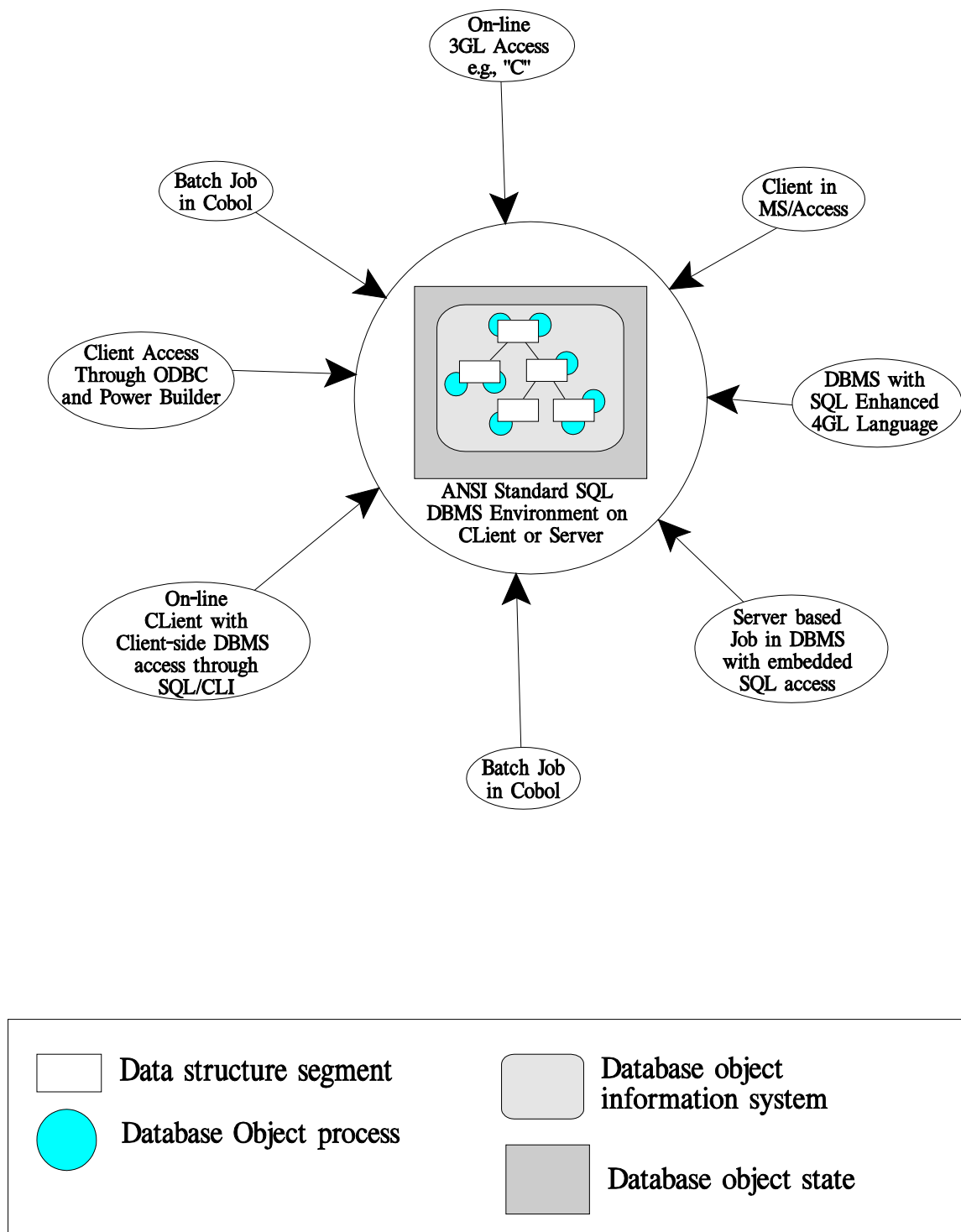


## Execution Sequence (effectively)

1. Data is obtained through 3GL and 4GL
2. Database object requested state is invoked through Business Information System. (Outside Square)
3. Data is passed to the database object information system (Rounded Edge Square)
4. Database object processes are passed data and they perform their actions (Circles to rear of white squares)
5. Database object data structure is modified through inserts, changes, and deletes (connected collection of white squares)
6. If success, then commit, otherwise rollback.



## 4.2 Why ANSI/SQL for Database Objects



### 4.3 Transportation Public Safety Database Objects

Database Object Class Type	Database Object Class
Facility or organization	Facility
	Transport Organization
	Government Agency
Highway	Highway
	Traffic Device
Incident	Incident
	Vehicle Checkpoint Incident
	Vehicle Accident Incident
	Emergency Shock Trauma Facility Medical Event Incident
	Emergency Medical Response Incident
	Vehicle Stop Incident
	Vehicle Inspection Incident
	Judical Case Incident



Database Object Class Type	Database Object Class
Person	Government Person
	Emergency Shock Trauma Treatment
	Insurance Policy
	Person
	Judical case
	Violation
	Interview
	Hazardous Material or Object
	Property
	(drug or alcohol) Medical Test
Vehicle	Drivers License
	Vehicle



## 4.2.1 Data Structure

### Single Segment Data Structure: INCIDENT

Incident Identifier  
Incident Date  
Incident Time  
Incident Call-In Site Identifier  
Incident Initiator Identifier





## **DATA STRUCTURE: EMERGENCY MEDICAL RESPONSE INVOLVED PERSON INJURY**

INCIDENT IDENTIFIER

EMERGENCY MEDICAL RESPONSE INCIDENT IDENTIFIER

EMERGENCY MEDICAL RESPONSE INVOLVED PERSON IDENTIFIER

CHIEF COMPLAINT DESCRIPTION

HISTORY OF PRESENT ILLNESS DESCRIPTION

PAST MEDICAL HISTORY

CODE

CURRENT MEDICATIONS

NAME

DOSAGE

ALLERGIES

NAME

DESCRIPTION

VITAL SIGNS ASSESSMENT

TIME

PULSE

RESPIRATION

BLOOD PRESSURE

EKG



## **DATA STRUCTURE: EMERGENCY MEDICAL RESPONSE INVOLVED PERSON INJURY (cont)**

### **OXYGEN APPLIED**

AT LPM QUANTITY  
VIA CODE

### **SURVEY**

#### **PRIMARY**

AIRWAY & CERVICAL SPINE  
ASSESSMENT CODES  
INITIAL INTERVENTION  
OUTCOME CODE

#### **BREATHING**

ASSESSMENT CODES  
INITIAL INTERVENTION CODES  
OUTCOME CODES

#### **CIRCULATION**

ASSESSMENT CODES  
INITIAL INTERVENTION CODES  
OUTCOME CODES

#### **EXPOSURE**

EYE OPENING ASSESSMENT CODES  
BEST VERBAL RESPONSE CODES  
BEST MOTOR RESPONSE CODES



## **DATA STRUCTURE: EMERGENCY MEDICAL RESPONSE INVOLVED PERSON INJURY (cont)**

### SECONDARY

#### SKULL & FACE FINDINGS

CODE

DESCRIPTION

#### PUPILS

RIGHT SIZE IN MM

RIGHT REACTIVE INDICATOR

LEFT SIZE IN MM

LEFT REACTIVE INDICATOR

#### JUGULAR VEINS CODE

#### NECK AND C SPINE

CODE

FINDINGS

#### LUNG

LEFT CODE

RIGHT CODE

#### ABDOMEN & PELVIS

CODE

FINDINGS



## **DATA STRUCTURE: EMERGENCY MEDICAL RESPONSE INVOLVED PERSON INJURY (cont)**

### Secondary Survey (cont)

EXTREMITIES BACK  
CODE  
FINDINGS

NEUROLOGICAL  
CODE  
FINDINGS

### INTUBATION

MEDIC NUMBER  
QUANTITY OF ATTEMPTS  
SUCCESS CODE  
ETT ORAL OR NASAL CODE  
TUBE SIZE IN MM  
EOA CODE  
TIME  
SOUNDS CODE  
GASTRIC COUNTS CODE  
COMPLICATIONS DESCRIPTION



## **DATA STRUCTURE: EMERGENCY MEDICAL RESPONSE INVOLVED PERSON INJURY (cont)**

### **IV STRATEGY ATTEMPT**

NSSS QUANTITY CODE  
MEDIC IDENTIFIER  
SITE  
TIME  
QUANTITY OF ATTEMPTS  
SIZE

### **OTHER MANAGEMENT**

SPECIAL IMMOBILIZATION CODE  
EXTRICATION  
CODE  
TIME  
AEROSOL TREATMENT  
MEASUREMENTS  
PREPEAK FLOW  
PRELUNG SOUND  
POSTPEAK FLO  
POSTLUNG SOUND

QUALITY INDICATION CODE  
FRACTURES SPLINTING CODE  
OTHER DESCRIPTION



## **DATA STRUCTURE: EMERGENCY MEDICAL RESPONSE INVOLVED PERSON INJURY (cont)**

### **MEDICATIONS**

DRUG NAME  
DRUG TIME  
DRUG AMOUNT  
DRUG ROUTE

STANDING ORDERS FOLLOWED DESCRIPTION

### **VERBAL ORDERS**

TIME  
DESCRIPTION

### **MEDICAL CONTROL CONTACTED CODE**

TRANSPORTED TO NAME  
BLS UNIT IDENTIFIER  
VERBAL REPORT GIVEN TO NAME  
SUPPLEMENTAL INFORMATION NARRATIVE

### **SIGNATURE PERSON**

NAME  
IDENTIFIER

### **WEATHER CONDITIONS CODE**

## **DATA STRUCTURE: EMERGENCY MEDICAL RESPONSE INVOLVED PERSON INJURY (cont)**

### **INVOLVED MEDIC PERSONNEL**

NAME  
IDENTIFIER

### **OTHER PERSONNEL**

NAME  
IDENTIFIER



## Database Object Data Structure Summary

- Simple that contain only single valued fields
- Complex that contain single valued, multiply valued, groups, repeating groups and nested repeating groups
- Co-owned data structures that are shared with other database objects
- Data structures that are contained exclusively within the cited database object but are co-owned by another contained database object segment.



## 4.2.2 Database Object Processes

Three basic types:

- Data structure instance add, delete and modify
- Field and data structure constraints
- Inter database object integrity clauses and their automatic actions

To adequately support database objects, database object processes must be able to:

- Receive data from database object information systems, for example, the data values for the various database object contained structures.
- Return messages that in turn request the information to store the next level of Emergency Response Harmful Event, or to store a sibling Emergency Response Harmful Event.
- Store data within a database object contained data structure upon database object process success, and reject storage or modification upon failure.
- Cascade failure, or perform a complete database object commit if the database object transaction is successful.





## **Collection of Database Object Processes (a.k.a., a database object information system)**

Insert Emergency Medical Response Incident

Insert Emergency Medical Response Harmful Event, else rollback

Insert Emergency Medical Response (validate Emergency Medical Response Organization existence)

Insert Emergency Response Involved Person (Validate Emergency Response Harmful Event if valued)

Insert Emergency Response Involved Person Injury

Insert Emergency Enroute Transport (validate  
Emergency/shock Trauma Facility existence, validate EMR  
Attendant existence)

Insert Ambulance Report, else rollback

Insert EMT/Paramedic Report, else rollback

Else rollback

Else rollback

Else rollback

Else rollback



Locate emergency medical response incident

Insert emergency response harmful event

Loop

Insert subevent until end

EndLoop

---

Locate Emergency Response Incident

Locate Emergency Response Harmful Event

Locate Emergency Medical Response

Locate Emergency Response Involved Person

Modify Emergency Response Involved Person foreign key to

Emergency Response Harmful Event primary key value



## Database Object Process Reuse

Database Object: Emergency Medical Response Incident		
Database Object Information Systems	Involved Database Processes	Involved and/or Affected Data Structure Segment
Enter Emergency Medical Response Incident	Insert Emergency Medical Response Incident	Emergency Medical Response Incident
Enter Emergency Medical Response	Locate Emergency Medical Response Incident Insert Emergency Medical Response	Emergency Medical Response Incident Emergency Medical Response
Enter Emergency Response Harmful Event	Locate Emergency Medical Response Incident Insert Emergency Response Harmful Event	Emergency Medical Response Incident Emergency Response Harmful Event
Enter Emergency Response Involved Person	Locate Emergency Medical Response Incident Locate Emergency Medical Response Insert Emergency Response Involved Person	Emergency Medical Response Incident Emergency Medical Response Emergency Response Involved Person
Enter Emergency Medical Response Involved Person Injury	Locate Emergency Medical Response Incident Locate Emergency Medical Response Locate Emergency Response Involved Person Insert Emergency Medical Response Involved Person Injury	Emergency Medical Response Incident Emergency Medical Response Emergency Response Involved Person Emergency Medical Response Involved Person
Enter Emergency Enroute Transport	Locate Emergency Medical Response Incident Locate Emergency Medical Response Locate Emergency Response Involved Person Insert Emergency Enroute Transport	Emergency Medical Response Incident Emergency Medical Response Emergency Response Involved Person Emergency Enroute Transport



### 4.2.3 Database Object Information Systems

#### Characteristics:

- Provide the overall control logic that is necessary to accomplish--through database object processes--the storage, modification, and deletion of different database objects
- Database object information systems are the mechanisms for controlling multiple database objects.
- Database object processes are employed to control the storage, modification, or deletion of a single database object.



## **Different from End-user Information Systems:**

- The database object information system is defined entirely through the SQL language. It is not programmed in any 4GL or 3GL.
- Database object information systems are entirely portable between instances of ANSI/SQL DBMSs.
- All the artifacts of the database object information system, once loaded in an ANSI/SQL DBMS are stored entirely within the schema information tables
- The database object information system does not directly interact with end users through GUI prompts or through non-SQL DBMS file reads and writes.
- The database object information system cannot invoke end-user or business information systems. Rather, database object information systems can be invoked only by other database object information systems or by other end-user business information systems



#### 4.2.4 Database Object States

- The business' requirements for a value state change (add, delete, or modify) in a business resource.
- A business resource is an essential aspect of a business that is transformed from one value state to another

Emergency medicine response states:

- Incident identified
- Incident response initiated
- Harmful event identified
- Involved person identified
- Involved person injury entry
- Emergency enroute transport and report

Each state change is achieved through one or more database object information systems.

Each information system involves one or more database object processes that in turn cause storage, manipulation, and/or deletion of one or more database objects.



<b>Business Resource</b>	<b>Resource State</b>	<b>Database Object Information System</b>	<b>Database Process</b>	<b>Affected Database Object data segment</b>
Emergency Medical Response	EMR incident identification	Emergency medical response incident idendentification	Insert Emergency Medical Response Incident	Emergency Medical Response Incident
	Incident response	Incident Medical Response	Locate Emergency Medical Reponse Response Incident	Emergency Medical Reponse Incident
			Validate Emergency Medical Response Organization	Emergency Medical Reponse Organization
			Insert Emergency Medical Response	Emergency Medical Response
	Harmful event recording	Record Harmful Event	Locate Emergency Medical Response Incident	Emergency Medical Reponse Incident
			Insert Emergency Medical Response Harmful Event	Emergency Medical Response Harmful event



<b>Business Resource</b>	<b>Resource State</b>	<b>Database Object Information System</b>	<b>Database Process</b>	<b>Affected Database Object data segment</b>
	Involved Person Identification	Record Involved Person	Locate Emergency Medical Reponse Response Incident	Emergency Medical Reponse Incident
			Locate Emergency Medical Response	Emergency Medical Response
			Insert Emergency Medical Reponse Involved Person	Emergency Medical Response Involved Person
		Connect Involved Person to Harmful Event	Locate Emergency Medical Response Harmful Event	Emergency Medical Response Harmful event
			Modify Emergency Response Involved Person	Emergency Medical Response Involved Person





Business Resource	Resource State	Database Object Information System	Database Process	Affected Database Object data segment
		Record Involved Person Injury	Insert Emergency Medical Response Involved Person Injury	Emergency Medical Response Involved Person Injury
	Emergency Enroute Transport	Record Emergency Enroute Report	Locate Emergency Medical Reponse Response Incident	Emergency Medical Reponse Incident
			Locate Emergency Medical Response	Emergency Medical Response
			Locate Emergency Medical Reponse Involved Person	Emergency Medical Response Involved Person
			Validate Emergency Shock Trauma Facility	Emergency Shock Trauma Facility
			Validate EMR Attendant	EMR Attendant
			Insert Emergency Enroute Transport	Emergency Enroute Transport



## 4.5 Business Information Systems

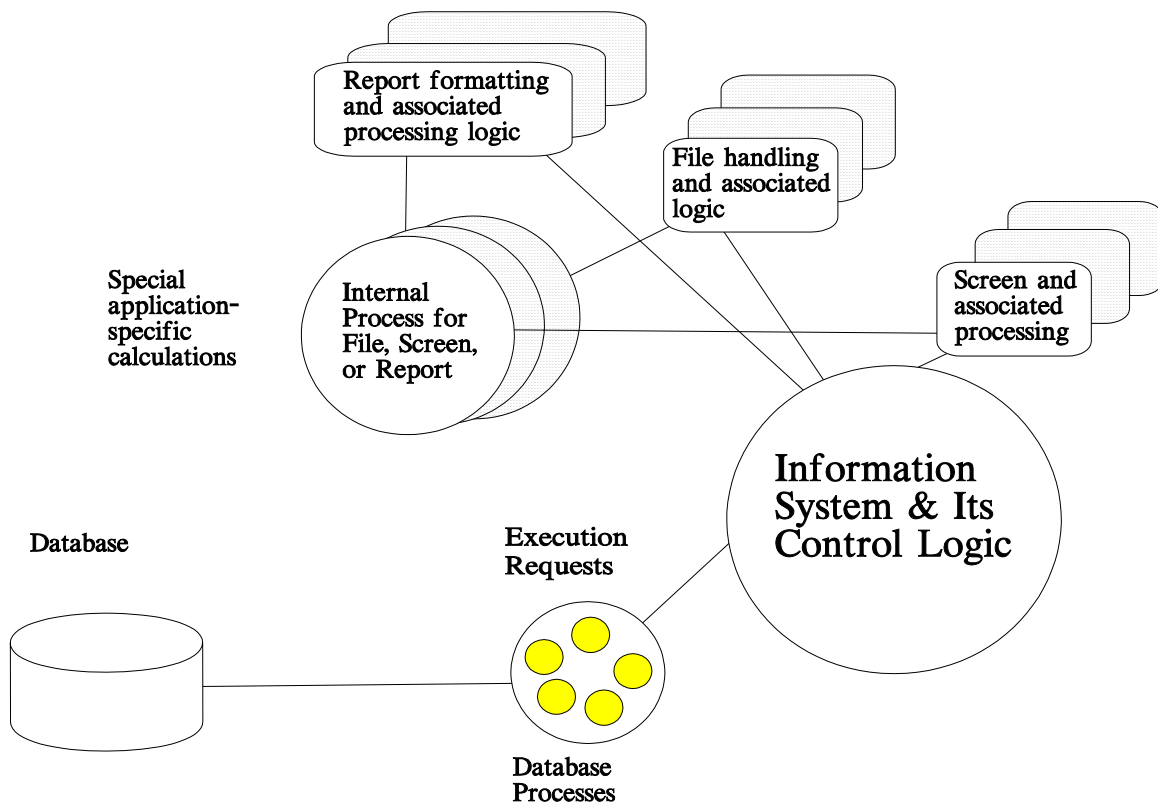
Business information systems consist of five subordinate components.

- Information system
- Report
- File
- Screen
- Internal process

Examples of acceptable information systems under this nonredundant, multiple-use model are:

- One that only contains one report specification
- One that contains a high-level reports menu, and then a selection of a number of lower-level reports
- One that collects the data represented by a single business form
- One that offers a high-level menu data collection menu, and then a selection of a lower-level number of data collections through business forms
- One that represents the transformation of some set of internally stored business data to a database, or formatted into a report, or presented through a screen, or that is extracted from one database and is stored in another





Selected combination of processing and I/O (report, file, screen) processes to handle the reporting, loading, updating, and deleting of database data.



**Business information system architecture that is:**

- Nonredundant, because each component is nonredundant
- Flexible, because each information system represents a *plug and play* use of one or more business information system components to support the other business information system components
- Robust, because each business information system component is well defined and is closed



## **The business information systems architecture supports:**

- Traditional systems, where input, process, and output are well defined into a monolithic system that implements traditional well-defined functions
- Data collection systems, where all enterprise data is collected once through a common motif, edited once, and then stored in transaction format in the most appropriate physically distributed locations for later use in operational systems, and then in warehouse systems, and can be obtained for use in special studies
- Data reporting systems, where all enterprise data, from wherever it is stored, is made available through a set of standard reports, or through a standard report writing language
- A COTS environment under either a specialized, probably redundant data collection scenario, or that uses data that has been commonly collected and edited



## 4.6 Business Events

Business events are the mechanisms that instigate information systems' services during the accomplishment of business functions.

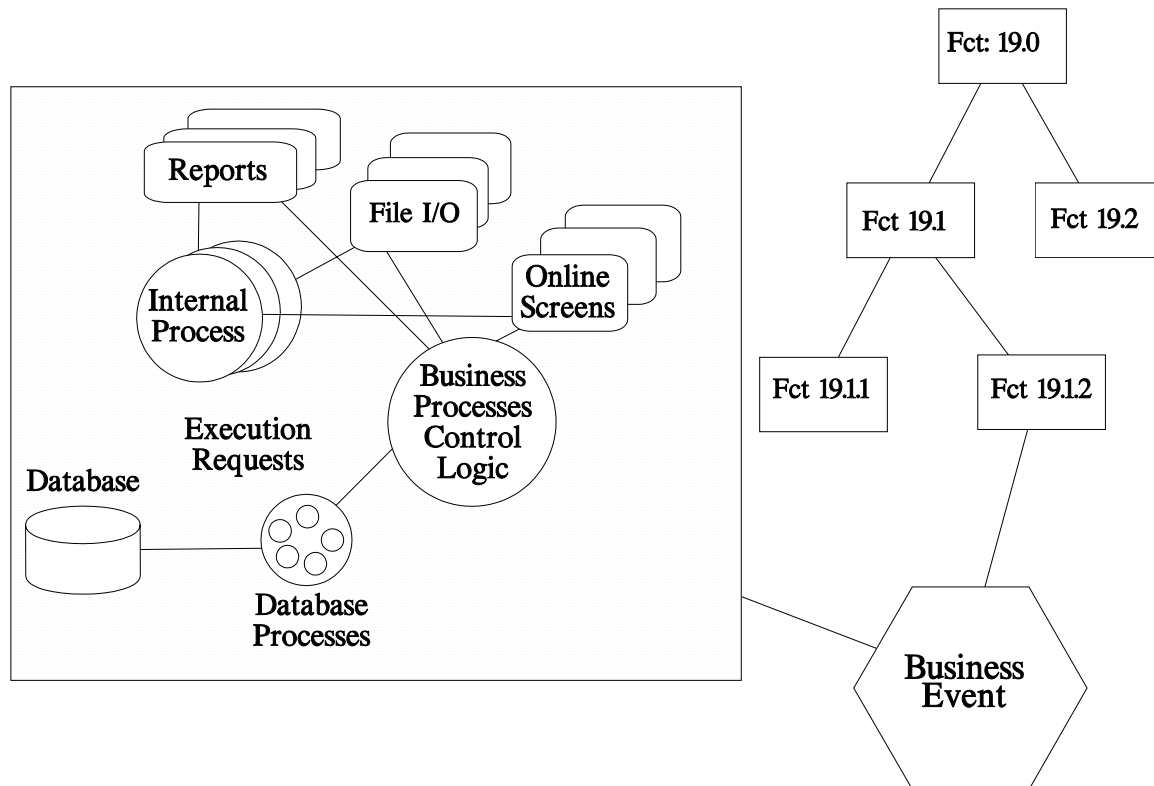
- A business event may be triggered through multiple business functions. Each business information system then invokes actions from database objects.
- A business event is not an activity of the enterprise.
- Activities of the enterprise are represented through the business function model.
- The business event model represents the automation interface between the business functions and the information systems.

Because of this separation through the business event, the following occurs:

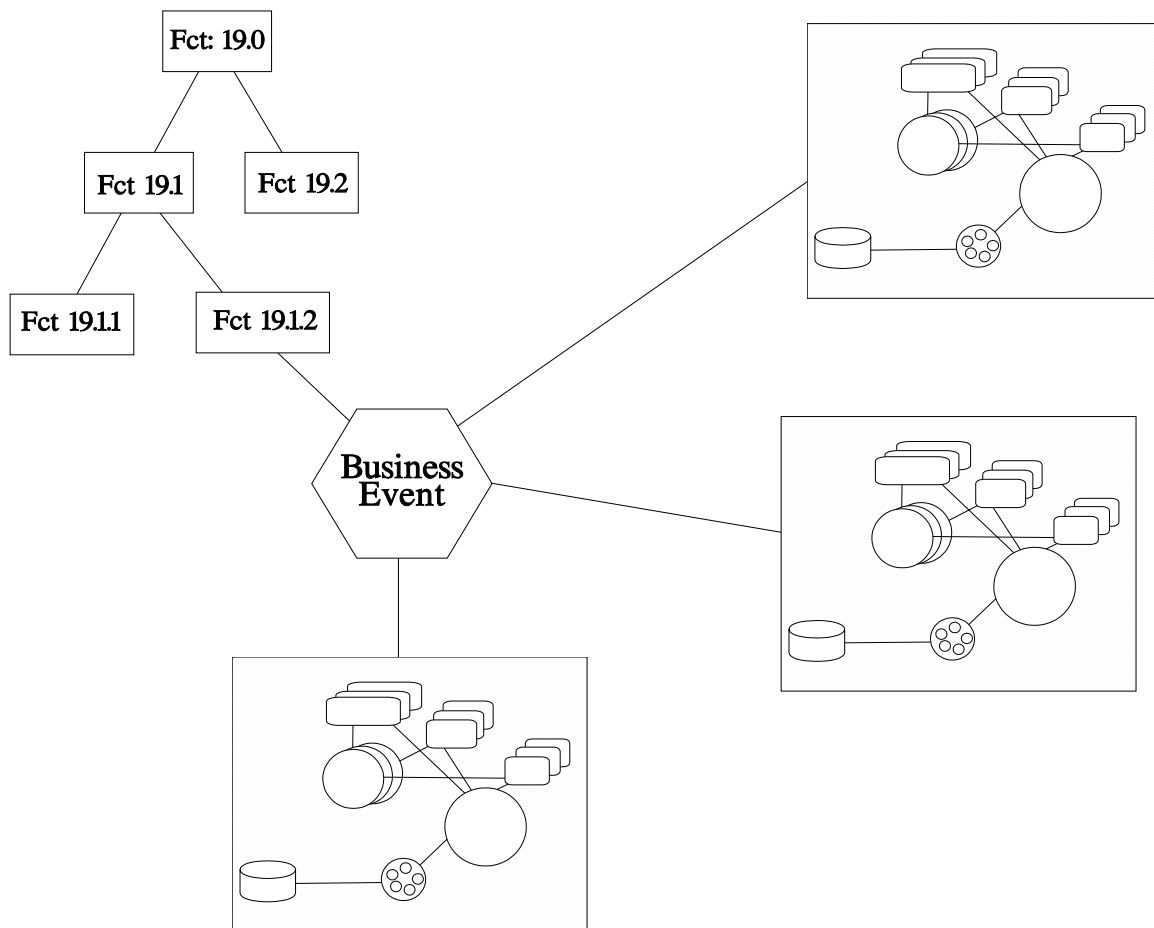
- There may be multiple styles of business functions that benefit from the information system.
- There may be information systems that are used by different business functions
- A business event hierarchy can employ different collections of information systems to serve the needs of different business functions.



## Business Function Use of a Business Information Systems Thru a Business Event



## Business Event Triggering Multiple Business Information Systems for a Given Business Function

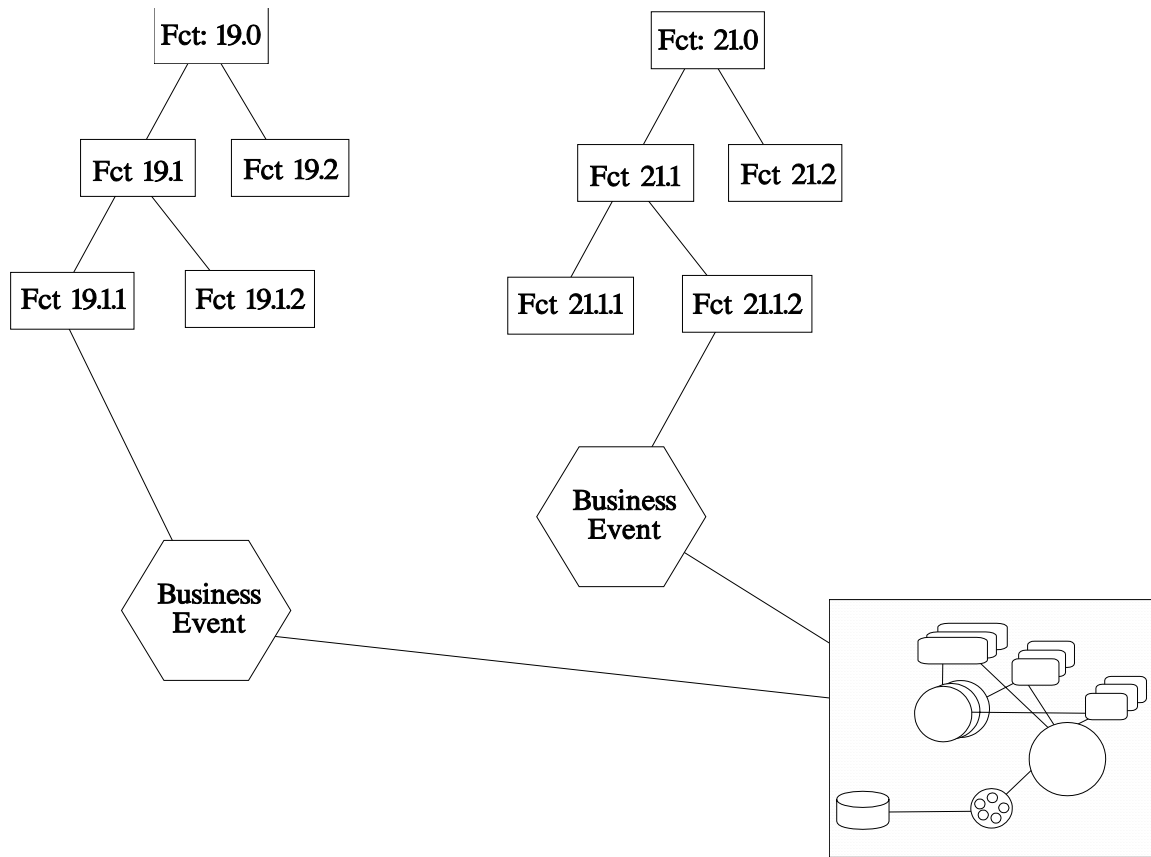


Multiple information systems servicing a business event.





## Multiple Business Functions Use of a Business Information System through Different Business Events



## 4.7 Business Functions

- Business functions identify the set of activities that accomplish one or more aspects of a business's activities.
- With respect to database objects, a business function and its activities are manual.
- When ever a business function requires business information systems support, it is instigated through the business event.

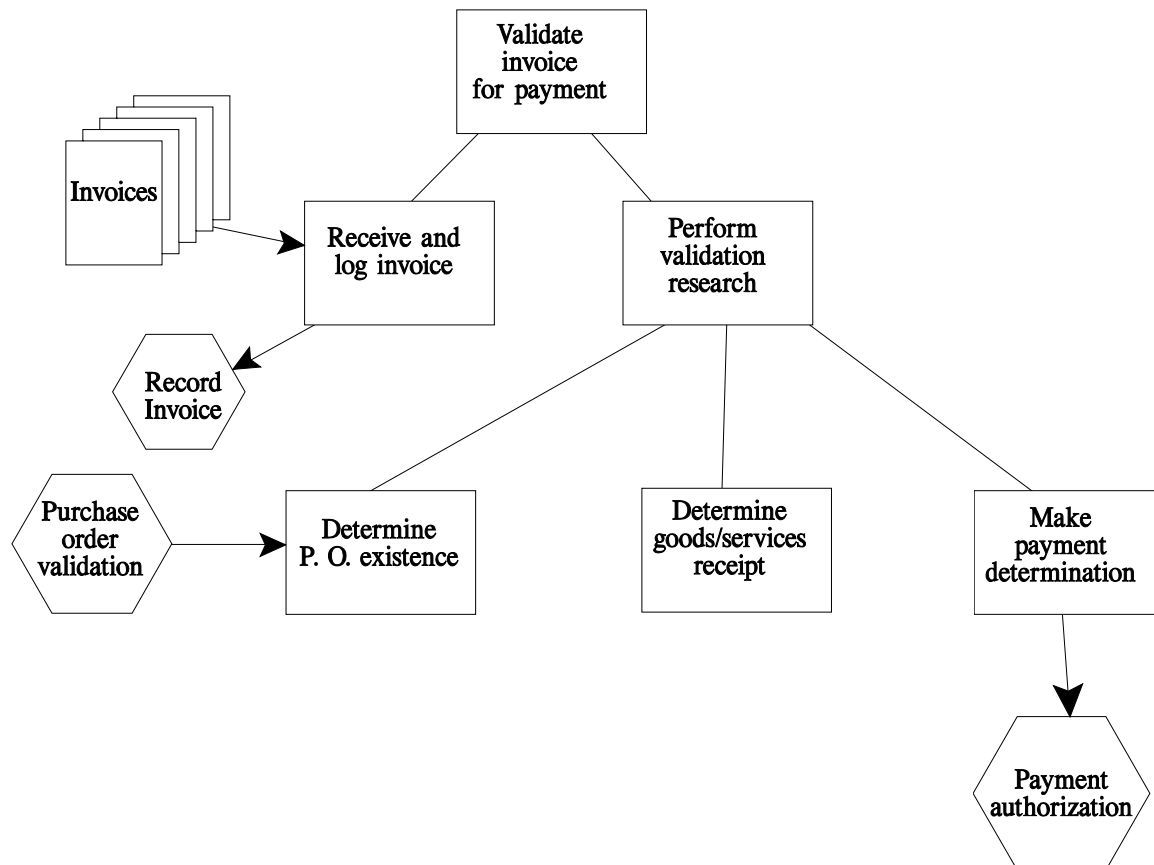
### Traditional Functional Decomposition Paradigm

- Identifies functions and then
- Decomposes into systems, subsystems, programs, and modules, which thus:
  - Maximizes the quantity of defined systems
  - Maximizes process redundancy
  - Maximizes programming, test, and documentation
  - Maximizes redundant maintenance

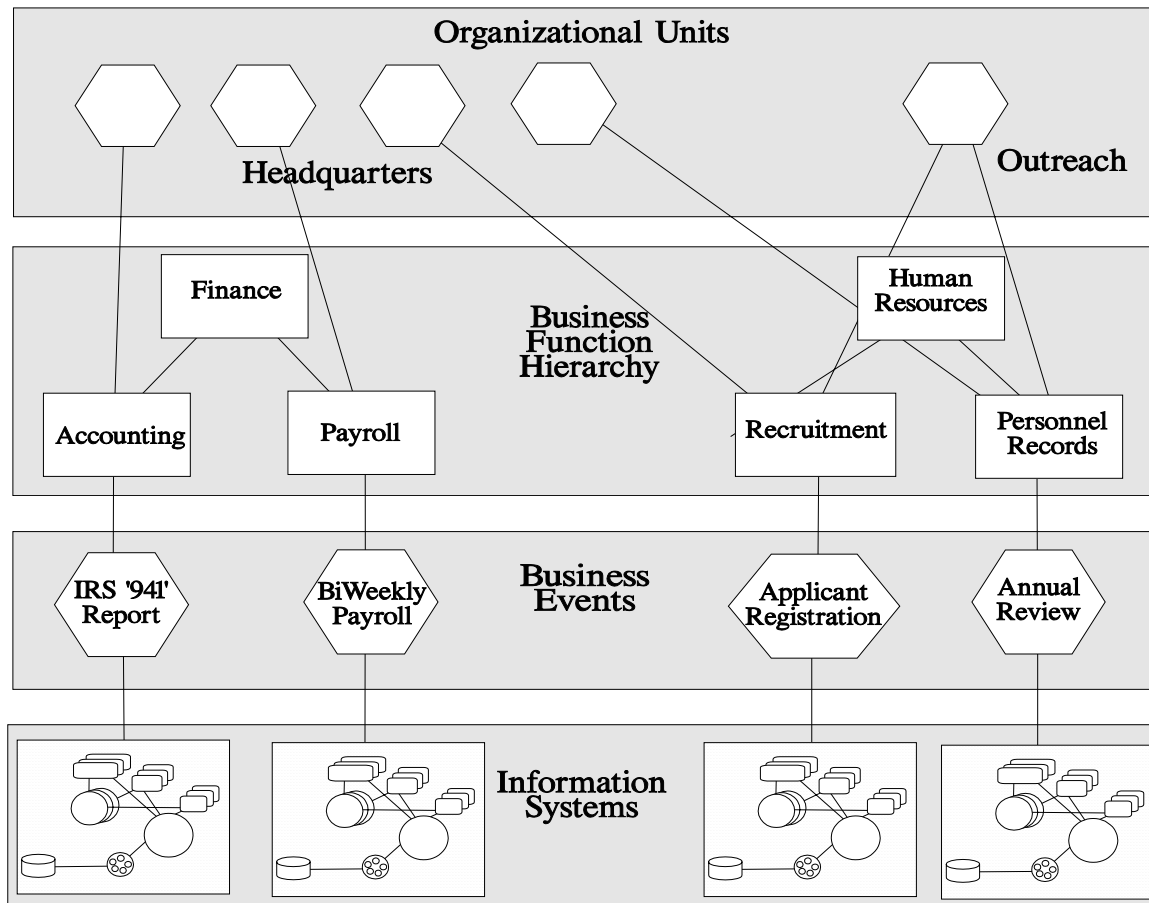
Differences and redundancy for no valid reason.



## Business Function Configuration



## 4.8 Business Organizations



## 4.9 Database Object Summary

Database object based information systems:

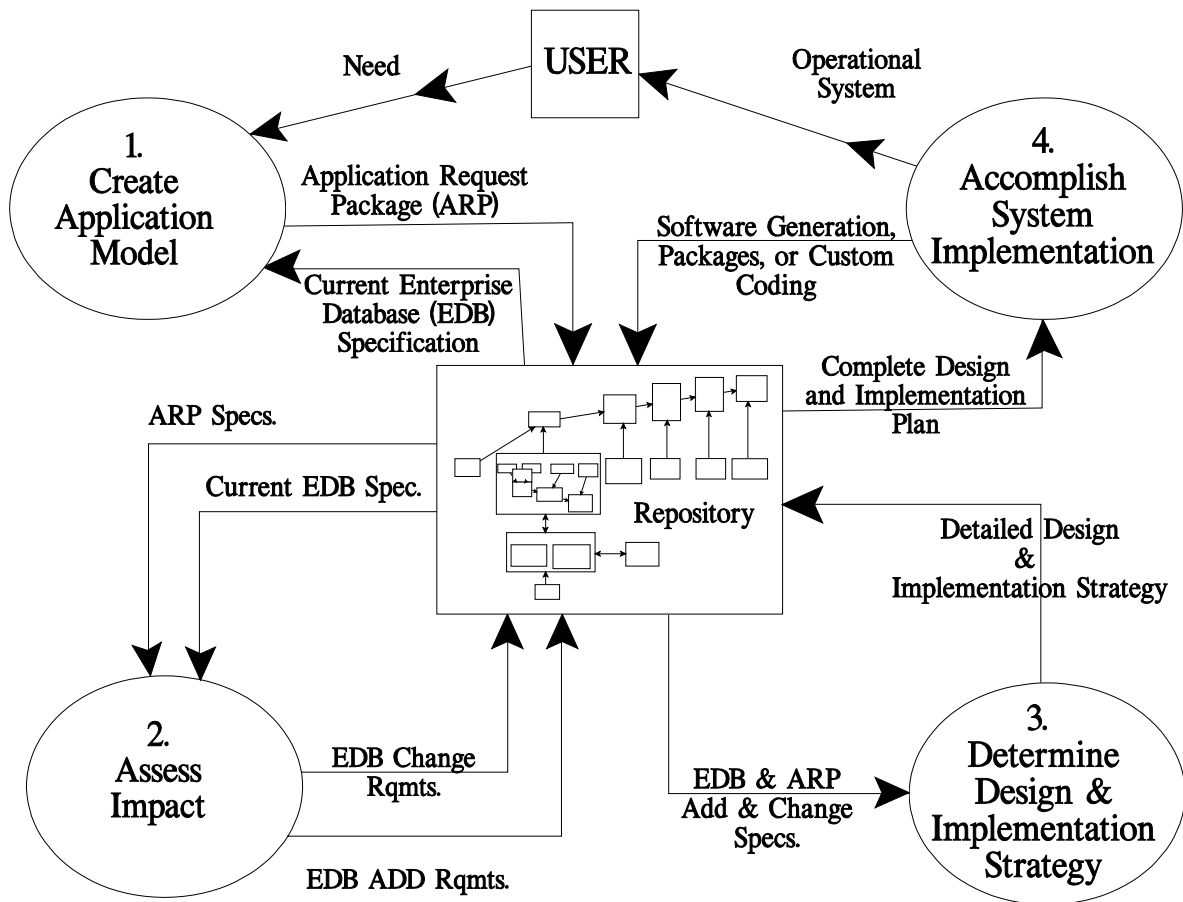
- Accomplish the maximum leveraging of prior work
- Minimize the political conflicts until they are seen as not mattering
- Force attention onto the fundamental policy issues of the enterprise that are derived from the enterprise's missions
- Allow differing styles of enterprise database accomplishment through individualized business function and organization models.



## 5 Methodology Support

Enterprise Database Metabase Models		Preliminary Analysis	Conceptual Specification	Binding	Implementation	Conversion & Deployment	Production & Administration
Mission Model		✓					✓
Database Object Components	Data Object Data Structures	✓	✓				✓
	Database Object Processes		✓				✓
	Database Object Information Systems		✓				✓
	Database Object States		✓				✓
Data Models	Semantic Hierarchies		✓				
	Data Elements		✓				✓
	Specified Data Model		✓				✓
	Implemented Data Model			✓			✓
	Operational Data Model				✓		✓
	Application Interface Model				✓		✓
Business Function Model		✓	✓				✓
Organization Model						✓	✓
Business Information Systems Model				✓	✓		
Security Model			✓		✓		
Deployed Hardware and Network Models			✓	✓	✓	✓	✓





**Because of continuous flow nature you can handle:**

- Multiple, concurrent, but differently scheduled projects against the same operations database or warehouse database
- Single-database projects that affect multiple operations and warehouse databases
- Projects that develop completely new MIS capabilities, that can assess required changes to existing MIS capabilities, and that can accommodate a variety of systems generation alternatives (COTS, package, and custom programming)





## 6 Repository Requirements

Enables:

- The recording and then subsequent utilization of the knowledge and expertise of business analysts in the requirements determination and/or modification process
- The utilization of recorded business requirements by systems analysts, and by data and process administrators to design business systems, and to record these design specifications for subsequent implementation, or to aid in the ongoing evolutions of business systems
- The utilization of recorded design specifications by programmers and database administration staff to implement these business systems, and to record the implementation details of these business systems for subsequent use in the impact analyses critical for the timely evolution of business systems

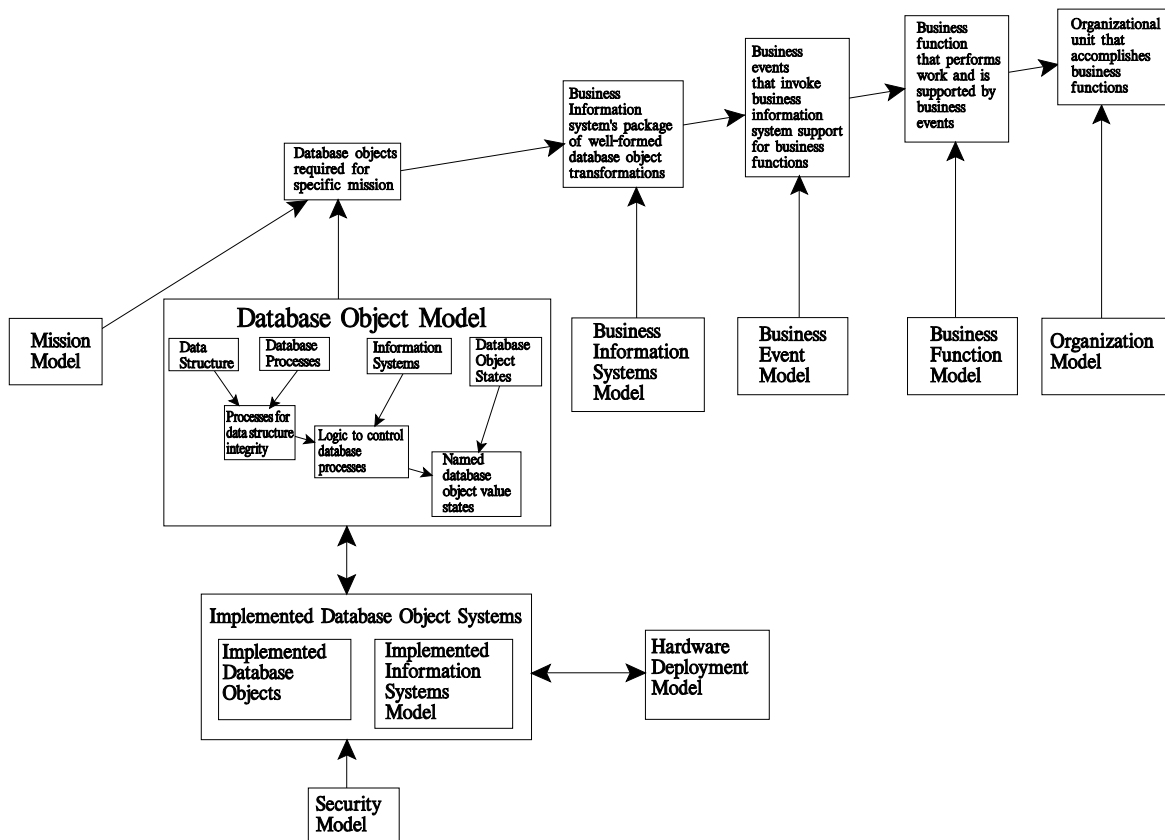


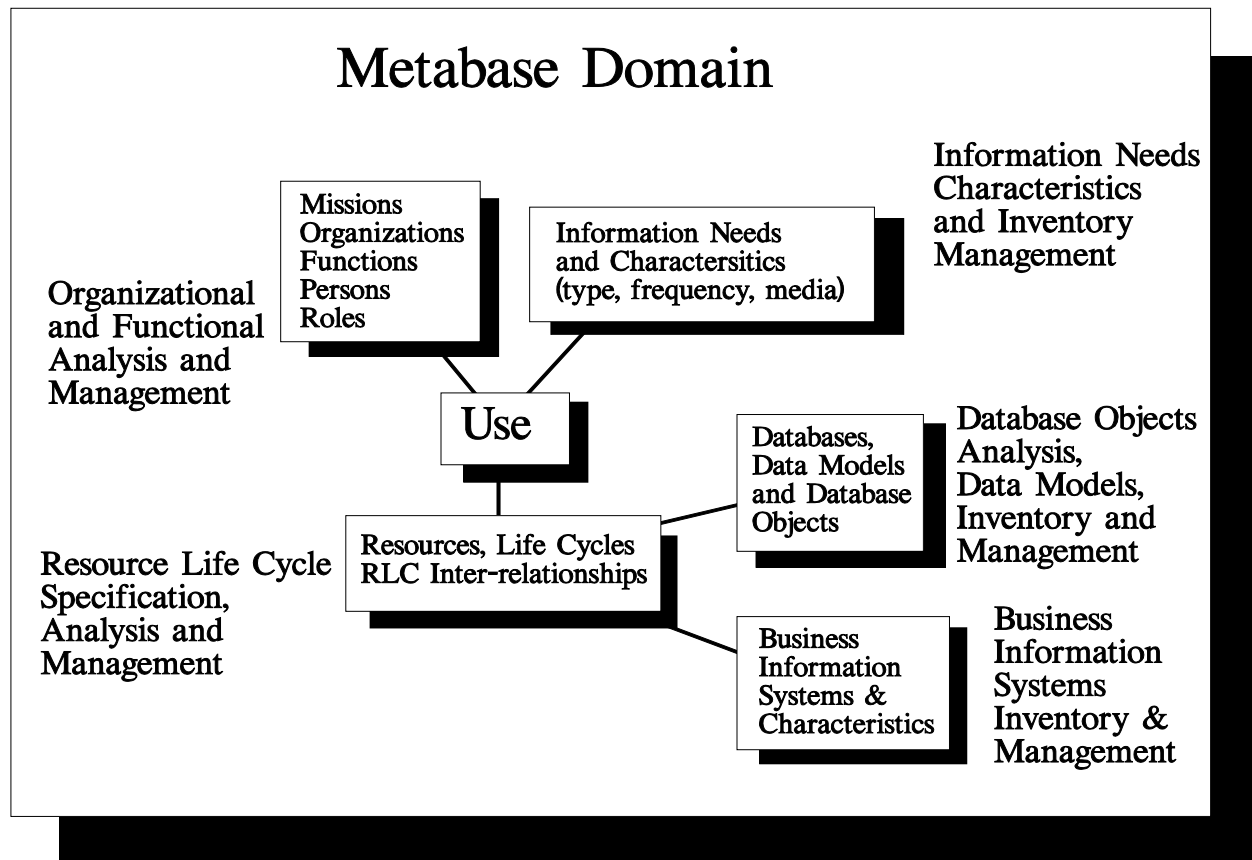
The mission of the enterprise repository environment is to:

- To maintain a single, non redundant storehouse for all enterprise meta database objects as they exist in their three forms: specification, implementation, and operation
- Be accessible on a need-to-know basis to staff as they employ tools for database object specification, implementation and operation
- Be supportive of enterprise database project identification, planning, scheduling, and progress reporting.
- Store, update, and report on the completeness of the enterprise data architecture, that is, the progress towards the instantiation of the five enterprise database classes (reference/strategic, original data collection, transaction data staging area, integrated subject area, and warehouses (wholesale and retail).
- Be accessible for comprehensive reporting of the type, kind, and extent of database object specification, implementation, and operation for individual database applications, organizational units, regions of organizational units, world-wide, and various levels of business functions.
- Support comprehensive reports addressing *where used*, *where same*, and *where different* for all database objects and their contained components for all three dimensions (specification, implementation, and operation)
- Be suitable for assessing the enterprise wide impact of policy and procedure changes that infer changes in database objects.



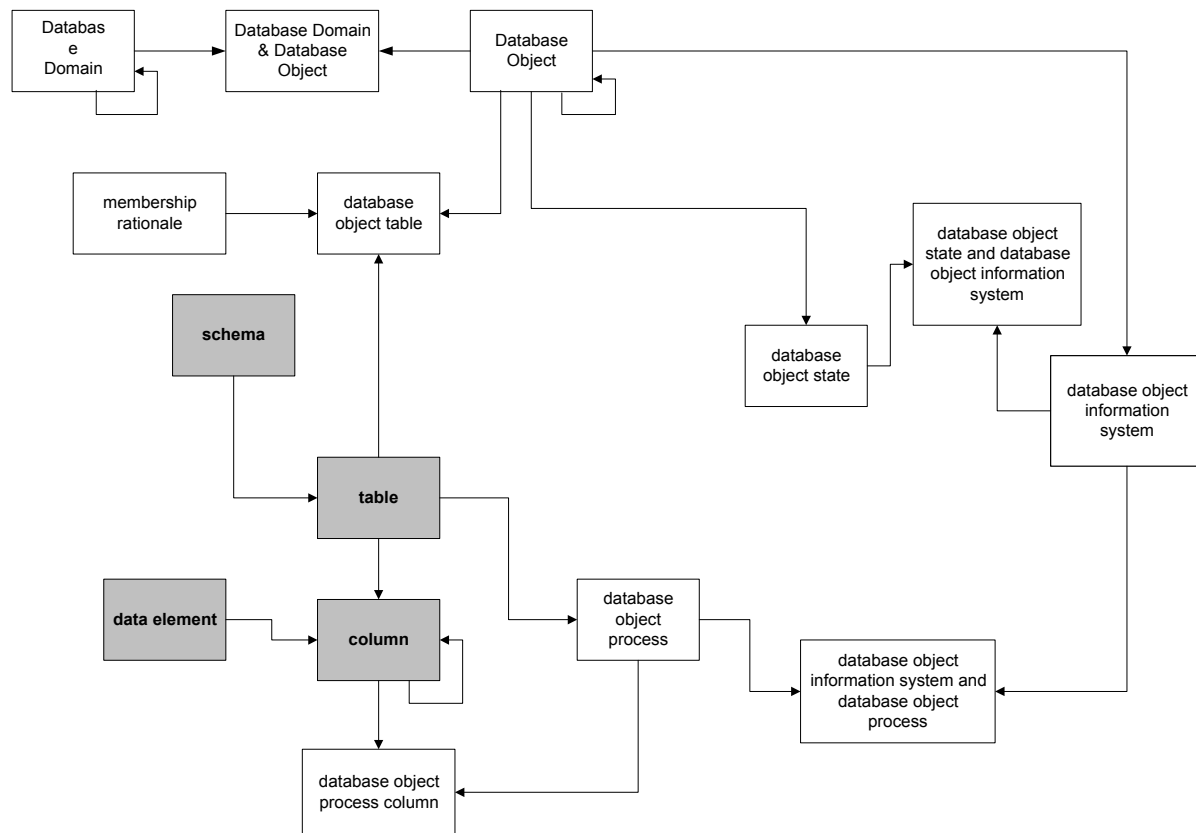
## Metabase Components





Metabase Software Module	Knowledge Worker Framework					
	Mission	Database Objects	Business Information System	Business Event	Business Function	Business Organization
Organizational Analysis	✓			✓	✓	✓
Resource Life Cycles		✓	✓			
Information Needs			✓			
Database		✓				
Data Modeler		✓	✓			
Business Information Systems			✓			
Information Systems Planning	✓	✓	✓	✓	✓	✓
Project Management	✓	✓	✓	✓	✓	✓

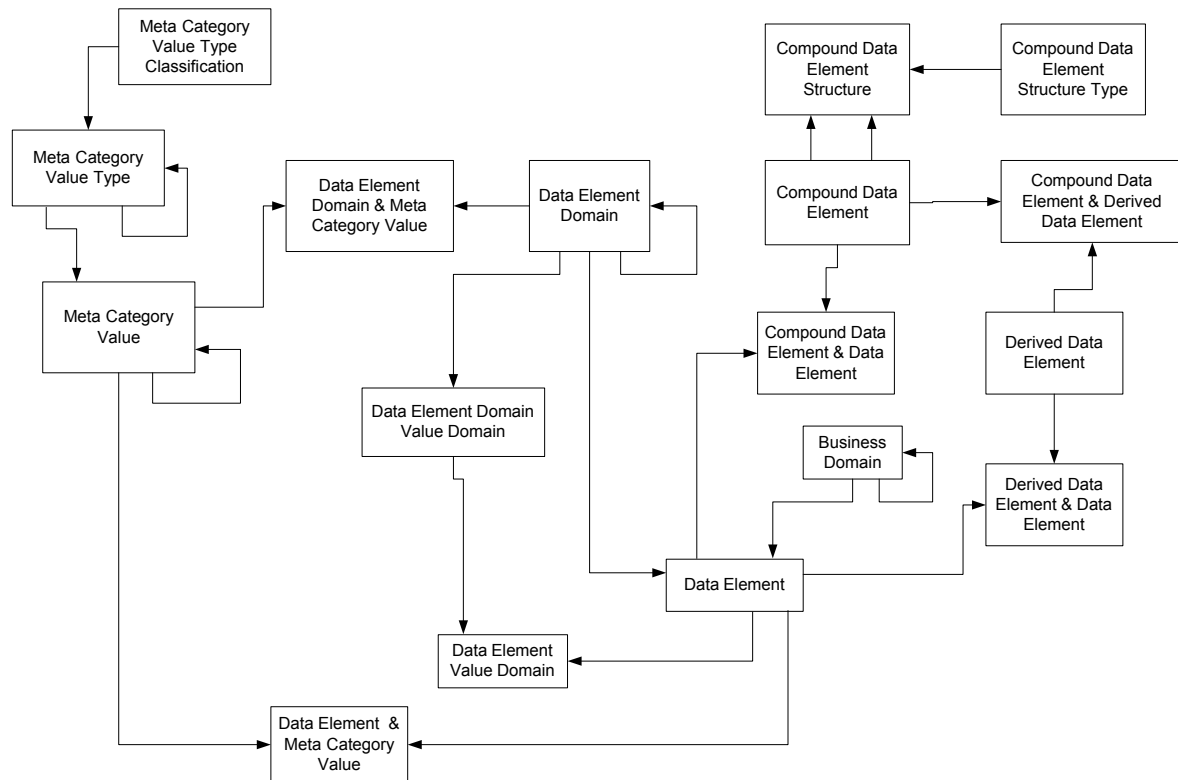




## Database Objects

Copyright 1999, Whitemarsh Information Systems Corporation,  
All Rights Reserved  
7/1/99

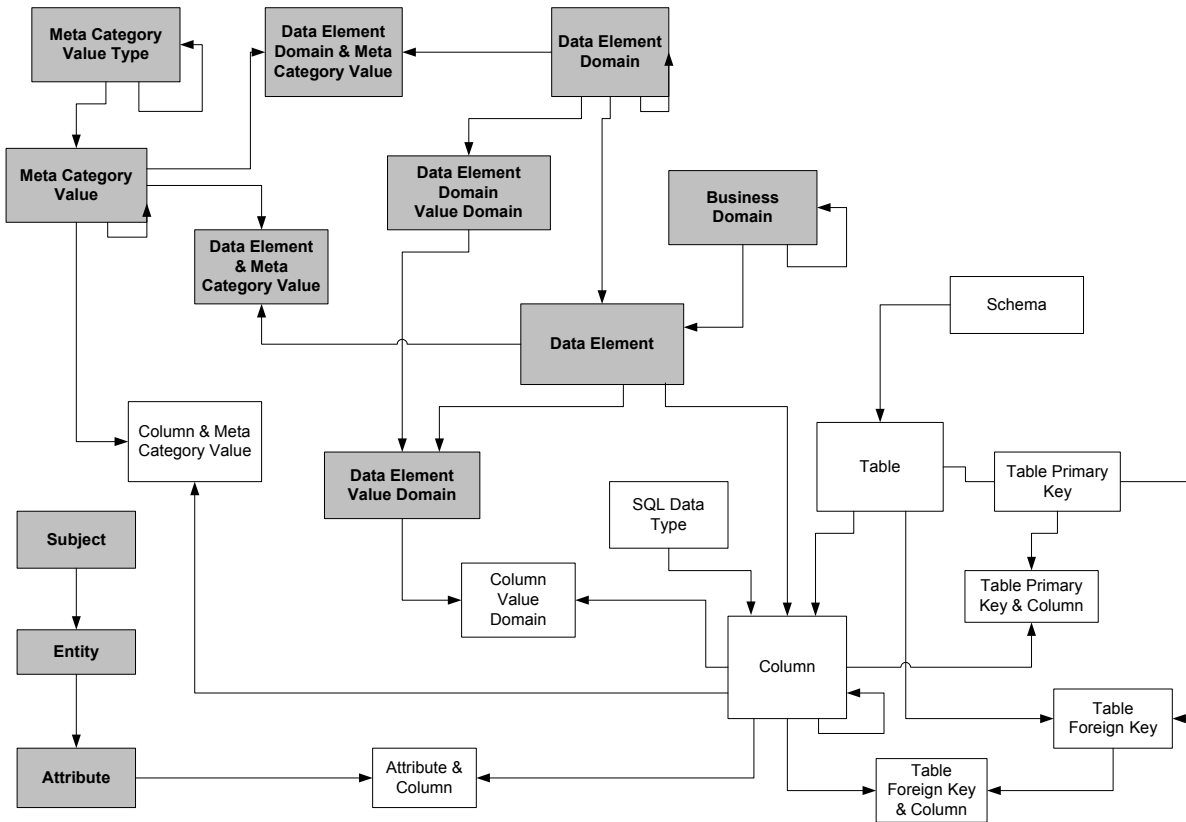




## Data Elements

Copyright 1999, Whitemarsh Information Systems Corporation,  
All Rights Reserved  
11/1/99





## Implemented Data Model

Copyright 1999, Whitemarsh Information Systems Corporation,  
All Rights Reserved  
9/24/99





## 7 ANSI/SQL Support

### 7.1 Data Structure

#### Data Types

- character
- integer
- decimal
- varchar (variable length character)
- bitmap
- enumerated (specific set of values)
- boolean
- blob (binary large object)
- clob (character large object)
- abstract data types (user defined data type with behavior, an encapsulated internal structure, and access characteristics of public, protected, or private)

#### Collection Types

- set
- list
- multiset



## **Subtables**

Multiple levels of nested subtables (real estate (vacation property (summer lodge)))). Note single root is not necessary

## **Row Types**

For groups and nested groups (table person (ssn, name(first, middle, last), address(street, city, state, zip(four, five)))))



## 7.2 Database Processes

### Constraints

- Assertions
- Table constraints
- Unique constraints
- Referential integrity constraints
- Check constraint
- Triggers

### Data Manipulation Operations

- Declare Cursor
- Open
- Fetch
- Close
- Select
- Delete
- Update



## **Transaction Management**

- start transaction
- set transaction
- test completion
- savepoint
- release savepoint
- commit
- rollback



## 7.3 Database Object Information Systems

### Persistent Stored Modules

- Call
- Return
- Compound Statements (Begin ... End)
- If Statements
- Case Statements
- Loop
- Repeat
- While
- For
- Leave
- Assignment
- Signal and Resignal



### **Triggers**

- Different triggering events, update, delete, and insert
- Optional condition
- Activation time: before and after
- Multiple statement action
- Several triggers per table
- User-defined ordering
- Condition and multiple statement action per each row or per statement

### **Transaction Management**

- Start transaction
- Set transaction
- Test completion
- Savepoint
- Release savepoint
- Commit
- Rollback

### **Connection Management**

- Connect
- Set connection
- Disconnect statement



## 7.4 Database Object States

No SQL Statements.



## 8 Summary

- Database Objects are a re-application of business data structures started in the 1960s-1980s
- Database objects returns power and responsibility to the business policy maker. Hurrah!
- Well structured business data are structured representations of business policies
- Database Objects, if implemented totally within the SQL “firewall” will enable world wide policy architectures.
- Database Objects force the need for data standardization -- the proper way, that is, map relationships among same policy data concepts. Don't standardize names (it cannot succeed anyway!)
- It will take longer to achieve enterprise database by waiting, so don't
- To be organized is to have database. Conversely...

