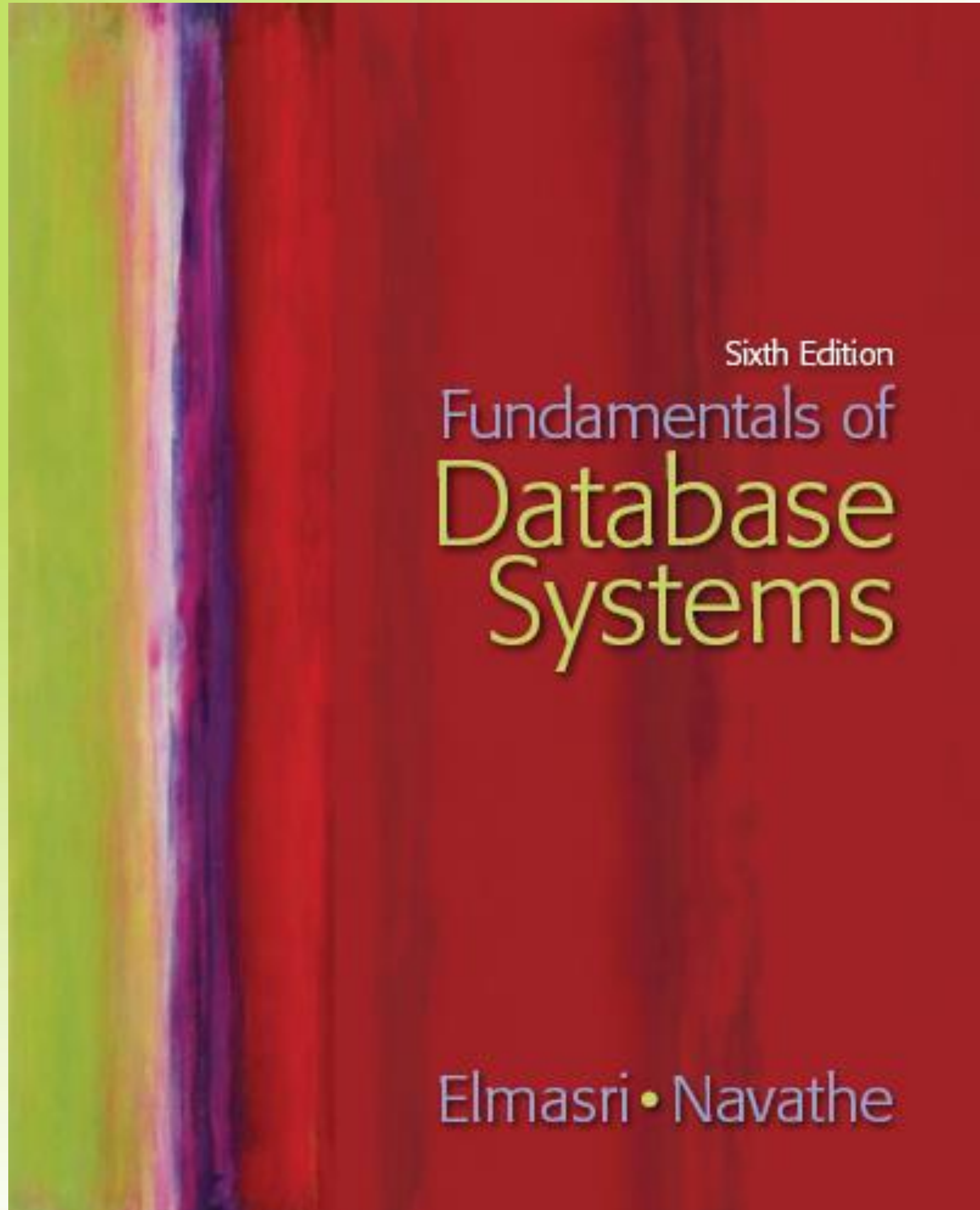
Chapter 3

The Relational Data Model and SQL



Addison-Wesley is an imprint of PEARSON

Chapter 3

The Relational Data Model and Relational Database Constraints

Addison-Wesley is an imprint of PEARSON

Chapter 3 Outline

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

Addison-Wesley is an imprint of PEARSON

Relational Model Concepts

- The relational Model of Data is based on the concept of a *Relation*
  – Has a formal mathematical foundation provided by set theory and first order predicate logic
- We review the essentials of the ***formal** relational model* in this chapter
- In *practice*, there is a *standard model* based on SQL (Structured Query Language) – described in Chapters 4 and 5
- <u>Note:</u> There are several important differences between the *formal* model and the *practical* model, as we shall see

Addison-Wesley is an imprint of PEARSON

Relational Model Origins

- The model was first proposed by Dr. E.F. Codd of IBM Research in 1970 in the following paper:
  - "A Relational Model for Large Shared Data Banks," Communications of the ACM, June 1970
- The above paper caused a major revolution in the field of database management
- Dr. Codd earned the coveted ACM Turing Award in 1981

Informal Definitions

- Informally, a **relation** looks like a **table** of values or a flat file of records (see Figure 3.1 on next slide).
- A relation contains a **set of rows**.
- The data elements in each **row** represent certain facts that correspond to a real-world **entity** or **relationship**
  - In the formal model, rows are called **tuples**
- Each **column** has a column header that gives an indication of the meaning of the data items in that column
  - In the formal model, the column header is called an **attribute name** (or just **attribute**)

Attributes

Relation Name

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

Tuples

Figure 3.1
The attributes and tuples of a relation STUDENT.

Informal Definitions

- Key of a Relation:
  - Each row (tuple) in the table is uniquely identified by the value of a particular attribute (or several attributes together)
    - Called the *key* of the relation
  - In the STUDENT relation, SSN is the key
  - If no attributes posses this uniqueness property, a new attribute can be added to the relation to assign unique row-id values (e.g. unique sequential numbers) to identify the rows in a relation
    - Called *artificial key* or *surrogate key*

Formal Definitions – Relation Schema

- **Relation Schema** (or description) of a Relation:
  - Denoted by R(A1, A2, ..., An)
  - R is the **name** of the relation
  - The **attributes** of the relation are A1, A2, ..., An
  - n is the **cardinality** of the relation
- Example:

  CUSTOMER (Cust-id, Cust-name, Address, Phone#)
  - CUSTOMER is the relation name
  - The CUSTOMER relation schema (or just *relation*) has four attributes: Cust-id, Cust-name, Address, Phone#
- Each attribute has a **domain** or a set of valid values.
  - For example, the domain of Cust-id can be 6 digit numbers.

Formal Definitions - Tuple

- A **tuple** is an ordered set of values (enclosed in angled brackets ‘< … >’)
- Each value is derived from an appropriate *domain*.
- A row in the CUSTOMER relation is a 4-tuple and would consist of four values, for example:
  - <632895, "John Smith", "101 Main St. Atlanta, GA 30332", "(404) 894-2000">
  - Called a 4-tuple because it has 4 values
  - In general, a particular relation will have n-tuples, where n is the number of attributes for the relation
- A relation is a **set** of such tuples (rows)

Formal Definitions - Domain

- A **domain** of values can have a **logical definition**:
  - Example: “USA_phone_numbers” are the set of 10 digit phone numbers valid in the U.S.
- A domain also has 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.
  - Dates have various formats such as year, month, date formatted as yyyy-mm-dd, or as dd:mm:yyyy etc.

- The attribute name designates the **role** played by a domain in a relation:
  - Used to interpret the meaning of the data elements corresponding to that attribute
  - Example: The domain Date may be used to define two attributes “Invoice-date” and “Payment-date” with different meanings (roles)

Formal Definitions – State of a Relation

- Formally, a **relation state** r(R) is a subset of the *Cartesian product* of the domains of its attributes
  - each domain contains the set of all possible values the attribute can take.
  - The *Cartesian product* contains all possible tuples from the attribute domains
  - The relations state r(R) is the subset of tuples that represent valid information in the mini-world at a particular time

Addison-Wesley is an imprint of PEARSON

Formal Definitions - Summary

- Formally (see Figure 3.1),
  - Given relation schema R(A1, A2, ..., An)
  - Relation state r(R) $\subset$ dom(A1) X dom(A2) X ....X dom(An)
- R(A1, A2, …, An) is the **schema** of the relation
- R is the **name** of the relation
- A1, A2, …, An are the **attributes** of the relation
- r(R): is a specific **state** (or "instance" or “population”) of relation R – this is a *set of tuples* (rows) in the relation at a particular moment in time
  - r(R) = {t1, t2, …, tn} where each ti is an n-tuple
  - ti = <v1, v2, …, vn> where each vj *element-of* dom(Aj)

Formal Definitions - Example

- Let R(A1, A2) be a relation schema:
  - Let dom(A1) = {0,1}
  - Let dom(A2) = {a,b,c}
- Then: The *Cartesian product* dom(A1) X dom(A2) contains all possible tuples from these domains:

  {<0,a> , <0,b> , <0,c>, <1,a>, <1,b>, <1,c> }
- The relation state r(R) $\subset$ dom(A1) X dom(A2)
- For example: One possible state r(R) could be {<0,a> , <0,b> , <1,c> }
  - This state has three 2-tuples: <0,a> , <0,b> , <1,c>

Relation Definitions Summary

Informal Terms		Formal Terms
Table		Relation
Column Header		Attribute
All possible Column Values or Data Type		Domain
Row		Tuple
Table Definition		Schema of a Relation
Populated Table		State of the Relation

Characteristics Of a Relation

- Ordering of tuples in a relation r(R):
  - The tuples are *not considered to be ordered*, because a relation is a ***set*** of tuples (elements of a set are unordered) – see Figure 3.2
- Ordering of attributes in a relation schema R (and of values within each tuple):
  - We will consider the attributes in R(A1, A2, ..., An) and the values in each t=<v1, v2, ..., vn> to be **ordered.** It is an ***ordered list*** of n values.
  - However, a *more general definition* of relation (**which we will not use**) does not require attribute ordering
  - In this case, a tuple t = { <ai, vi>, ..., <aj, vj> } is an unordered set of n <attribute, value> pairs – one pair for each of the relation attributes (see Figure 3.3)
  - This is needed in query optimization (see Chapter 19)

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

Figure 3.1

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 3.3
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.)

- Values in a tuple:
  - All values are considered atomic (indivisible). Composite and multivalued attributes not allowed
  - Each value must be from the domain of the attribute for that column
    - If tuple t = <v1, v2, …, vn> is a tuple (row) in the relation state r of R(A1, A2, …, An)
    - Then each *vi* must be a value from *dom(Ai)*
  - A special **null** value is used to represent values that are *unknown, exist but not available or inapplicable or undefined* to certain tuples.

Characteristics Of Relations (cont.)

- Notation:
  - We refer to **component values** of a tuple t by:
    - t[Ai] or t.Ai
    - This is the value vi of attribute Ai for tuple t
  - Similarly, t[Au, Av, ..., Aw] refers to the subtuple of t containing the values of attributes Au, Av, ..., Aw, respectively in t

Relational Integrity Constraints

- Constraints are **conditions** that must hold on **all** *valid* relation states.
- Constraints are *derived* from the mini-world semantics
- There are three *main types* of built-in constraints in the relational model (Explicit Constraints):
  - **Key** constraints
  - **Entity integrity** constraints
  - **Referential integrity** constraints
- Another implicit constraint is the **domain** constraint
  - Every value in a tuple must be from the *domain of its attribute* (or it could be **null**, if allowed for that attribute)

Key Constraints

- **Superkey** SK of R:
  - Is a set of attributes SK of R with the following condition:
    - No two tuples in any valid relation state r(R) will have the same value for SK
    - That is, for any distinct tuples t1 and t2 in r(R), t1.SK ≠ t2.SK
    - This condition must hold in *any valid state* r(R)

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

Key Constraints

- **Key** K of R:
  - Is a "minimal" superkey
  - Formally, a key K is a superkey such that *removal of any attribute* from K results in a set of attributes that is not a superkey (or key) any more (does not possess the superkey uniqueness property)
  - Hence, a superkey *with one attribute* is always a key

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

Key Constraints (cont.)

- Example: Consider the CAR relation schema:
  - CAR(State, Reg#, SerialNo, Make, Model, Year)
  - CAR has two keys (determined from the mini-world constraints):
    - Key1 = {State, Reg#}
    - Key2 = {SerialNo}
  - Both are also superkeys of CAR
  - However, {SerialNo, Make} is a superkey but *not* a key.
- In general:
  - Any *key* is a *superkey* (but not vice versa)
  - Any set of attributes that *includes a key* is a *superkey*
  - A *minimal* superkey is also a key

Key Constraints (cont.)

- If a relation has several keys, they are called **candidate keys;** one is chosen to be the **primary key;** the others are called **unique** (or **secondary**) keys
  - The primary key attributes are <u>underlined</u>.
- Example: Consider the CAR relation schema:
  - CAR(<u>State, Reg#</u>, SerialNo, Make, Model, Year)
  - We choose License_number (which contains (State, Reg#) together) as the primary key – see Figure 3.4
- The primary key value is used to *uniquely identify* each tuple in a relation
  - Provides the tuple *identity*
  - Also used to *reference* the tuple from other tuples
- General rule: Choose the smallest-sized candidate key (in bytes) as primary key
  - Not always applicable – choice is sometimes subjective (as in Figure 3.4 – see next slide)

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 Database Schema

- **Relational Database Schema:**
  - A set S of relation schemas that belong to the same database.
  - S is the name of the whole **database schema**
  - S = {R1, R2, ..., Rn}
  - R1, R2, …, Rn are the names of the individual **relation schemas** within the database S
- Figure 3.5 shows a COMPANY database schema with 6 relation schemas

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.

Relational Database State

- **Next two slides show an example of a COMPANY database state (Figure 3.6)**
  - Each relation has a set of tuples
    - The tuples in each table satisfy key and other constraints
    - If all constraints are satisfied by a database state, it is called a **valid state**
  - The database state changes to another state whenever the tuples in any relation are changed via insertions, deletions, or updates

Addison-Wesley is an imprint of

PEARSON

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

Continued next page...

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

Entity Integrity Constraint

- **Entity Integrity:**
  - The *primary key attributes* PK of each relation schema R in S cannot have null values in any tuple of r(R).
    - This is because primary key values are used to *identify* the individual tuples.
    - t.PK $\neq$ null for any tuple t in r(R)
    - If PK has several attributes, null is not allowed in any of these attributes
  - Note: Other attributes of R may be also be constrained to disallow null values (called NOT NULL constraint), even though they are not members of the primary key.

Addison-Wesley is an imprint of PEARSON

Referential Integrity Constraint

- A constraint involving **two** relations
  - The previous constraints (key, entity integrity) involve a single relation.
- Used to specify a **relationship** among tuples in two relations:
  - The **referencing relation** and the **referenced relation**.

Addison-Wesley is an imprint of PEARSON

Referential Integrity (cont.)

- Tuples in the **referencing relation** R1 have attributes FK (called **foreign key** attributes) that reference the primary key attributes PK of the **referenced relation** R2.
  - A tuple t1 in R1 is said to **reference** a tuple t2 in R2 if t1.FK = t2.PK
  - FK and PK have to be from same domain
- Referential integrity can be displayed as a directed arc from R1.FK to R2.PK – see Figure 3.7

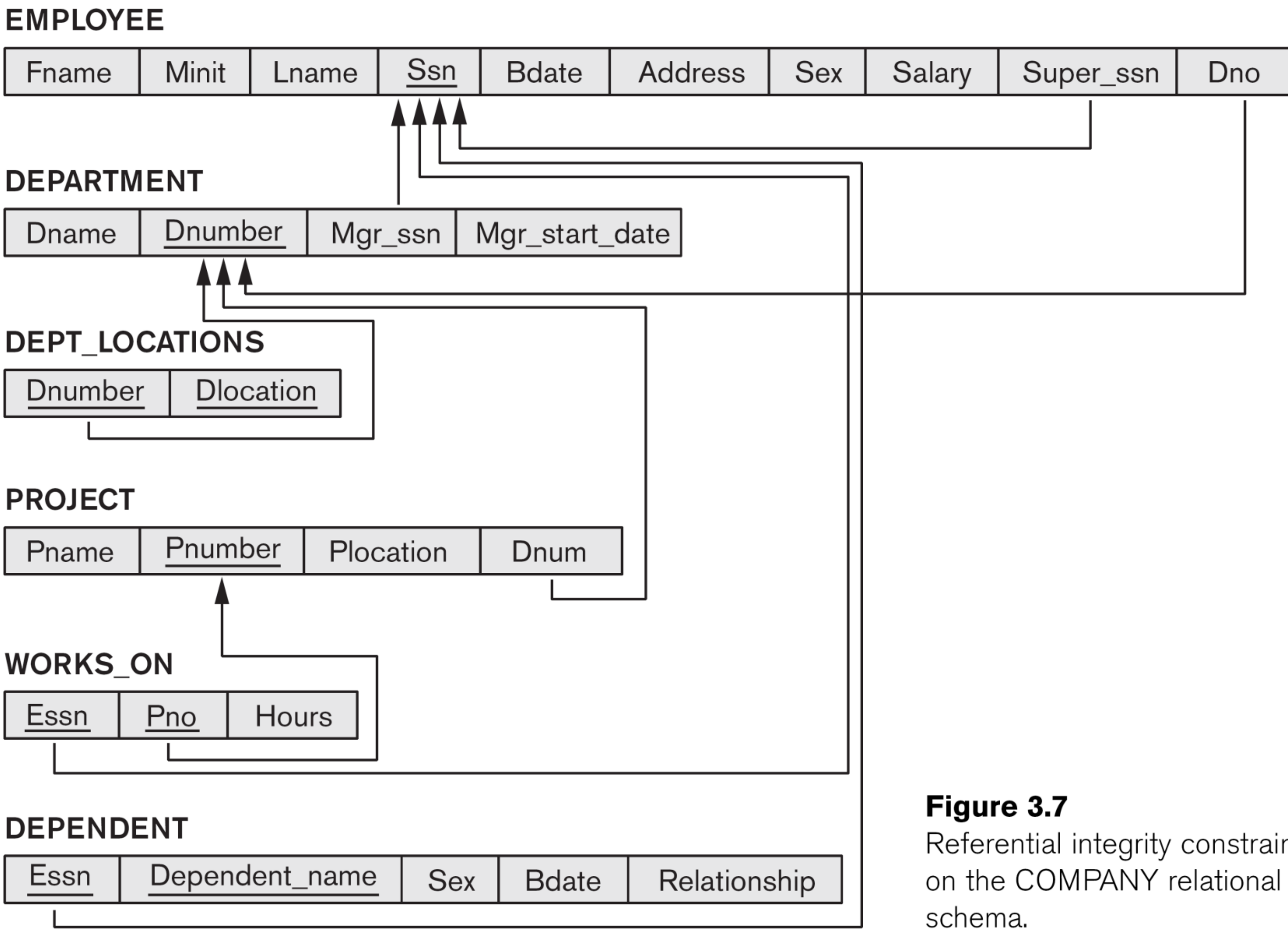


Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

Referential Integrity (or foreign key) Constraint (cont.)

- Statement of the constraint
  - For a particular database state, the value of the foreign key attribute (or attributes) FK in each tuple of the **referencing relation** R1 can be **either**:
    - (1) An existing primary key (PK) value of a tuple in the **referenced relation** R2, <u>or</u>
    - (2) a **null**.
- In case (2), the FK in R1 should **not** be a part of its own primary key, and cannot have the NOT NULL constraint.

Other Types of Constraints

- Semantic Integrity Constraints:
  - cannot be expressed by the built-in model constraints
  - Example: “the max. no. of hours per employee for all projects he or she works on is 56 hrs per week”
- A **constraint specification** language can be used to express these
- SQL has **TRIGGERS** and **ASSERTIONS** to express some of these constraints (see Section 5.2)

Addison-Wesley is an imprint of

PEARSON

Relational Update Operations

- Each *relation* will have many tuples in its current relation state
- The *relational database state* is a union of all the individual relation states at a particular time
- Whenever the database is changed, a new state arises
- Basic operations for changing the database:
  - INSERT new tuples in a relation
  - DELETE existing tuples from a relation
  - UPDATE attribute values of existing tuples

Operations to Modify Relations

- Three basic operations:
  - INSERT
  - DELETE
  - UPDATE
- Integrity constraints should not be violated by the update operations.
- Several update operations may have to be grouped together into a **transaction**.
- Updates may **propagate** to cause other updates automatically. This may be necessary to maintain integrity constraints.

Update Operations (cont.)

- In case of integrity violation, several actions can be taken:
  - Cancel the operation that causes the violation (RESTRICT or REJECT option)
  - Perform the operation but inform the user of the violation
  - Trigger additional updates so the violation is corrected (CASCADE option, SET NULL option)
  - Execute a user-specified error-correction routine

Addison-Wesley is an imprint of

PEARSON

INSERT operation

- INSERT one or more new tuples into a relation
- INSERT may violate any of the constraints:
  – Domain constraint:
    - if one of the attribute values provided for a new tuple is not of the specified attribute domain
  – Key constraint:
    - if the value of a key attribute in a new tuple already exists in another tuple in the relation
  – Referential integrity:
    - if a foreign key value in a new tuple references a primary key value that does not exist in the referenced relation
  – Entity integrity:
    - if the primary key value is null in a new tuple

DELETE operation

- DELETE one or more existing tuples from a relation
- DELETE may violate only referential integrity:
  - If the primary key value of the tuple being deleted is referenced from other tuples in the database
    - Can be remedied by several actions: RESTRICT, CASCADE, SET NULL (see Chapter 4 for more details)
      - RESTRICT option: reject the deletion
      - CASCADE option: propagate the deletion by automatically deleting the referencing tuples
      - SET NULL option: set the foreign keys of the referencing tuples to NULL (the foreign keys cannot have NOT NULL constraint)
  - One of the above options must be specified during database design for each referential integrity (foreign key) constraint

UPDATE operation

- UPDATE modifies the values of attributes in one or more existing tuples in a relation
- UPDATE may violate domain constraint and NOT NULL constraint on an attribute being modified
- Other constraints may also be violated:
  - Updating the primary key (PK):
    - Similar to a DELETE followed by an INSERT
    - Need to specify similar options to DELETE
    - The CASCADE option propagates the new value of PK to the foreign keys of the referencing tuples automatically
  - Updating a foreign key (FK) may violate referential integrity
  - Updating an ordinary attribute (neither PK nor FK):
    - Can only violate domain or NOT NULL constraints

Summary

- Presented Relational Model Concepts
  - Definitions
  - Characteristics of relations
- Discussed Relational Model Constraints and Relational Database Schemas
  - Domain constraints'
  - Key constraints
  - Entity integrity
  - Referential integrity
- Described the Relational Update Operations and Dealing with Constraint Violations

In-Class Exercise

(Taken from Exercise 3.16)

Consider the following relations for a database that keeps track of student enrollment in courses and the books adopted for each course:

STUDENT(<u>SSN</u>, Name, Major, Bdate)

COURSE(<u>Course#</u>, Cname, Dept)

ENROLL(<u>SSN</u>, <u>Course#</u>, <u>Quarter</u>, Grade)

BOOK_ADOPTION(<u>Course#</u>, <u>Quarter</u>, Book_ISBN)

TEXT(<u>Book_ISBN</u>, Book_Title, Publisher, Author)

Draw a relational schema diagram specifying the foreign keys for this schema.

Addison-Wesley is an imprint of PEARSON