CONSTRAINTS AND UPDATES

CHAPTER 3 (6/E)

CHAPTER 5 (5/E)

LECTURE OUTLINE

- Constraints in Relational Databases
- Update Operations
- Brief History of Database Applications (from Section 1.7)

RELATIONAL MODEL CONSTRAINTS

- **Constraints**
  - Restrictions on the permitted values in a database state
  - Derived from the rules in the miniworld that the database represents
- **Inherent model-based constraints** or **implicit constraints**
  - Inherent in the data model
  - e.g., duplicate tuples are not allowed in a relation
- **Schema-based constraints** or **explicit constraints**
  - Can be directly expressed in schemas of the data model
  - e.g., films have only one director
- **Application-based** or **semantic constraints**
  - Also called **business rules**
  - Not directly expressed in schemas
  - Expressed and enforced by application program
  - e.g., this year's salary increase can be no more than last year's

DOMAIN CONSTRAINTS

- Declared by specifying the data type for each attribute:
  - 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

- Uniqueness constraints on tuples
- SK $\subseteq \{A_1, A_2, ..., A_n\}$ is a **superkey** of $R(A_1, A_2, ..., A_n)$ if
  - In any relation state r of R, no two distinct tuples can have the same values for SK
    - $t_1[SK] = t_2[SK] \Rightarrow t_1 = t_2$
- K is a **key** of R if
  1. K is a superkey of R
  2. Removing any attribute from K leaves a set of attributes that is not a superkey of R any more
     - No proper subset of K is a superkey of R
- If K is a key, it satisfies two properties
  1. No two distinct tuples have the same values across all attributes in K (i.e., it is a superkey)
  2. It is a minimal superkey (i.e., no subset of K has this uniqueness constraint)

KEY CONSTRAINTS (CONT'D.)

- What are some possible keys for the following relation?

Film

title	genre	year	director	minutes	budget	gross
The Company Men	drama	2010	John Wells	104	15,000,000	4,439,063
Lincoln	biography	2012	Steven Spielberg	150	65,000,000	181,408,467
War Horse	drama	2011	Steven Spielberg	146	66,000,000	79,883,359
Argo	drama	2012	Ben Affleck	120	44,500,000	135,178,251
Fire Sale	comedy	1977	Alan Arkin	88	1,500,000	NULL
Lincoln	biography	1992	Peter W. Kunhardt	240	NULL	NULL
Life	comedy	1999	Ted Demme	108	75,000,000	63,844,974
Life	drama	1999	Eun-Ryung Cho	19	NULL	NULL

- Note that the instance can show that something is *not* a key, but we need to declare as part of the schema that something *is* a key.
  - Uniqueness must hold in all valid relation states.
  - Serves as a constraint on updates.

KEY CONSTRAINTS (CONT'D.)

- **Primary key** of the relation
  - Relation schema may have more than one key.
  - Declare one chosen key among candidates as *primary*
    - Its values will be used to refer to specific tuples
    - Cannot have the value NULL for any tuple
  - Diagrammatically, underline attribute
- Other candidate keys are designated as **unique**
  - Non-null values cannot repeat, but values may be NULL

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.

SATISFYING INTEGRITY CONSTRAINTS

- **Relational database schema** *S*
  - Set of relation schemas $S = \{R_1, R_2, ..., R_m\}$
  - Set of integrity constraints IC
- **Valid relational database state**
  - Set of relation states $DB = \{r_1, r_2, ..., r_m\}$
  - Each r_i is a state of R_i such that r_i satisfies integrity constraints specified in IC
- **Invalid state**
  - Does not obey all the integrity constraints

OTHER INTEGRITY CONSTRAINTS

- Already defined *domain constraints* and *key constraints*
- **Entity integrity constraint**
  - No primary key value can be NULL
- **Referential integrity constraint**
  - Specified between two relations
    - Allows tuples in one relation to *refer to* tuples in another
  - Maintains consistency among tuples in two relations
  - **Foreign key** rules:
    - Let PK be the primary key in one relation R_1 (set of attributes in its relational schema declared to be primary key)
    - Let FK be a set of attributes for another relation R_2
    - The attribute(s) FK have the same domain(s) as the attribute(s) PK
    - Value of FK in a tuple t_2 of the current state $r_2(R_2)$ either occurs as a value of PK for some tuple t_1 in the current state $r_1(R_1)$ or it is NULL

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

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

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

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

DIAGRAMMING REF CONSTRAINTS

- Show each relational schema
  - Underline primary key attributes in each
- Directed arc from each foreign key to the relation it references

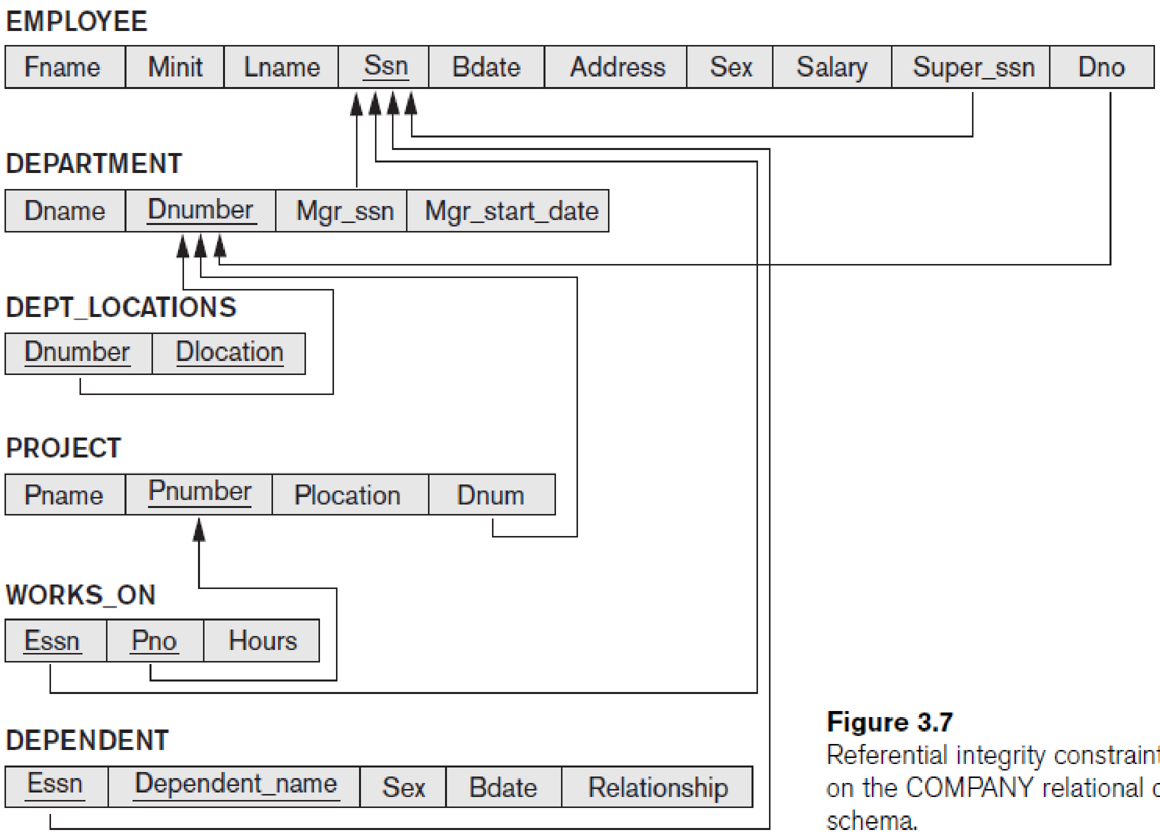

Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

MORE INTEGRITY CONSTRAINTS

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

(more later in the course)

- **Semantic integrity constraints**
  - Specified by business rules outside the schema
  - Sometimes declared using database **triggers** and **assertions**
  - Often undeclared but checked within application programs
  - **State** (**static**) **constraints**
    - Define conditions that a valid state of the database must satisfy
  - **Transition** (**dynamic**) **constraints**
    - Define valid state *changes* in the database

UPDATE OPERATIONS

- Operations of the relational model are *retrievals* or *changes*
- Basic operations that change the states of relations in the database:
  - Insert
  - Delete
  - Update (or Modify)
- Updates must be consistent with constraints

THE INSERT OPERATION

- Select a relation *R* and provide a list of attribute values for a new tuple *t* to be inserted into (appended to) *R*
- Need to check against *all* constraints
    - If an insertion violates one or more constraints
        - Default option is to reject the insertion

THE DELETE OPERATION

- Select the tuple(s) to be deleted
- Can violate referential integrity only
  - If tuple being deleted is referenced by foreign keys from other tuples
  - **Restrict**
    - Reject the deletion
  - **Cascade**
    - Propagate the deletion by deleting tuples that reference the tuple being deleted
  - **Set null** or **set default**
    - Modify the referencing attribute values that cause the violation

THE UPDATE OPERATION

- 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 possible constraint violations as with Insert/Delete

THE TRANSACTION CONCEPT

- **Transaction**
  - Executing program includes some database operations
  - To be considered as if it were just a single operation
  - Must leave the database in a valid or consistent state
- **Online transaction processing (OLTP) systems**
  - Examples: reservation systems, purchase systems
  - Execute transactions at rates that reach several hundred per second

BRIEF HISTORY

- Relational model
  - Formulated by E.F.Codd (IBM) before 1970
  - First commercial implementations available in early 1980s
  - Predominant database model used today
- (earlier) Hierarchical and network models
  - Preceded the relational model
  - Pointer-based
  - Access relied on record-at-a-time navigation
- (later) Object-oriented applications and more complex databases
  - Object-relational model
  - Used in specialized applications: engineering design, multimedia publishing, and manufacturing systems

RECENT HISTORY

- Interchanging data on the Web for e-commerce using XML
  - Extended markup language (XML) primary standard for interchanging data among various types of databases and Web pages
  - Moving to cloud-based services
- Extending database capabilities for new applications
  - Extensions to support specialized requirements for applications
  - Enterprise resource planning (ERP)
    - e.g., SAP
  - Customer relationship management (CRM)
    - e.g., SAP
  - Enterprise content management (ECM)
    - e.g., Open Text
    - includes extensions to information retrieval (IR) to deal with documents (proposals, reports, news articles, etc.)

LECTURE SUMMARY

- 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
- Database applications have evolved
  - Current trend: “Big data” involving Web, social networks, scientific or financial streams