Chapter 3: The Relational Data Model and Relational Database Constraints

CS-6360 Database Design

Chris Irwin Davis, Ph.D.

Email: chrisirwindavis@utdallas.edu
Phone: (972) 883-3574
Office: ECSS 4.705

Chapter 3 Outline

- 3.1 – The Relational Data Model and Relational Database Constraints
- 3.2 – Relational Model Constraints and Relational Database Schemas
- 3.3 – Update Operations, Transactions, and Dealing with Constraint Violations

3.1 – The Relational Data Model and Relational Database Constraints

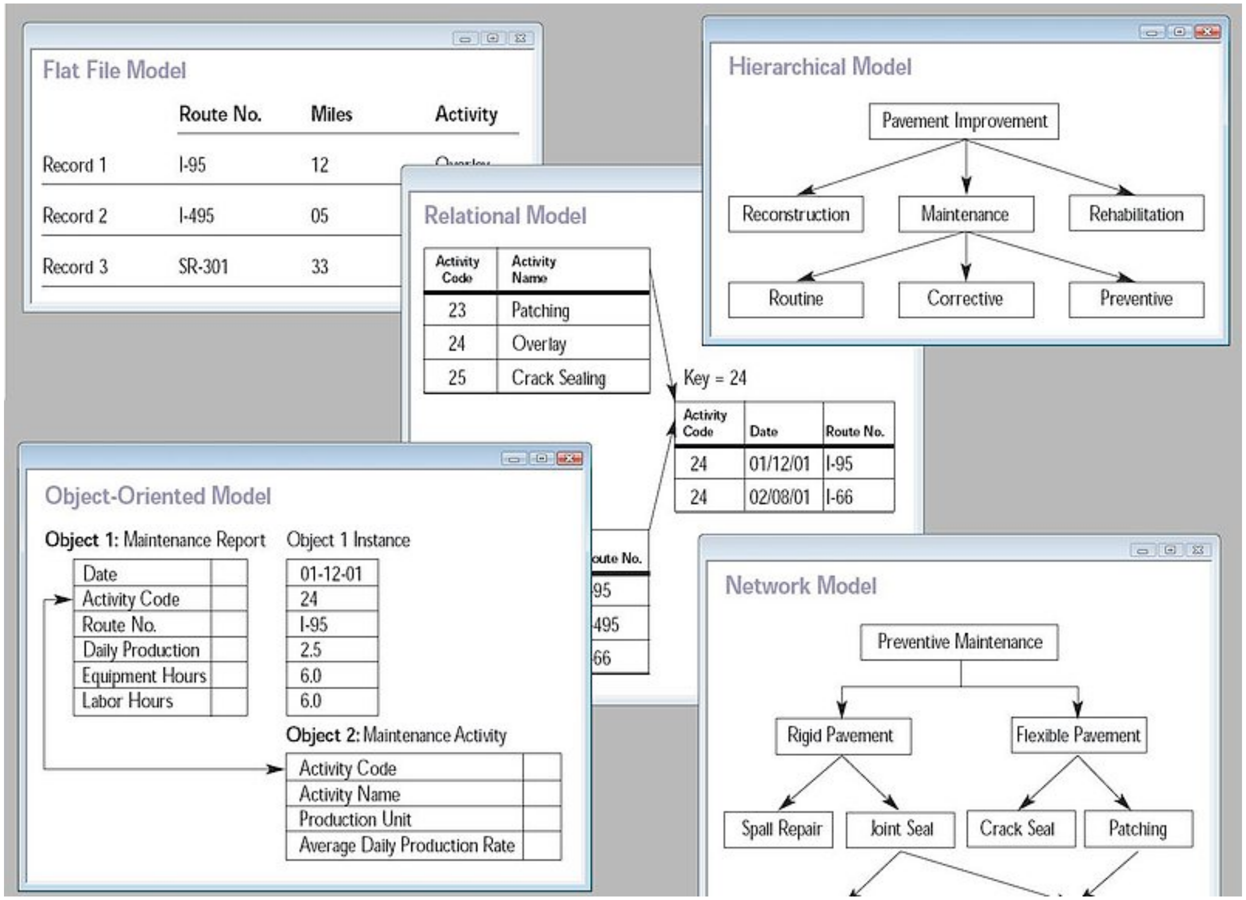
The Relational Data Model

- **Relational model**
  - Based on Relational Algebra
  - First commercial implementations available in early 1980s
  - Has been implemented in a large number of commercial system

Other Database Models

- Physical Data Models
  - Flat File Model
  - Inverted Index Model
- Logical Data Models
  - Network Model
  - Hierarchical Model
  - Object-oriented Model
- Other Models
  - XML Database

Database Models



Relational Model Concepts

- Represents data as a collection of relations
  - i.e. Relational Algebra “relations” (Set Theory)
- Table of values
  - Row
    - Represents a collection of related data values
    - Fact that typically corresponds to a real-world entity or relationship
    - Tuple
  - Table name and Column names
    - Interpret the meaning of the values in each row attribute

Relational Model Concepts (cont'd.)

Relation Name

Attributes

STUDENT

Tuples

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 3.1
The attributes and tuples of a relation STUDENT.

Relational Model Concepts (cont'd.)

Table Name
Relation Name

Table Columns
Attributes

Table Rows
Tuples

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 3.1
The attributes and tuples of a relation STUDENT.

Database Implementation Names

Domains, Attributes, Tuples, and Relations

- Domain D
  - Set of valid atomic values
- **Atomic**
  - Each value indivisible
- Specifying a domain
  - Data type specified for each domain

Examples of Domains

- Usa_phone_numbers: The set of ten-digit phone numbers valid in the U.S.
- Social_security_numbers: The set of valid nine-digit Social Security numbers.
- Names: The set of character strings that represent names of persons.
- Grade_point_averages: The set of possible values of computed grade point averages; must be a real number 0 to 4.
- Employee_ages: The set of possible ages of employees in a

Domains, Attributes, Tuples, and Relations (cont'd.)

- Relation schema R
  - Denoted by R(A1, A2, ..., An)
  - Made up of a relation name R and a list of attributes, A1, A2, ..., An
- Attribute Ai
  - Name of a role played by some domain D in the relation schema R
- Degree (or arity) of a relation
  - Number of attributes n of its relation schema

Relation Schema Example

- Relation with arity degree 7
  - STUDENT(Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa)
- Using Data Types, Relations sometimes written as…
  - STUDENT(Name: string, Ssn: string, Home_phone: string, Address: string, Office_phone: string, Age: integer, Gpa: real)

Domains, Attributes, Tuples, and Relations (cont'd.)

- Relation (or relation state)
  - Set of n-tuples $r = \{t_1, t_2, \ldots , t_m\}$
  - Each n-tuple t
    - Ordered list of n values $t = \langle v_1, v_2, \ldots , v_n \rangle$
    - Each value v_i, $1 \leq i \leq n$, is an element of $dom(A_i)$ or is a special NULL value
  - Based on a Relation Schema

Domains, Attributes, Tuples, and Relations (cont'd.)

- Relation (or relation state) r(R)
  - Mathematical (i.e. Set Theory) relation of degree n on the domains dom(A1), dom(A_2), … , $dom(A_n)$
  - Subset of the Cartesian product of the domains that define R:
    - $r(R) \subseteq (dom(A_1) \times dom(A_2) \times ... \times dom(A_n))$

Domains, Attributes, Tuples, and Relations (cont'd.)

- **Current relation state**
  - Relation state at a given time
  - Reflects only the valid tuples that represent a particular state of the real world
- **Attribute names**
  - Indicate different roles, or interpretations, for the domain

Characteristics of Relations

- Ordering of tuples in a relation… NOPE!
  - Indices have an order, relations do not
  - Relation defined as a set of tuples
  - Set elements (members) have no order among them
- Ordering of values within a tuple and an alternative definition of a relation
  - Order of attributes and values is not that important
  - As long as correspondence between attributes and values maintained
  - Default attribute order

Characteristics of Relations (cont'd.)

Figure 3.2
The relation STUDENT from Figure 3.1 with a different order of tuples.

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21

Characteristics of Relations (cont'd.)

- Alternative definition of a relation
  - Tuple considered as a set of (⟨attribute⟩, ⟨value⟩) pairs
  - Each pair gives the value of the mapping from an attribute A_i to a value v_i from $dom(A_i)$
  - Order is explicitly not important
- However, we use the first definition of relation by convention
  - Attributes and the values within tuples are ordered
  - Simpler notation

Attribute-Value Pairs

Figure 3.3
Two identical tuples when the order of attributes and values is not part of relation definition.

t = < (Name, Dick Davidson),(Ssn, 422-11-2320),(Home_phone, NULL),(Address, 3452 Elgin Road),
(Office_phone, (817)749-1253),(Age, 25),(Gpa, 3.53)>

t = < (Address, 3452 Elgin Road),(Name, Dick Davidson),(Ssn, 422-11-2320),(Age, 25),
(Office_phone, (817)749-1253),(Gpa, 3.53),(Home_phone, NULL)>

Characteristics of Relations (cont'd.)

- Values and NULLs in tuples
  - Each value in a tuple is atomic. This is called…
  - **Flat relational model**
    - Composite and multivalued attributes not allowed
    - First normal form assumption
  - Multivalued attributes
    - Must be represented by separate relations
  - Composite attributes
    - Represented only by simple component attributes in basic relational model

Characteristics of Relations (cont'd.)

- NULL values
  - Represent the values of attributes that may be unknown or may not apply to a tuple
  - Meanings for NULL values
    - Value unknown
    - Value exists but is not available
    - Attribute does not apply to this tuple (also known as value undefined)

Characteristics of Relations (cont'd.)

- Two alternative interpretations (meanings) of a relation
  - **Declarative Assertion (of facts)**
    - Each tuple in the relation is a fact or a particular instance of the assertion. For example, the schema of the STUDENT relation of Figure 3.1 asserts that, in general, a student entity has a Name, Ssn, Home_phone, Address, Office_phone, Age, and Gpa.
    - Each tuple in the relation can then be interpreted as a fact or a particular instance of the assertion. For example, the first tuple in Figure 3.1 asserts the fact that there is a STUDENT whose Name is Benjamin Bayer, Ssn is 305-61-2435, Age is 19, and so on.
    - Notice that some relations may represent facts about entities, whereas other relations may represent facts about relationships.

Characteristics of Relations (cont'd.)

- Two alternative interpretations (meanings) of a relation
  - **Logical Predicate**
    - Values in each tuple interpreted as values that satisfy the predicate
    - Predicate Logic, First Order Logic (FOL)
    - Relation name is the predicate, which has relation arity (i.e. degree) number of parameters
    - For example, the predicate STUDENT (Name, Ssn, ...) is true for the five tuples in relation STUDENT of Figure 3.1. These tuples represent five different propositions or facts in the real world.
  - **Triple Store?**

Relational Model Notation

- Relation schema R of degree n
  - Denoted by $R(A_1, A_2, \ldots, A_n)$
- Uppercase letters Q, R, S
  - Denote relation names
- Lowercase letters q, r, s
  - Denote relation states
- Letters t, u, v
  - Denote tuples

Relational Model Notation

- Name of a relation schema: STUDENT
  - Indicates the current set of tuples in that relation
- Notation: STUDENT(Name, Ssn, ...)
  - Refers only to relation schema
- Attribute A can be qualified with the relation name R to which it belongs
  - Using the dot notation R.A
  - e.g. STUDENT.Name, STUDENT.Ssn, etc.

Relational Model Notation

- n-tuple t in a relation r(R)
  - Denoted by $t = <v_1, v_2, \ldots, v_n>$
  - vi is the value corresponding to attribute A_i
- Component values of tuples:
  - $t[A_i]$ and $t.A_i$ refer to the value vi in n-tuple t for attribute A_i
  - $t[A_u, A_w, \ldots, A_z]$ and $t.(A_u, A_w, \ldots, A_z)$ refer to the sub-tuple of values $<v_u, v_w, \ldots, v_z>$ from t corresponding to the attributes specified in the list

3.2 – Relational Model Constraints and Relational Database Schemas

Relational Model Constraints

- Constraints
  - Restrictions on the actual values in a database state
  - Derived from the rules in the mini-world that the database represents

Relational Model Constraints (cont'd.)

- Inherent model-based constraints or implicit constraints
  - Inherent in the data model
- Schema-based constraints or explicit constraints
  - Can be directly expressed in schemas of the data model by specifying them in the DDL
- Application-based or semantic constraints or business rules
  - Cannot be directly expressed in schemas
  - Expressed and enforced by application program

Domain Constraints

- Typically include (but not limited to):
  - Numeric data types for integers and real numbers
  - Characters
  - Booleans
  - Fixed-length strings
  - Variable-length strings
  - Date, time, timestamp
  - Money
  - Other special data types

Key Constraints and Constraints on NULL Values

- No two tuples can have the same combination of values for all their attributes.
- **Superkey**
  - No two distinct tuples in any state r of R can have the same value for SK
  - Each whole tuple is a superkey
- **Key**
  - Superkey of R
  - Removing any attribute A from key K leaves a set of attributes K' that is not a superkey of R any more

Key Constraints and Constraints on NULL Values

- A Key satisfies two properties:
  - **Uniqueness**
    - Two distinct tuples in any state of relation cannot have identical values for (all) attributes in key
  - **Minimal superkey**
    - Cannot remove any attributes and still have uniqueness constraint in above condition hold

Key Constraints and Constraints on NULL Values

- **Candidate key**
  - Relation schema may have more than one key
- Primary key of the relation
  - Designated among candidate keys
  - Underline attribute
- Other candidate keys are designated as unique keys (even though keys are unique by definition)

Key Clarifications

- A key uniquely identifies a single record. May be more than one key. Each key is…
- A candidate key. Each candidate key may be a different number of attributes (key arity).
- A superkey also uniquely identifies a single record, but may contain more information than necessary. For example, SSN is enough to identify a person, but a superkey may have gender, as well.
- A minimal superkey is the smallest key (that is, number of fields) that uniquely identifies a record

Key Constraints and Constraints on NULL Values

CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

Figure 3.4
The CAR relation, with two candidate keys: License_number and Engine_serial_number.

Relational Databases and Relational Database

- Relational database schema S
  - Set of relation schemas S = {R1, R2, ..., Rm}
  - Set of integrity constraints IC
- **Relational database state**
  - Set of relation states DB = {r1, r2, ..., rm}
  - Each ri is a state of Ri and such that the ri relation states satisfy all integrity constraints specified in the set IC

Relational Databases and Relational Database

- **Invalid state**
  - Does not obey all the integrity constraints
  - i.e. violates at least one integrity constraint
- **Valid state**
  - Satisfies all the constraints in the defined set of Integrity Constraints (IC)

Integrity, Referential Integrity,

- **Entity integrity constraint**
  - No primary key value can be NULL

Integrity, Referential Integrity,

- **Entity integrity constraint**
  - No primary key value can be NULL
- Referential integrity constraint
  - Specified between two relations
  - Maintains consistency among tuples in two relations

Integrity, Referential Integrity,

- Foreign key rules to maintain referential integrity:
  - The attributes in FK have the same domain(s) as the primary key attributes PK
  - Value of FK in a tuple t1 of the current state r1(R1) either occurs as a value of PK for some tuple t2 in the current state r2(R2) or is NULL

Integrity, Referential Integrity,

- Managing Foreign Keys
  - Diagrammatically display referential integrity constraints
    - Directed arc from each foreign key to the relation it references
  - All integrity constraints should be specified on relational database schema

Other Types of Constraints

- **Semantic integrity constraints**
  - May have to be specified and enforced on a relational database
  - Use triggers and assertions
  - "More common to check for these types of constraints within the application programs" – Elmasri
    - Bad Idea™ to generalize – Davis

Other Types of Constraints (cont'd.)

- **Functional dependency constraint**
  - Establishes a functional relationship among two sets of attributes X and Y
  - Value of X determines a unique value of Y

Other Types of Constraints (cont'd.)

- **Functional dependency constraint**
  - Establishes a functional relationship among two sets of attributes X and Y
  - Value of X determines a unique value of Y
- State constraints
  - Define the constraints that a valid state of the database must satisfy

Other Types of Constraints (cont'd.)

- **Functional dependency constraint**
  - Establishes a functional relationship among two sets of attributes X and Y
  - Value of X determines a unique value of Y
- State constraints
  - Define the constraints that a valid state of the database must satisfy
- **Transition constraints**
  - Define to deal with state changes in the database

3.3 – Update Operations, Transactions, and Dealing with Constraint Violations

Update Operations, Transactions, and Dealing with

- Operations of the relational model can be categorized into Retrievals and Modifications
- Retrieval
  - Query
- Basic modification operations that change the states of relations in the database:
  - Insert
  - Delete
  - Update

The Insert Operation

- Provides a list of attribute values for a new tuple t that is to be inserted into a relation R

The Insert Operation

- Provides a list of attribute values for a new tuple t that is to be inserted into a relation R
- Can violate any of the four types of constraints
  - Domain Constraints
  - Key Constraints
  - Entity Integrity
  - Referential Integrity
- If an insertion violates one or more constraints
  - Default option is to reject the insertion
  - Other options?

The Delete Operation

- Can violate only Referential Integrity
- If tuple being deleted is referenced by foreign keys from other tuples
- Options
  - **Restrict – Reject the deletion**
  - **Cascade – Propagate the deletion by deleting tuples that reference the tuple that is being deleted**
  - Set null or set default – Modify the referencing attribute values that cause the violation

The Update Operation

- Necessary to specify a condition on attributes of relation
  - Select the tuple (or tuples) to be modified
- If attribute not part of a primary key nor of a foreign key
  - Usually causes no problems
- Updating a primary/foreign key
  - Similar issues as with Insert/Delete

Example: COMPANY schema

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 3.5
Schema diagram for the COMPANY relational database schema.

Example: COMPANY Referential Constraints

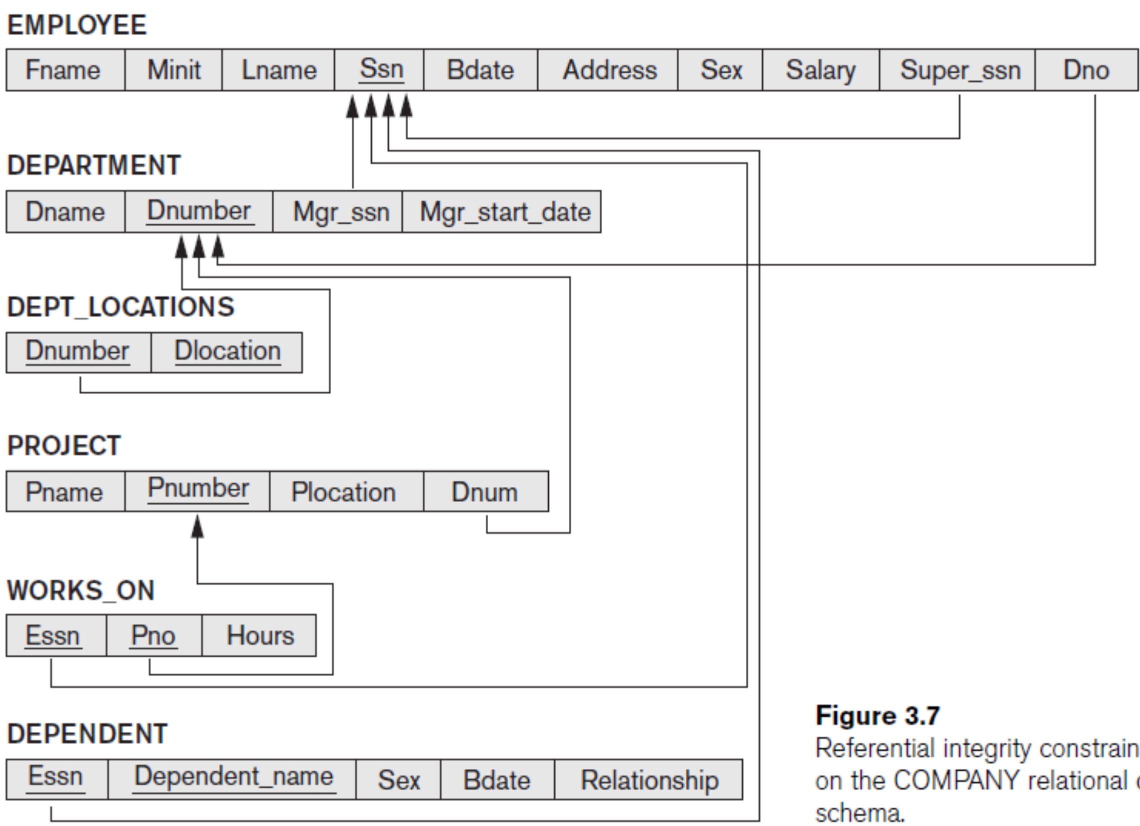


Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

Example: COMPANY Relations

Figure 3.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Example: COMPANY Relations (cont'd)

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Chapter 3 Summary

- Characteristics differentiate relations from ordinary tables or files
- Classify database constraints into:
  - Inherent model-based constraints,
  - Explicit schema-based constraints, and
  - Application-based constraints
- Modification operations on the relational model:
  - Insert, Delete, and Update