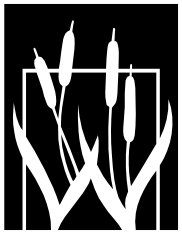


Whitemarsh
Information Systems Corporation

Database Management Systems: Understanding and Applying Database Technology

Chapters 3

Logical Database

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

Logical Database

- Logical Database Introduction
- DBMS Data Models
- Static Data Models
- Dynamic Data Models
- SQL Data Model
- Schemas & Subschemas
- Logical Database Summary

1.0 Logical Database Introduction

- Context
- Definitions
- Analytic Model
- DBMS Model
- Data Models

1.1 Logical Database Context

Managing Database Viewpoint	Interrelationships Among Database Viewpoints			
	Logical	**Physical**	**Interrogation**	**System Control**
Technology (this course)	Data model	Database creation and maintenance	Data selection and reporting	Audit trails, protection and evolution etc.
Staffing (Other Whitemarsh materials)	Database specialist	DBMS specialist	Interrogation specialist	System control specialist
Project (Other Whitemarsh materials)	Conceptual specification phase Implementation phase Production and administration phase			
DBMS (Oracle manuals)	Schema and sub-schema	Access methods, data loading, update, and maintenance	Query, host language, report writers	Utilities, languages and techniques

1.2 Definition

- Logical Database
  - A Combination of Graphics and Analytics That Describes Business Information Set down By:
    - Top and Middle Management as High Level Business Goals, Objectives and Policies to Set Direction and Scope
    - Then Translated by Database Team Members into Detailed Business Policies to Enable These High Level Activities to Be Set down into Business Practice
  - In Short, the Logical Database Is A Business Policy Based Analytical Model That Is Prerequisite to DBMS Implementation

1.3 Logical Database -- Analytic Model

- High Level Definitions (Specified by Analysts)
- Table: Represents Independent, Stand Alone Function Within an Organization That Serves or Fulfills a Useful Role
  - Example:
    - Contract
    - Salesman
    - Project
- Column: Attribute or Partial Description of a Table
  - Example:
    - Sign-date

Relationship

- An Expression of Important Interaction Within/between/among Tables
  - Example:
    - Enrollment of Students in a Course Or Gpa = (Sum Grades)/(Nbr Courses)

Operation

- Distinct and Allowable Action to Retrieve/update Tables/columns
  Within a Database Utilizing Specific Relationships
- Example:
  - Add Employee
  - Compute Commission

Analytical Model Example

- Table: Contract

- Columns:
  - Contract-number
  - Type Is Numeric
  - Valid Value Range 00001-99999

- Relationship
  - Contract-salesperson
  - Owner Is Contract
  - Member Is Salesperson
  - Constraint Is Contract-salesperson-id Eq Salesperson-id

- Operation
  - Add Purchase Order
  - Locate Contract, If Error Goto
  - Store Purchase Order
  - Insert into Open-order-set
  - Insert into Customer-business-set

Personnel Involved	Logical Database	
	Analytical Model	**DBMS Model**
Top and Middle	High Level, Business Policy Expressed by Graphics and Analytics	Not sufficiently detailed for DBMS Representation
Database Specialist	Detailed business policy expressed by data and relationship definitions	DBMS logical data models around which a DBMS is designed.

Now, We'll Discuss the DBMS's Logical Data Model

1.4 Logical Database -- DBMS Model

- Table Organizations
- Relationships:
  - Static (Bound at Update)
  - Dynamic (Bound at Retrieval)
- Operations
  - Record: (Store, Modify, Delete, Find, Get)
  - Relationship: (Connect, Disconnect)
  - (Join, Project, Divide)

1.4.1 Table Organizations

- Tables
  - Collection of simple columns (single valued)
  - Collection of complex columns (multi-valued, etc.)
  - Collection of simple and complex columns
- Column Data Types
  - Fixed
  - Float
  - Integer
  - Character
- Column Integrity Rules

- Simple Columns
  - Name
  - Social Security Number
  - Birth Date
- Complex Columns
  - Single-valued Columns, and
  - Multi-valued Items
  - Multi-dimension Columns
  - Groups
  - Repeating Groups

Complex Columns

- Multi-valued Columns

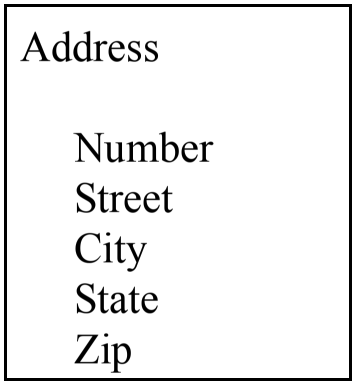
TELEPHONE NUMBERS			
3012495300	3012491142	7176485913	...

- Multi-dimension Columns

Sales Group Element (12 X 10 X 1)			
Dimension	**12**	**10**	**1**
Sub-Element	**Month**	**District**	**Value**
Data value	JAN	1	$2500
	JAN	2	$1800
	JAN	3	$1700
	JAN	4	$2900
	JAN	5	$1100
	JAN	6	$2300
	JAN	7	$2400
	JAN	8	$1600
	JAN	9	$1800

Complex Columns (Cont.)

Group Column



Nested Repeating Group Column

Employee - ID
- .
- .
- .

Dependents (RG)

- Name
- Sex
- Birth-date
- Hobbies (RG in Dependents)
  - Hobby name
  - Annual cost

Column Data Types

- Standard
  - Character
  - Fixed
  - Float
  - Complex
- Specialized
  - Time
  - Date

Column Integrity Clauses

Columns are governed by integrity rules. These rules typically involve:

- Valid value ranges
- Specific valid values
- Invalid value ranges
- Specific invalid values
- Numbers of occurrences
- Duplicates allowed or disallowed
- Default values
- Conversion rules
- Null
- Encodes and decodes

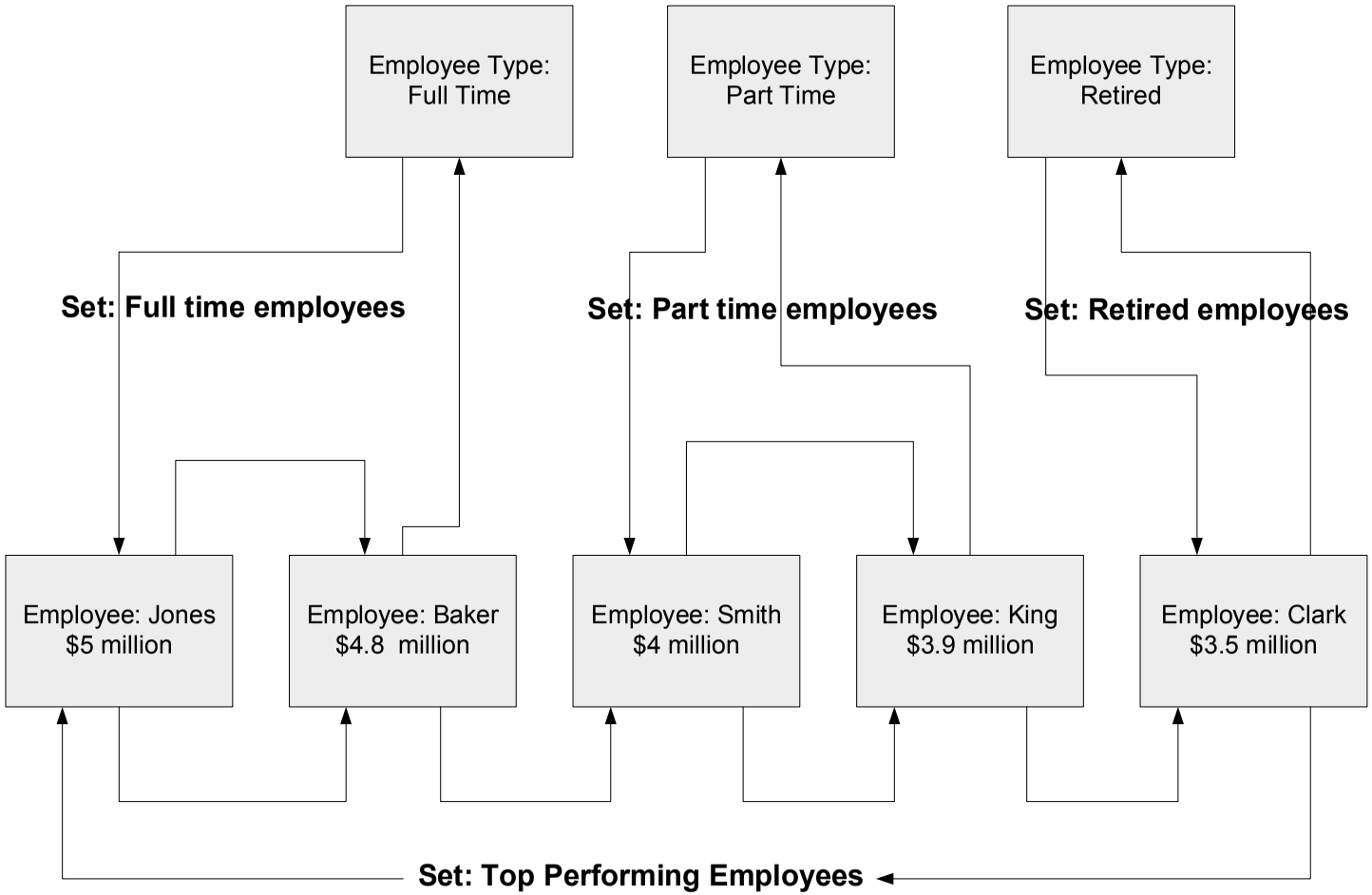
1.4.2 Relationships

- A Mechanism That Causes Records from the Same or Different Tables to Be Related.
  - Static Relationship
    - One in Which Participants Are Bound into the Relationship During Update or Load.
  - Dynamic Relationship
    - One in Which Participants Are Bound into the Relationship During Retrieval.

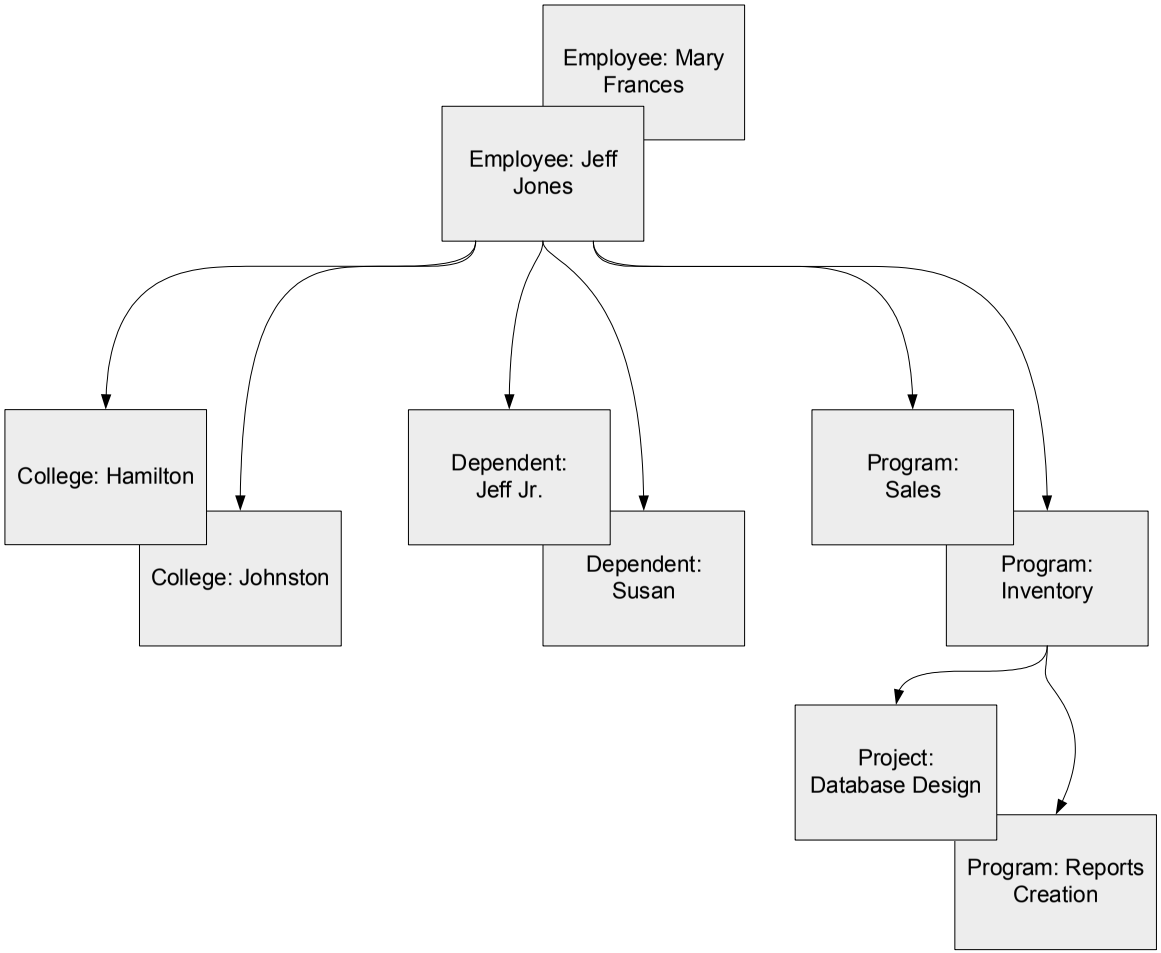
Static Relationship

- An Addressing Mechanism Controlled by the DBMS
- Stored by the DBMS in Records
- Arbitrarily Established Through Program Logic
- Records Bound into Relationship by the User During Load/update Operations

Static Relationship Example (Network)



Static Relationship Example (Hierarchy)



Dynamic Relationships

- Controlled by the User
- Stored by the User As Shared Column Values
- Based Completely on Data Values
- Records Bound into Relationship By the DBMS During Retrieval Operations

Dynamic Relationship Example

Employee				
Employee Nbr	**First Name**	**Last Name**	**Title**	**Gender**
01	Harry	Smith	Engineer	M
02	Sue	Jones	Supervisor	F
03	George	Jefferson	Engineer	M

Dependents				
Dependent Nbr	**First Name**	**Last Name**	**Gender**	**Employee Nbr**
01	Mary	Smith	M	01
02	John	Smith	F	01
03	Jack	Jones	M	02

1.4.3 Operations

- Definition
  - An Action That Is Allowed to Take Place on Records & Relationships Through Either Static or Dynamic Rules.

Row Operations	
Operation	**Description**
FIND	SELECT According to STORED Order, or SELECT and PUT into DML Specified Order
GET	Obtain Record From Find
ADD	Install a New Row Into Database
DELETE	Remove an Existing Row From Database
MODIFY	Change Some Column Values in Existing Row

Relationship Operations

Operation	Description
CONNECT	Add to a Named RELATIONSHIP in Specific Oder
DISCONNECT	Delete From RELATIONSHIP
GET OWNER	Obtains The Parent of the Row That is Current
GET MEMBER	Obtains the First Child of the Owner For the Named Relationship
GET NEXT	Obtains the Next Row Within The Named Relationship
INTERSECT	Find and Keep Only the Common
DIFFERENCE	Find and Keep Only the Not Common
JOIN	"Append" Relations to Each Other
DIVIDE	Subset
PRODUCT	Cross-Product
UNION	Merge and Drop Duplicates

Combination Operations

- Fetch = Find + Get
- Modify = Find + Get + Store
- Insert = Store + Connect

2.0 DBMS Data Model

- Fundamental Alternatives

Table Organization

- Simple Columns
- Complex Columns

Relationships

- Static: “members” known to “owners” at load or update type
- Dynamic: “members” and “owners” know of each other only at retrieval time
- Referential integrity and referential actions: a DBMS facility for both static and dynamic relationships

Operations

- Row actions (add, delete, or modify)
  Relationship actions on Owner modify and delete, and Member insert and modify

2.1 Fundamental Data Model Alternatives

Data Model Type	Data Definition Language Components	Data Manipulation Language Components
Static	Row Organization and Relationships	Operations
Dynamic	Row Organization	Relationships and Operations

Data Model Components	{ Row Organization } + { Inter-row Relationships } + { Operations }
Data Model Languages	{Data Definition Language } + { Data Manipulation Language }
Dynamic Data Model	{ DDL { RO} } + { DML { REL + OPS } }
Static Data Model	{ DDL {RO + REL} } + { DML { OPS } }

WHERE:

DDL: Data Definition Model
DML: Data Manipulation Language
RO: Row Organization
REL: Relationships
OPS: Operations

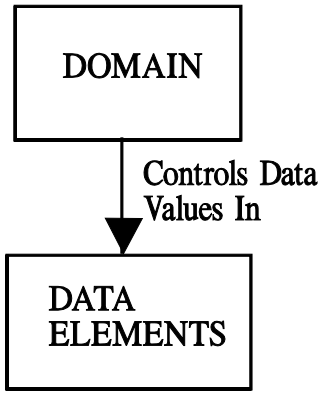
2.2 Table Organizations

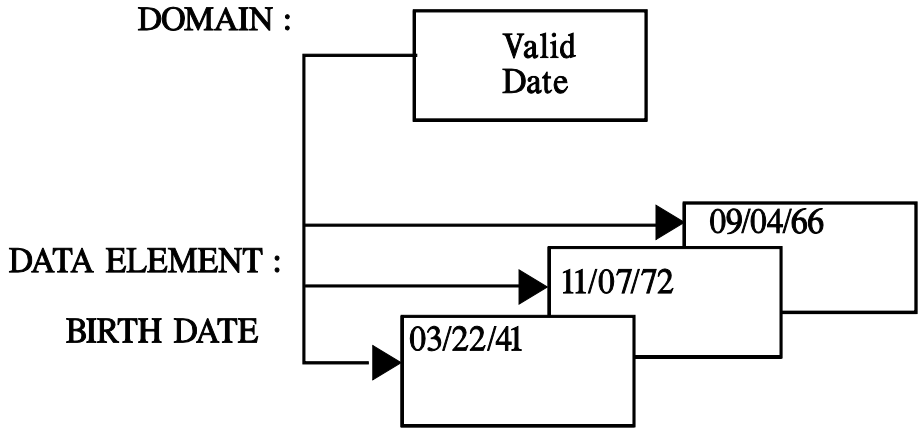
Column Types	Definition
Single Value	Each component represents a single value such as Birthdate with the value 11/11/1987
Multi-value	Eeach component represents multiple values such as Nicknames with values “Buddy, Guy, Mac”
Groups	Each component has subcomponents to represent a single-set of values such as Address with Street-1, Street-2, City, State, Zip
Repeating Groups	Each component has subcomponents to represent multi-sets of values such as Dependents that contains subcomponents, Dependent Name, Dependent Birth date, Dependent SSN.
Nested Repeating Groups	Each component has subcomponents to represent multi-sets of values such as Hobby (Hobby Name, Hobby Description, Hobby Annual Cost) within each multi-set component, Dependent, that also contains single value components such as Dependent Name, Dependent Birth Date, Dependent SSN that is within the Employee table that also contains single-valued components, Employee SSN, Employee birth-date, etc.

2.3 Relationship Types

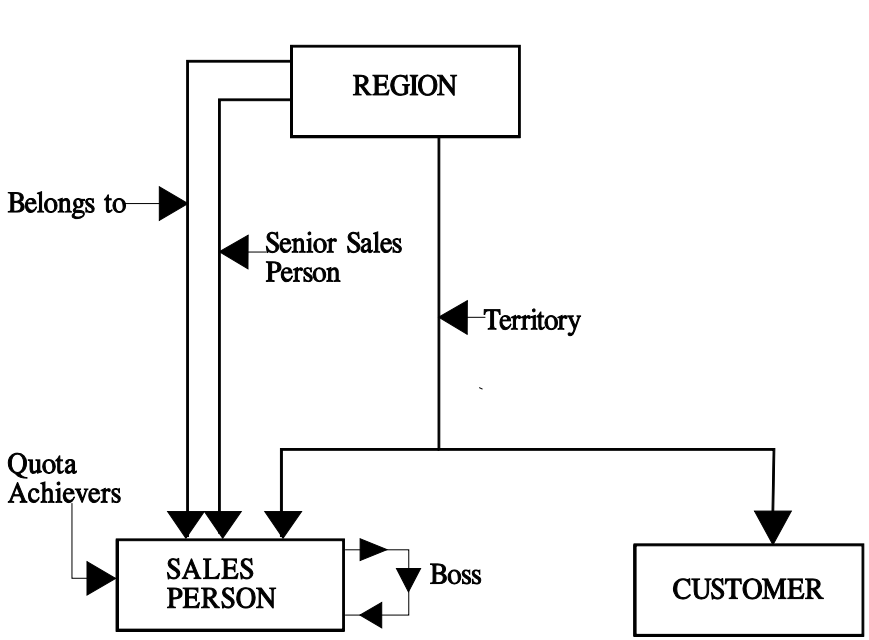
Name	Example
One-to-many	Employee to dependents
Owner-multiple-member	Territory contains salesmen and customers
Singular-one-member	Top performing employees
Singular-multiple-member	Top performing current, former, part-time, and retired employees
Recursive	Organization contains organization
Many-to-many	Automobiles and owners
One-to-one	Table and its primary key
Inferential	House with location, and buyer & desired location

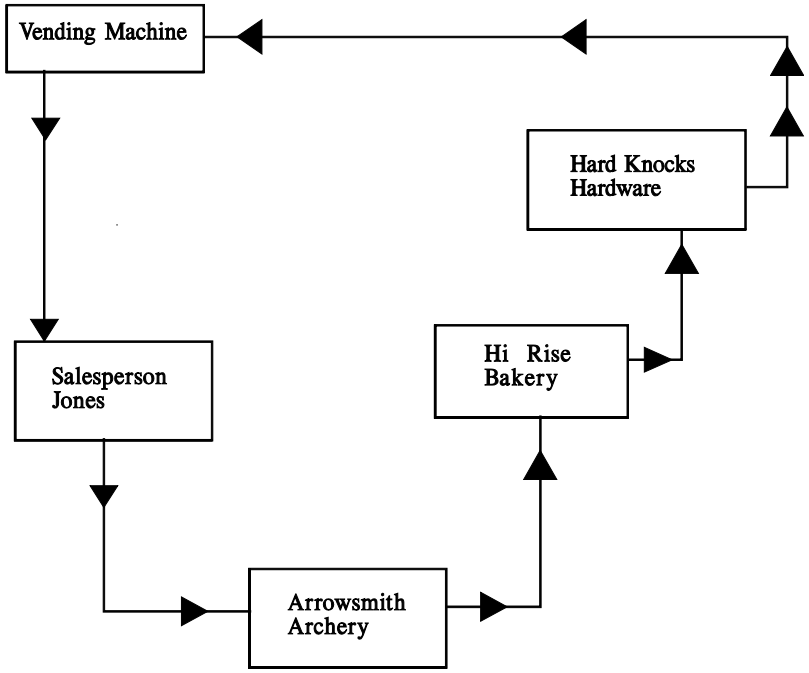
One-to-Many





Owner-Multiple-Member





Relationship Types :

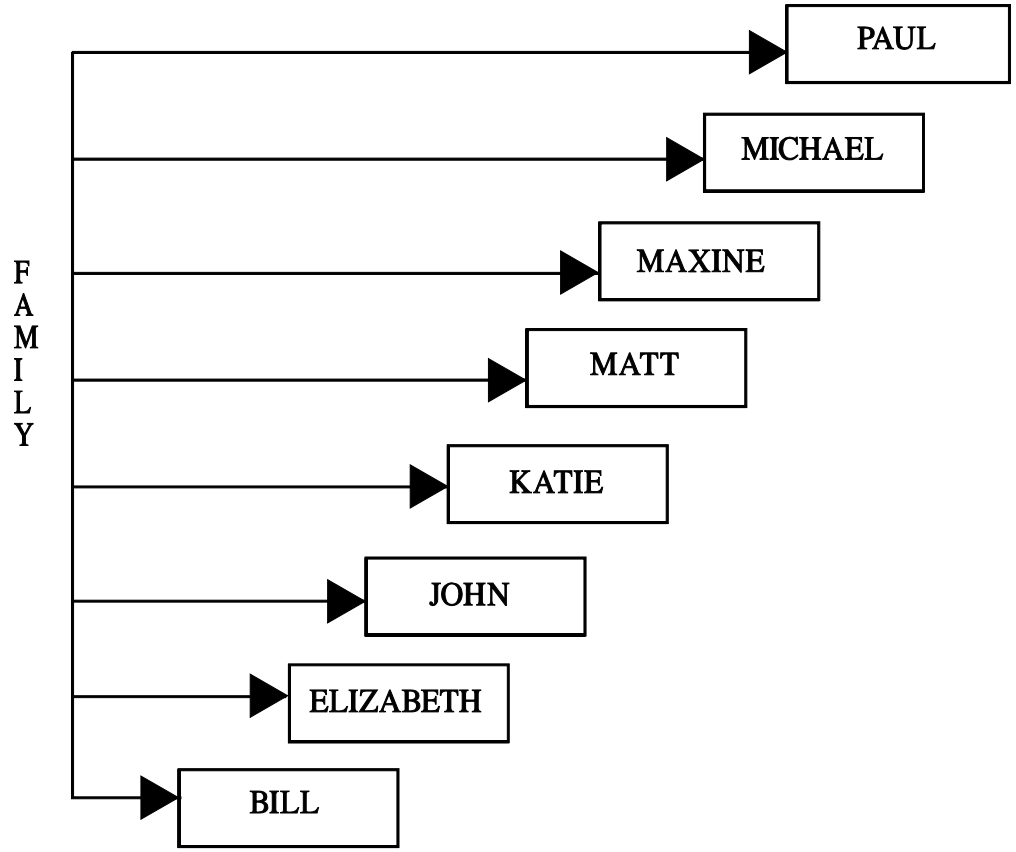
Singular - Quota Achievers
Single - Member - Senior Sales Person
Multiple - Member - Territory
Recursive - Boss

Set : Territory

Owner : Machine Type

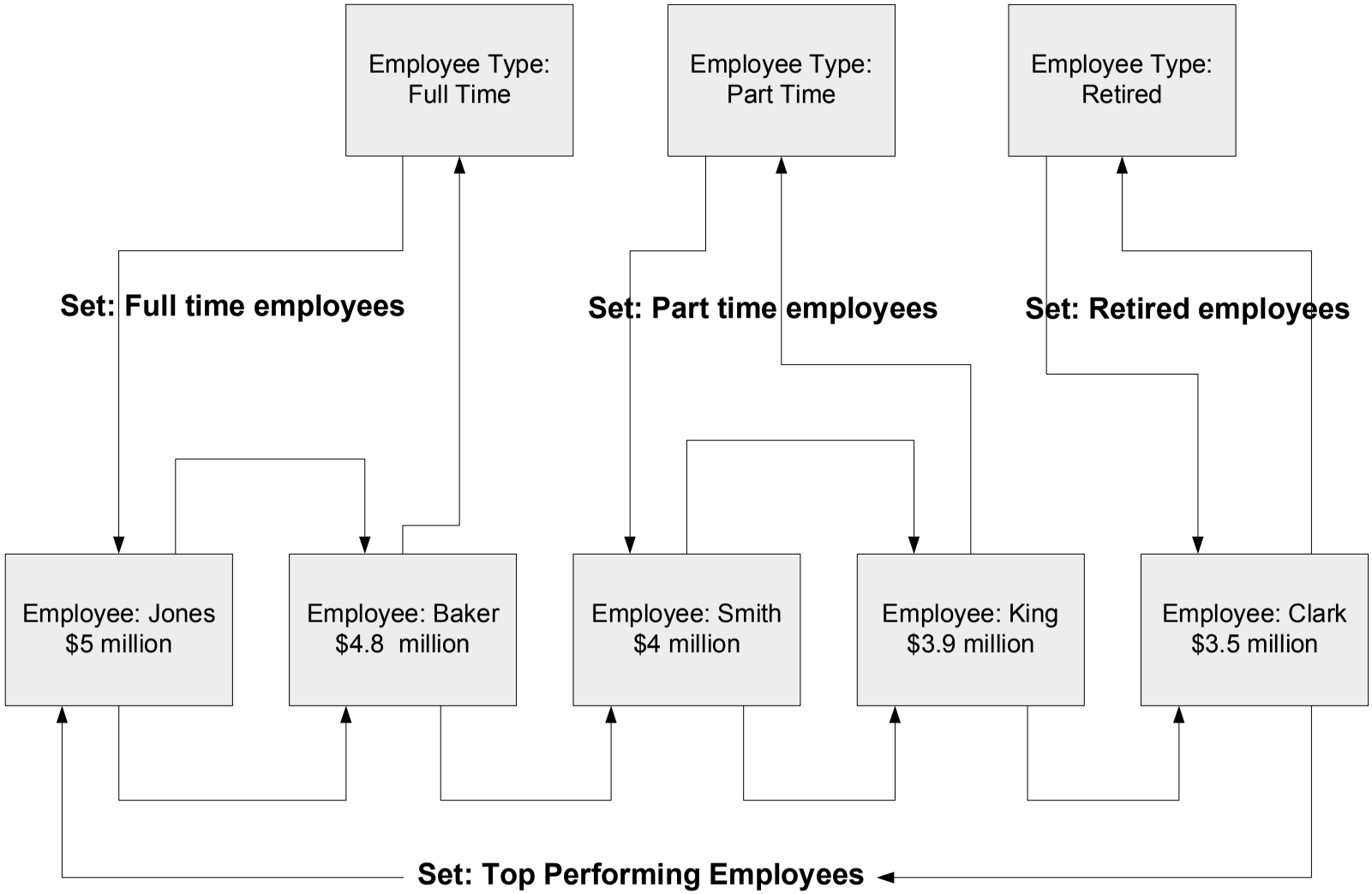
Member : Salesperson, Store

Singular-One-Member

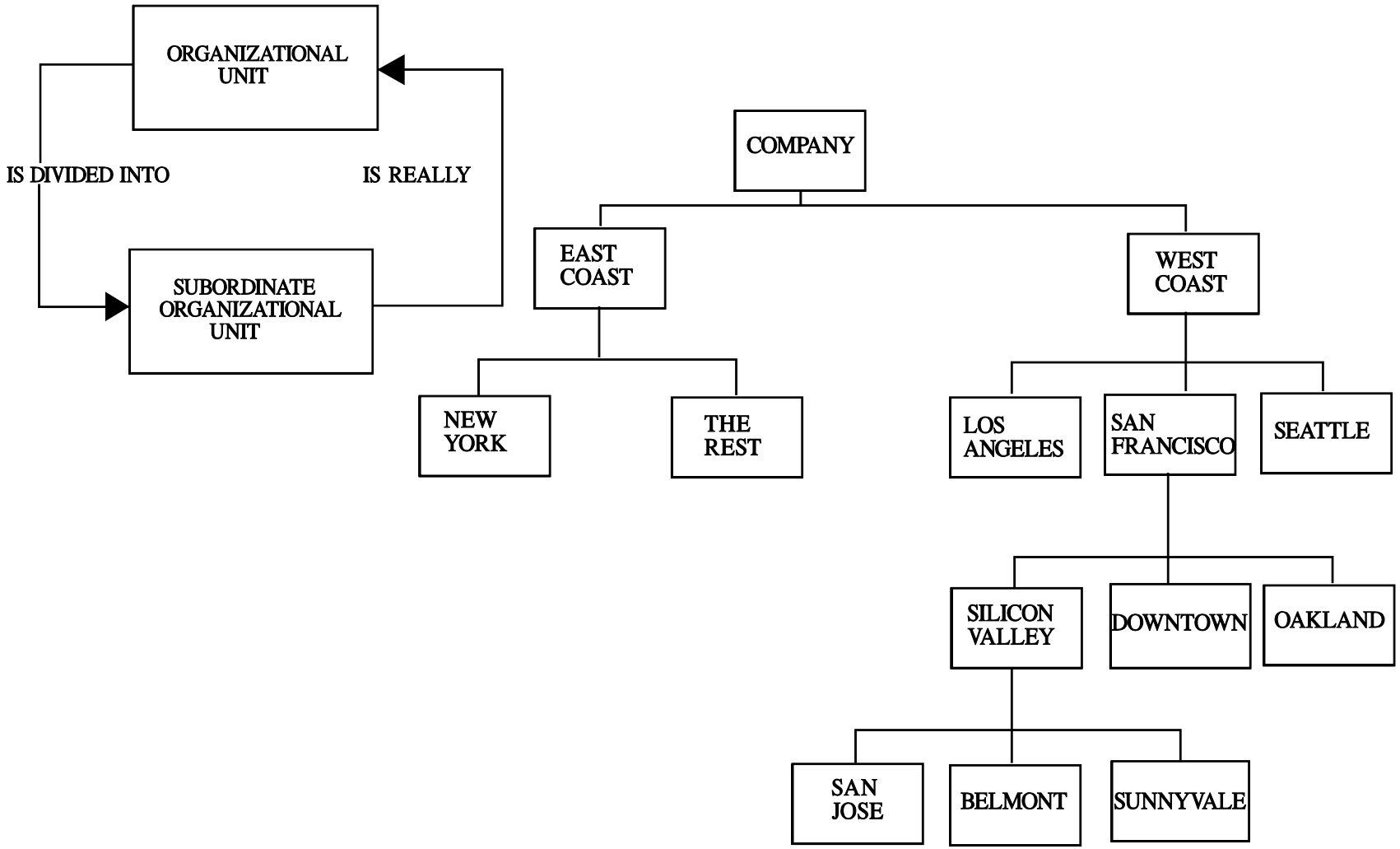


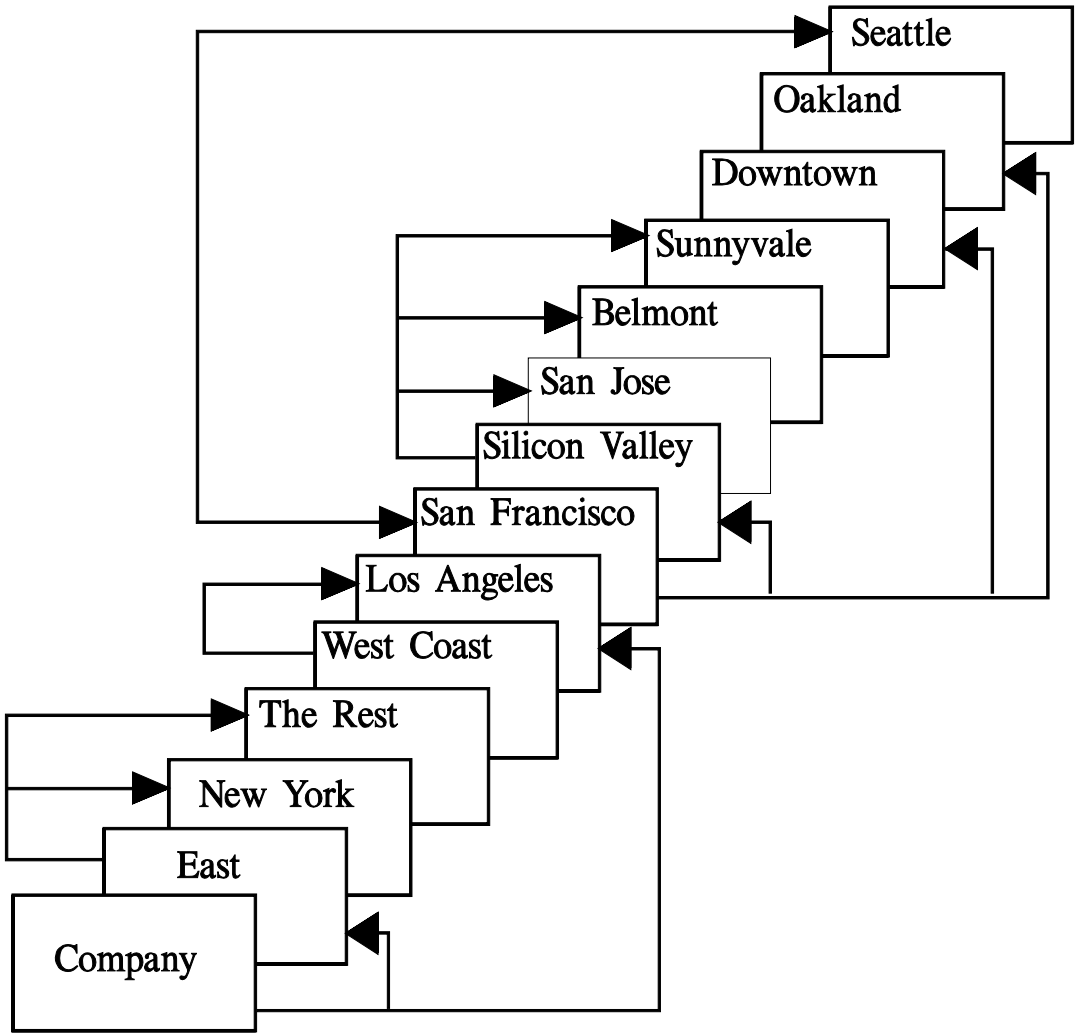
Set Type : Singular, single member

Singular-Multiple-Member



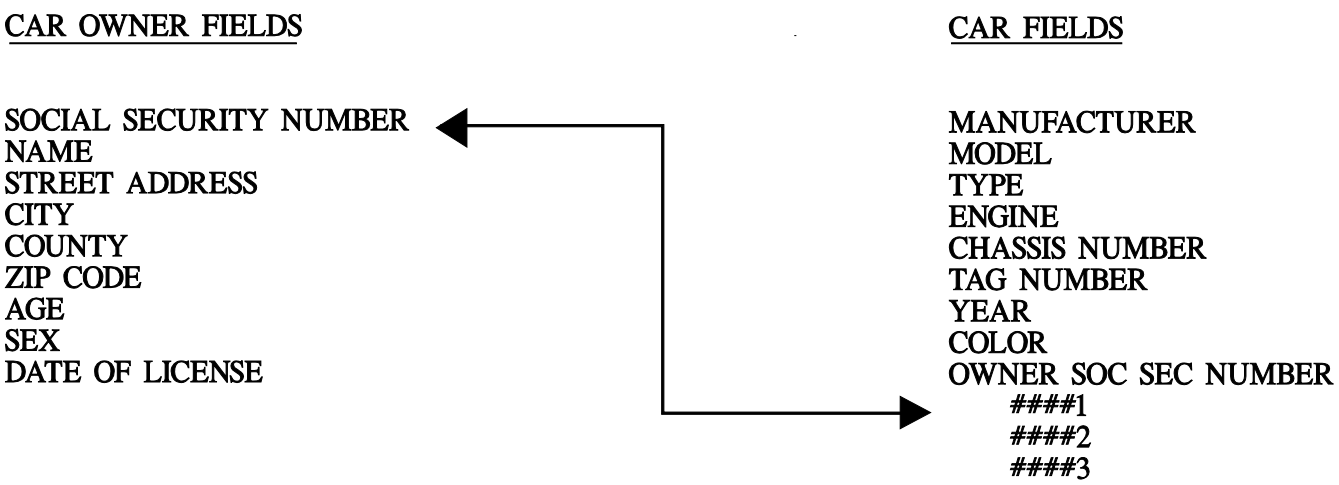
Recursive



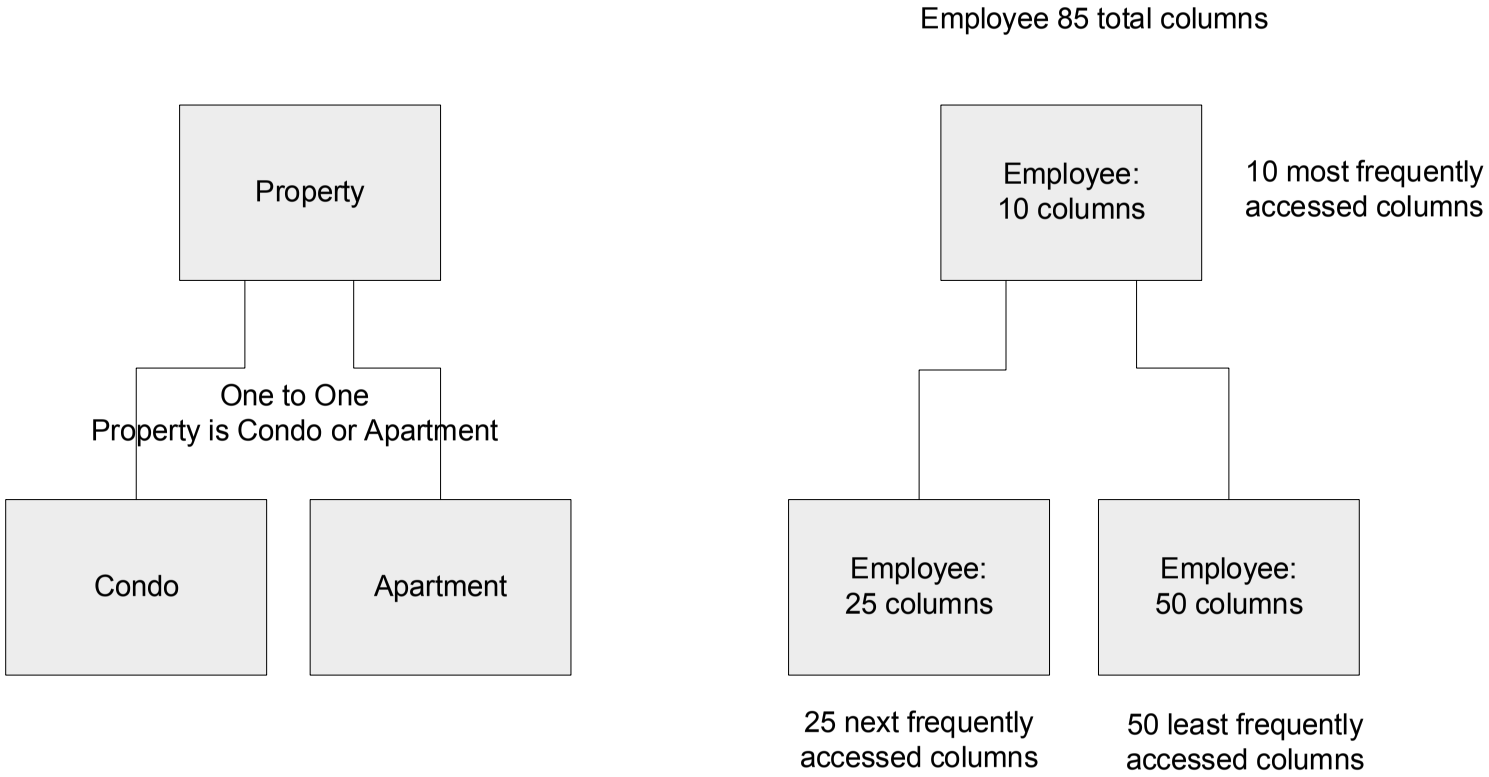


Many-to-Many

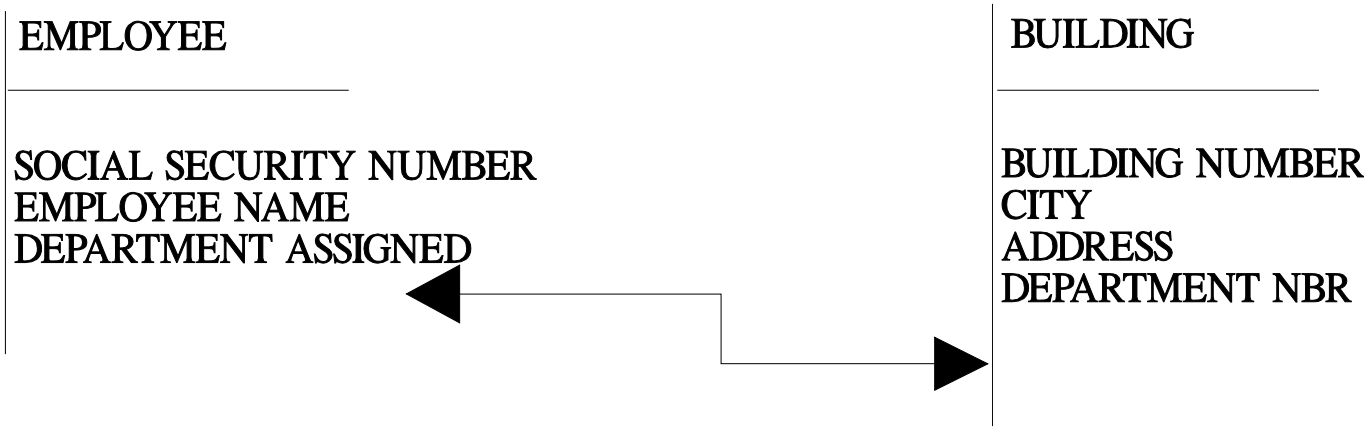
Student	Course		
	Math 203	English 701	Biology 107
Jones		Yes	Yes
Smith	Yes	Yes	Yes
Jackson	Yes		



One-to-One



Inferential



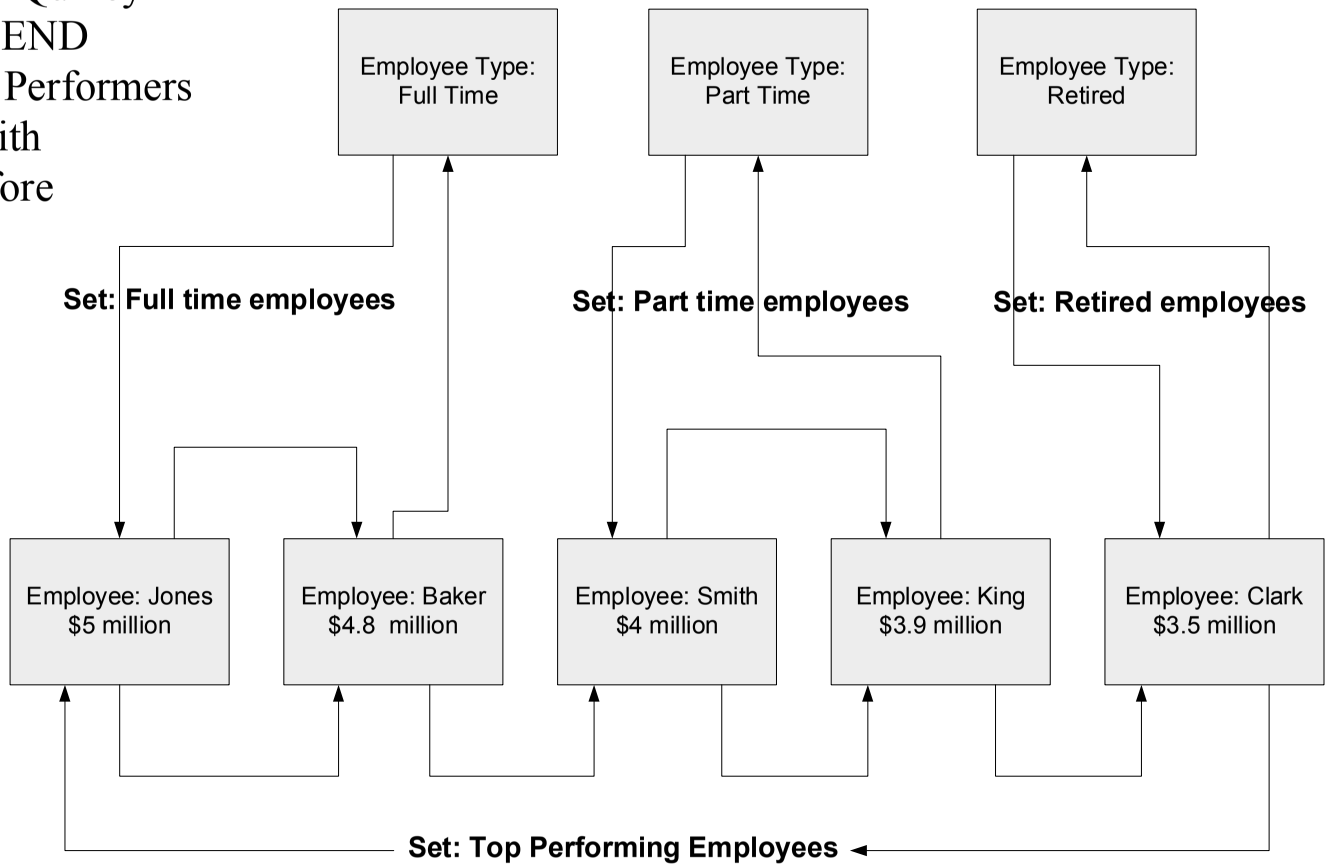
2.4 Operations

Row Operations		
Operation	**Static**	**Dynamic**
Find	SELECT According to STORED Order	SELECT and PUT into DML Specified Order
Get	Obtain Row From Find	Ditto
Add	Install a New Row Into Database	Ditto
Delete	Remove an Existing Row From Database	Ditto
Modify	Change Some Data Column Values in Existing Row	Ditto

Relationship Operations		
Operation	**Static**	**Dynamic**
Connect	Add to a Named RELATIONSHIP in Specific Order	N/A
Disconnect	Delete From RELATIONSHIP	N/A
Get Owner	Obtains The Parent of the Row That is Current	N/A
Get Member	Obtains the First Child of the Owner For the Named Relationship	N/A
Get next	Obtains the Next Row Within The Named Relationship	N/A
Intersect	N/A	Find and Keep Only the Common
Difference	N/A	Find and Keep Only the Not Common
Join	N/A	"Append" Relations to Each Other
Divide	N/A	Subset
Product	N/A	Cross-Product
Union	N/A	Merge and Drop Duplicates

2.4.1 Static Relationship Operations

1. Determine that Quincy had sales of $4.2 million.
2. Get Employee Type: Full Time
3. Insert Record for Quincy
4. Insert Quincy at END
5. Access Set: Top Performers
6. Get Next till Smith
7. Do an Insert Before



2.4.2 Dynamic Relationship Operations

OPERATION	DYNAMIC
INTERSECT	Find and Keep Only the Common
DIFFERENCE	Find and Keep Only the Not Common
JOIN	"Append" Relations to Each Other
DIVIDE	Subset
PRODUCT	Cross-Product
UNION	Merge and Drop Duplicates

2.5 DBMS Data Model Types

- Combinations of Record Organization, plus Relationships, plus Operations

- Static Relationship Data Models
  - Network
  - Hierarchical

- Dynamic Relationship Data Models
  - Independent Logical File
  - Relational

- SQL Data Model

3.0 Static Data Models

- Common Characteristics
- Network
- Hierarchy
- Summary

3.1 Common Characteristics

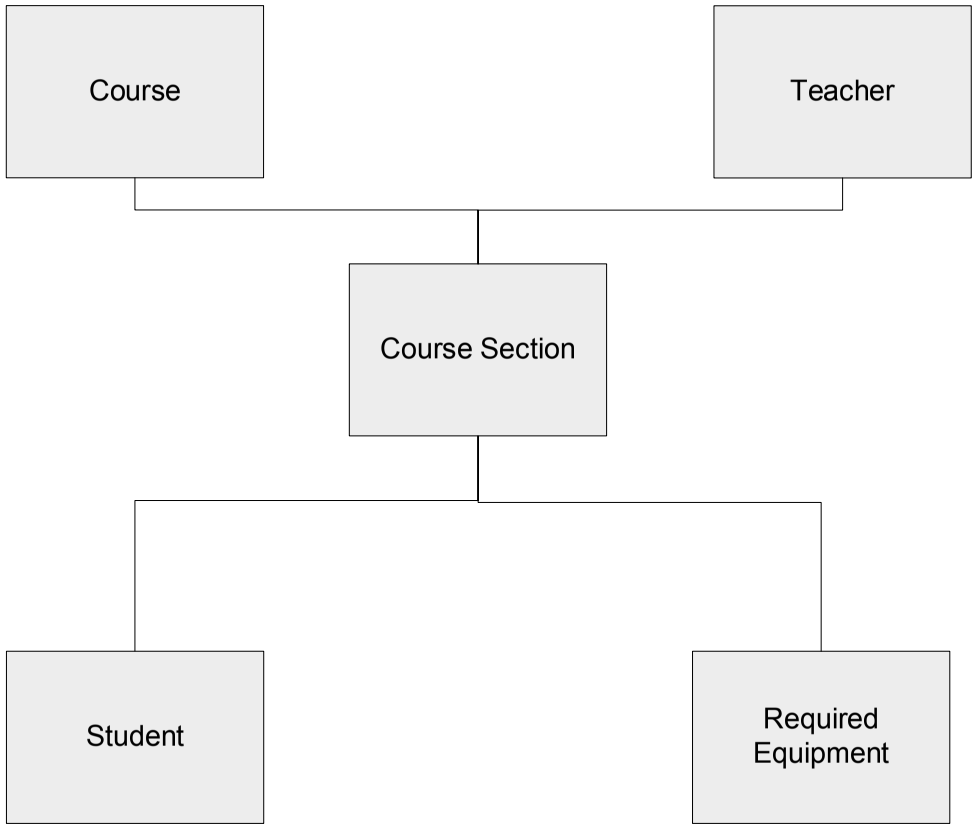
- Relationships via DBMS Generated Pointers
- Loading via Structure Dictates
- Relationships Bound at Load/update Time

3.2 Network Data Model

- ANSI/CODASYL
  - IDS II
  - DMS 1100
  - IDMS
  - VAX/DBMS
  - Prime DBMS
  - Data General DBMS
  - NCR DBMS
  - Total/TIS/SUPRA
  - Burroughs DMS-2

3.2.1 Network Definition

- A Collection of Records Is a Network When Records Are Relatable to Multiple Owners and Members



3.2.2 Network Table Organization

Models		Column Structure	Column Example
Codasyl	ANSI		
Yes	Yes	Primary Key	SSN
Yes	Yes	Non-repeated fields	Name, Address, Sex
Yes	Yes	Vectors	Nick names
Yes	Yes	Matrix	Monthly Division-sale
Yes	No	Repeating Group	Job(Title, Start Date, Stop Date)
Yes	No	Nested Repeating Group	Family(Kids(Hobbies))

3.2.3 Network Relationships

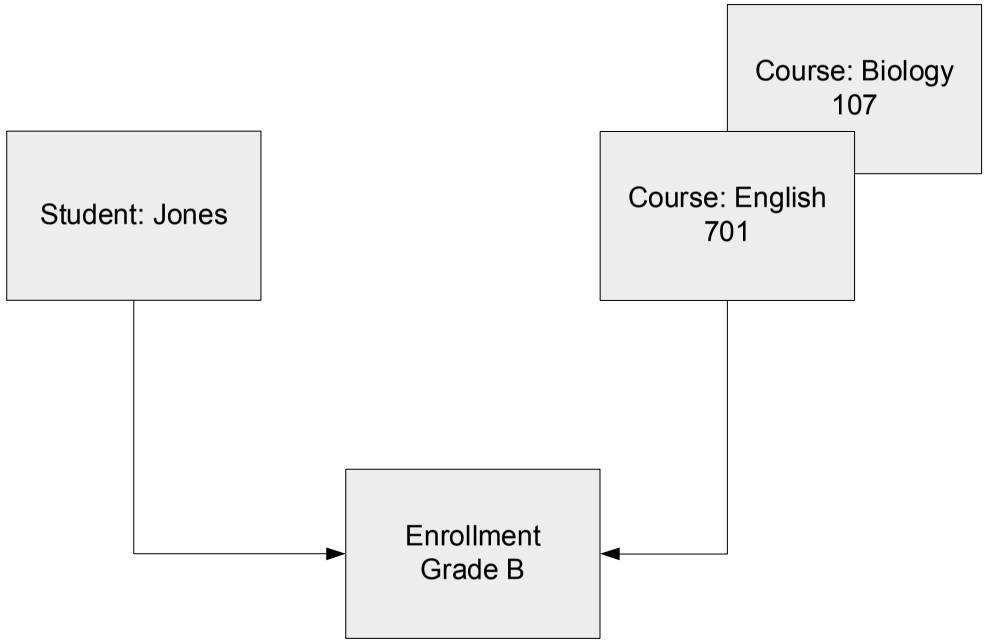
- Direct Network (Not Possible in ANSI)

  N : M

Student	Course		
	Math 203	**English 701**	**Biology 107**
Jones		Yes	Yes
Smith	Yes	Yes	Yes
Jackson	Yes		

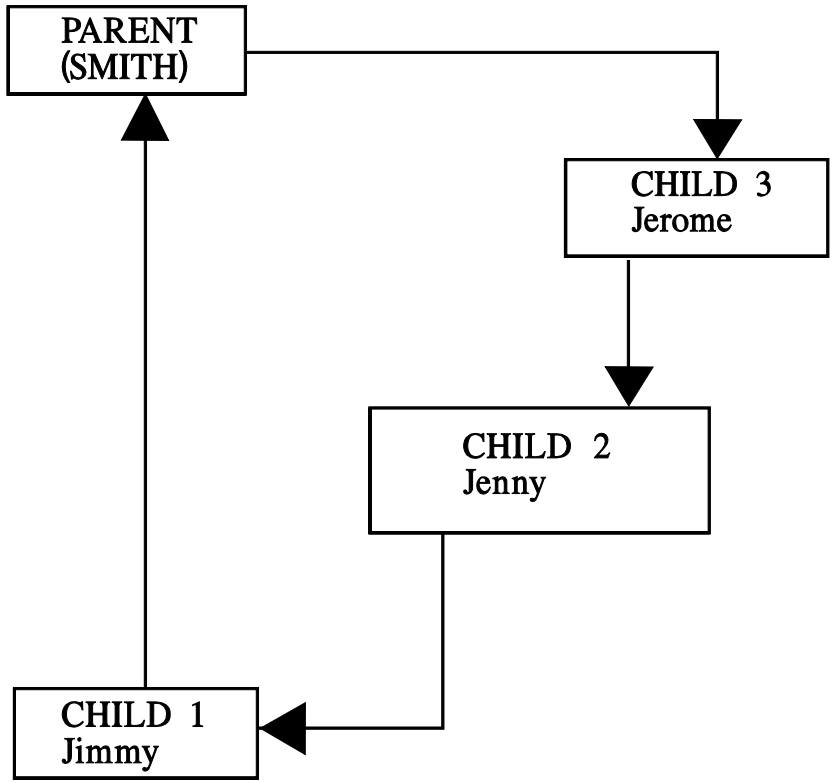
Indirect Network (ANSI/CODASYL)

- Two Set Types
  - Student with Grade Records
  - Course Section with Grades



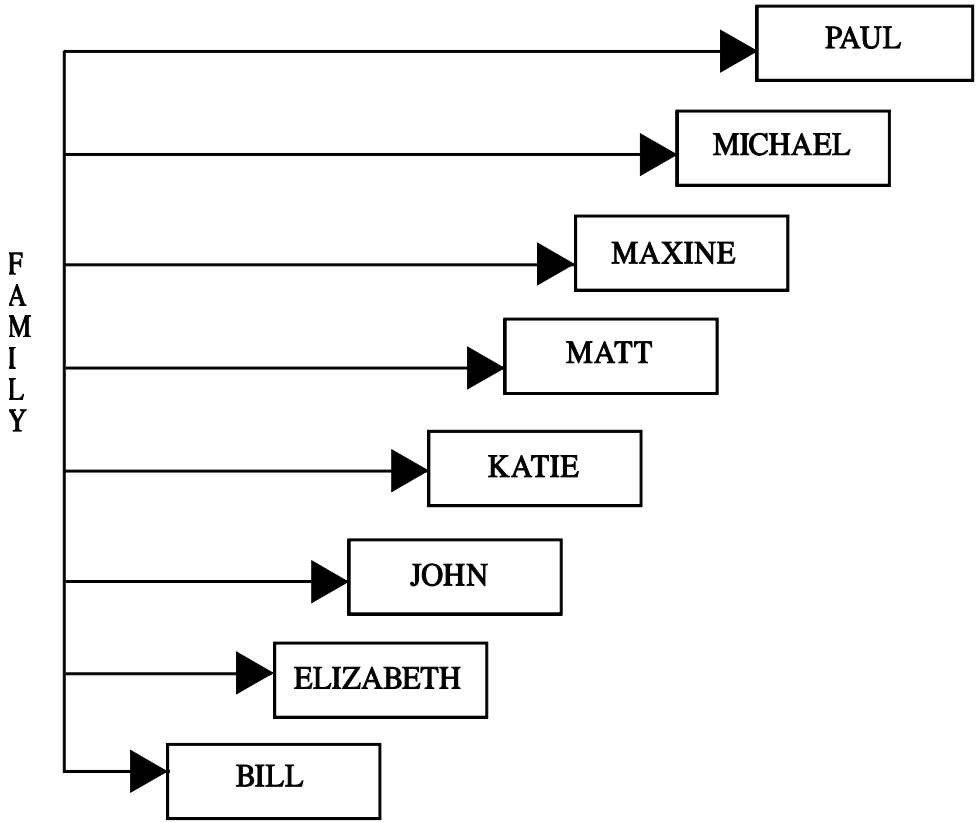
Regular Owner-Member Set

- Members
  - Set Type: Regular, Single Member
  - Information Bearing



Singular Set

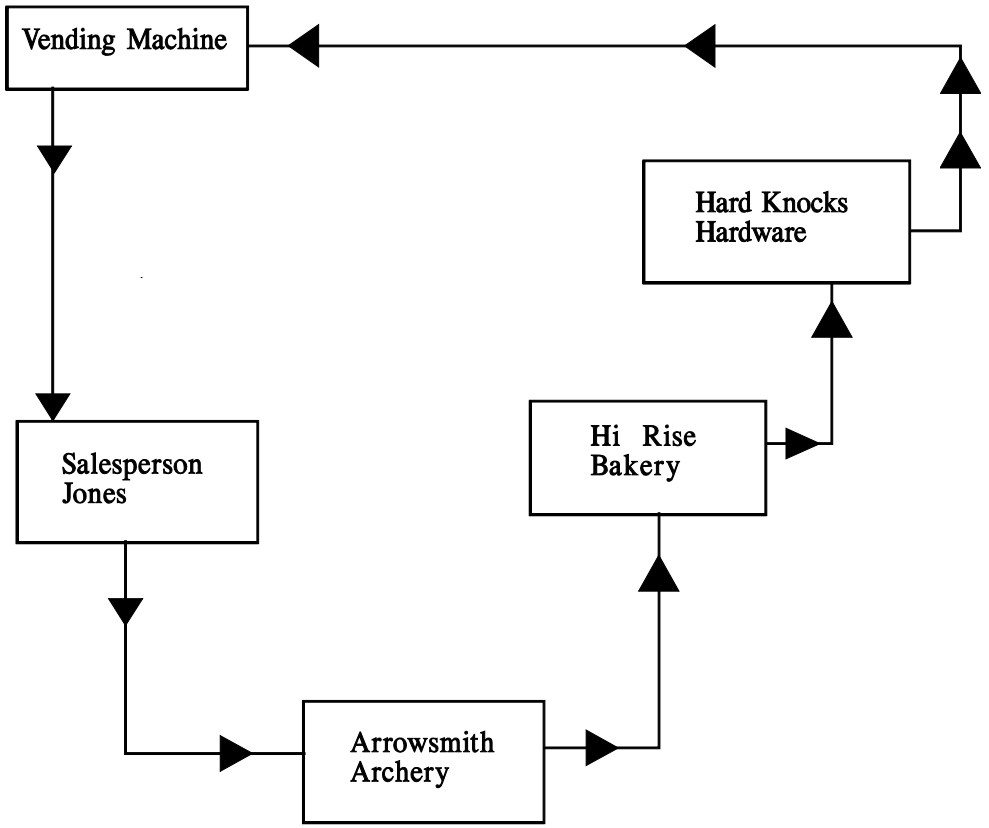
- Set Type: Singular, Single Member



Set Type : Singular, single member

Multi-member Set

- Set: Territory
- Owner: Machine Type
- Member: Salesperson, Store

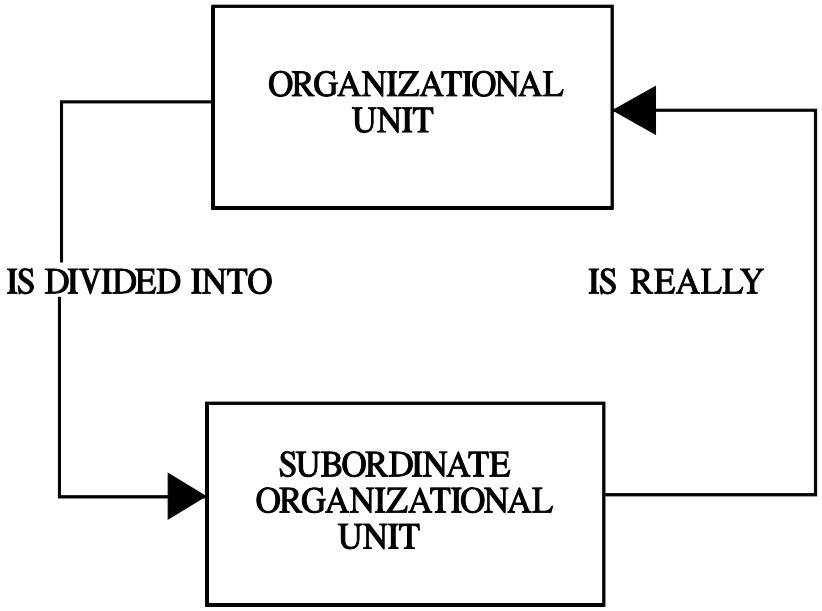


Set : Territory

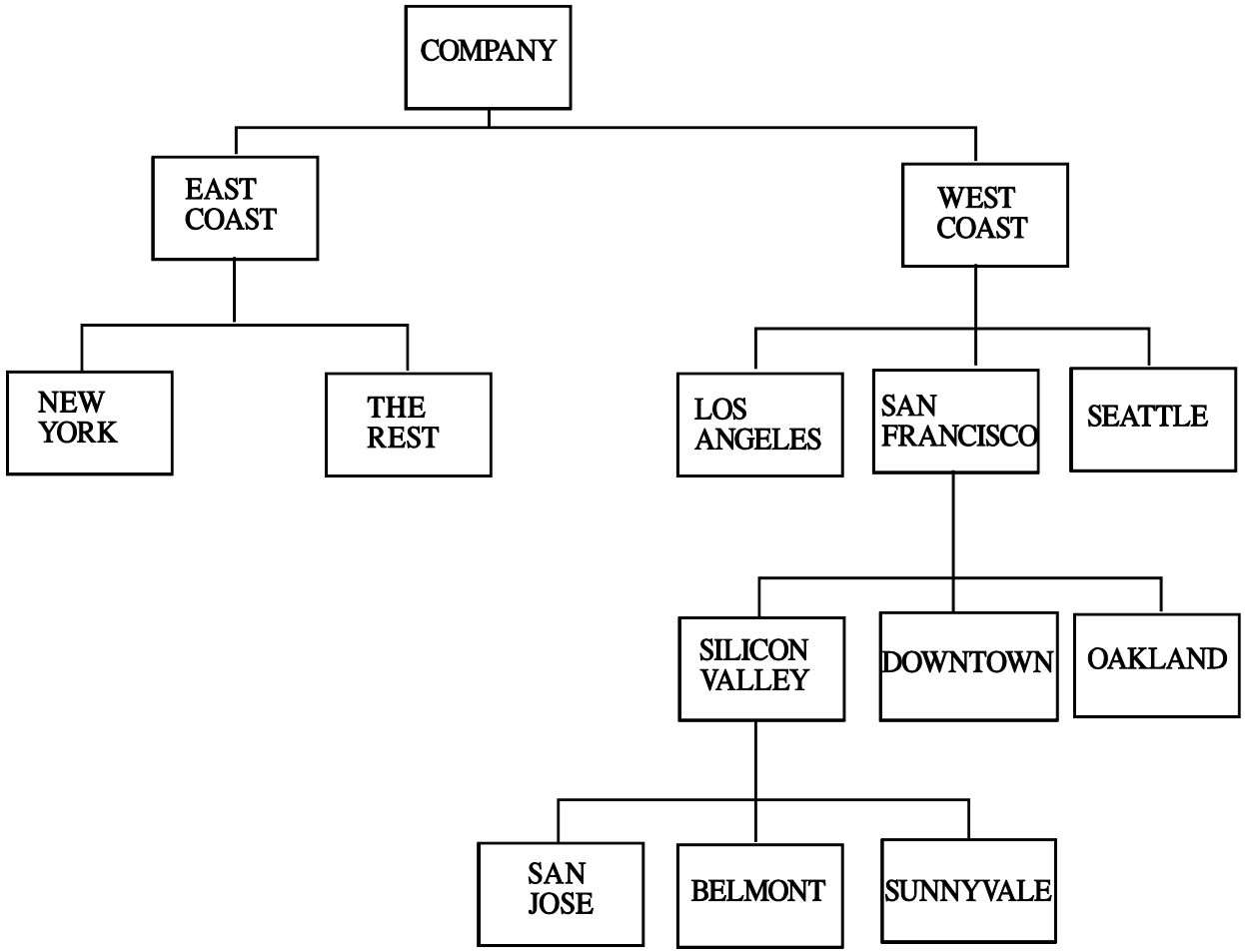
Owner : Machine Type

Member : Salesperson, Store

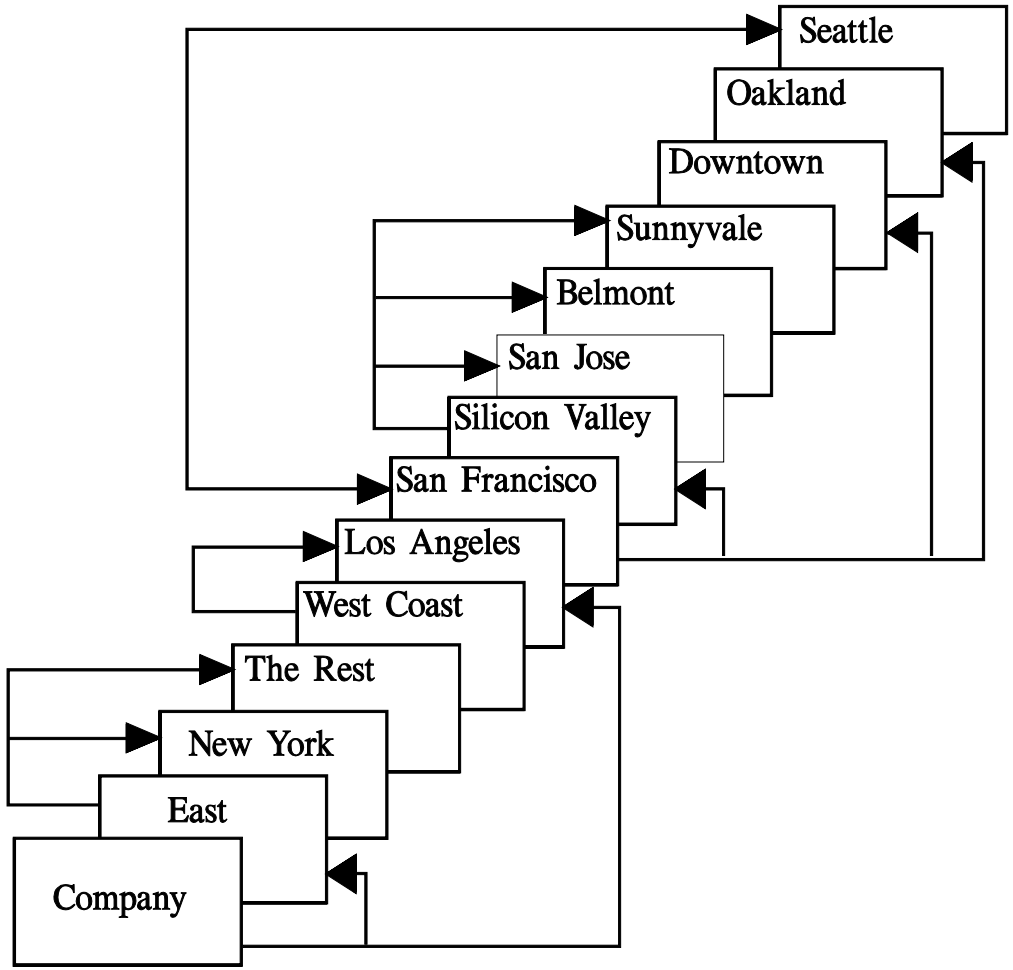
Recursive Set (Organization contains Organization)



Display of Hierarchically Linked Rows



Display of Hierarchically Linked Rows Within Organization Table



Record Insert Order Within Set May Be:

- First
- Last
- Next
- Prior
- System Default
- <Sorted>

Summary of ANSI NDL Relationships

- A Set Is a Collection of Related Records.
- There Are an Arbitrary Number of Sets in the Database.
- Each "Regular" Set Has One Owner Table And One Member Table.
- Each "Multiple-member" Set Has More than One Defined Member Table.
- Each "Singular" Set Has No Owner, and One or More Member Types.
- A "Recursive" Set Consists of One Table, Which Is Both Owner and Member Type.

ANSI NDL Summary (Cont).

- Each Owner Record Occurrence Defines a Set Occurrence
- There May Be an Arbitrary Number of Member Record Occurrences in One Set Occurrence
- Member Records May Be Ordered Within a Set.
- Set Records Can Be Accessed Directly by Specifying
- The Values of Record Data-items Set Records Can Be Accessed Positionally or Relatively.
- A Record Occurrence May Be a Member of More than One Set Type.
- A Record Occurrence May Not Be a Member of Two Occurrences of the Same Set.
- A Record Occurrence May Not Be a Member and an Owner of the Same Set Type.

3.2.4 Network Operations

- Row Operations
  - Retrieval
  - Find
  - Get
  - Update
    - Store
    - Delete

- Relationship Operations
  - Get Owner
  - Member
  - Next
  - Connect
  - Disconnect

- Combination Operations
  - Insert
  - Fetch

3.2.5 Network Summary

- Record Structure
- Simple or Complex Records
- Relationships

Type	Member	
	Singular	**Multiple**
Regular	Yes	Yes
Singular	Yes	Yes
Recursive	Yes	No

- Operations
  - ♦ Row Operations (Find, Store, Delete)
  - ♦ Relationship Operations (Connect, Disconnect)
  - ♦ Combination Operations (Fetch, Modify, Insert)

3.3 Hierarchy Data Model

- System 2000
- IBM's IMS

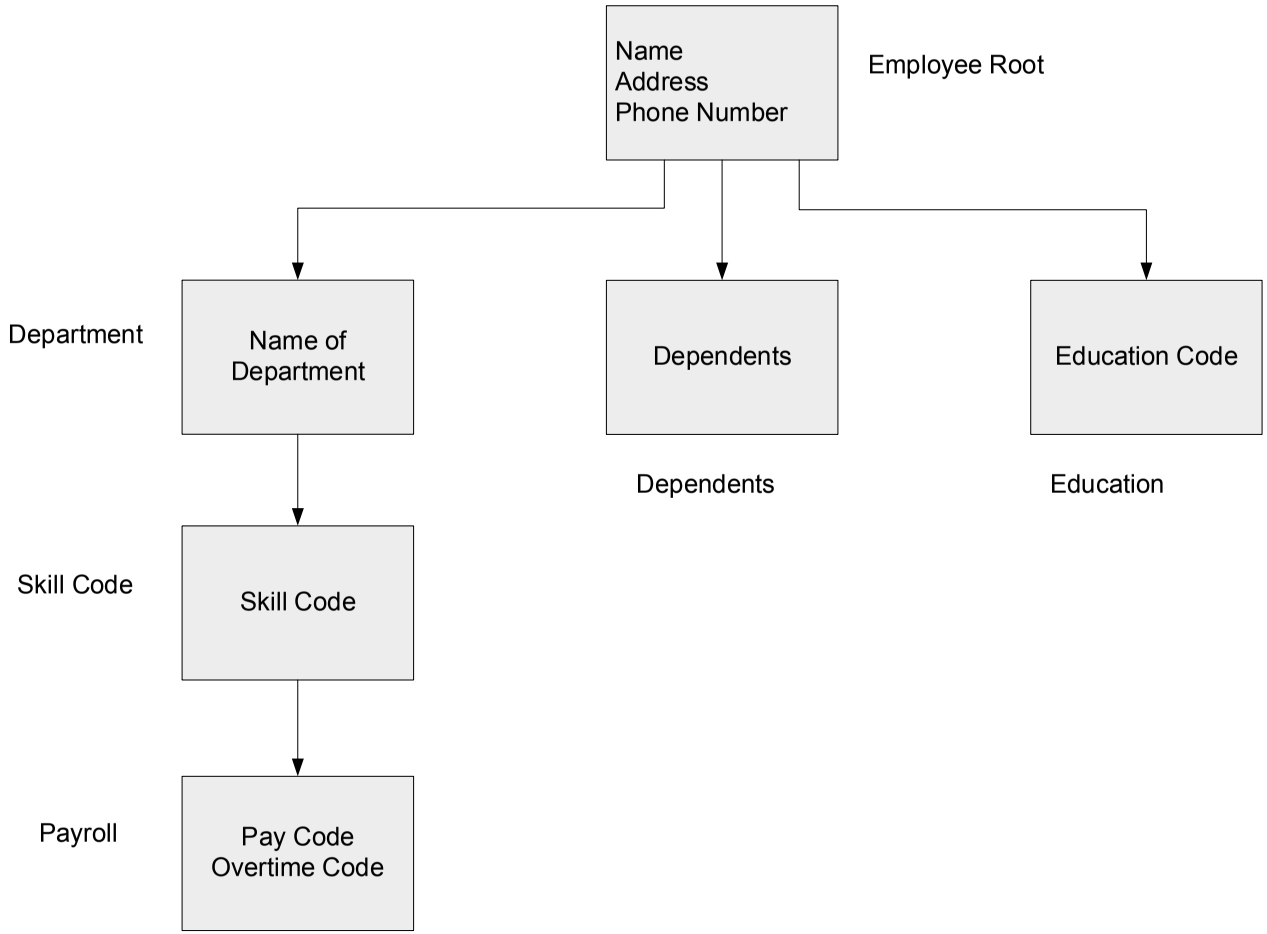
3.3.1 Hierarchy Definition

- A Data Structure in Which
- A Record May Be Related to Multiple Dependents, but Cannot Be Related to Multiple Owners.

3.3.2 Hierarchical Table Organization

Primary Key	SSN
Single Valued Columns	Birth Date Gender Current Job Degree Major

3.3.3 Hierarchy Relationship



3.3.4 Hierarchical Operations

- Row Operations
  - Retrieval
    - Find
    - Get
    - Insert
  - Modify
  - Delete
- Relationship Operations
  - Get Owner
  - Get Member
  - Get next
- Combination Operations
  - Modify
  - Delete Tree

3.3.5 Static Hierarchy Summary

- Record Structure
  - Flat Records (Single Valued Columns)
- Relationships
  - Owner-to-members
  - Members-to-owner
- Operations
  - Record (Find, Store, Delete)
  - Relationship (Get Owner, Get Member, Get next)
  - Combination (Modify, Delete Tree)

3.4 Network Summary

- Record Structure
  - ♦ Simple or Complex Records
  - ♦ Relationships

Type	Member	
	Singular	**Multiple**
Regular	Yes	Yes
Singular	Yes	Yes
Recursive	Yes	No

- Operations
  - ♦ Row Operations (Find, Store, Delete)
  - ♦ Relationship Operations (Connect, Disconnect)
  - ♦ Combination Operations (Fetch, Modify, Insert)

3.5 Hierarchical Summary

- Record Structure
  - Flat Records (Single Valued Columns)

- Relationships
  - Owner-to-members
  - Members-to-owner

- Operations
  - Record (Find, Store, Delete)
  - Relationship (Get Owner, Get Member, Get next)
  - Combination (Modify, Delete Tree)

3.6 Static Data Model Summary

- Relationship via DBMS Generated Pointers
- Loading via Structure
- Relationships Bound at Update Time
- Relationships Changed via Deletes & Re-adds

4.0 Dynamic Data Models

- Common Characteristics
- Independent Logical File
- Relational
- Summary

4.1 Common Characteristics

- Relationships Through DDL (Optional)
- Data Loading via Table
- Record Independence
- Concurrent Update
- Separate Processing
- Relationships Through Language Expression

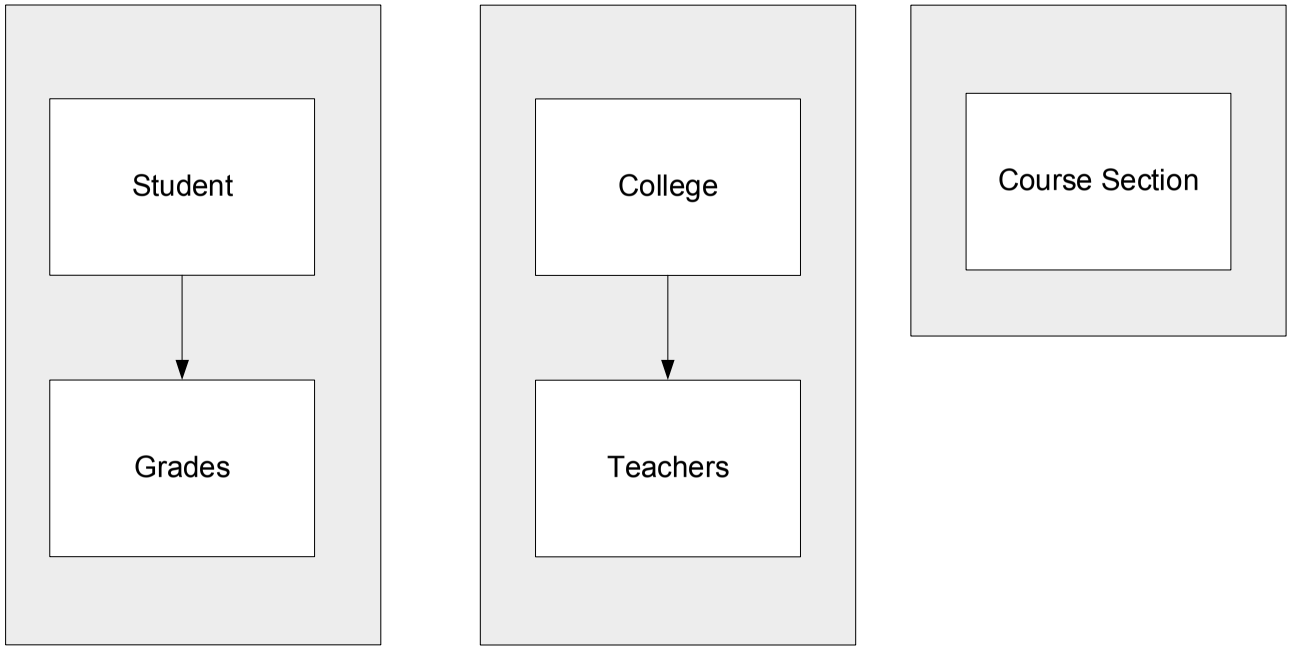
4.2 Independent Logical File

- Adabas
- Model 204
- Inquire
- Datacom-DB
- Focus

4.2.1 Definition

- Independent Logical File Data Model:
  - A Series of Independently Defined Tables (Called Files)
  - That Can Have a Complex Structure, and
  - Are Related to Each Other Through Common Column Values

4.2.2 Independent Logical File Table Structure

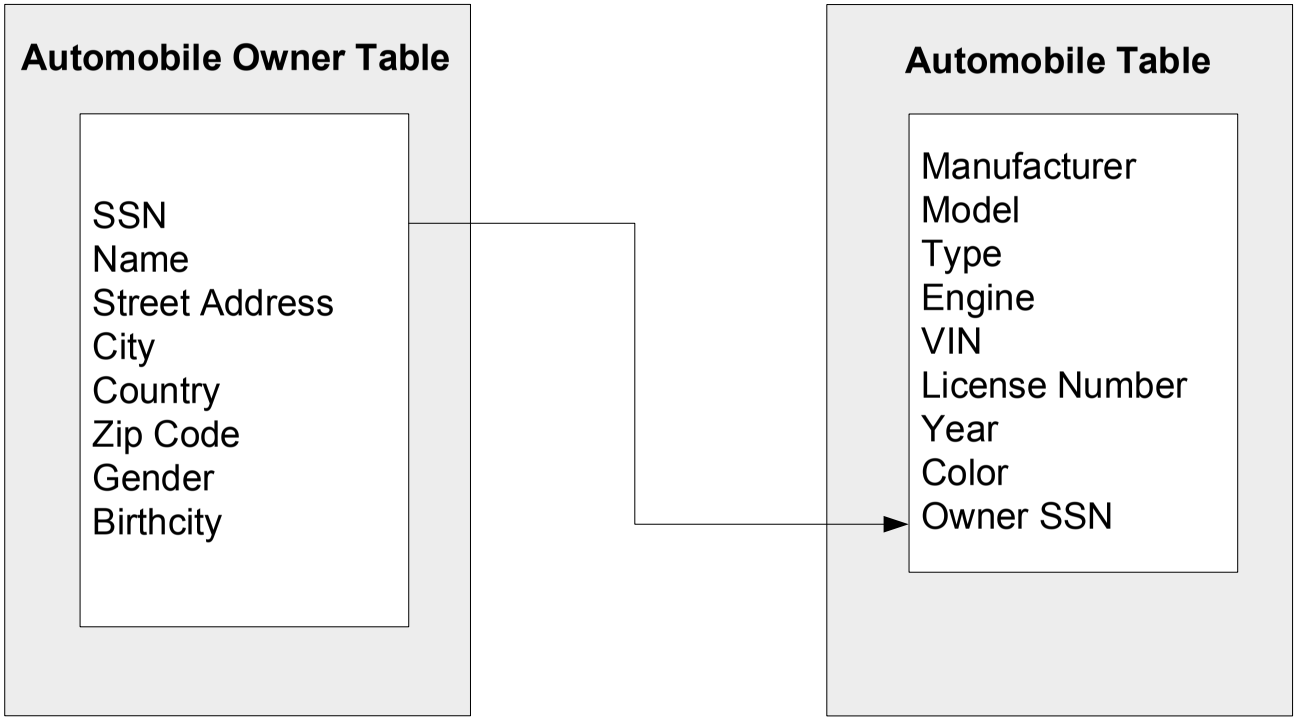


- Each Table Can Be Simple or Complex

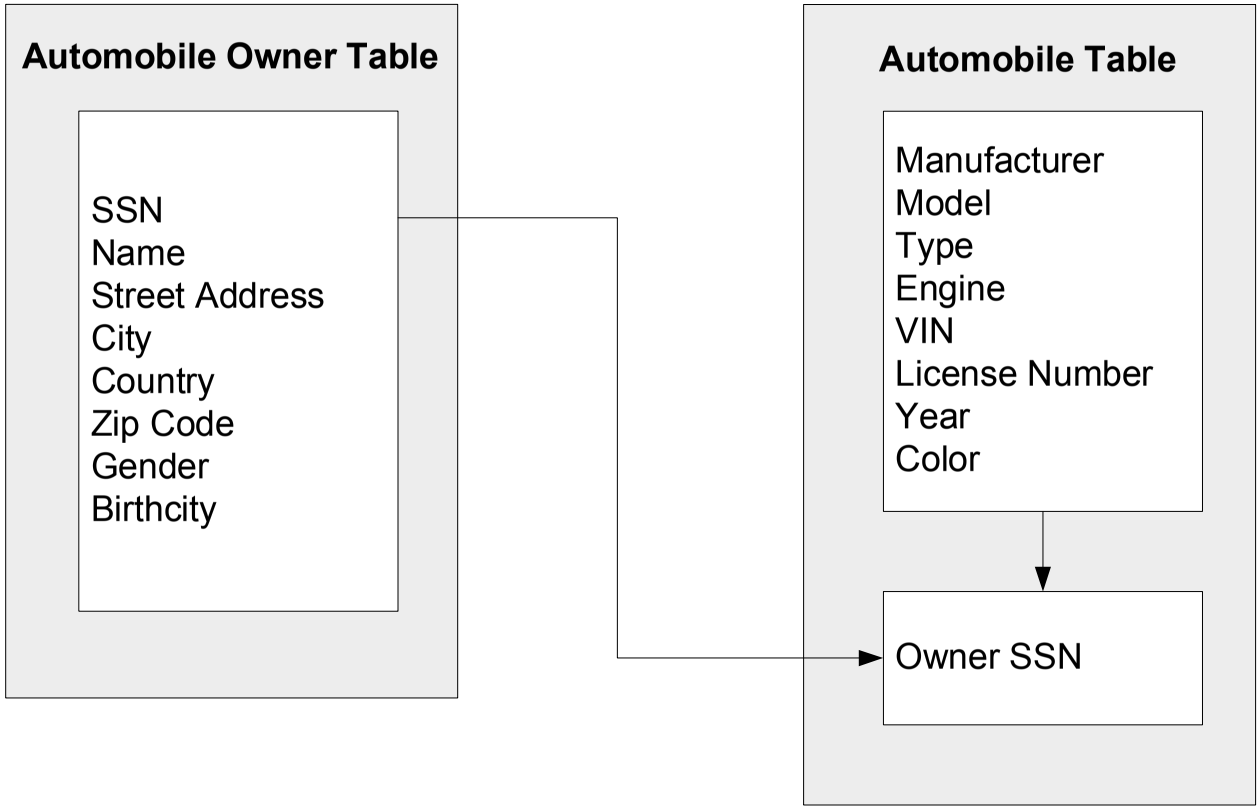
Independent Logical Table Organization	
Structure	**Example**
Primary Key	SSN
Non-Repeated Columns	Name Address Gender
Vectors	Nicknames
Matrix	Monthly Sales by Division
Repeating Group	Job Title Start Date End Date
Nested Repeating Group	Project Task Work

4.2.3 Independent Logical File Relationships

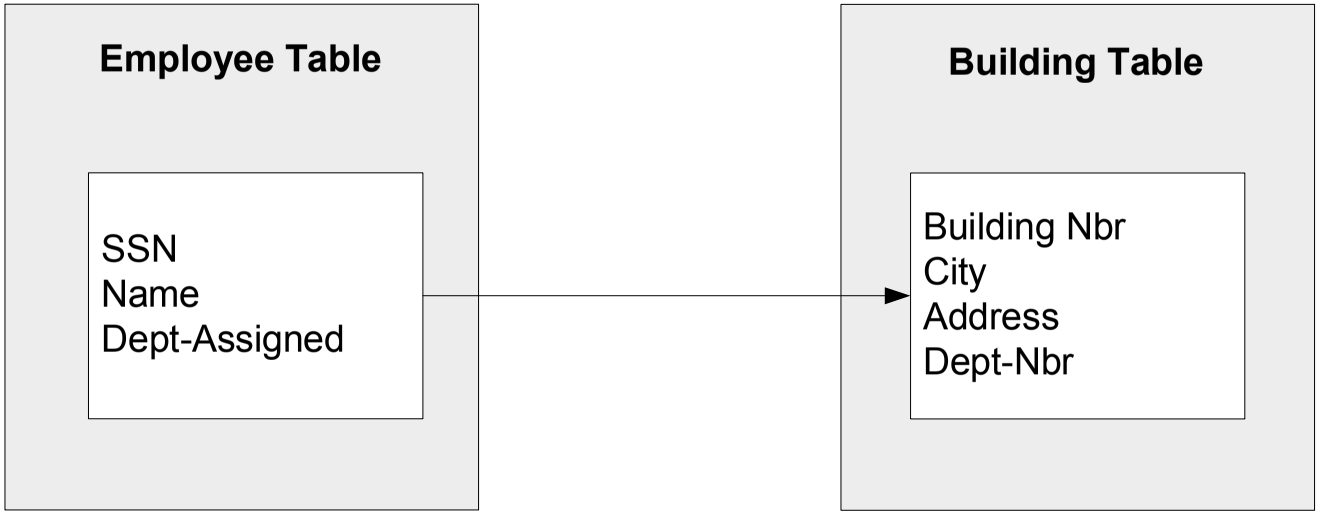
- One to Many



Many to Many



Inferential



4.2.4 Independent Logical File Operations

- Record
  - Retrieval
  - Find
  - Add
  - Delete

- Relationship
  - Connect <Table-name> to <Table-name>
  - Via <Table-name-1-column> Equal <Table-name-2-column>

- Combination Operations
  - Select <Table-name>
  - Ordered by <Column-name>(S)
  - Where <Column-name> <Relational Operator> <Table-name-2-column>

- Modify

4.2.5 Independent Logical File Summary

- Record Structure
  - Two Level Structures (Sometimes More)

- Relationships (Value Based)
  - One to Many
  - One to One
  - Many to Many
  - Inferential

- Operations
  - Record (Find, Add, Delete)
  - Relationship (Connect....Then Find)
  - Combination (Modify)

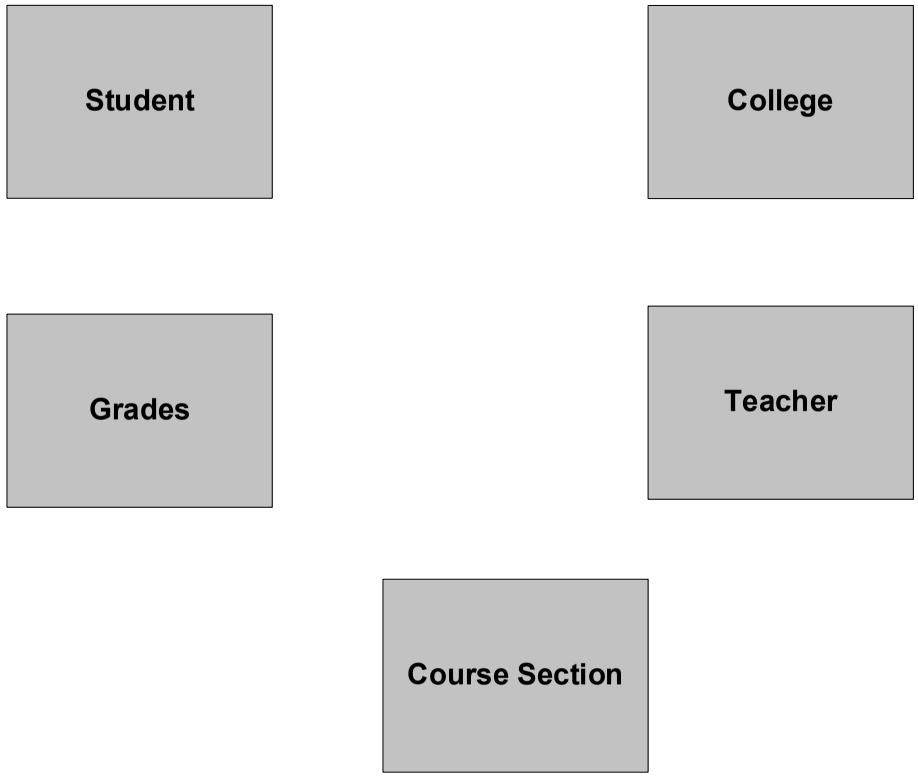
4.3 Relational Model

- Oracle
- DB-2
- Sybase
- Informix
- Ingress
- SQL Server 2000

4.3.1 Definition

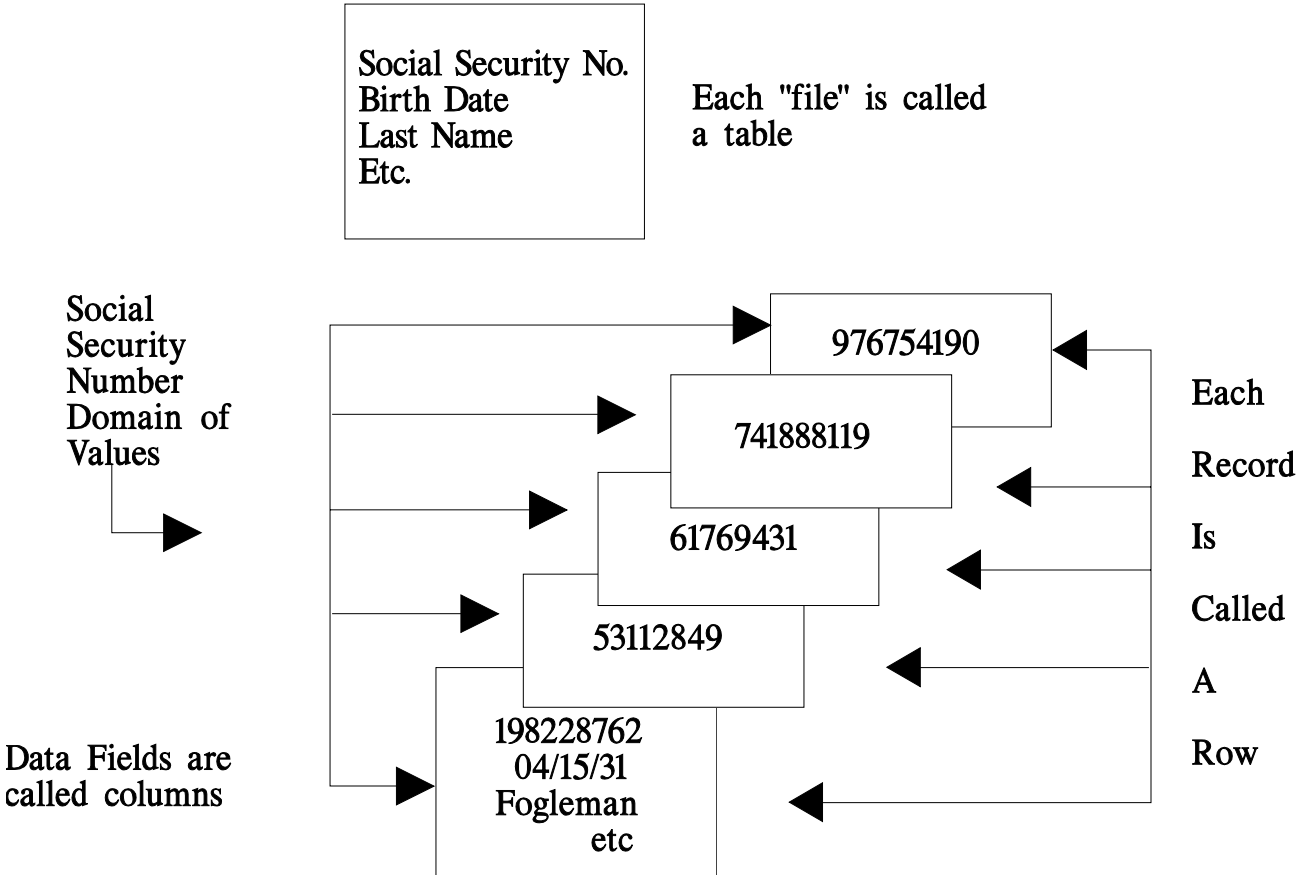
- the Relational Model Consists of a Series of Independently Defined Tables
- That Have Simple Structures
- Related to Each Other Through Common Column Values

4.3.2 Record Structure–Simple Records Only



- Each Table (Or Record or Relation) must Be Simple

4.3.3 Relational Terminology



As in Tables

Row Id	Columns		
	SSN	Birthdate	Last Name
1	423-89-2635	04/15/1931	Fogleman
2	235-53-8026	07/15/1965	Johnson
3	643-90-8719	06/25/1955	Smith

4.3.4 Three Normal Forms

Why 3NF?

Advantages

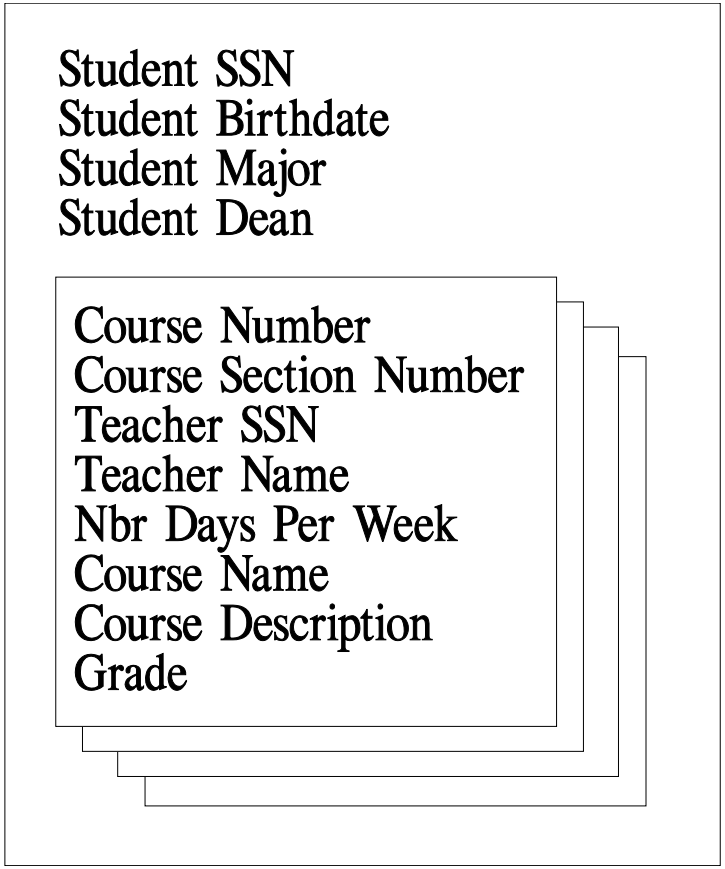
- Cleaner structure
- Column & value in single table.
- Faster, surer update delete, etc.
- Controlled data redundancy & that means greater integrity

Disadvantages

- "logical record" scattered among many small tables
- Re-collection of logical big job --especially i/o

Three Normal Forms

This is not in any normal form because of nested groups of Courses within Student and for other reasons.



1st Normal Form

After eliminating the nested repeating group, ...

Student SSN Course Number Course Section Number
Student Birthdate Student Major Student Dean Teacher SSN Teacher Name Nbr Days Per Week Course Name Course Description Grade

Over 25,000 students, how many times is the same Dean, Teacher Name, Course Name, and Course Description repeated?

2nd Normal Form

This not second normal form because part of the primary key determines some of the table's column values.

Student SSN Course Number Course Section Number
Student Birthdate Student Major Student Dean Teacher SSN Teacher Name Nbr Days Per Week Course Name Course Description Grade

- Student Number determines birth-date and major.
- Course Number determines course name and description
- Course Section Number determines Nbr Days Per Week
- Course Number & Course Section Number determines Teacher SSN

2nd Normal Form

Elimination of partial primary key dependencies. Created, for example, three tables for some of the "offending" dependencies.

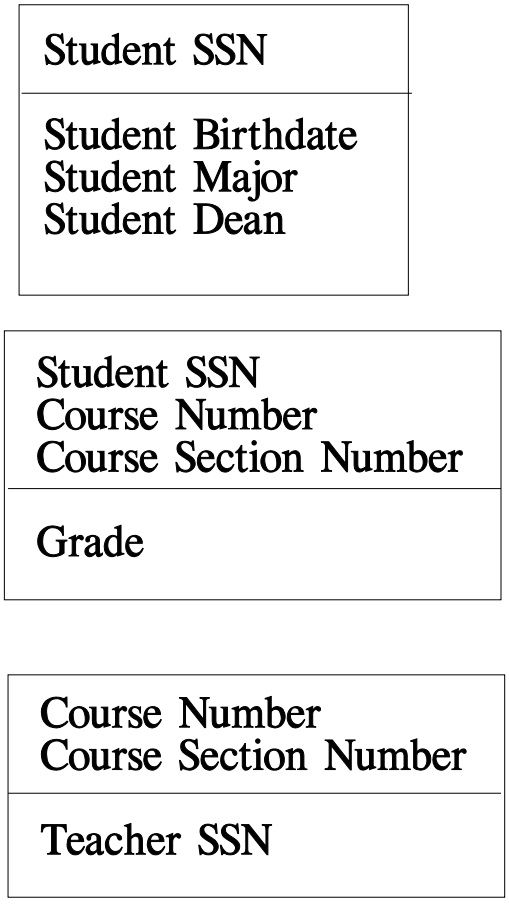
Student SSN
Student Birthdate Student Major Student Dean

Student SSN Course Number Course Section Number
Grade

Course Number Course Section Number
Teacher SSN

3rd Normal Form

Top Student table is not in Third Normal Form because a Non-key field, Major determines who the



Dean is. E.g, Engineering and Nursing both have a Dean Smith!

3rd Normal Form (cont.)

- To solve that problem, divide the Student table into two different tables, Major, and Student

Student SSN
Student Birthdate Student Major Student Dean

Major
Student Dean

Student SSN
Student Birthdate Student Major

3rd Normal Form (cont.)

Student SSN
Student Birthdate Student Major

Major
Student Dean

Student SSN Course Number Course Section Number
Grade

Course Number
Course Name Coures Description

Teacher SSN
Teacher Name

Course Number Course Section Number
Nbr Days Per Week Teacher SSN

Started with 1 file structure

Ended with 6 tables, all in third normal form

Normal Forms Summary

Normal Form	Key Considerations
None	Groups, arrays, repeating groups and nested repeating groups
1st Normal Form	No Embedded Repeats/groups
2nd Normal Form	1st Normal Form and No Partial-key Field Determinates
3rd Normal Form	2nd Normal Form and No Non-key Field Determinates

4.3.5 Relational Operations

- Record
  - Add
  - Delete
  - Modify

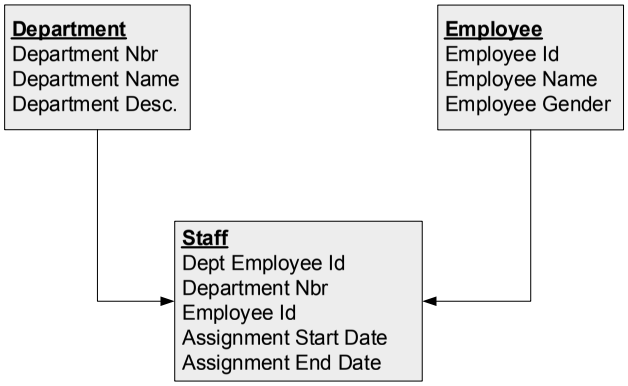
- Relationship
  - Select
  - Project
  - Product
  - Union
  - Intersection
  - Difference
  - Join
  - Divide

Critical Requirement for Some Operations

- Union Compatible
- Same Degree (Number of Columns)
- Attributes from Same Domain

4.3.6 Referential Integrity

Cases	Referential Integrity Specification							
	Update				Delete			
	Cascade	Set		No Action	Cascade	Set		No Action
		Null	Default			Null	Default	
Department. For Update the Department Nbr is changed from one value to a new, non-existing value. For Delete, the Department row of a certain Department NBR is deleted	Corresponding Staff Department rows are also updated	Old Staff Department Number is set to Null	Old Staff Department Number is set to pre-determined default value	Department number cannot be changed	Corresponding Staff Department rows are also deleted	Department row is deleted. Staff rows of Department Number are set to Null	Department row is deleted. Staff rows of Department Number are set to Null	Department cannot be changed



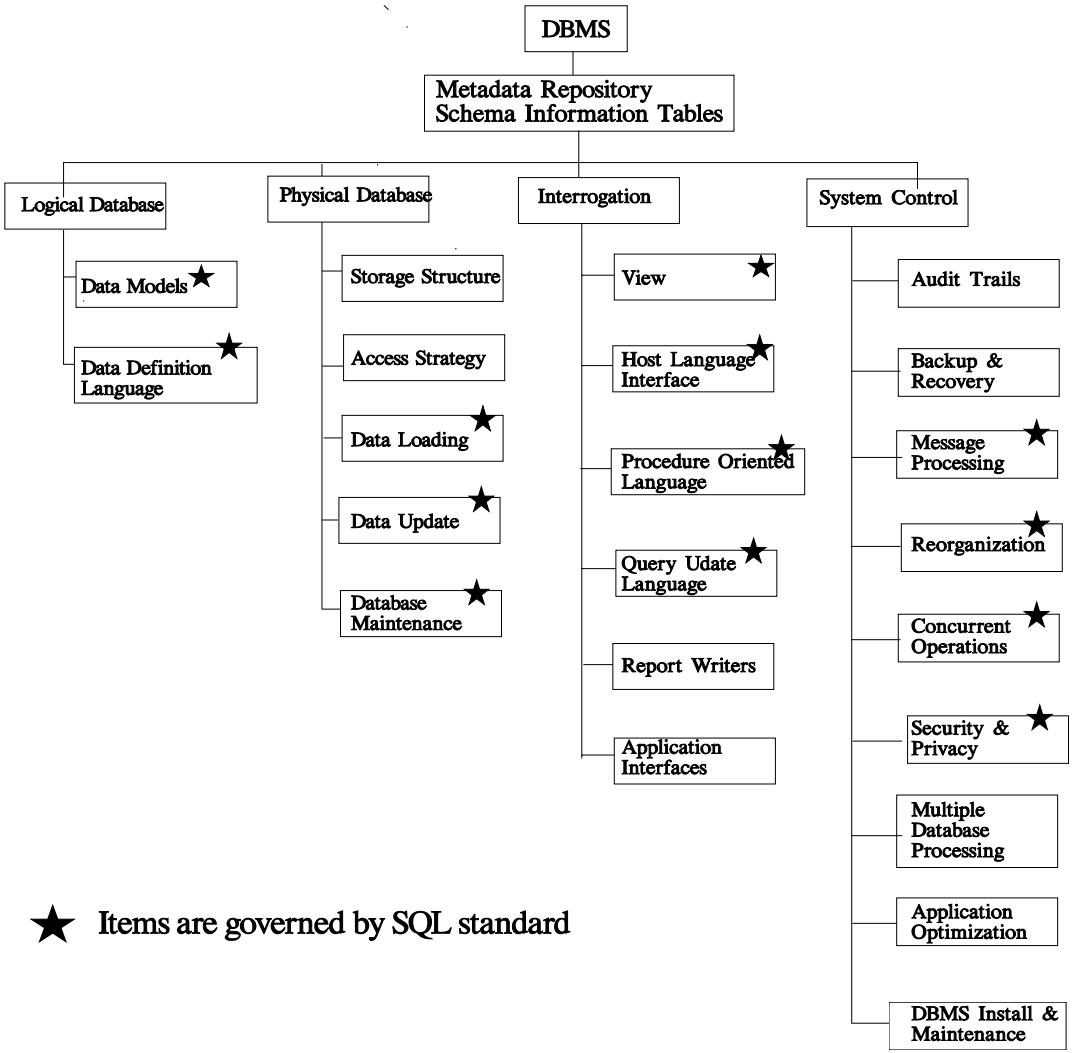
4.3.7 Relational Model Summary

- Table Structure
  - ♦ Flat Records
  - ♦ Relationships
  - ♦ Value Based
- Operations
  - ♦ Record (Add, Modify, Delete)
  - ♦ Relationship (Select, Project, Product, Union, Intersection, Difference, Join, Divide)
- Referential Integrity

4.4 Dynamic Model Characteristics Summary

- Independent Record/relation Processing
- Dynamic Ordering
- Dynamic Relationships (Sets)

5.0 SQL Data Model



5.1 Evolution of the SQL Language

SQL/1986	SQL/1989	SQL/1992
Basic Tables	SQL/1986 plus	SQL/1989 plus
Some integrity constraints Language bindings to COBOL, FORTRAN, C, etc.	Partial Referential Integrity	Assertions Bit data type CASE Character Sets Connection Management DATETIME Domains Dynamic SQL Enhanced constraints Full Referential Integrity Get Diagnostics Grouped operations Information Schema Multiple module support National character sets Natural joins (inner & outer) Row & Table constraints Schema manipulation

SQL/1986	SQL/1989	SQL/1992
Basic Tables	SQL/1986 plus	SQL/1989 plus
		Subqueries in check clauses Table constraints Temporary tables Transaction Management Union and intersect

5.2 SQL:1999 Data Model

- Data Structures
- Relationships
- Operations

5.2.1 SQL Data Model: Data Structures

Column Data Type	Definition	SQL:1999
Single Value	Each component represents a single value such as Birthdate with the value 11/11/1987	✔
Multi-value	Each component represents multiple values such as Nicknames with values “Buddy, Guy, Mac”	✔-Note 1
Groups	Each component has subcomponents to represent single-set of values such as Address with Street-1, Street-2, City, State, Zip	✔-Note 2
Repeating Groups	Each component has subcomponents to represent multi-sets of values such as Dependents that contains subcomponents, Dependent Name, Dependent Birth date, Dependent SSN.	✔-Note 3
Nested Repeating Groups	Employee (Dependents (Hobbies))	✔-Note 4

Notes:

1. Arrays as a data type for a column

2. ROW data structure of a column

3. ROW data structure for a column wherein each Row structure field has the data type, ARRAY

4. ROW data structure for a column with contained ARRAYs where each ARRAY column is a ROW data structure with contained ARRAYs where each ARRAY column , etc....

5.2.2 SQL Data Model: Relationships

Name	Example	SQL:1999
One-to-many	Employee to dependents	✔-Note 1
Owner-multiple-member	Territory contains salesmen and customers	✔-Note 2
Singular-one-member	Top performing employees	✔-Note 1
Singular-multiple-member	Top performing current, former, part-time, and retired employees	✔-Note 3
Recursive	Organization contains organization	✔-Note 4
Many-to-many	Automobiles and owners	✔-Note 5
One-to-one	Table and its primary key	✔-Note 6
Inferential	Many houses each with a location, and then buyer with desired location	✔-Note 7

Notes:

1. Traditional relational data model (SQL:1986, 1989, & 1992)

2. Implemented as a ROW(TerritoryId, Salesman REF (SalesmanId),

3. Developed using Subtables where Employees are partitioned off into their common columns (employee) and their unique columns (current, former, part-time, and retired)

4. Recursion operations built into the language (WITH RECURSION...)

5. Cross joins from within columns of ARRAYS contained as data types of column in different tables

6. Effectively as tables and subtables. Most directly with UNIQUE Fkey

7. A single valued non-primary key Location within House and same for Buyer

5.2.3 SQL Data Model: Operations

SQL: 1999 Operations

- **Row Operations –** traditional insert, delete, and modify of rows and columns within rows
- **Relationship Operations** – traditional operations that affect the relationships between rows of the same or different tables

Row Operations			
Operation	**Static**	**Dynamic**	**SQL:1999**
Find	SELECT According to STORED Order	SELECT and PUT into DML Specified Order	✔
Get	Obtain Row From Find	Ditto	✔
Add	Install a New Row Into Database	Ditto	✔
Delete	Remove an Existing Row From Database	Ditto	✔
Modify	Change Some Data Column Values in Existing Row	Ditto	✔

Relationship Operations			
Operation	**Static**	**Dynamic**	**SQL:1999**
Connect	Add to a Named RELATIONSHIP in Specific Order	N/A	✔-Note 1
Disconnect	Delete From RELATIONSHIP	N/A	✔-Note 2
Get Owner	Obtains The Parent of the Row That is Current	N/A	No
Get Member	Obtains the First Child of the Owner For the Named Relationship	N/A	✔-Note 3
Get next	Obtains the Next Row Within The Named Relationship	N/A	✔-Note 4
Intersect	N/A	Find and Keep Only the Common	✔
Difference	N/A	Find and Keep Only the Not Common	✔

Relationship Operations			
Operation	**Static**	**Dynamic**	**SQL:1999**
Join	N/A	"Append" Relations to Each Other	✔
Divide	N/A	Subset	✔
Product	N/A	Cross-Product	✔
Union	N/A	Merge and Drop Duplicates	✔

SQL:1999 Relationship Operations(cont.)

Notes:

1	Value an Column in an ARRAY. Data type of Array is REF type pointing elsewhere.
2	Set value of column in array to null
3	Get the first column of an array. Data type of Array is REF type pointing elsewhere
4	Get next column in array. Data type of Array is REF type pointing elsewhere

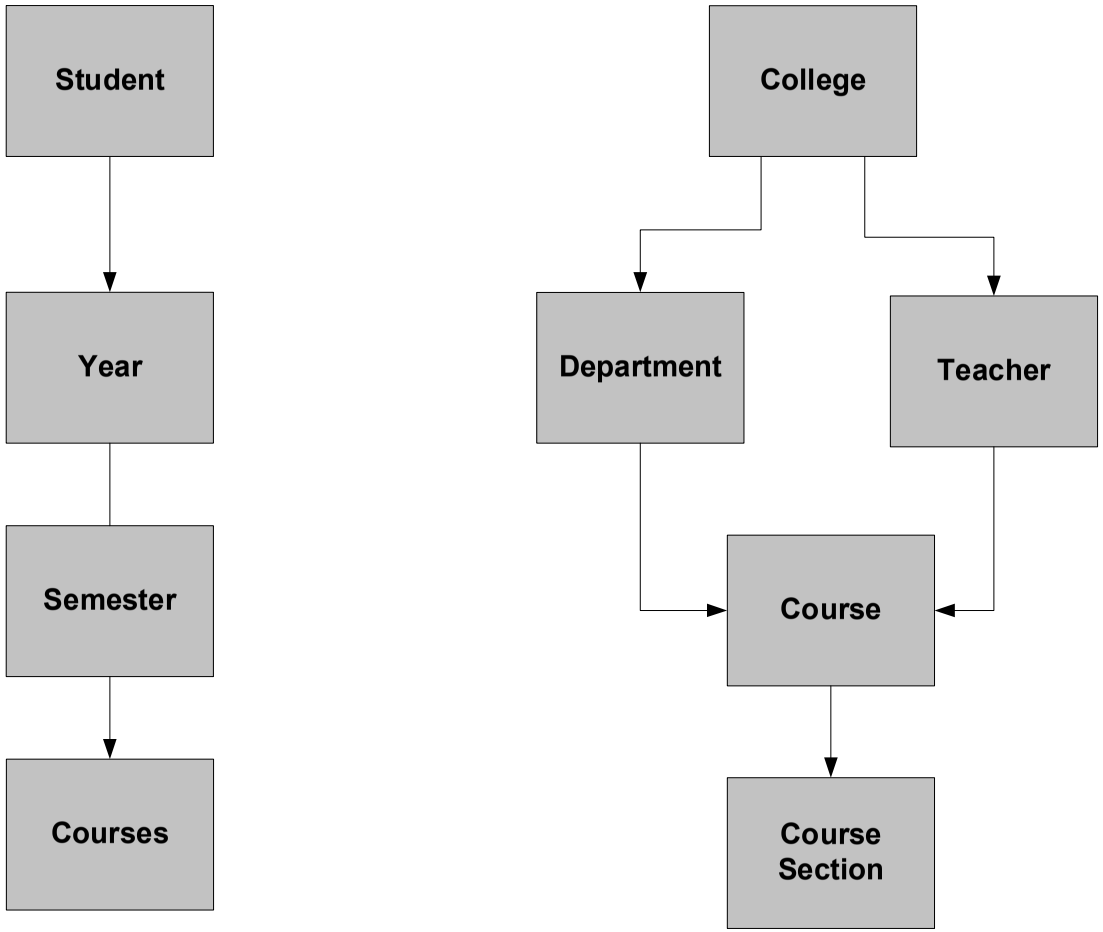
6.0 Schemas and Subschemas

- Schema Types
- Logical Database Schemas
- User Schema (Aka Subschema)
- View Facilities
- Summary

6.1 Schema Types

- Logical (Database) Schema – Created Through Database Definition
- User Schema (A.K.A. Subschema) – Created by Dba for Specific Use or User
- Physical (Database) Schema – Defines Physical Structure Layout
- Security Schema – Permits Specific Restricted Usages

6.2 Database Schema (Logical Database)



6.2.1 Schema Content

Schema Name Clause

- Record Name Clauses

- Column Clauses
  - ♦ Descriptions
  - ♦ Conditions

- Relationship Specification Clauses
- Owner Clause
- Member Clause
- Condition Clauses

Record and Column Definitions

- Record Is Salesperson
- On Store, Call Store-rec-proc Indexes Are Ssn, Job-title, Region
- Duplicates Not Allowed
- Fields Are
  - Ssn, Type Is...
  - Employee-name, Type Is...
  - Job-title, Type Is....
  - Region, Type Is....
  - Record Is ....

Column Definition (Detail)

- Field Is Job-title Type Is Char, Length Is 10,
- Null Is Not Allowed, Duplicates Are Allowed
- Valid Values Are Senior, Trainee, Manager
- Job-type Column Definition

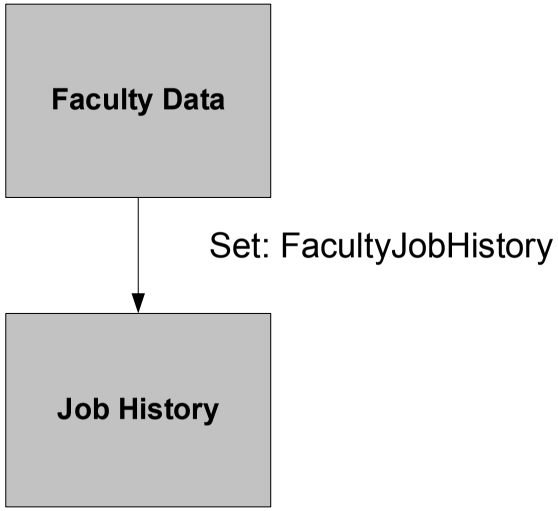
Relationship Definitions

- Relationship Is Territory
- Owner Is Region
- Order Is Salesperson, Customer
- Member Is Salesperson
  - Duplicates Prohibited
  - Key Is Salesperson-id
  - Connection Rules Are
  - Assign-date Ge Emp-hire-date
  - Assign-date Le Emp-term-date
- Member Is Customer
  - Duplicates Are Prohibited
  - Key Is Customer-id

.

6.2.2 Network Schemas

- ANSI-network Schemas



ANSI Network DDL

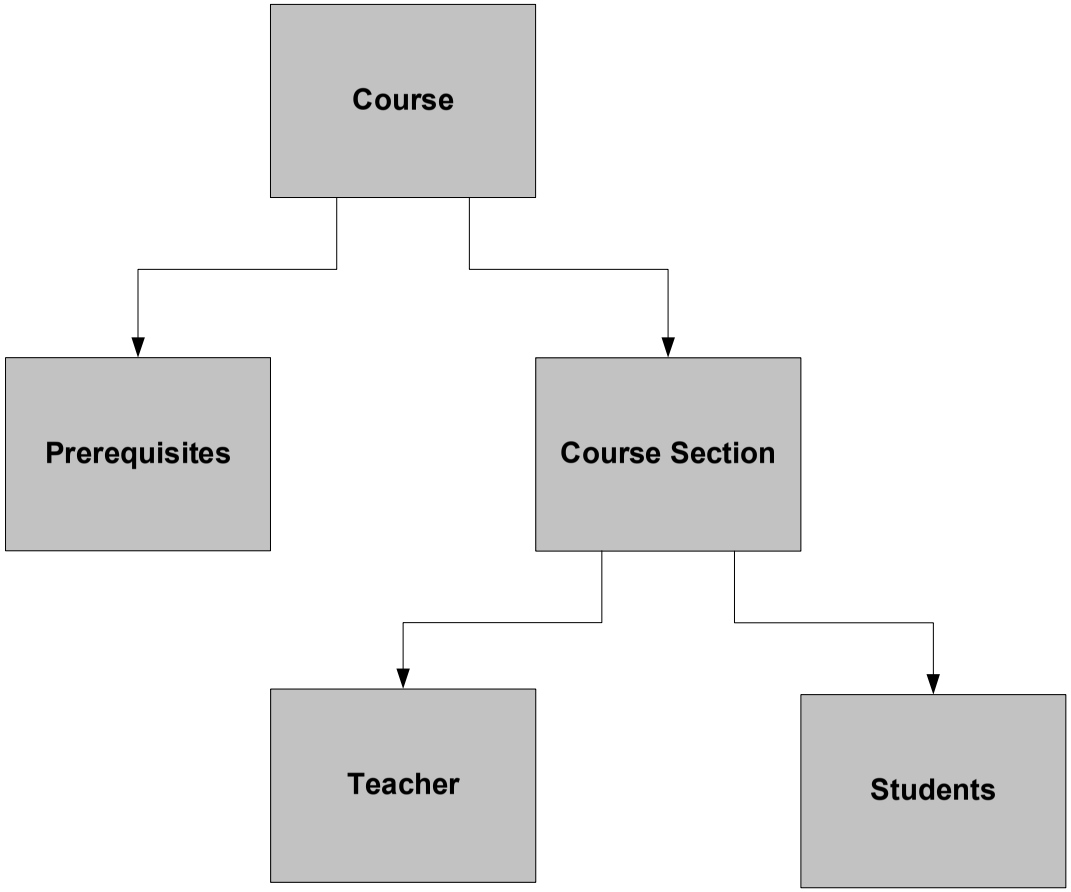
- Schema Name Is Faculty-data

- Record Name Is Faculty
  - Duplicates Are Not Allowed
  - Name Type Is Character 25
  - Address Type Is Character 40
  - SSN Type Is Character 11

- Record Name Is Jobhist
  - Jobtitle Type Is Character 25
  - Jobcode Type Is Fixed Decimal 2

- Set Name Is Fac-jobhist
  - Order Is Sorted
  - Owner Is Faculty
  - Member Is Jobhist
  - Automatic Mandatory
  - Ascending Key Is Jobcode
  - Duplicates Are Allowed

6.2.3 Hierarchical Schemas



IMS/DLT Database Definition

- Dbd Name=ecudpdbd
- Segm Name=course, Bytes=256
- Field Name=(Coursenbr,seq),Bytes=3,start=1
- Field Name=title,bytes=33,state=4
- Field Name=description,bytes=220,start=37
- Segm Name=prereq,parent=course,bytes=36
- Field Name=(Coursenbr,seq),Bytes=3,start=1
- Field Name=title,bytes=33,start=4
- Segm Name=offering,parent=course,bytes=20
- Field Name=(Date,seq,m),Bytes=6,start=1
- Field Name=location,bytes=12,start=7
- Field Name=format,bytes=2,start=19
- Segm Name=teacher,parent=offering,bytes=24
- Field Name=(Empnbr,seq),Bytes=6,start=1
- Field Name=name,bytes=18,start=7
- Segm Name=student,parent=offering,bytes=25
- Field Name=(Empnbr,seq),Bytes=6,start=1
- Field Name=name,bytes=18,start=7
- Field Name=grade,bytes=1,start=25

Static Hierarchy (System 2000)

Number	Component
1	Course Title (Int 9(6))
2	Course Description (Name X (25))
3	Prerequisites (RG)
4	Prereq Course Nbr (Int 9(6) in 3)
5	Prereq Course Title (Int 9(6) in 3)
6	Offering (RG)
7	Section Nbr (Int 99 in 6)
8	Date (Days of Week (Name X (12) in 6
9	Teachers (RG in 6)
10	Teacher Nbr (Int 9(9) in 9)
11	Students (RG in 6)
12	Student Id (Int X (9) in 11)
13	Final Grade (Name Xx in 11)

6.2.4 Dynamic -- Independent Logical File

```
Record Is Teacher
Primary Key Is T-ssn
Columns Are
        T-ssn Type Is Numeric
        T-name Type Is Character 30
```

```
Record Is Course
Primary Key Is Course-nbr
Columns Are
        Course-nbr Type Is Numeric
        Course-name Type Is Character 30
```

```
Record Is Course-section
Primary Key Is C-s-nb
Alternates Keys Are
        Course-nb
        Teacher-ssn
Columns Are
        C-s-nb Type Is Group
        Course-nbr Type Is Numeric in C-s-nb
        Section-nb Type Is Numeric in C-s-nb
        Teacher-ssn Type Is Numeric
```

6.2.5 Relational

- Create Table S
  (S# Char(5) Not Null
  Sname Char(20),
  Status Smallint,
  City Char(15));

- Create Table P
  (P# Char(6) Not Null
  Pname Char(20),
  Color Char(6),
  Weight Smallint,
  City Char(15));

- Create Table J
  (J# Char(4) Not Null
  Jname Char(10),
  City Char(15));

- Create Table SPJ
  - (S# Char(5) Not Null,
  - P# Char(6) Not Null,
  - J# Char(4) Not Null,
  - Qty Integer)

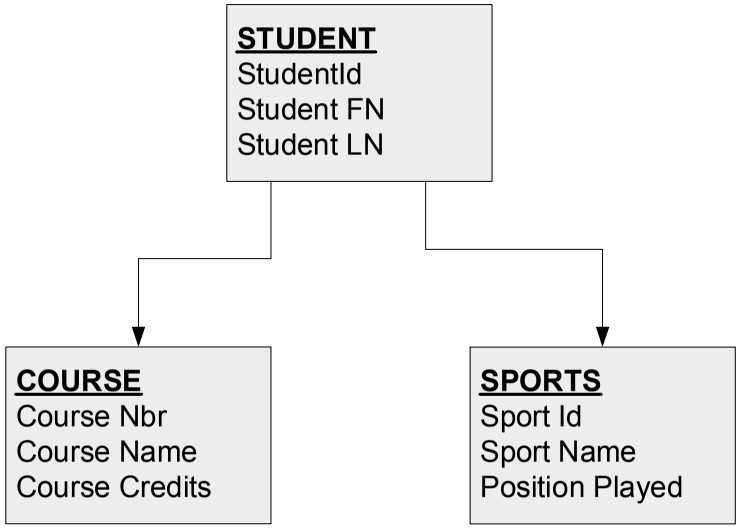
6.2.6 Schema Summary:

- Describes Entire Database
- Many Different Forms
  - ♦ All Essentially the Same
- Process
  - ♦ Data Definition Language Is Fed to DBMS
  - ♦ DBMS Creates "Schema"
  - ♦ Stores it into Dictionary Portion of Database's
  - ♦ Storage Structure

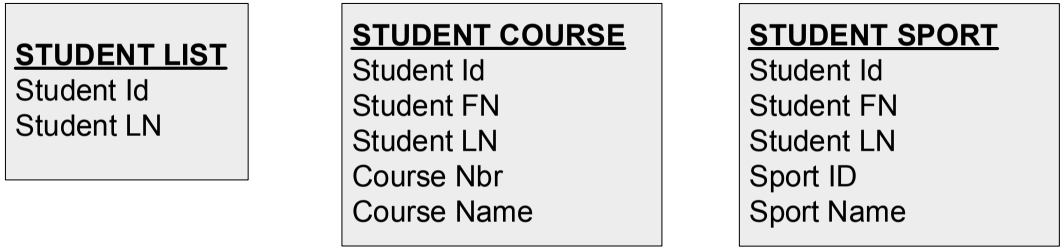
6.3 User Schema (a.k.a., Subschema or View)

- Definition
  - A Subordinate Part of a Database, A Logical Sub-division.
- Purpose
  - Logically Segment Large Schema into Smaller, More Easily Understood "Pieces"
- Role
  - User Interface Mechanism:
    - It Contains Local Names and Formats

Given: Student Schema,



The Three Subschemas Might Be:



6.3.1 Subschema Content

Common Part

- Subschema Name Is Subpart
  - Record Is Employee
    - Person Id
    - First Name
    - MI
    - Last Name
  - Record Is Leave
    - Leave Type
    - Hours Remaining

Subschema Content (cont.)

- Language Specific Section
  - Language Is Fortran
  - Record Employee
    - Person Id Is Integer
    - Rename to Iperid
  - Language Is Query
  - Record Is Employee
    - Person Id Is Chr 9(x)
    - Rename to Employee Identifier
    - Null Is Error
    - Blank Is Error

6.3.2 Network Subschema

- **Subschema Employee**
  - Record Name Is Exposition
    - Store Is Not Allowed
    - Erase Is Not Allowed
    - Columns Are
    - Pos-finish-date
    - Pos-start-date

6.3.3 Hierarchical Subschema

- (System 2000 Subschema)
- Schema Education of Personnel.
    - Education
    - Degree Picture Is Name X(20)
    - Year Picture Is Integer 9999
- Schema Study Areas of Personnel
    - Study Areas
    - Subject Picture Is Name X(20)
    - Credits Picture Is Integer 99
- Schema Employee of Personnel
    - Employee
    - Employee Number Picture Is Integer 9(9)
    - Employee Name Picture Is Name X(30)

6.3.4 Independent Logical File

- CSC's DBMS Manage Example
  - Subschema Demo3 Within Demo
  - Record Supply
  - Record Invty
  - Record Orders
  - Record Purord
  - Structure Inv Contains Supinv, Invord, Purinv
  - Structure Pur Contains Suppur
  - Report Record Description
  - Record Stock Renames Invty
  - Part Number Renames Stocknum
  - Heading Is "Part", "Number"
  - Report Structure Description

6.3.5 Relational

- Create View <View Name>
- As Select S#, P#, Sum(qty)
  - From Spj
  - Group by S#, P#

6.4 Schema and Sub-schema Summary

- Schema Summary
- Describes Entire Database
- Logical
- Many Different Forms
- All Essentially the Same

Schema - Summary (Cont.)

- Data Definition Language Is Fed to DBMS
- DBMS Creates "Schema"
- Stores it into Dictionary Portion of Database's Storage Structure

Sub-schema Summary

- A Subordinate Part of a Schema
  - Records to Be Included
  - Columns to Be Included
- Interface Language Requirements
  - Local Names
  - Data Type Conversions

7.0 Logical Database Summary

- Data Models
  - Network
  - Hierarchy
  - Independent Logical File
  - Relational
  - SQL 1999 and 2003
- Data Model Components Review
- Data Model Effects on Applications
- Summary

7.1 Network

- Table Structure – Simple or Complex Records

- Relationships

Member Table	Owner Table	
	Singular	Multiple
Regular	Yes	Yes
Singular	Yes	Yes
Recursive	Yes	No

- Operations
  - Row Operations (Find, Store, Delete)
  - Relationship Operations (Connect, Disconnect)
  - Combination Operations (Fetch, Modify, Insert

7.2 Hierarchy

- Table Structure
  - ♦ Flat Rows (i.e., Single Valued Columns)
- Relationships
  - ♦ Owner-to-members
  - ♦ Members-to-owner
- Operations
  - ♦ Record (Find, Store, Delete)
  - ♦ Relationship (Get Owner, Get Member, Get next)
  - ♦ Combination (Modify, Delete Tree)

7.3 Independent Logical File

- Table Structure
  - ♦ Single valued columns and/or
  - ♦ Two Level Structures (Sometimes More)
- Relationships (Value Based)
  - ♦ One to Many
  - ♦ One to One
  - ♦ Many to Many
  - ♦ Inferential
- Operations

- Table (Find, Add, Delete
- Relationship (Connect....Then Find...)
- Combination (Modify)

7.4 Relational

- Row Structure
  - Flat Rows
- Relationships
  - Value Based
- Operations
  - Row
    - Add, Modify, Delete
  - Relationship
    - Select, Project, Product
    - Union, Intersection, Difference

- Join,divide

- Referential Integrity

7.5 Data Model Components Review

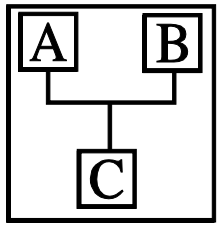
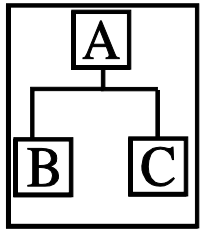
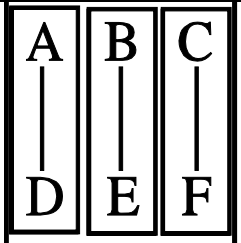
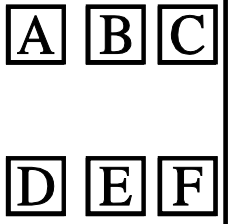
Data Model Type	Data Definition Language Components	Data Manipulation Language Components

Static	Row Organization and Relationships	Operations
Dynamic	Row Organization	Relationships and Operations

Data Model Components	{ Row Organization } + { Inter-row Relationships } + { Operations }
Data Model Languages	{Data Definition Language } + { Data Manipulation Language }
Dynamic Data Model	{ DDL { RO} } + { DML { REL + OPS } }
Static Data Model	{ DDL {RO + REL} } + { DML { OPS } }

WHERE:

DDL: Data Definition Model
DML: Data Manipulation Language
RO: Row Organization
REL: Relationships
OPS: Operations

Characteristics		Data Model: Network	Hierarchy	Independant Logical File	Relational
Record Organization		SV, MV, MD, G, RG	SV segments	SV, MV, RG	SV
Relationship (REL)					
Operations (OPS)	Record	A, D, F, M	A, D, F, M	A, D, F, M	A, D, F, M, P
	Relation-ships	C, D, GO, GM, GN	GO, GM, GN	J, INT DIV, UN DIF	J, INT, DIV, UN PR, DIV
DDL		REL, RO	REL, RO	RO	RO
DML		OPS	OPS	REL & OPS	REL & OPS

Record Organization	SV = Single Value		
	MV = Multiple Value		
	MD = Multiple Dimension		
	G = Group		
Operations	Record	A = Add	
		D = Delete	
		M = Modify	
		P = Project	
		F = Find	
	Relation-ship	C = Connect	Dis = Disconnect
		P = Project	J = Join
		DIV = Divide	GO = Get Owner
		GM = Get Member	GN = Get Next
		INT = Intersection	PR = Product
		UN = Union	DIF = Difference

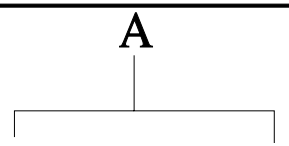
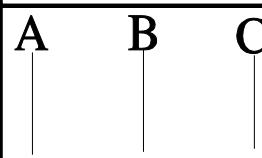
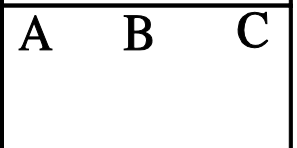
DATA MODEL				
Relationship Type	DDL Declaration		DML Simulation	
	ANSI Network	Hierarchy	ILF	ANSI-SQL Relational
Owner 1 Member	DS	DS	DD	DD
Owner >1 Member	DS	NO	NO	NO
Owner 1 Member	DS	NO	NO	NO
Owner >1 Member	DS	NO	NO	NO
Recursive	DS	ID	DD	ID
Many to Many	IS	ID	DD	ID
Inferential	ID	NO	DD	DD
One to One	DS	ID	DD	DD

DS means direct, static relationships
IS means indirect, static relationships
DD means direct, dynamic relationships
ID means indirect, dynamic relationships
No means no practical method

Data Model Effects on Application

Characteristic	Data model			
	Network	**Hierarchy**	**ILF**	**Relational**
Structure	Complex	Moderate	Complex	Simple
Record Order Control	Update	Update Retrieval	Retrieval	Run-Unit
Relationship Binding	Load or Update	Load or Update	Only at Retrieval	Only at Retrieval
Mandatory Design Work	High	Moderate	Low	Lowest
DBMS Control Over Programmer	High	Moderate	Low	Lowest
Ease of Database Design Change	Poor	Less Poor	Better	Best
Effort to Design a Quality Data Database	Great	Great	Great	Great

Static and Dynamic Data Model Summary

DBMS TYPE			
STATIC		DYNAMIC	
NETWORK	HIERARCHICAL	ILF	RELATIONAL
			
Supra IDMS/R* IDS* DMS-2 DMS 1100 IMAGE VAX-DBMS* PRIME-DB*	IMS System 2000	Manage Model 204 Inquire Adabas Nomad Focus Ramis DMS 170 Datacom/DB	Ingres* Oracle* DB/2 Sybase Informix SQL Server
*ANSI-NDL			*ANSI-SQL/1986, 1989, and 1992
		*ANSI-SQL/1999 and 2003