Chapter 6

The Relational Data Model and Relational Database Constraints (from E&N and my editing)

Chapter Outline

- Relational Model Concepts
- Relational Model Constraints and
- Relational Database Schemas
- Update Operations and Dealing with Constraint Violations

Relational Model Concepts

- The model was first proposed by Dr. E.F. Codd of IBM in 1970 in the following paper: "A Relational Model for Large Shared Data Banks," Communications of the ACM, June 1970.

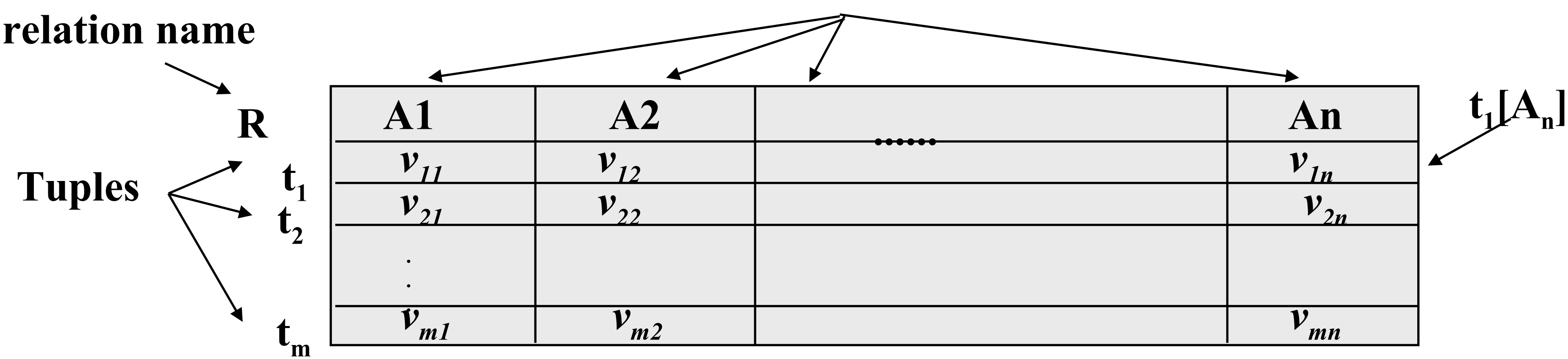
Relational Model Concepts

- The relational Model of Data is based on the concept of a Relation.
- A Relation is a mathematical concept based on the ideas of sets.
- The strength of the relational approach to data management comes from the formal foundation provided by the theory of relations.

Relations

- Relational DBMS products store data in the form of relations, a special type of table
- A relation is a two-dimensional table that has the following characteristics
  - Rows contain data about an entity
  - Columns contain data about attributes of the entity
  - Cells of the table hold a single value
  - All entries in a column are of the same kind
  - Each column has a unique name
  - The order of the columns is unimportant – concept of set
  - The order of the rows is unimportant
  - No two rows may be identical

- Set of Tuples and Typically Shown as a Table With Columns and Rows.
- Column (Field) Represents an Attribute
- Row (Tuple) Represents an Entity Instance



- Although not all tables are relations, the terms table and relation are normally used interchangeably

Relational Model	Programmer	User
Relation	File	Table
Tuple (Row)	Record	Row
Attribute	Field	Column

Ex: Relation

Figure 4.2 Sample Relation

EmployeeNumber	FirstName	LastName	Department	Email	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	236-9987
200	Mary	Abernathy	Finance	MA@somewhere.com	444-8898
300	Liz	Smathers	Finance	LS@somewhere.com	777-0098
400	Tom	Caruthers	Accounting	TC@somewhere.com	236-9987
500	Tom	Jackson	Production	TJ@somewhere.com	444-9980
600	Eleanore	Caldera	Legal	EC@somewhere.com	767-0900
700	Richard	Bandalone	Legal	RB@somewhere.com	767-0900

Ex: “just” Table

Figure 4.3b Tables but Not Relations — Multiple Entries per Cell

EmployeeNumber	FirstName	LastName	Department	Email	Phone
100	Jerry	Johnson	Accounting	JJ@somewhere.com	236-9987
200	Mary	Abernathy	Finance	MA@somewhere.com	444-8898
300	Liz	Smathers	Finance	LS@somewhere.com	777-0098
400	Tom	Caruthers	Accounting	TC@somewhere.com	236-9987
				Fax:	266-9987
				Home:	555-7171
500	Tom	Jackson	Production	TJ@somewhere.com	444-9980
600	Eleanore	Caldera	Legal	EC@somewhere.com	767-0900
				Fax:	236-9987
				Home:	555-7171
700	Richard	Bandalone	Legal	RB@somewhere.com	767-0900

(b)

- Are the Following Relations in a Relational Model? Why?

R1

A	B	C	D
a2	{b1, b2}	c1	d5
a2	b7	c9	d5
a2	b23	c22	d1
……			

R2

A	B	C	D
a2	b2	c6	d1
a2	b7	c9	d5
a2	b7	c9	d5
……			

Employee

E#	Ename	AGE	ADDRESS
E2	Diamond	45	1888 Buford Hyw.
E1	Smith	30	3302 Peachtree Rd., Atlanta, GA
E3	Evan		Baker Ct. Atlanta

FORMAL DEFINITIONS

- A **Relation** may be defined in multiple ways.
- The **Schema** of a Relation: *R* (A1, A2, ...An)

  Relation schema *R* is defined over **attributes** A1, A2, ...An

  For Example -

  CUSTOMER (Cust-id, Cust-name, Address, Phone#)

  Here, CUSTOMER is a relation defined over the four attributes Cust-id, Cust-name, Address, Phone#, each of which has a **domain** or a set of valid values. For example, the domain of Cust-id is 6 digit numbers.

FORMAL DEFINITIONS

- A **tuple** is an ordered set of values
- Each value is derived from an appropriate domain.
- Each row in the CUSTOMER table may be referred to as a tuple in the table and would consist of four values.

  <632895, "John Smith", "101 Main St. Atlanta, GA 30332", "(404) 894-2000">

  is a tuple belonging to the CUSTOMER relation.
- A relation may be regarded as a ***set of tuples*** (rows).
- Columns in a table are also called attributes of the relation.

FORMAL DEFINITIONS

- A **domain** has a logical definition: e.g., "USA_phone_numbers" are the set of 10 digit phone numbers valid in the U.S.
- A domain may have a data-type or a format defined for it. The USA_phone_numbers may have a format: (ddd)-ddd-dddd where each d is a decimal digit. E.g., Dates have various formats such as monthname, date, year or yyyy-mm-dd, or dd mm,yyyy etc.
- An attribute designates the **role** played by the domain. E.g., the domain Date may be used to define attributes "Invoice-date" and "Payment-date".

FORMAL DEFINITIONS

- The relation is formed over the cartesian product of the sets; each set has values from a domain; that domain is used in a specific role which is conveyed by the attribute name.
- For example, attribute Cust-name is defined over the domain of strings of 25 characters. The role these strings play in the CUSTOMER relation is that of the name of customers.
- Formally,

  Given $R(A_1, A_2, \ldots\ldots\ldots, A_n)$

  $r(R) \subset dom(A_1) \times dom(A_2) \times \ldots \times dom(A_n)$

  - R: schema of the relation
  - r of R: a specific "value" or population of R.
  - R is also called the **intension** of a relation
  - r is also called the **extension** of a relation

FORMAL DEFINITIONS

- Let S1 = {0,1}
- Let S2 = {a,b,c}

- Let R ⊂ S1 X S2

- Then for example: r(R) = {<0,a> , <0,b> , <1,c> } is one possible "state" or "population" or "extension" r of the relation R, defined over domains S1 and S2. It has three tuples.

DEFINITION SUMMARY

Informal Terms	Formal Terms
Table	Relation
Column	Attribute/Domain
Row	Tuple
Values in a column	Domain
Table Definition	Schema of a Relation
Populated Table	Extension

Example - Figure 5.1

Relation name: STUDENT; Attributes: Name, SSN, HomePhone, Address, OfficePhone, Age, GPA; Tuples: the five rows below.

STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Relation Schemes

- Example
  - EMP(ENO, ENAME, TITLE, SAL)
  - PROJ (PNO, PNAME, BUDGET)
  - WORKS(ENO, PNO, RESP, DUR)
- Underlined Attributes are Relation Keys which Uniquely Distinguish Among Tuples (Rows)
- Tabular Fo

EMP

ENO	ENAME	TITLE	SAL

PROJ

PNO	PNAME	BUDGET

WORKS

ENO	PNO	RESP	DUR

Relation Instances

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

WORKS

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E7	P5	Engineer	23
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000
P5	CAD/CAM	500000

- Exercise:
  - R(A, B) is a Relation Schema Defined over A and B
  - Let domain(A) = {a1, a2} and domain(B) = {0, 1, 2}
  - Which of the Following are Relations of R?
    - {(a1, 1), (a1, 2), (a2, 0)}
    - {(a1, 0), (a1, 1), (a1, 2)}
    - {(a1, 1), (a2, 2}, (a0, 0)}
    - {(a1, 1), (a2, a2}, (a0, a0)}
    - {(a1, 1, c1), (a2, 2)}
- What if Attribute A is a Key?

Characteristics of Attr

- Attribute Name
  - An Attribute Name Refers to a Position in a Tuple by Name Rather than Position
  - An Attribute Name Indicates the Role of a Domain in a Relation
  - Attribute Names must be Unique Within Relations
  - By Using Attribute Names we can Disregard the Ordering of Field Values in Tuples
- Attribute Value - Must have a Value
  - Must Be an Atomic Value
  - Can Be a Null Value Meaning “Not Known”, “Not Applicable” ...

Constraint in Database

- Inherent Constraint
  - Constraint that are inherent in the data model
  - Characteristics relation
- Schema-based constraint
  - Constraint that can be directly expressed in the schemas of the data model, typically specifying in DDL
  - Domain constraint, key constraint, etc
- Application-based constraint
  - Constraint that can not be directly expressed in the data model and must be expressed and enforced by the application program
  - Trigger, assertion, etc

CHARACTERISTICS OF RELATIONS

- Ordering of tuples in a relation r(R): The tuples are *not* considered to be ordered, even though they appear to be in the tabular form.
- Ordering of attributes in a relation schema R (and of values within each tuple): We will consider the attributes in R(A_1, A_2, ..., A_n) and the values in t=<v_1, v_2, ..., v_n> to be *ordered* .

  (However, a more general *alternative definition* of relation does not require this ordering).
- Values in a tuple: All values are considered *atomic* (indivisible). A special **null** value is used to represent values that are unknown or inapplicable to certain tuples.

CHARACTERISTICS OF RELATIONS

- Notation:

- We refer to **component values** of a tuple t by $t[A_i] = v_i$ (the value of attribute A_i for tuple t).

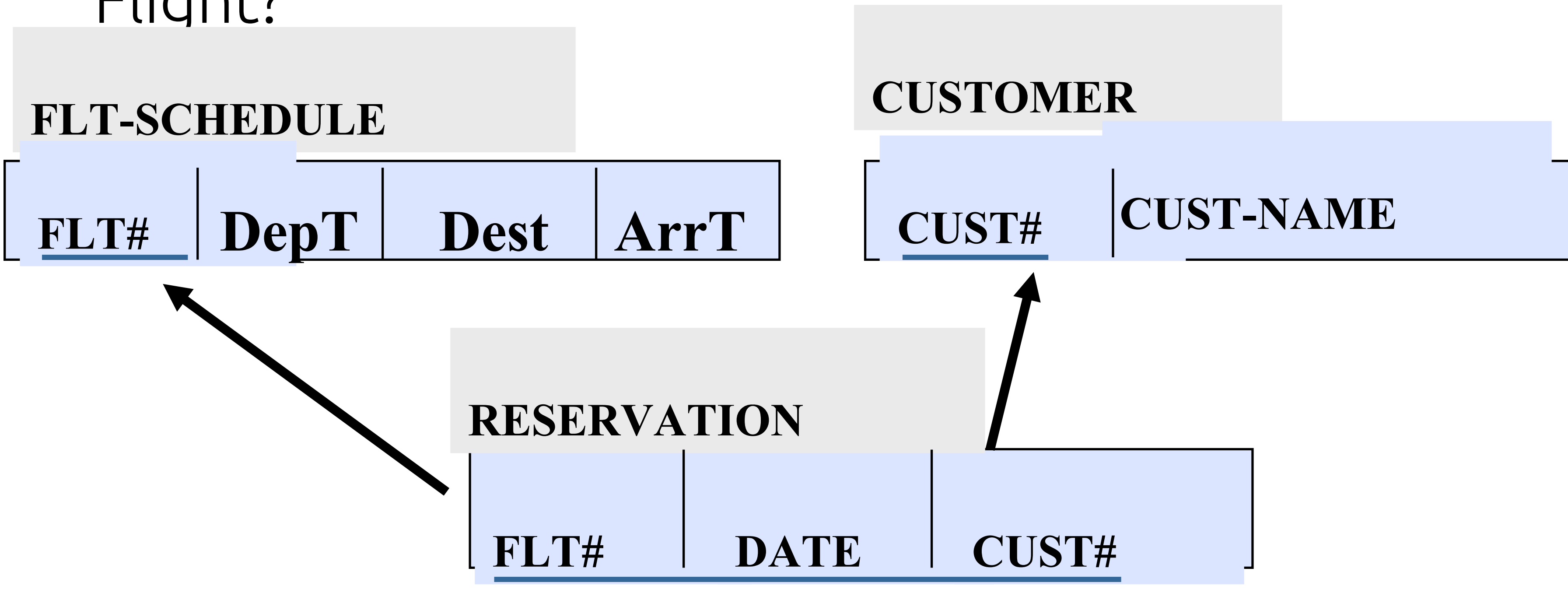
  Similarly, $t[A_u, A_v, ..., A_w]$ refers to the subtuple of t containing the values of attributes $A_u, A_v, ..., A_w$, respectively.

CHARACTERISTICS OF RELATIONS

STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21

Relational Integrity Constraints

- IC: Conditions that Must Hold on All Valid Relation Instances at Any Given Database State. Why are Integrity Constraints Needed? What Happens when we try to Delete a Flight?



Relational Integrity Constraints

- There are three main types of constraints:
  - Key constraints
  - Entity integrity constraints
  - Referential integrity constraints
- Other Types of Semantic Constraints:
  - Domain Constraints
  - Transition Constraints
  - Set Constraints
- DBMSs Handle Some But Not All Constraints

Types of Key

- A **key** is one or more columns of a relation that identifies a row
- **Composite key** is a key that contains two or more attributes
- A relation has one unique **primary key** and may also have additional unique keys called **candidate keys**
- Primary key is used to
  - Represent the table in relationships
  - Organize table storage
  - Generate indexes

Key Constraints

- Superkey (SK):
  - Any Subset of Attributes Whose Values are Guaranteed to Distinguish Among Tuples
- Candidate Key (CK):
  - A Superkey with a Minimal Set of Attributes (No Attribute Can Be Removed Without Destroying the Uniqueness -- Minimal Identity)
  - A Value of an Attribute or a Set of Attributes in a Relation That Uniquely Identifies a Tuple
  - There may be Multiple Candidate Keys

- Primary Key (PK):
  - Choose One From Candidate Keys
  - The Primary Key Attributed are Underlined
- Foreign Key (FK):
  - An Attribute or a Combination of Attributes (Say A) of Relation R1 Which Occurs as the Primary Key of another Relation R2 (Defined on the Same Domain)
  - Allows Linkages Between Relations that are Tracked and Establish Dependencies
  - Useful to Capture ER Relationships

Key Constraints

Figure 7.4 The CAR relation with two candidate keys: LicenseNun 5.4 ınd EngineSerialNumber.

CAR	LicenseNumber	EngineSerialNumber	Make	Model	Year
	Texas ABC-739	A69352	Ford	Mustang	96
	Florida TVP-347	B43696	Oldsmobile	Cutlass	99
	New York MPO-22	X83554	Oldsmobile	Delta	95
	California 432-TFY	C43742	Mercedes	190-D	93
	California RSK-629	Y82935	Toyota	Camry	98
	Texas RSK-629	U028365	Jaguar	XJS	98

- Example:
  - The CAR relation schema:

    CAR(State Reg#, SerialNo, Make, Model, Year)
- Its primary key is {State Reg#}
- It has two candidate keys
  - Key1 = {State Reg#}
  - Key2 = {SerialNo}
  - {SerialNo, Make} is a Superkey but not a Candidate Key

    Why?If Remove SerialNo, Make is not a Primary Key

Schema with Key

CAR(<u>License#</u>, EngineSerialNumber, Make, Model, Year)

CAR	LicenseNumber	EngineSerialNumber	Make	Model	Year
	Texas ABC-739	A69352	Ford	Mustang	96
	Florida TVP-347	B43696	Oldsmobile	Cutlass	99
	New York MPO-22	X83554	Oldsmobile	Delta	95
	California 432-TFY	C43742	Mercedes	190-D	93
	California RSK-629	Y82935	Toyota	Camry	98
	Texas RSK-629	U028365	Jaguar	XJS	98

What are Typically Used as Keys for Cars?

Complete Schema with Key

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

Keys Allow us to Establish Links Between Relations

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

What is This Similar to in ER?

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

Corresponding DB Table

- Which Represent Tuples/Instances of Each Relation

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John		Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin		Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia		Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer		Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh		Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce		English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad		Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James		Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
		Houston
		Stafford
		Bellaire
		Sugarland

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

Remaining DB Tables

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

Ex:

- Relational Schema PROJ(PNO, PNAME, BUDGET), we Assume that PNO is the Primary Key
- The Two Tables Below are Relations of PROJ
- Questions:
  - Is (PNO,PNAME) a Superkey in Either? Both?
  - Is PNAME a Candidate Key? Explain Your Answer.
  - Is (PNAME,BUDGET) a Superkey in Either? Both?

r1(PROJ)

PNO	PNAME	BUDGET
P11	Instrumentation	450000
P12	Database Develop.	145000
P13	CAD/CAM	150000
P14	Maintenance	450000
P15	Wireless Web	350000

r2(PROJ)

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000
P5	CAD/CAM	500000

Entity Integrity

- **Relational Database Schema**:
  - A Set *S* of Relation Schemas (R_1, R_2, ..., R_n) That Belong to the Same Database
  - *S* is the *Name* of the **Database**
  - *S* = {R_1, R_2, ..., R_n}
- Entity Integrity:
  - For Any R_i in *S*, Pk_i is the *Primary Key of* R
  - Attributes in Pk_i Cannot Have Null Values in any Tuple of $R(r_i)$
    - $T[Pk_i]$ **< >** Null for Any Tuple T in R(r)

Referential Integrity

- A Constraint Involving *Two* Relations Used to Specify a *Relationship* Among Tuples in
- Referencing Relation and Referenced Relation

- Definition: R_1 and R_2 have a Referential Integrity Constraint If
  - Tuples in the *Referencing Relation* R_1 have a Set of **Foreign Key (FK)** Attributes That Reference the Primary Key PK of the *Referenced Relation* R_2
  - A Tuple T_1 in $R_1(A_1, A_2, \ldots, A_n)$ is Said to **Reference** a Tuple T_2 in R_2 if $\exists\, FK \subseteq \{A_1, A_2, \ldots, A_n\}$ such that $T_1[fk] = T_2[pk]$

Ex:

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

WORKS

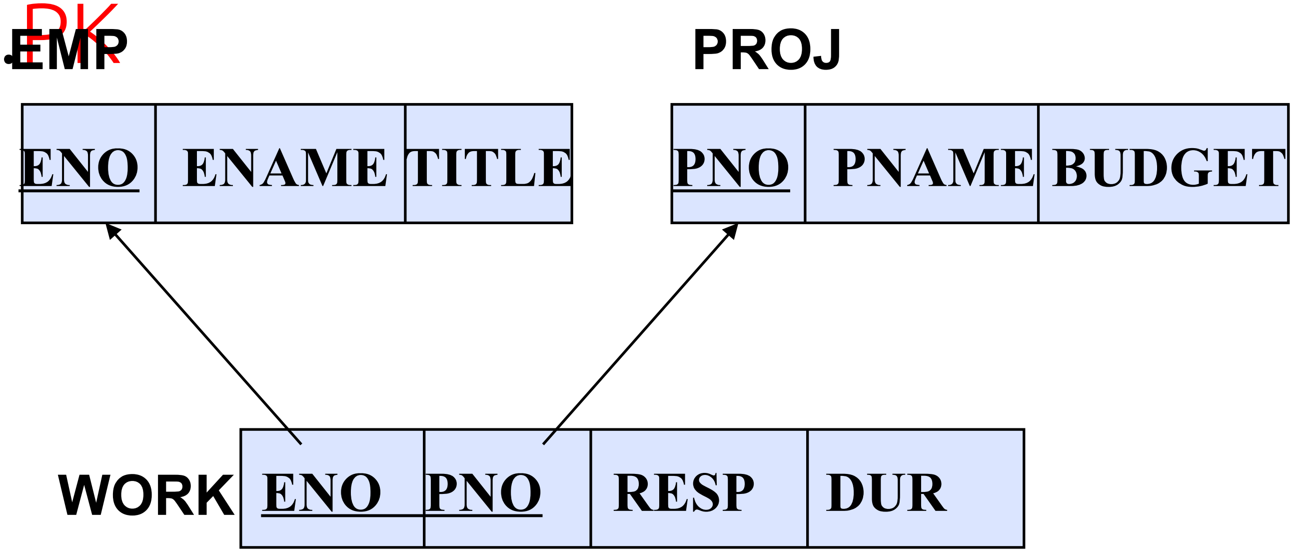
ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E7	P5	Engineer	23
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000
P5	CAD/CAM	500000

E9 P3 Engineer 30

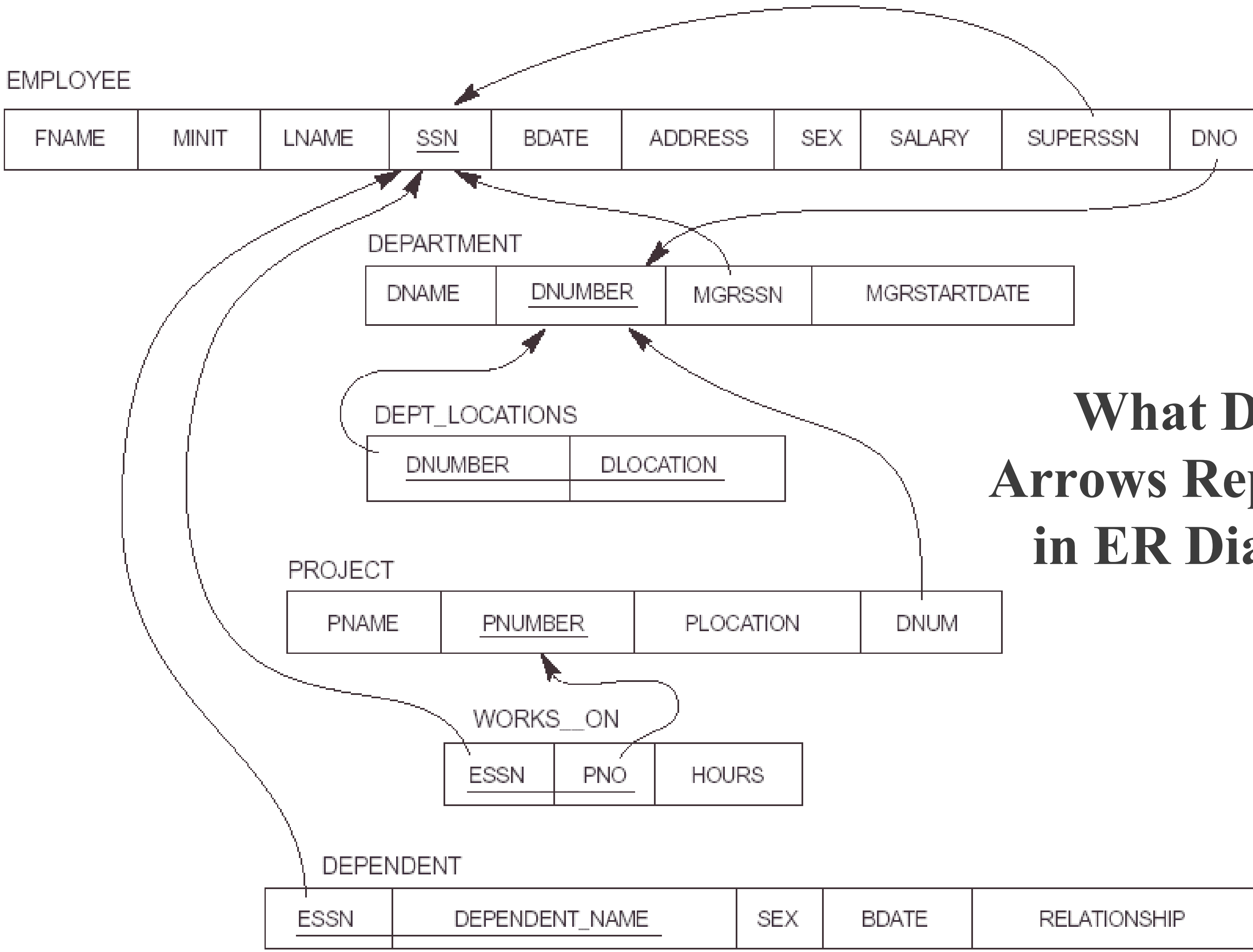
- A Referential Integrity Constraint Can Be Displayed in a Relational Database Schema as a Directed Arc From R_1.FK to R_2.PK



WORK[ENO] is a subset of EMP[ENO]

WORK[PNO] is a subset of PROJ[PNO]

Another one



What Do these Arrows Represent in ER Diagram?

Transition Integrity Constraint

- Can be defined to deal with state changes in the database
- Sometimes called dynamic constraints
- Example: “the salary of an employee can only increase”

Integrity Constraints Summary

- Relational Database: Set of Relations Satisfying the Integrity Constraints
- Integrity Constraints (ICs): Conditions that Must Hold on All Valid Relation Instances
  - Key Constraints - Uniqueness of Keys
  - Entity ICs - No Primary Key Value is Null
  - Referential ICs Between Two Relations, Cross References Must Point to Existing Tuples
  - Domain ICs are Limits on the Value of Particular Attribute
  - Transition ICs Indicate the Way Values Changes Due to Database Update

Operations on Relations

- A DBMS Operates via User Queries to Read and Change Data in a Database
- Changes Can be Inserting, Deleting, or Updating (Equivalent to a Delete followed by Insert)
- One Critical Issue in DB Operations is Integrity Constraints Maintenance in the Presence of
  - INSERTING a Tuple
  - DELETING a Tuple
  - UPDATING/MODIFYING a Tuple.

Problem Statements

- Integrity Constraints (ICs) Should Not Be Violated by Update Operations
- To Maintain ICs, Updates may Need to be Propagated and Cause Other Updates Automatically
  - Common Method: Group Several Update Operations Together As a Single Transaction
- If Integrity Violation, Several Actions Can Be Taken:
  - Cancel Operation that Caused Violation (REJECT)
  - Perform the Operation but Inform User of Violation
  - Trigger Additional Updates So the Violation is Corrected (CASCADE Option, SET NULL Option)
  - Execute a User-specified Error-Correction Routine

Update Operations on Relations

- INSERT a tuple.
- DELETE a tuple.
- MODIFY a tuple.

- Integrity constraints should not be violated by the update operations.
- Several update operations may have to be grouped together.
- Updates may *propagate* to cause other updates automatically. This may be necessary to maintain integrity constraints.

Inserting Operations

- Insert a Duplicate Key Violates Key Integrity:
  - Check If Duplicates Occur
- Insert a Null Key Violates Entity Integrity:
  - Check If Null is in Any Key
- Insert a Tuple Whose Foreign Key Attribute Pointing to an Non-existent Tuple Violates Referential Integrity:
  - Check the Existence of Referred Tuple

- Correction Actions:
  - Reject the Update
  - Correct the Violation - Change Null, Duplicate, Etc.
  - Cascade the Access - Insert a New Tuple That Did Not Exist

Ex:

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.

E6	L. Chu	
E3	R. Davis	Mech. Eng.

WORKS

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000
P5	CAD/CAM	500000

E1		Engineer	36
E1	P5	Engineer	
E8	P3	Manager	40

Deletion Operations

- Deleting a Tuple Referred to by Other Tuples in Database (via FKs) would Violate Referential Integrity
- Action:
  - Check for Incoming Pointers of the Deleted Tuple.
  - Group the Deletion and the Post-processing of the Referencing Pointers in a Single Transaction

- Three Options If Deletion Causes a Violation
  - Reject the Deletion
  - Attempt to Cascade (Propagate) the Deletion by Deleting the Tuples which Reference the Tuple being or to be Deleted
  - Modify the Referencing Attribute Values that Cause the Violation; Each Values is Set to Null or Changed to Reference to Another Valid Tuple

Ex:

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
		Elect. Eng.

WORKS

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P5	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48

Deleting this tuple?

1. Cascading

2. reference revision?

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000
P5	CAD/CAM	500000

Modify Operations

- Modify Operation Changes Values of One or More Attributes in a Tuple (or Tuples) of a Given Relation R
- Maintaining ICs Requires to Check If the Modifying Attributes Are Primary Key or Foreign Keys.

- Integrity Check Actions:
  - Case 1:
    - If the Attributes to be Modified are Neither a Primary Key nor a Foreign Key, Modify Causes No Problems
- Must Check and Confirm that the New Value is of Correct Data Type and Domain
  - Case 2:
    - Modifying a Primary Key Value Similar to Deleting One Tuple and Insert Another in its Place

Other Types of Constraints

Semantic Integrity Constraints:

- based on application semantics
- E.g., "the max. no. of hours per employee for all projects he or she works on is 56 hrs per week"
- A *constraint specification language* may have to be used to express these
- SQL-99 allows triggers and ASSERTIONS to allow for some of these

Figure 5.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.

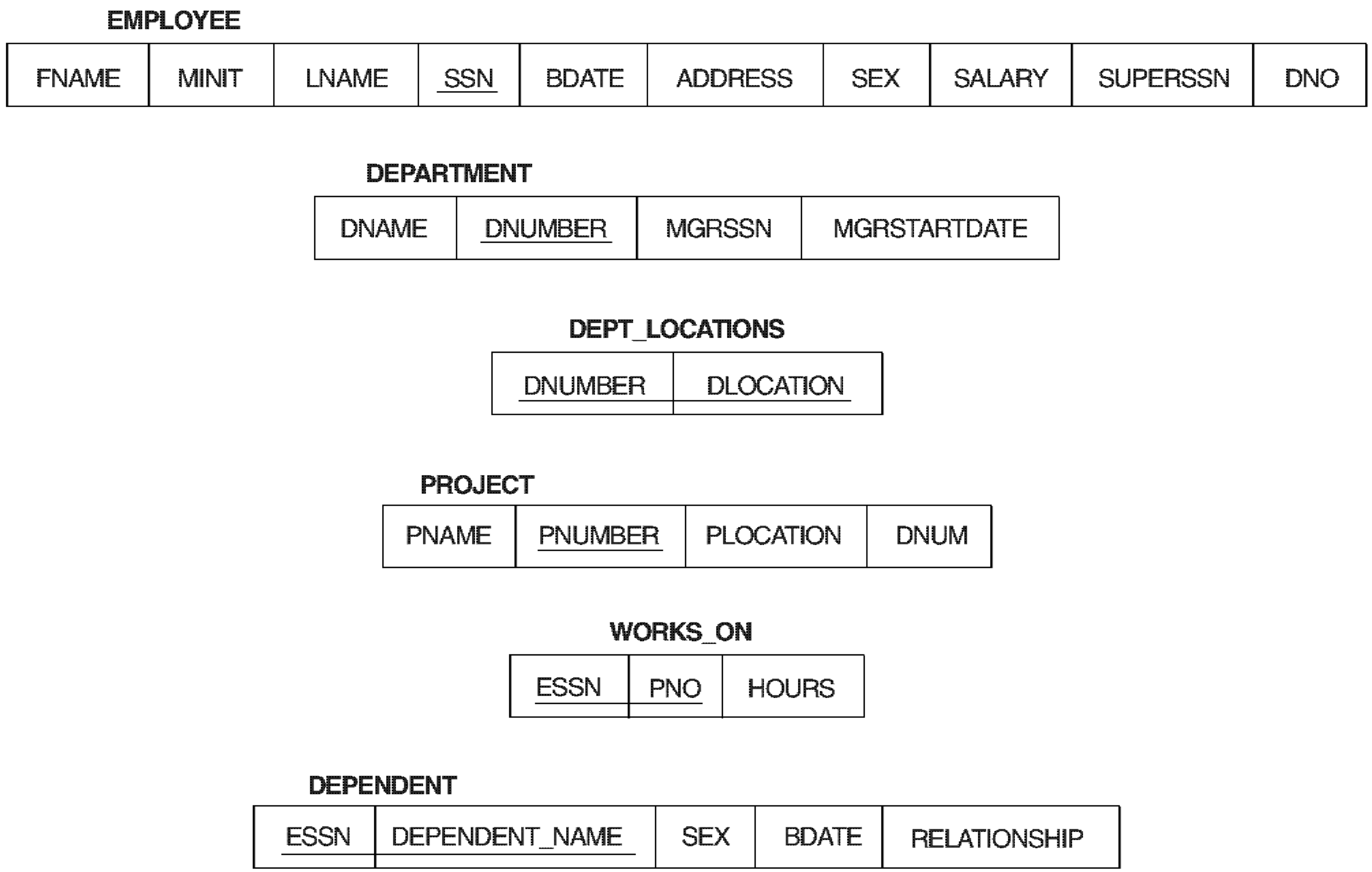


Figure 5.6 One possible relational database state corresponding to the COMPANY schema.

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John		Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin		Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia		Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer		Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh		Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce		English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad		Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James		Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
		Houston
		Stafford
		Bellaire
		Sugarland

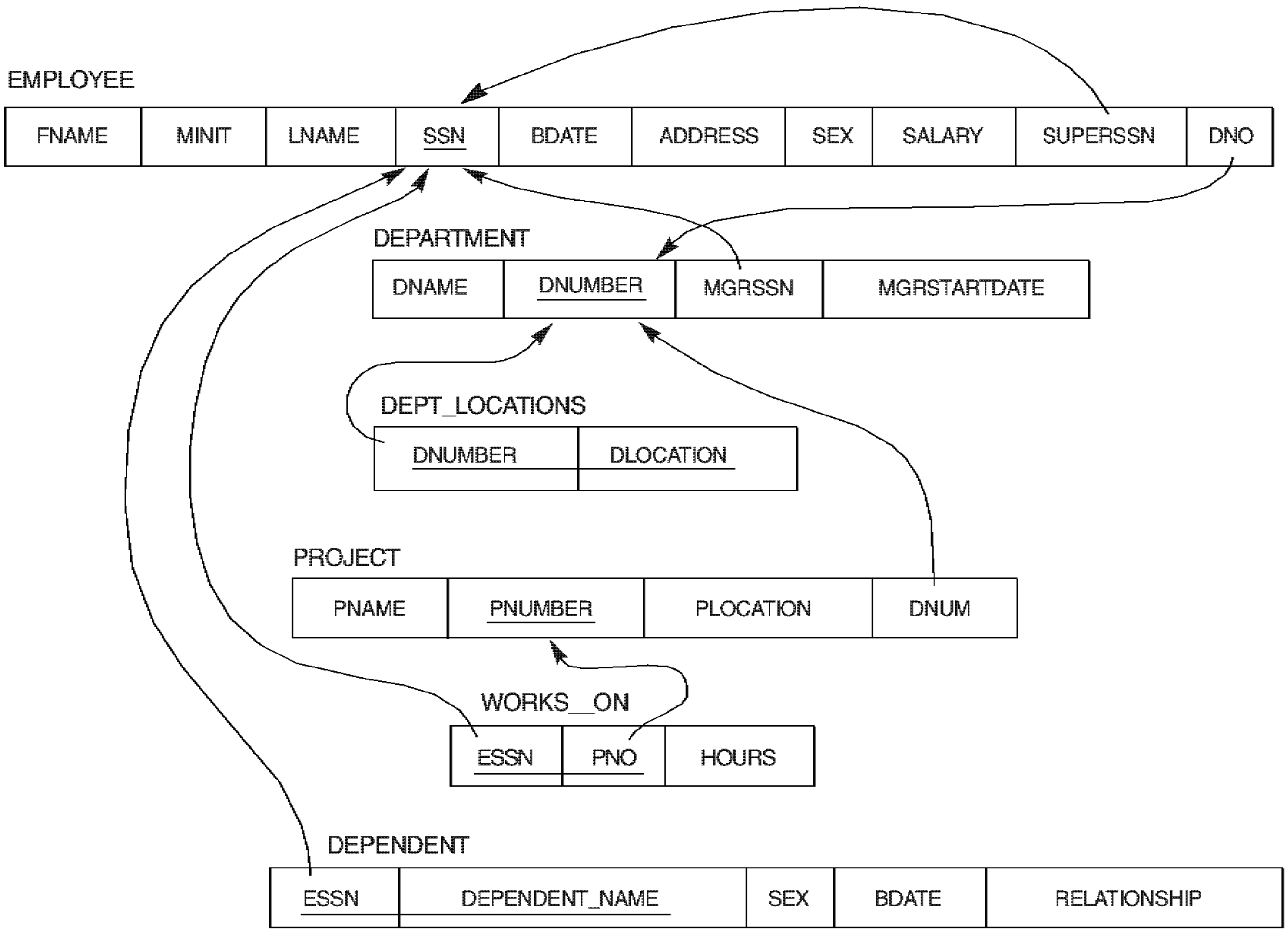
DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

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

Figure 5.7 Referential integrity constraints displayed on the COMPANY relational database schema diagram.



Relational Language

- A Relational Language
  - Defines Operations to Manipulate Relations
  - Used to Specify Retrieval Requests (Queries)
  - Query Result is Expressed in the Form of a Relation
- Classification
  - Relational Algebra
  - Relational Calculus
  - Structured Query Language