

Whitemarsh
Information Systems Corporation

*Database Management Systems: Understanding and
Applying
Database Technology
Chapters 5*

Interrogation

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

Table of Contents

1.0 Interrogation	1
2.0 Interrogation Introduction	2
2.1 Introduction	3
2.2 User View (Subschema)	4
2.3 Types of Language	6
2.4 Interrogation Summary	12
3.0 Host Language Interface	14
3.1 Introduction	15
3.2 Components of Interface	16
3.2.1 Data Interface Area	17
3.2.2 Message Interface	18
3.2.3 Interface Language Types	19
3.3 Cursors (Currency)	21
3.4 HLI Data Types	24
3.5 Static DBMS--Network Data Model	25
3.5.1 Network Interface Components	26
3.5.2 Network Cursor Model	29
3.5.3 Network Manipulating Capabilities	30
3.5.4 Network Example	39
3.6 Static DBMS--Hierarchy Data Model	47
3.6.2 System 2000 (Currency)	51
3.6.3 Hierarchy Manipulation Capabilities	52
3.6.4 Hierarchical Example	57
3.7 Dynamic DBMS--Independent Logical File Data Model	59
3.7.1 Interface Components	60
3.7.3 Manipulating Capabilities	66
3.7.4 Dynamic HLI Example	70



3.8 Dynamic DBMS-Relational (SQL) Data Model	74
3.8.1 Interface Components	75
3.8.2 Data and Message Interface	76
3.8.3 Relational Cursors	78
3.8.4 Manipulating Capabilities	79
3.9 View Processing	83
3.10 Static Vs, Dynamic HLI Difference	84
3.11 HLI Summary	85
4.0 Procedure Oriented Language	86
4.1 POL Introduction	87
4.2 Capabilities	88
4.3 Static vs Dynamic Comparison	93
4.4 POL Summary	94
5.0 Query-update Language	95
5.1 Introduction	96
5.2 Capabilities	97
5.3 Static vs Dynamic Comparison	104
5.4 Query-update Examples:	105
5.5 Query-update Summary	108
6.0 Report Writers	109
6.1 Report Writer Introduction	110
6.2 Report Writer Capabilities	111
6.3 Report Writer Example	112
6.4 Report Writer Comparison	114
6.5 Report Writer Summary	115
7.0 Application Interfaces	116
7.1 Introduction	117



7.2 Example	118
7.3 Static vs Dynamic Comparison	119
7.4 Application Interface Summary	120
8.0 DBMS Vendor Independent Development Environments	121
8.1 DBMS Interface Drivers	122
8.2 Integrated Development Environments	125
8.2.1 Clarion Demonstration	126
8.2.2 Crystal Reports Demonstration	127
8.3 Summary	128
9.0 Interrogation Summary	129
9.1 Host Language Interface	130
9.2 Procedure Oriented Language (Pol)	131
9.3 Query-update Language	132
9.4 Report Writer	133
9.5 Application Interfaces	134
9.6 Interrogation Summary	135



1.0 Interrogation

- Interrogation Introduction
- Host Language Interface
- Procedure Oriented Language
- Query-update Language
- Report-writer
- Application Interface
- Interrogation Summary



2.0 Interrogation Introduction

- User Schema (Subschema)
- Types of Language
- Control Contrast



2.1 Introduction

Definition: The Process of Formulating Selection, Retrieval, and Then Update Clauses to Obtain Data from the Database Through User Views (Subschemas)

Key Topics

- User View Definition
- Selection
- Retrieval
- Update (Process, Not Effects)



2.2 User View (Subschema)

Two Alternatives:

- Subschema Facility
 - ◆ Data Interface to Schema Tables
 - ◆ User Encodes All Navigation and Access in Program

Or

- Derived View Facility
 - ◆ Normalized View of Tables with
 - ◆ With Navigation thru DBA Specified Verbs
 - ◆ User Interaction Consists of Only: Get, Store, Modify, and/or Delete



Data Interface (Subschema / Derived)

Common Part

Subschema Name Is Subpart

Row Is Contract

Contract Id

Contract Effective Date

Contract End Date

Maximum Order Amount

Row Is Order

Order Id

Order Amount

Language Specific Section

Language Is Fortran

Row Contract

Contract Id Is Integer

Rename to Icontid

Language Is Query

Row Is Contract

Contract Id Is Char 9(x)

Rename to Contract Number

Null Is Error

Blank Is Error



2.3 Types of Language

- Host Language Interface
- Procedure Oriented Language
- Query/update Language
- Report Writer
- Applications & Packages Interfaces



Host Language Interface

- Database Access Through Cobol, Fortran, PL/1, Ada, C
- Principle Purpose
 - ◆ Data Processing Intensive Applications
 - ◆ Data Migration and Transformation To, from and Between Databases and DBMS's.
Provides Maximum Control of User Database Interaction



Procedure Oriented Language (POL)

- ◆ Database Use Through a Vendor Invented Natural Language.
- ◆ Capabilities like Cobol or Fortran At 1/10 the Coding.
- Principle Purpose
 - ◆ Data Selection and Processing.
 - ◆ Data Migration and Transformation To, from and Between Databases of the Same DBMS.
 - ◆ Controlled User Database Access.



Query-update Language

- Another DBMS Vendor Natural Language for Reporting and Updating
- Principle Purpose
 - ◆ Simple Reports/updates from a Single Database
 - ◆ User-controlled Database Access



Report Writer

- Another DBMS Vendor Natural Language
- Principle Purpose
 - ◆ Moderate to Complex Reports With-
 - Row and Column Titles
 - Sorting
 - Various Levels of Break Totals
 - Table-look-up
 - ◆ Great Prototyping for HLI Report Writing.
- Human Resources for Design, Implementation and Maintenance Is 1/10 of HLI



Application Packages

- Specially Created Packages Written by DBMS Vendors to Expedite Use of DBMS
- Typical Packages Include:
 - ◆ Statistics
 - ◆ Financial Management & Accounting
 - ◆ Materials Handling
 - ◆ Plant Management
 - ◆ Human Resource Management
- Principle Purpose
 - ◆ Specialized Packages for Classes of Functional Users
 - ◆ Database Access Allowed Through Functionally Oriented Packages
 - ◆ All DBMS Vendor Natural Languages



2.4 Interrogation Summary

Language Selection Criteria	Interrogation Language Type			
	Host Language Interface	Procedure Oriented Language	Report Writers	Query Update Language
Task Development Effort	Hi	Medium	Medium	Low
Relative Work Units	100	10	10	1
Level of user control over database interaction	Low	Medium	Medium	High
Range of portability from one DBMS to another of same Data Model	Medium to High	Low	Low	Low



Interrogation

- Pick the Right Interrogation Language One for the Job
- Static -- Natural Languages Could Be Better
- Dynamic -- HLI Could Be Better



3.0 Host Language Interface

- Introduction
- Components of Interface
 - ◆ Cursors (Currency)
 - ◆ HLI Data Types
- Data Model Interfaces
 - ◆ Network
 - ◆ Hierarchy
 - ◆ Independent Logical File
 - ◆ Relational
- Derived View Facility – Static - Dynamic Differences
- Summary



3.1 Introduction

- Host Language Interface (HLI)
 - ◆ COBOL, FORTRAN, Assembler Programs
 - ◆ Sophisticated Data Selection
 - ◆ Row Insertion, Modification, Deletion
 - ◆ Row Sorting
 - ◆ Structure Navigation

- Maximum Flexibility for Data Selection, Manipulation, and Reporting

- Maximum Resources Required for Program Design, Implementation and Maintenance

- Minimum Computer Resources for Execution

- Vehicle for Updating If DBMS Has No Schema-based Editing and Validation



3.2 Components of Interface

- Data Area
- Message Area
- Interface Language Types



3.2.1 Data Interface Area

Schema/subschema Systems

- Id Division
- Environment Division
- Data Division
 - ◆ Schema Section
 - DB Staff Within Personnel
 - ◆ File Section
 - ◆ Working Storage Section
 - 01 Subschema - Employee
 - 05 Employee-id
 - 05 Employee-name
 - 05 Etc



3.2.2 Message Interface

- Program-name
- Error-status
- Dbkey Value
- Row-name
- Working Storage Area-name
 - Error-set
 - Error-row
 - Error-area
 - Direct-dbkey
 - Database-status

|
|
|
|



3.2.3 Interface Language Types

- **Specialized Call**

Call 'Ifput' Using Err Workarea Edit-1

- **Generalized Call**

Call 'Adabas' Using

Control-block

Format-buffer

Row-buffer

Search-buffer

Value-buffer

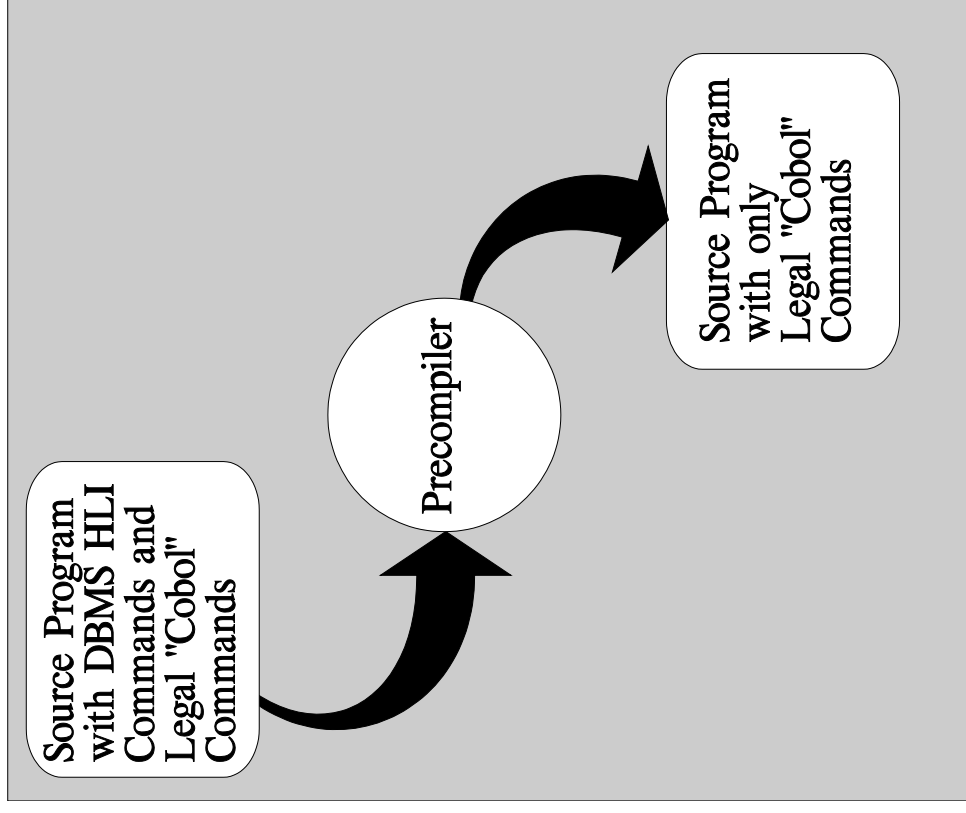
Isn-buffer



3) Pre-compiler

Locate Copies Where
Copyright-date Eq Publication-date

Move 10 to S2kdu1
Move 1 to S2kdu2
Call 'S2kpl' Using Stacks-0
Copies-0 S2r5
Copies-4 Copies-5



3.3 Cursors (Currency)

- Where Am I?
- Where Can I Now Go?
 - ◆ Up?
 - ◆ Down?
 - ◆ Right?
 - ◆ Left?
- Which Program Am I?
- Which Relationship (Set) am I traversing?
- Which Sub-Schema am I using?



Cursor Complexity			
Static		Dynamic	
Network	Hierarchical	Independent Logical File	Relational
Complex	=====	=====	>>>>>> Simple



Domain of the Cursor				
Cursor Manipulated on behalf of	DBMS Type			
	Static		Dynamic	
	Network	Hierarchical	Independent Logical File	Relational
Database	Yes	Yes	Yes	Yes
Area	Yes	No	No	No
Set Type	Yes	No	No	No
Set Instance	Yes	Yes	No	No
Table	Yes	Yes	Yes	Yes
Row Instance	Yes	Yes	Yes	Yes



3.4 HLI Data Types

Language	Schema	Program
COBOL	Character Fixed Numeric Integer Float Real Double-Precision	Numeric Character
Fortran		Integer Real Character Double-Precision



3.5 Static DBMS--Network Data Model

- Interface Components
- Cursors (Currency) Model
- Manipulation Capabilities
- Example



3.5.1 Network Interface Components

- Data Area
- Message Area



Data Area

- Identification Division
Program-id.
Author: Ted Codd and Chris Date
Remarks this Program...
- Environment Division
Input-output Section
File-control
DBMS-control Section
Protocol. Mode Is Batch Debug
IDMS-rows Within Working-storage Section
Schema Section
DB Demoss01 Within Demoschm
01 Subschema-ctrl
01 Subschema-setnames
01 Subschema-areanames
01 Product
03 Prod-cost Pic 59(5)v99
Usage Comp-3
03 Prod-price Pic 59(5)v99
Usage Comp-3



Message Area

01 Subschema-ctrl	
03 Program-name	Picture x(8)value Spaces
04 Error Status	Picture x(4)value "1400"
88 Db-status-ok	Value "0000"
88 Any-status	Value "0000"_"9999"
88 Any-error-status	Value "0001"_"9999"
88 Db-end-of-set	Value "0307"
88 Db-rec-not-found	Value "0326"
03 Dbkey	Picture S9(8)
	Usage Comp
	Sync
03 Row-name	Picture x(16)value Spaces
03 Row-name	Redefines Row-name
05 Ssc-nodn	Picture x(8)
05 Ssc-dbn	Picture x(8)
03 Area-name	Picture x(16)value Spaces
03 Error-set	Picture x(16)value Spaces
03 Error-row	Picture x(16)value Spaces
03 Error-area	Picture x(16)value Spaces
03 DBMSscom-area	Picture x(100)value Low-value
03 DBMSscom	Redefines IDBMScom-area
	Picture x
01 Subschema-ssname	Picture x(8) Value "Demoss01"
01 Subschema-renames	



3.5.2 Network Cursor Model

* ANSI (Codasyl) Network Systems

- Database
- Area (Codasyl Only)
- Set Type
- Set Instance
- Table
- Row Instance



3.5.3 Network Manipulating Capabilities

- Selection
- Sorting
- Retrieval
- Navigation
- Updating
- System Control



Selection, Sorting, Navigation & Retrieval

(1) Find Using a Key

FIND $\left[\begin{array}{c} \text{Keep [Exclusive]} \\ \text{record type name} \end{array} \right]$

DB-Key Is <db-key-field>

(2) Find "Current"

Find $\left[\begin{array}{c} \text{Keep [Exclusive]} \\ \text{Current} \end{array} \right]$

<record-type-name>
Within <Set-name>
Within <Area-name>



(3) Find Within Set/area

Find $\left[\begin{array}{c} \text{Keep [Exclusive]} \end{array} \right]$

$\left\{ \begin{array}{l} \text{Next} \\ \text{Prior} \\ \text{First} \\ \text{Last} \\ \text{"Nth"} \end{array} \right\} \langle \text{record-type-name} \rangle$

Within $\langle \text{Set-name} \rangle$
Within $\langle \text{Area-name} \rangle$

(4) Find Owner

Find $\left[\begin{array}{c} \text{Keep [Exclusive]} \end{array} \right]$

Owner Within $\langle \text{Set-name} \rangle$



(5) Find with Calc

Find	Keep [Exclusive]	Calc	Duplicate
			«record-type-name»

(6) Find Within Sorted Set

Find **[** **Keep [Exclusive]** **]** **<record-type-name>**

Within <set-name> [Current] **Using <sort-name-field>**

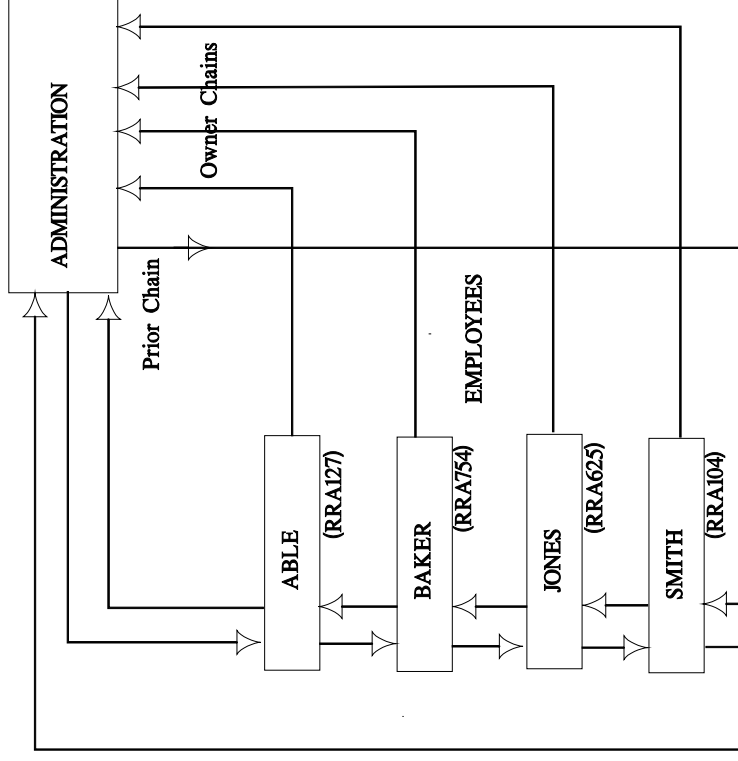


Update Statements

**[Only
Selective
All]**

Delete <record-type-name> Record

Set: Dept-empl
Owner: Department
Member: Employees
Order: Sorted on Last Name



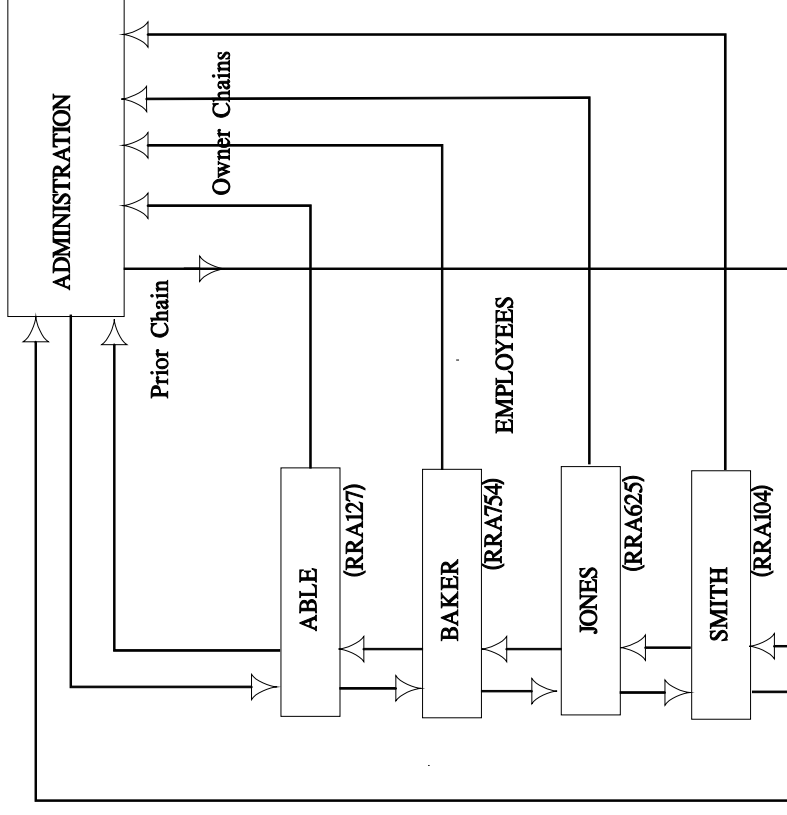
SET: DEPT-EMPL
OWNER: DEPARTMENT
MEMBER: EMPLOYEES
ORDER: SORTED ON LAST NAME



Update Statements

Modify <row-type-name> Row.

Set: Dept-empl
Owner: Department
Member: Employees
Order: Sorted on Last Name



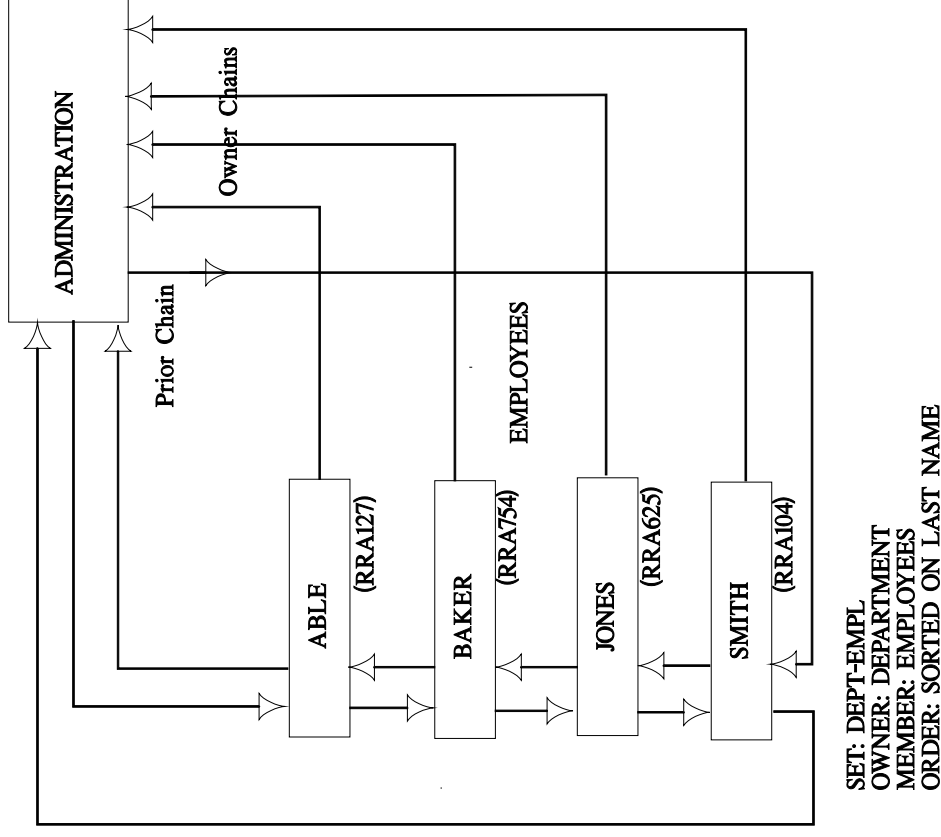
SET: DEPT-EMPL
OWNER: DEPARTMENT
MEMBER: EMPLOYEES
ORDER: SORTED ON LAST NAME



Update Statements

Disconnect <row-type-name> Row.

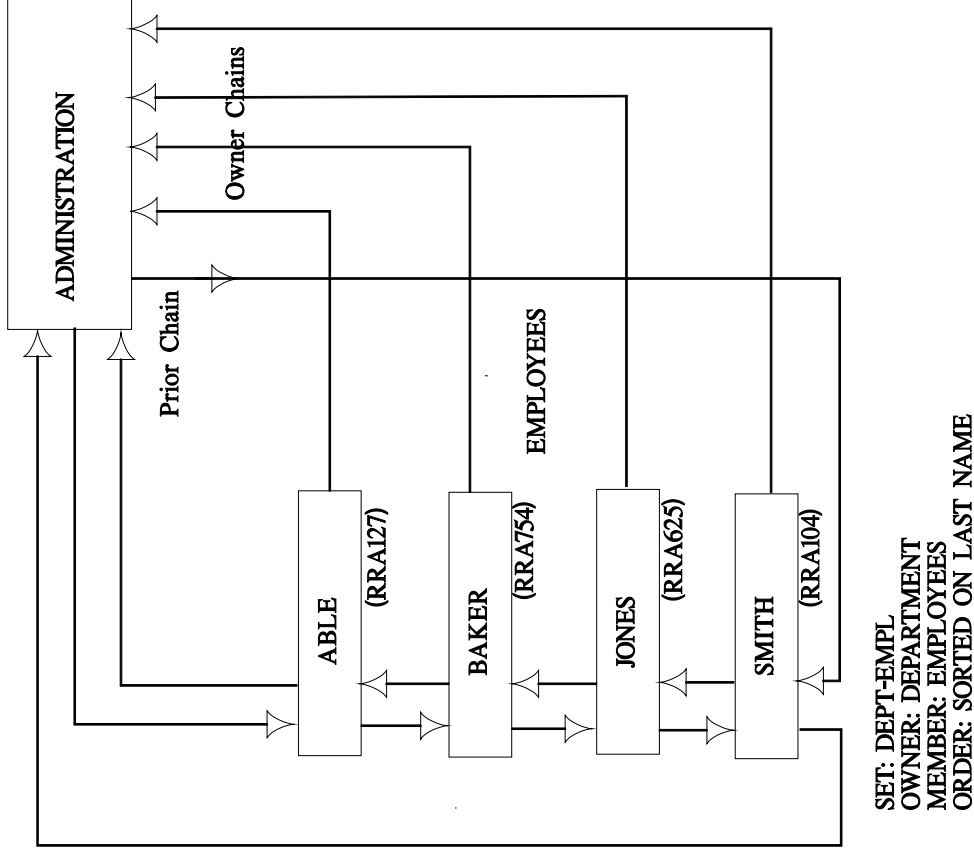
Set: Dept-empl
Owner: Department
Member: Employees
Order: Sorted on Last Name



Update Statements

Store <row-type-name> Row.

Set: Dept-empl
Owner: Department
Member: Employees
Order: Sorted on Last Name



Network System Control Statements

Schema Section

Invoke Subschema <Subschema-name> of <Schema-name>

Open	All Areas Area <area-name>	Usage Mode Is	Retrieval Protected Retrieval Exclusive Retrieval Update Protected Update Exclusive Update
------	-------------------------------	---------------	---



3.5.4 Network Example

Id Division

Program-id.

Author: Ted Codd and Chris Date

Remarks this Program...

Environment Division

Input-output Section

File-control

DBMS-control Section

Protocol. Mode Is Batch Debug

Idms-rows Within Working-storage Section

Schema Section

Db Demoss01 Within Demoschm



Message Interface

01 Subschema-ctrl	
03 Program-name	Pic x(8)value Spaces
04 Error Status	Pic x(4)value"1400"
88 Db-status-ok	Value"0000"
88 Any-status	Value"0000"_"9999"
88 Any-error-status	Value"0001"_"9999"
88 Db-end-of-set	Value "0307"
88 Db-rec-not-found	Value "0326"
03 Dbkey	Pic 59(8) Usage Comp Sync
03 Row-name	Pic x(16)value Spaces
03 Row-name	Redefines Row-name
05 Ssc-nodn	Pic x(8)
05 Ssc-dbn	Pic x(8)
03 Area-name	Pic x(16)value Spaces
03 Error-set	Pic x(16)value Spaces
03 Error-row	Pic x(16)value Spaces
03 Error-area	Pic x(16)value Spaces
03 DBMSscom-area	Pic x(100)value Low-value
03 DBMSscom	Redefines IDBMScom-area Pic x
01 Subschema-ssname	Pic x(8) Value "Demoss01"
01 Subschema-renames	



Set Interface

01 Subschema-setnames

03 Customer-order	Pic x(16) Value"customer-order"
03 Order-oremark	Pic x(16) Value"order-oremark"
03 Order-item	Pic x(16) Value"order-item"
03 Product-item	Pic x(16) Value"product-item"
03 Calc	Pic x(16) Value"calc"



Area Interface

01 Subschema-areanames

03 Customer-region	Pic x(16) Value"customer-region"
03 Order-region	Pic x(16) Value"order-region"
03 Product-region	Pic x(16) Value"product-region"



Table Interface

01 Product	
03 Prod-cost	Pic 59(5)v99 Usage Comp-3
03 Prod-price	Pic 59(5)v99 Usage Comp-3
03 Prod-number	Picx(8)
03 Prod-desc	Picx(20)
03 Prod-specs	
05 Prod-unit-code	Pic X
88 Prod-unit-units	Value"1"
88 Prod-unit-lbs	Value"2"
88 Prod-unit-gals	Value"3"
05 Prod-weight	
05 Prod-color-code	Pic 9(5)v99 Usage Comp-3
03 Filler	Pic x(3)

Move "demoprogram" to Program-name

Bind Run-unit
Bind Product
Bind Oremark
Bind Item
Bind Order
Bind Customer

DBMS-status Section



Data Manipulation

Procedure Division

Copy DBMS Subschema-binds
Ready Usage-mode Is Retrieval

0300-GET-CUST-INFO

Accept Card-image

If Ci-end Go to 9200-EOJ

Move Ci-cust-number to Cust-number, Rpt-cust-no.

Obtain Calc Customer

If Db-rec-not-found

Move "no Customer Row "To Rpt Name

Perform 9000-WRITE THRU 9010-EXIT

Go to 0300-GET-CUST-INFO

Perform Idms-status

Obtain next Order Within Customer-order

Finish



Error Processing

IDMS-status

If Db-status-ok Go to Isabex
Perform Idms-abort
Display

```
*****  
"Aborting -"Program-name  
" " Error-status  
" " Error-row  
****recover Idms*****
```

Upon Console

```
Display "program Name-----" Program-name  
Display "error Status-----" Error-status  
Display "error Row-----" Error-row  
Display "error Set-----" Error-set  
Display "error Area-----" Error-area  
Display "last Good Row--" Row-name  
Display "last Good Area----" Area-name  
Display "dml Sequence-----" Dml-sequence
```

Rollback
Call"abort"
Isabex.exit.



Operation Effects on Rows:

DML Operation	Retention	Mandatory		Optional	
	Insertion	Auto	Manual	Auto	Manual
Set Membership established when the member row is stored		Yes	No	Yes	No
Set membership established by the connect statement		N/A Error	Yes	Yes	Yes
Set membership terminated by the disconnect statement		No Error	No Error	Yes	Yes
Member row deleted by the delete only statement		Yes	Yes	No	No
Member row deleted by the delete selective statement		Yes	Yes	Yes	Yes



3.6 Static DBMS--Hierarchy Data Model

- Interface Components
- Cursors (Currency) Model
- Manipulation Capabilities
- Example



3.6.1 Interface Components

- Data Interface
- Message Areas



Data Interface

System 2000 Subschema

Schema Education of Personnel.

01 Education.

02 Degree Picture Is Name X(20).

02 Year Picture Is Integer 9999.

Schema Study Areas of Personnel.

01 Study Areas.

02 Subject Picture Is Name X(20).

02 Credits Picture Is Integer 99.

Schema Employee of Personnel.

01 Employee

02 Employee Number Picture Is Integer 9(9).

02 Employee Name Picture Is Name X(30).



Message Interface (IMS)

Hierarchy - - DBMS: IMS

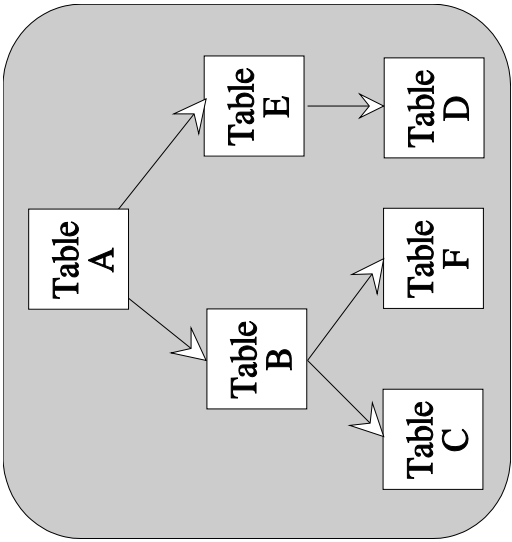
Mask Written in Cobol

```
01 Pcbname
  02 Dbd-name      Pic X(8)
  02 Seg-level     Pic XX
  02 Status-code   Pic XX
  02 Proc-options  Pic XXXX
  02 Reserve-DL/1  Pic S9(5)
  02 Seg-name-fb   Pic X(8)
  02 Length-fb-key Pic S9(5)
  02 Numb-sens-segs Pic S9(5)
  02 Key-fb-area
    03 Name-key    Pic X(12)
    03 Name-addr-key Pic X(14)
    03 Name-payroll-key Pic X(17)
```



3.6.2 System 2000 (Currency)

LEVEL	STACK NUMBER			
	0	1	2	3 15
0	a,1			
1	b,1		e,1	
2	c,1	d,1	f,1	
3				
4				
5				
6				
.				
.				
.				
.				
31				



3.6.3 Hierarchy Manipulation Capabilities

- Selection
- Sorting
- Retrieval
- Navigation
- Updating
- System Control



Selection & Retrieval

Get1 <Subschema-name>

Get1 <Subschema-name> Where.....

Locate <Subschema-name> Where....



Sorting

Locate.....

Order by <Field-name>.....

Navigation

- Get Member <Row-name>
- Get next <Row-name>
- Get Owner
- Link <Cousin-1> to <Cousin-2> via <Cousin-1-field> Eq <Cousin-2-field>



Update

- Insert <Row-type-name>
- Delete <Row-type-name>
- Modify <Row-type-name>
- Remove Tree



System Control

- Open/lock
- Get/hold
- Remove Tree



3.6.4 Hierarchical Example

Step 1: /* Initialization */

```
Password = "Demo";
Start S2k ;
If S2krtc <= 0 Then
Do;
    Put Skip List ("S2krtc <= 0 on Start S2k");
    Stop S2k;
    Exit;
End;

Open Employee;
If Damage = 1 | Return Code <= 0 Then
Do;
    Call Displaycomm("Open");
    Goto Step7;
Open File (Reader);
On Endfile (Reader) Go to Step7;
```



Step2: /* Read One Input Card*/

Get File(reader) List (Number, payday, total Hours, expected
Input.vacation, sickness, incometax);

Step3: /* Find the Entry in the Data Base for this Employee.

```
/* Also, find Current Salary
Get1 Entry Where Enumer Eq Number;
If Return Code <= 0 Then
    Do;
        Call Displaycomm("Get1");
        Goto Step7;
    End;
Getd Salary First;
If Return Code <= 0 Then
    Do;
        Call Displaycomm("Getd");
        Goto Step7;
    End;
```



3.7 Dynamic DBMS--Independent Logical File Data Model

- Interface Components
- Cursors
- Manipulation Capabilities
- Examples



3.7.1 Interface Components

- Data Interface
- Message Areas



Data and Message Interface

Adabas

General Purpose

| |

Row-buffer

| |

General Purpose

| |

Format-buffer

| |



Data and Message Interface (Cont)

Inquire

		Working Storage Section.
01	Pcb-jobhist	
02	Fieldname	
02	Subscript-no	
02	Action	
02	Length	
02	Jobhist-data	
03	Startdte	
03	Enddte	
03	Histsal	
03	Histtitl	
03	Histdept	



Data and Message Interface (Cont)

Skeletal Cobol DMLC Program

```
01 DMSA.  
    05 Db-retcod           Pic Si(36).  
    05 Db-elt-err          Pic X(16).  
    05 Db-rec-err          Pic X(16).  
    05 Db-struc-err        Pic X(16).  
    05 Db-funct-code       Pic 9(02).  
    05 Db-column-number    Pic 9(03).  
    05 Db-line-number      Pic X(09).  
    05 Db-program-id       Pic X(06).  
    05 Db-error-sub1       Pic 9(03).  
    05 Db-error-sub2       Pic 9(03).  
    05 Db-error-code-ct    Pic 9(02).  
    05 Db-exist            Pic 1.  
    05 Db-null            Pic 1.  
    05 Db-test-ind         Pic 1.  
    05 Filler              Pic 1(33).
```



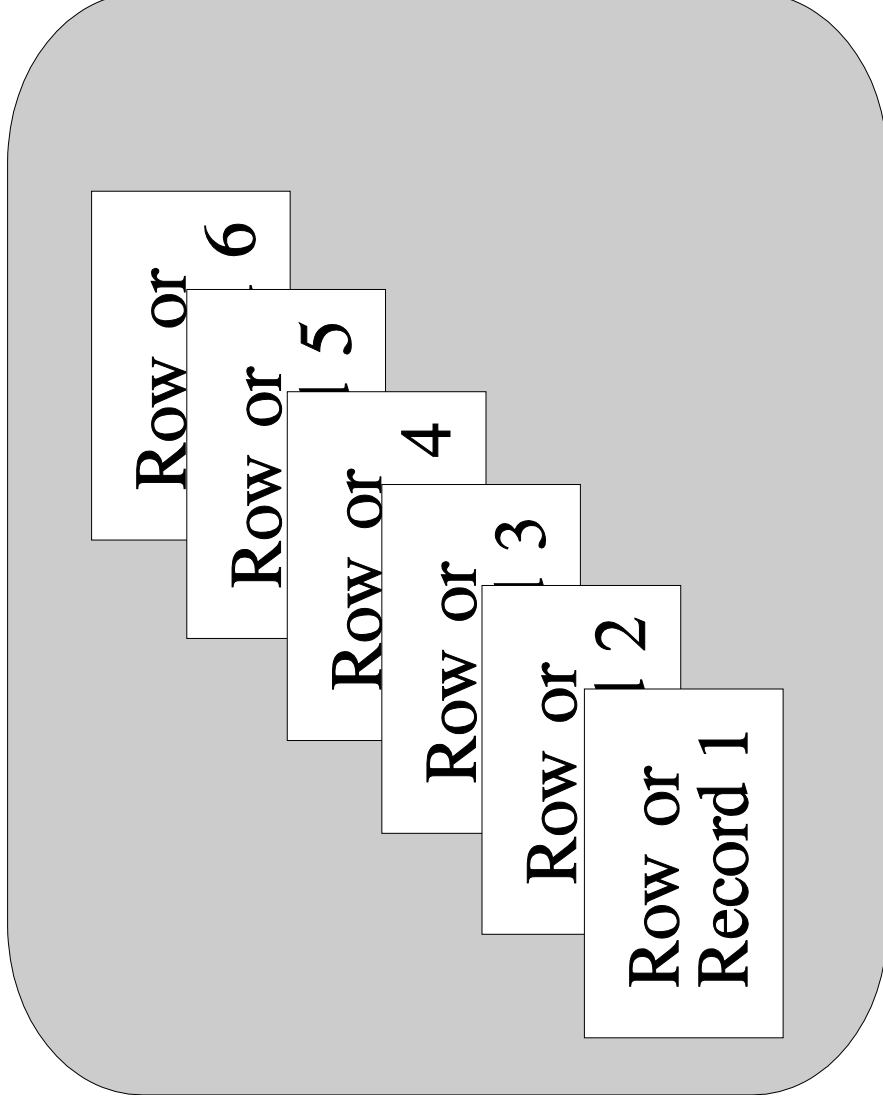
Data and Message Interface (Cont)

```
05 Db-customer.  
  10 Db-customer-nl Pic 1.  
  10 Filler Pic 1 (35).  
  10 Db-div.  
    15 Db-div-nl Pic 1.  
    15 Filler Pic 1 (35).  
  10 Db-sale.  
    15 Db-sale-nl Pic 1.  
    15 Filler Pic 1 (35).  
  10 Db-empl.  
    15 Db-salary-history-ct Pic S9(10) Usage Comp.  
    15 Db-empl-nl Pic 1.  
    15 Filler Pic 1 (35).
```



3.7.2 ILF/relational Cursors

- Domain: Single Table
- Position: Within Table
- Or SQL View Contained set of database tables



3.7.3 Manipulating Capabilities

- Selection
- Sorting
- Retrieval
- Navigation
- Updating
- System Control



Selection

Find <Row-name> Where.....

Sorting

Find-sorted <Row Name> Where

Retrieval

Get	First	Logical
	Next	
	Last	



Navigation (DML Based)

- Connect.....
- Find Coupled....

Updating

- Update/modify <Field-name>
- Add/delete <Row-type>



System Control

- Audit Trails
- Recovery
- Reorganization
- DDL Creation



3.7.4 Dynamic HLI Example

Identification Division.

Program-id. Dml001

Remarks.

Environment Division.

Configuration Section

Input-output Section

File-control

Select Rpt001 Assign to Printer

Data Division

Db Section

Subschema Dem04

File Section



Precompiled Reference to Messages Area

01 DMSA.

05 Db-retcod	Pic Si (36).
05 Db-elt-err	Pic X (16).
05 Db-rec-err	Pic X (16).
05 Db-struc-err	Pic X (16).
05 Db-funct-code	Pic 9 (02).
05 Db-column-number	Pic 9 (03).
05 Db-line-number	Pic X (09).
05 Db-program-id	Pic X (06).
05 Db-error-sub1	Pic 9 (03).
05 Db-error-sub2	Pic 9 (03).
05 Db-error-code-ct	Pic 9 (02).
05 Db-exist	Pic 1.
05 Db-null	Pic 1.
05 Db-test-ind	Pic 1.
05 Filler	Pic 1 (33).



Precompiled Reference to Tables Within Subschema

05 Db-customer.
 10 Db-customer-nl Pic 1.
 10 Filler Pic 1 (35).
10 Db-div.
 15 Db-div-nl Pic 1.
 15 Filler Pic 1 (35).
10 Db-sale.
 15 Db-sale-nl Pic 1.
 15 Filler Pic 1 (35).
10 Db-empl.
 15 Db-salary-history-ct Pic S9(10) Usage Comp.
 15 Db-empl-nl Pic 1.
 15 Filler Pic 1 (35).



Statements from the Procedure Division (Not Precompiled)

Procedure Division.

Init Section.

I1.

*Invoke on Error Display Msg1 Go to Fini
Open Output Rpt001.
Perform Heading-rtn.

I2

*Ready Adm1 Usage-mode Shared Retrieval
*On Error Display Msg2 Go to End-it.

I3.

Move Spaces to Q-value.
Display "please Enter Quota as 8 Char Alphanumeric,
Left-justified W/trailing Blanks".
Accept Q-value.
*Scan Sale Using Selct.
End-it.

Close Rpt001.

Display "Report Complete".

Fini.



3.8 Dynamic DBMS-Relational (SQL) Data Model

- Interface Components
- Cursors
- Manipulation Capabilities
- Examples



3.8.1 Interface Components

- Data Interface
- Message Areas



3.8.2 Data and Message Interface

- ANSI/SQL

* Declare the SQLSTATE status variable.

EXEC SQL BEGIN DECLARE SECTION END-EXEC.

...

01 SQLSTATE PIC X(5).

...

EXEC SQL END DECLARE SECTION END-EXEC.

00 successful completion
01 warning
02 no data
07 dynamic SQL error
08 connection exception
0A feature not supported
21 cardinality violation
22 data exception
23 integrity constraint violation
24 invalid cursor state
25 invalid transaction state
etc...



SQL Communications Area

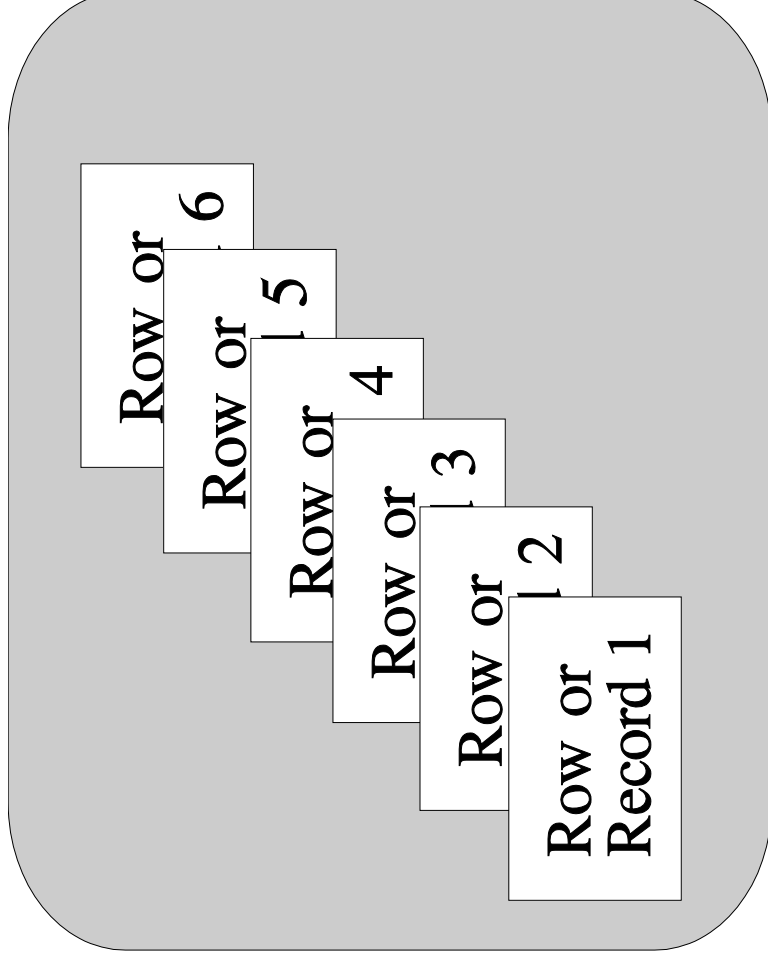
01 SQLCA.
05 SQLCAID PIC X(8).
05 SQLCABC PIC S9(9) COMPUTATIONAL.
05 SQLCODE PIC S9(9) COMPUTATIONAL.
05 SQLERRM.
49 SQLERRML PIC S9(4) COMPUTATIONAL.
49 SQLERRMC PIC X(70)
05 SQLERRP PIC X(8).
05 SQLERRD OCCURS 6 TIMES
PIC S9(9) COMPUTATIONAL.
05 SQLWARN.
10 SQLWARNO PIC X(1).
10 SQLWARN1 PIC X(1).
10 SQLWARN2 PIC X(1).
10 SQLWARN3 PIC X(1).
10 SQLWARN4 PIC X(1).
10 SQLWARN5 PIC X(1).
10 SQLWARN6 PIC X(1).
10 SQLWARN7 PIC X(1).
05 SQLEXT PIC X(8).



3.8.3 Relational Cursors

Domain: Single Table

Position: Within Table



3.8.4 Manipulating Capabilities

- Selection
- Sorting
- Retrieval
- Navigation
- Updating
- System Control



Selection, Retrieval and Sorting

Exec SQL Select Status, City..

Into :Rank, :City

From S

Where S# = :Givens#

Order by S#



Navigation (DML Based)

Exec SQL Declare X Cursor for

Select S#, Sname, Status

From S

Where City = :Y;

Exec SQL Open X;

Do While

Exec SQL Fetch X into :S#, :Sname, :Status;

End;

Exec SQL Close X;



Updating

Exec SQL Update S

```
Set Status = Status + :Raise  
Where City = 'London';
```

Exec SQL Delete

```
From Sp  
Where :City =  
(Select City  
From S  
Where S.s# = Sp.s#);
```

Exec SQL Insert

```
Into P (P#, Pname, Weight)  
Values (:Pno, :Pname, :Pwt);
```



3.9 View Processing

Create View MissionOrganizationFunction as

```
select "mis"."missionid", "mis"."mbname", "misorg"."missionorganizationid", "fct"."functionid", "fct"."mbname",  
"org"."organizationid", "org"."mbname"  
from (((("sysadm"."org" org inner join "sysadm"."misorg" misorg on org. "Organizationid" = misorg. "Organizationid")  
Inner join "sysadm"."mis" mis on mis.missionid = misorg.missionid)  
Inner join "sysadm"."fct" fct on fct.functionid = misorgfc.functionid)  
Inner join "sysadm"."misorgfc" misorgfc on misorg.missionorganizationid = misorgfc.missionorganizationid);
```

Data Manipulation Verbs

Exec SQL Select MissionOrganizationFunction Where



3.10 Static Vs, Dynamic HLI Difference

Capability	DBMS Type	
	Static	Dynamic
Selection	Med to Low	High
Sorting	Low	High
Navigation	High	Low
Retrieval	OK	OK
Updating	OK	OK
System Control	Varied to Low	Varied to High



3.11 HLI Summary

- Database Interaction

Through Cobol, Fortran, PL/1, Etc.

- Principle Role

Data Processing Intensive Applications

Data Migration and Transformation To, from and Between Databases and DBMS's

Maximum Control of User Data Base Interaction



4.0 Procedure Oriented Language

Introduction

Capabilities

Static - Dynamic Comparison

Summary



4.1 POL Introduction

- A Natural Language (Vendor Proprietary) for
 - ◆ Analytic Users
 - ◆ Programmers
 - ◆ Analysts
- For Moderate to Sophisticated
 - ◆ Reports
 - ◆ Updates
 - ◆ Analysis
- Vendor Proprietary because there is no ANSI Standards Committee for these “4GLs”
- When POLs access databases via ODBC then they are SQL Vendor Independent.



4.2 Capabilities

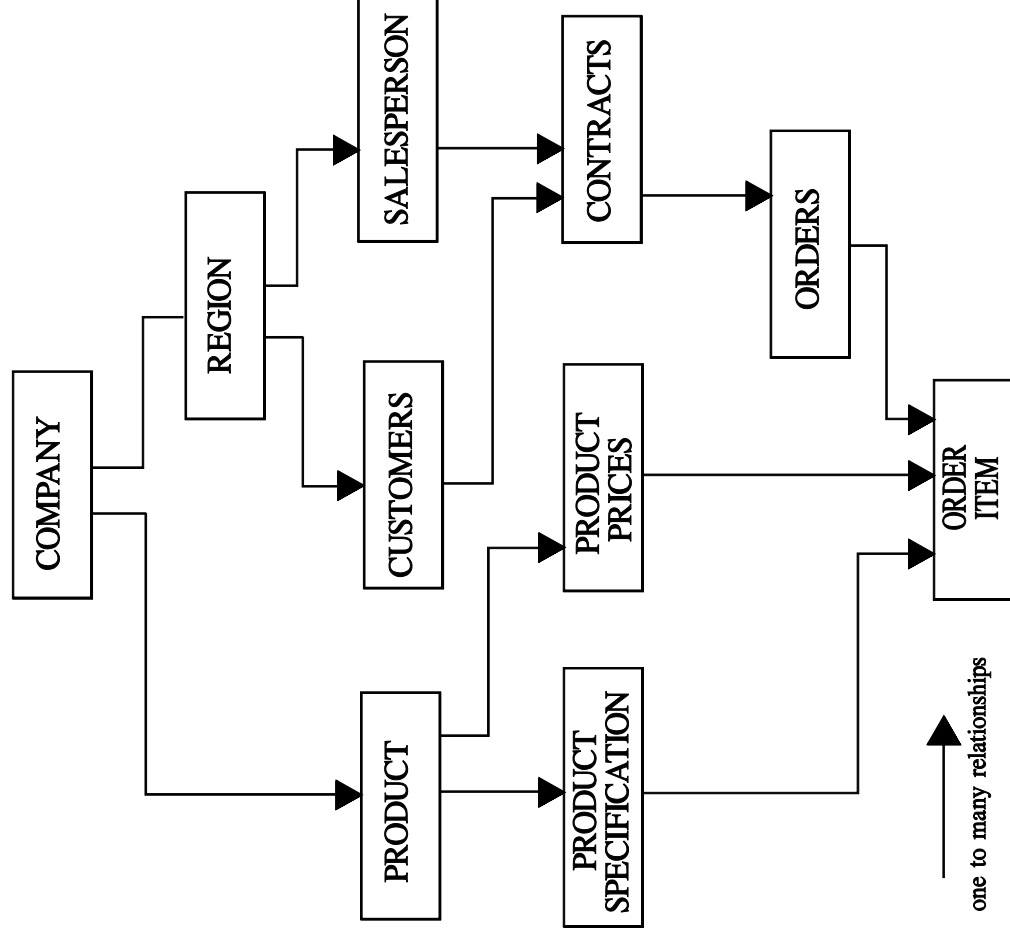
- Screen Development and Prompting,
- Data Validation,
- Detailed Data Editing,
- Selection,
- Branching,
- Looping
- Page Headers, and Break Totals,
- Sorting,
- Computations,
- Character String Search,
- Arithmetic,
- Relational, &
- Logical Selection Operators

In short, complete application system development.



Network Data Model

One Database with Nine Tables



Static POL

Problem: Print Salespersons Who Sell Products Manufactured Where They Were Born

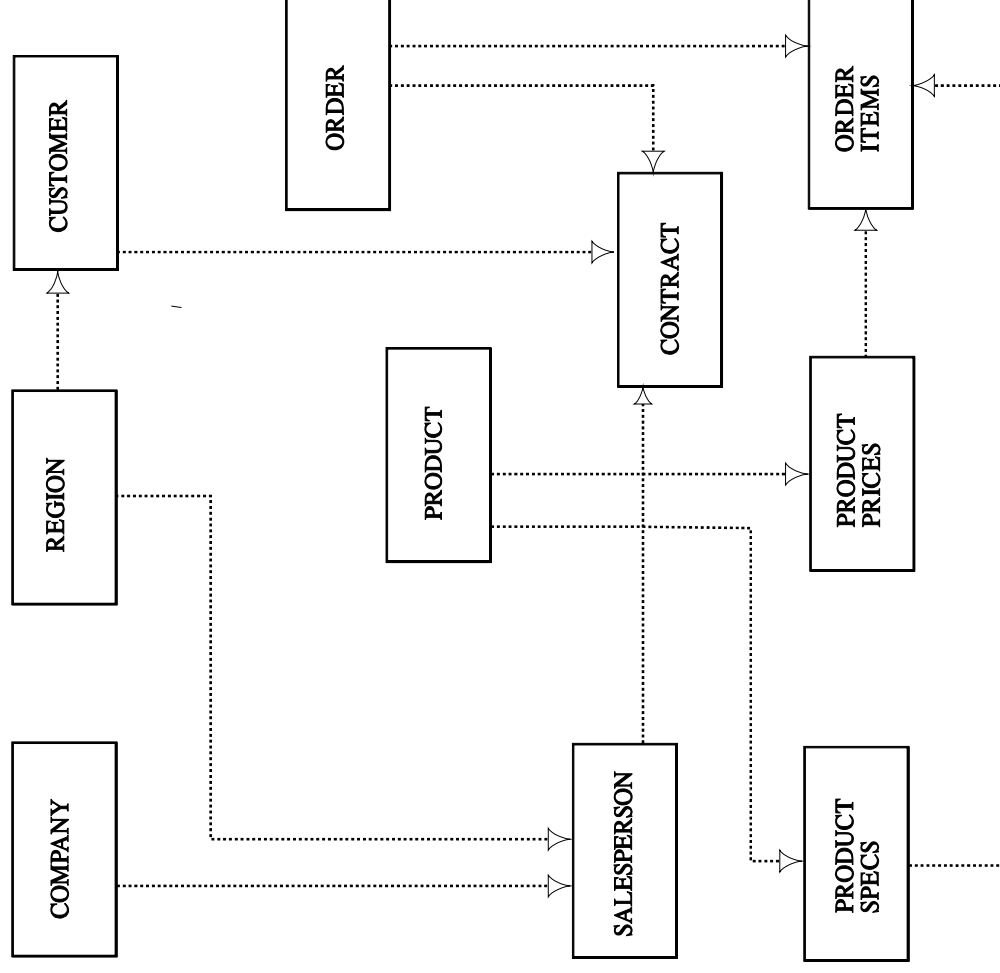
~~~~~

```
10  Select Salesperson, on Error Goto 100, at End Goto 110
20  Get Member Contract, on Error Goto 100, at End Goto 10
30  Get Member Order, on Error Goto 100, at End Goto 20
40  Get Member Order-items, on Error Goto 100, at End Goto 30
50  Get Owner Product-specifications, on Error Goto 100
60  Get Owner Product, on Error Goto 100
70  If Product-mfg-city Eq Salesperson-birth-city and
    Product-mfg-state Eq Salesperson-birth-state
    Then Print Salesperson-name, Product-name,
    Salesperson Birth-city, Salesperson-birth-state
    Else Goto 40
100 Print "Error"
110 End
```



## Relational Data Model

### Nine Simple Row-type Databases



## **Dynamic POL**

Problem: Print Salespersons Who Sell Products Manufactured Where They Were Born

~~~~~  
Connect Product to Order-item Via Product-nbr Eq Order-item-product-nbr

Find Match and Keep Product-name, Order-id, Product-mfg-city, Product-mfg-state in List-1

Connect Salesperson to Contract Via Salesperson-nbr Eq Contract-salesperson-id

Find Match and Keep Salesperson-name, Contract-id, Salesperson-birth-city, Salesperson-birth-state
In List-2

Connect List-2 to Order via Contract-id of List-2 Eq Order-contract-id

Find Match and Keep Order-id, Sales-person-name, Salesperson-birth-city, Salesperson-birth-state
In List-3

Connect List-1 to List-3 Via Order-id of List-1 Eq Order-id of List-3 and Salesperson-birth-city Eq
Product-mfg-city and Salesperson-birth-state Eq Product-mfg-state

Find Match and Print Salesperson-name, Product-name, Salesperson-birth-city,
Salesperson-birth-state



4.3 Static vs Dynamic Comparison

Capability or Facility	DBMS Type	
	Static	Dynamic
Selection	Low to None	High
Sorting	Low to None	High
Navigation	Low to None	Medium
Reporting	Low to None	High
Updating	Low to None	High
System Control	Low to None	Medium to High



4.4 POL Summary

- Capabilities Similar to HLI
- Proprietary Natural Language
- Great for Prototyping HLI
- Human Resources (1/3 of HLI) for
 - ◆ Design,
 - ◆ Implementation,
 - ◆ And Maintenance
- Consumes Much More Computer Resources



5.0 Query-update Language

- Introduction
- Capabilities
- Static/dynamic Comparison
- Examples
- Summary



5.1 Introduction

Another DBMS Vendor Natural Language for Reporting and Updating

- Principle Role
 - ◆ Simple Reports/queries from a Single Database
 - ◆ User-controlled Database Access
 - ◆ Quick and Immediate Capability for User Updates
- ANSI Standard SQL is NOT-Proprietary. However Vendor Extensions are.



5.2 Capabilities

- Retrieval
- Updating
- Selection
- Sorting
- Sorting,
- Statistical Functions
- Output Formatting



Retrieval Commands

Print <Table Name>

Example: Print Student

Print <Column Name,>.....

Example: Print Ssn, Sex, Gpa



Updating

- Single Column Changes
Change Status to "Closed" If Store Eq "Aq94871".
- Multiple Column Changes
Change Status to "Closed", Reason to "Bankrupt" If Store Eq "Aq94871".
- Single Column, Multiple Value
Change Salary to Salary*1.10 Where Employee Exists
- Single Row Deletion
Remove Employee Where Employeeid Eq "12a154121".
- Multiple Row Deletion
Remove Employee Where Employee Status Eq "Terminated".



Selection Clause

- If/where <Expression>
- Print Course Name If Department Is....
- Print Student Name Where Gpa Eq 3.0 And Sex Is Male



Sorting Clause

By Column Name,.....

Example Print Course Name by Department

Or

By Department, Course Name, Section,

Print Student Name, Letter Grade If College Is 'Business'.



Statistical Functions

Function	Example
Total	Print total annual budget
Tally	Print tally teacher number
Count	Print count course section id
Maximum	Print maximum GPA
Minimum	Print Minimum GPA
Average	Print Average GPA
Standard Deviation	Print SIGMA GPA

- Simple Arithmetic Operations Using Columns
- Print Class Length = Stop Time - Start Time



Report Formatting

Title 'Fall Semester Grades'/date
Column Spacing Is 5
Device Is Terminal
Page Is 132 Wide by 55 Long
Double Space
Suppress Blanks
Page Ejects on Value Change
Skip after Row Print
Skip Before Row Print
Skip after Control Break on <Column>
Skip Before Control Break on <Column>



5.3 Static vs Dynamic Comparison

Capability	DBMS Type	
	Static	Dynamic
Selection	Low to Medium	High
Sorting	Low to Medium	Medium to High
Navigation	Low to Medium	Medium to High
Retrieval	Low to Medium	Medium to High
Updating	Low to Medium	Medium to High
System Control	Low to None	Low to None



5.4 Query-update Examples:

List/title L(5)customer-id, L(15)customer-name, L(40)contract-id, R(50)total-order-cost,
Ordered by Customer-id, Contract-id Where Contract-id Exists:

Assign Sales-person-id Eq 123-45-6789 Where Contract-id Eq 76498:

Remove Contract Where Contract-id Eq 76498:



Query-update Examples (cont.)

Find All Rows in Personnel-file with Education EQ
Lawyer or Engineer
And Age from 40 thru 60
And Sort by Name
And Display Name, Profession, Age, Salary.

Result: 11 Rows Found

Name	Education	Age	Salary

Brennen	Lawyer	45	35000
Davis	Engineer	55	28000
Edwards	Lawyer	51	33000
Jackson	Lawyer	40	30000
Murray	Engineer	44	22000



Add Date Returned Eq 10/20/1981 Where Card Number Is Equal to 262728:

Columns	Data Before Command	Data After Command
Borrower	A. Ferguson Billfinger	A. Ferguson Bullfinger
Card Number	262728	262728
Date Borrowed	10/06/1981	10/06/1981
Date Due	10/20/1981	10/20/1981
Date Returned	-Null-	10/20/1981
Outstanding Fine	-Null-	-Null-



5.5 Query-update Summary

- Another Proprietary Natural Language
- Designed for the Non-technical User
- 1/10 Human Resources Required for
- Design, Implementation, and Maintenance
- Great Prototyping Tool for HLI, POL, or Report Writer



6.0 Report Writers

- Introduction
- Capabilities
- Example
- Static/dynamic Comparison
- Summary



6.1 Report Writer Introduction

- A Special DBMS Natural Language
- Principle Role
 - ◆ Good for Moderate to Complex Reports With-
 - ◆ Row and Column Titles Sorting
 - ◆ Various Levels of Break Totals
 - ◆ Table-look-up
- Great Prototyping for HLI Report Writing.
 - ◆ Human Resources for Design,
 - ◆ Implementation and Maintenance
 - ◆ Is 1/10 of HLI

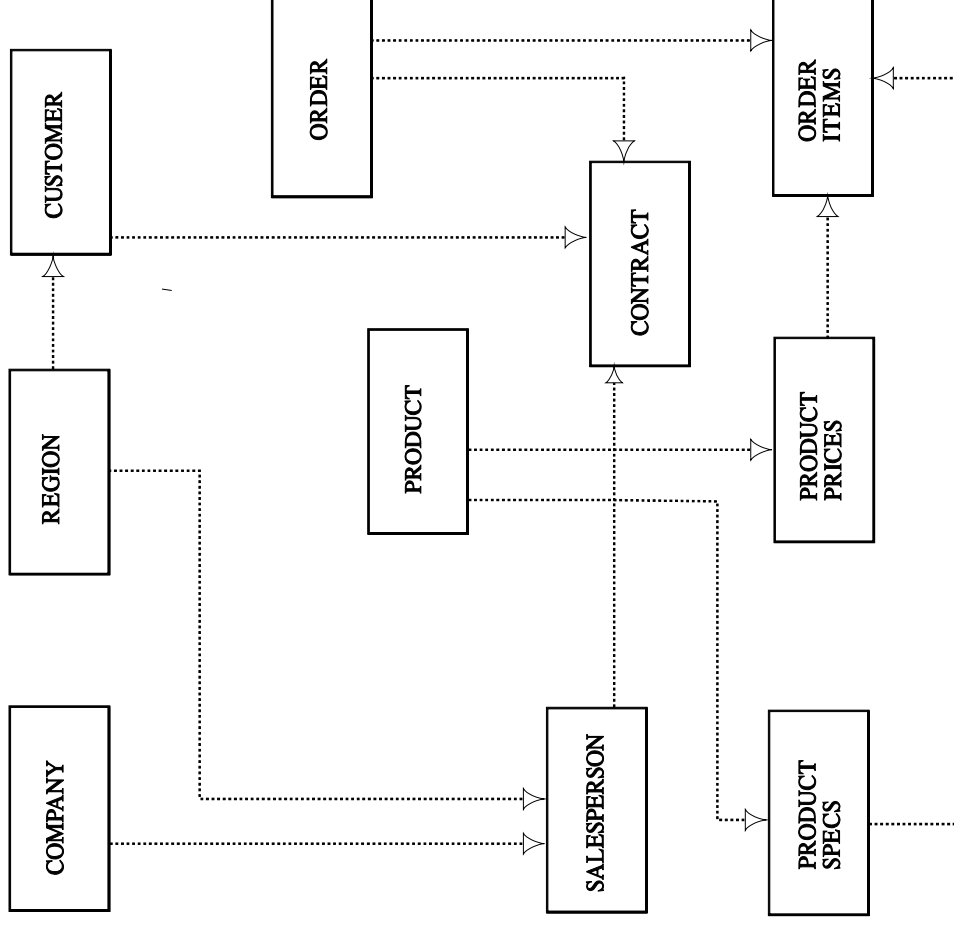


6.2 Report Writer Capabilities

- Precise Data Placement
- Many Levels of Subtotals
- Page Size, Width, Etc.
- Table Lookups
- Complex Computations



6.3 Report Writer Example



Report Writer Example

For Report Sales,
Physical Page Is 55 by 55:
Declare Sum1 = Rcount of Order-id:
Declare Sum2 = Rcount of Contract-id:
Select Row If Order Occurs:
Order by Name of Customer-id,
Contract-id, Order-id:
For Customer, Skip to New Page,
Skip 2 Lines,
At End, Print (15)\$total Number of
Contracts\$, R(45)sum2:
For Contract, Skip 2 Lines, Compute Sum2,
Print (15)\$contract-id\$,
(30)\$date-signed\$, (45)\$maximum-cost \$,
At End, Skip 1 Line, Print (15)\$total Orders\$,
R(28)sum1, (40)\$total Value\$,R(55)total-order-cost.
For Order, at End, Compute Sum1
End Report:
Generate Activity Where Customer-id Exists:



6.4 Report Writer Comparison

Capability	DBMS Type	
	Static	Dynamic
Selection	Medium to High	Medium to Low
Sorting	Medium to High	Medium to Low
Navigation	Implicit	Medium to Low
Retrieval	Medium to High	Medium to Low
Update	None	None
System Control	None	None



6.5 Report Writer Summary

- A Special DBMS Natural Language
- Principle Role
 - ◆ Good for Moderate to Complex Reports With-
 - Row and Column Titles Sorting
 - Various Levels of Break Totals
 - Table-look-up
- Great Prototyping for HLI Report Writing.
- Human Resources for Design, Implementation and Maintenance, Is 1/10 of HLI



7.0 Application Interfaces

- Introduction
- Example
- Static/dynamic Comparison
- Summary



7.1 Introduction

Definition: Interface with Outside Application

- Text Processing
- KWIC & KWOC
- Thesari
- List Selection & Processing
- Special Output Processing
- Special Output Processing
- Plots/graphics
- Statistics
- Financial
- Management Science
- Engineering Science



7.2 Example

```
0 Report Name Is Base Year Profit Summary
1 Columns 1-4, Yr1, Pct1
2 "
3 Center 1979 Base Year Profit Summary
4 "
5 Print Column Titles
6 Underline-
7 Print Sales Volume, Sales Price
8 Underline-
9 Print Total Sales = Sales Volume * Sales Price
10 Space 1
11 Print Total Variable Expenses =
12 Sum (Material Cost, R * D Expenses,
13 Sales Expenses, Manufacturing Exp.)
14 Print Total Fixed Expenses =
15 Sum (Space Rental, Leases, Admin Exp.)
15 Underline
16 Print Net Profit
17 Underline=
18 End
```

Generate Base Year Profit Summary Where Year Is 1976 to 1978 for Product = "Answer Machine"



7.3 Static vs Dynamic Comparison

Capability	DBMS Type	
	Static	Dynamic
Selection	As required	As required
Sorting	As required	As required
Navigation	Automatic	Automatic
Retrieval	As required	As required
Updating	As required	As required
System Control	None	None



7.4 Application Interface Summary

- Specialized Language Interfaces to
- Special User Oriented Packages
- Growing Part of Database Industry
 - ◆ Text Processing
 - ◆ Engineering
 - ◆ Financial Modeling
 - ◆ Graphics



8.0 DBMS Vendor Independent Development Environments

- DBMS Interface Drivers
- Integrated Development Environments
- Summary



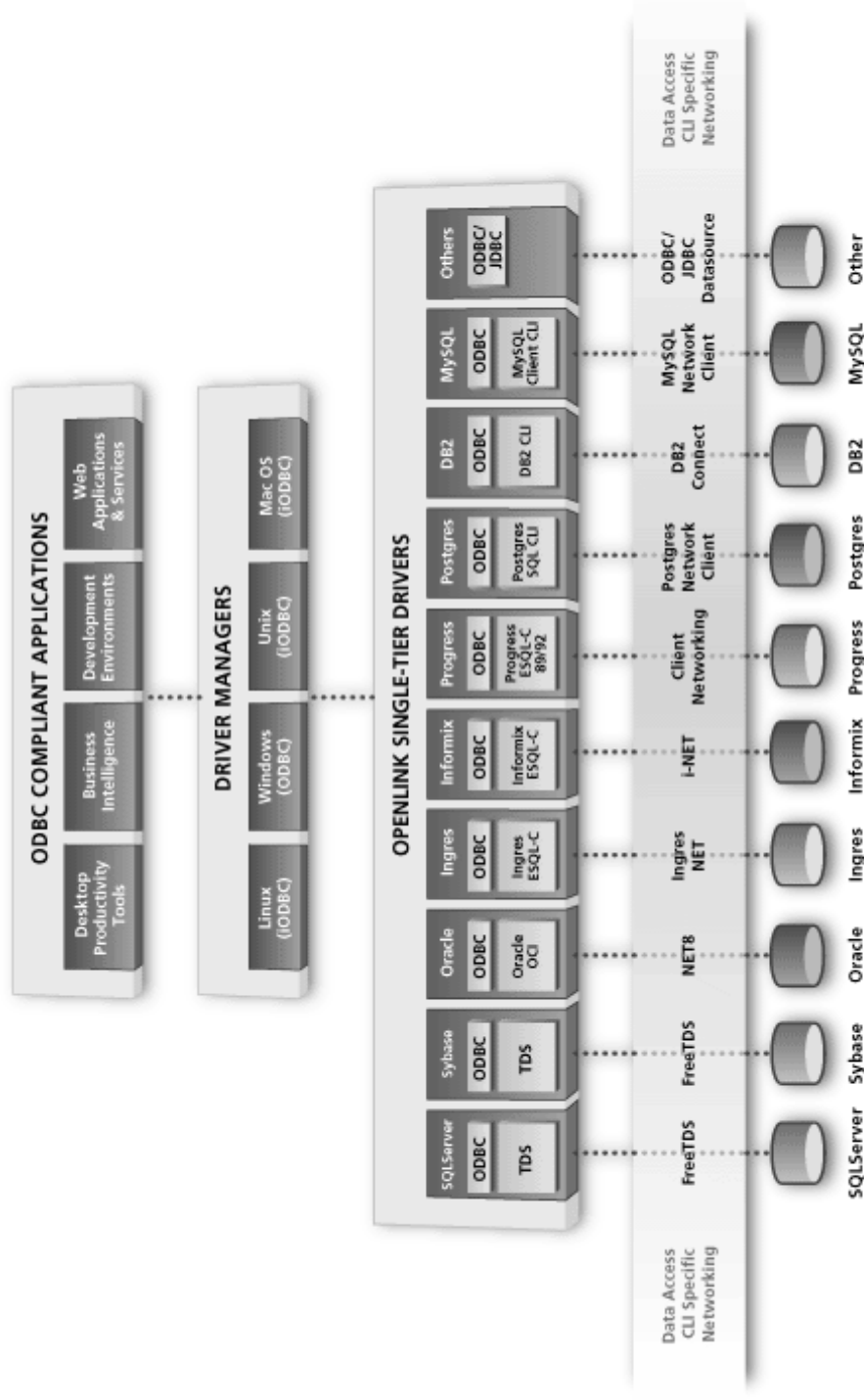
8.1 DBMS Interface Drivers

ODBC

- ODBC (Open DataBase Connectivity) is a Windows "strategic interface" for accessing data from a variety of Relational Database Management Systems (RDBMS) across a variety of networks and platforms.
- The ODBC standard was developed and is maintained by Microsoft, which publishes an ODBC Software Development Kit (SDK), geared for use with its Visual C++ product. ODBC support is another way to provide an extensible platform for you to create applications

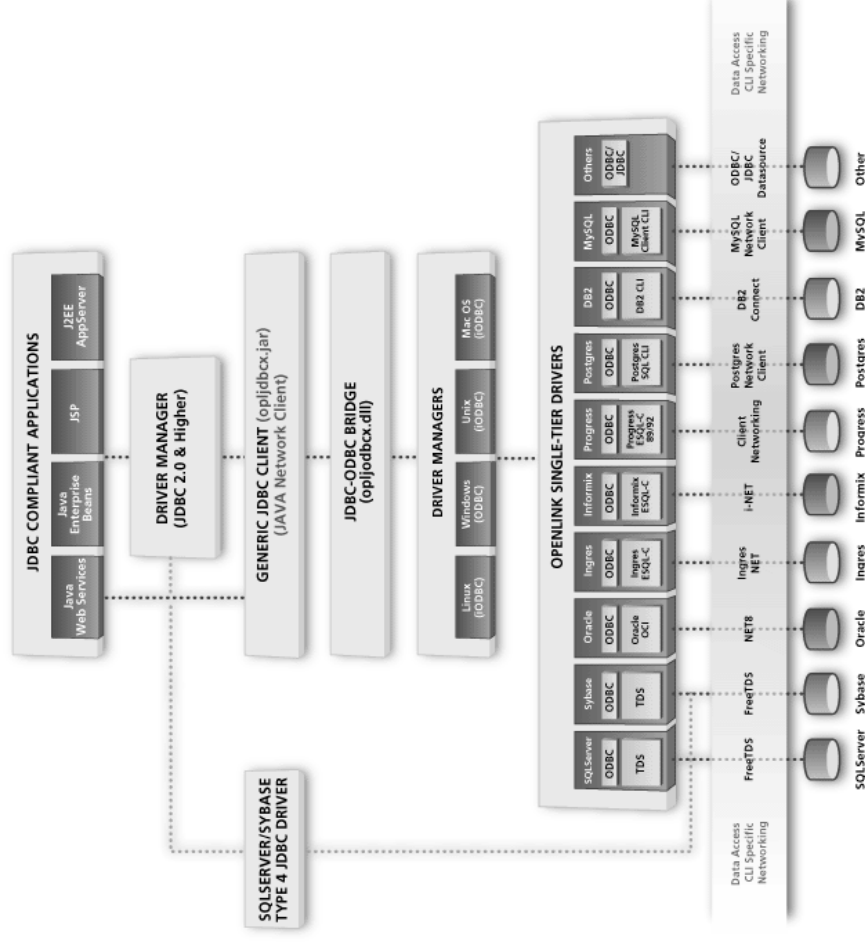


ODBC (cont)



JDBC

- Java Database Connectivity, standardized database interface for Java. This technology allows to write an application once, and use it with any SQL database that has a JDBC-driver.



8.2 Integrated Development Environments

- Application Development
 - ◆ Application Structure Development
 - ◆ Screen Development
 - ◆ Application Code Development
 - ◆ ODBC and JDBC Access to DBMSs
- Report Writers
 - ◆ Crystal Reports
 - ◆ Information Builders FOCUS



8.2.1 Clarion Demonstration

- Starting the System
- Creating a single table Dictionary
- Creating the single table application
- Adding a table to the Dictionary
- Building a relationship between two tables
- Modifying the Application Menu
- Using the Wizard to Generate for the new Table
- Building Referential Integrity between the two tables
- Using the Clarion Help System to Learn about Clarion



8.2.2 Crystal Reports Demonstration

- Building a Report through ODBC
- Executing a Report



8.3 Summary

- DBMS Independence
- Dependence on either ODBC or JDBC
- Additional contracts with other vendors
- Certainly not ANSI Standard Access



9.0 Interrogation Summary

- ◆ Host Language Interface
- ◆ Query-update Language
- ◆ Procedure Oriented Language (Pol)
- ◆ Report Writer
- ◆ Application Interfaces
- ◆ Interrogation Summary



9.1 Host Language Interface

- ◆ Database Interaction

Through Cobol, Fortran, Pl/1, Etc.

- ◆ Principle Role

Data Processing Intensive Applications

Data Migration and Transformation To, from and Between Databases and DBMS's

Maximum Control of User Data Base Interaction



9.2 Procedure Oriented Language (Pol)

- ◆ Capabilities Similar to HLI
- ◆ Proprietary Natural Language
- ◆ Great for Prototyping HLI
- ◆ Human Resources (1/3 of HLI) for
 - ◆ Design,
 - ◆ Implementation,
 - ◆ And Maintenance
- ◆ Consumes Much More Computer Resources



9.3 Query-update Language

- ◆ Another Proprietary Natural Language
- ◆ Designed for the Non-technical User
- ◆ 1/10 Human Resources Required for
- ◆ Design, Implementation, and Maintenance
- ◆ Great Prototyping Tool for HLI, POL, or Report Writer



9.4 Report Writer

- ◆ A Special DBMS Natural Language
- ◆ Principle Role
 - ◆ Good for Moderate to Complex Reports With-
 - Row and Column Titles Sorting
 - Various Levels of Break Totals
 - Table-look-up
- ◆ Great Prototyping for HLI Report Writing.
- ◆ Human Resources for Design, Implementation and Maintenance, Is 1/10 of HLI



9.5 Application Interfaces

- ◆ Specialized Language Interfaces to
- ◆ Special User Oriented Packages
- ◆ Growing Part of Database Industry
 - ◆ Text Processing
 - ◆ Engineering
 - ◆ Financial Modeling
 - ◆ Graphics



9.6 Interrogation Summary

Language Selection Criteria	Interrogation Language Type			
	Host Language Interface	Procedure Oriented Language	Report Writers	Query Update Language
Task Development Effort	Hi	Medium	Medium	Low
Relative Work Units	100	10	10	1
Level of user control over database interaction	Low	Medium	Medium	High
Range of portability from one DBMS to another of same Data Model	Medium to High	Low	Low	Low



Interrogation

- Pick the Right Interrogation Language One for the Job
- Static -- Natural Languages Could Be Better
- Dynamic -- HLI Could Be Better

