Chapter 3
The Relational Data Model and Relational Database Constraints
Chapter 3 Outline

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

- Relational model
  - Based on original work by Edgar F. Codd.
  - First commercial implementations available in early 1980s
  - Has been implemented in a large number of commercial systems

- Hierarchical and network models
  - Preceded the relational model
The Relational Data Model and Relational Database Constraints

- Hierarchical and network models

Nodes and links are allowed
The Relational Data Model and Relational Database Constraints

- Relational data model

*Only tables are allowed*
Relational Model Concepts

- Represents data as a collection of relations
- **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*
### Figure 3.1
The attributes and tuples of a relation STUDENT.

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

- **Domain** $D$
  - Set of atomic values

- **Atomic**
  - Each value indivisible

- **Specifying a domain**
  - **Data type** specified for each domain
Domains, Attributes, Tuples, and Relations (cont’d.)

- **Relation schema** $R$
  - Denoted by $R(A_1, A_2, ..., A_n)$
  - Made up of a relation name $R$ and a list of attributes, $A_1, A_2, ..., A_n$

- **Attribute** $A_i$
  - 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
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 $t = <v_1, v_2, \ldots, v_n$
    - Each value $v_i$, $1 \leq i \leq n$, is an element of dom($A_i$) or is a special NULL value
Domains, Attributes, Tuples, and Relations (cont’d.)

- Relation (or relation state) \( r(R) \)
  - Mathematical relation of degree \( n \) on the domains \( \text{dom}(A_1), \text{dom}(A_2), \ldots, \text{dom}(A_n) \)

- Subset of the Cartesian product of the domains that define \( R \):
  - \( r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \ldots \times \text{dom}(A_n)) \)
Domains, Attributes, Tuples, and Relations (cont’d.)

- **Cardinality**
  - Total number of values in domain

- **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
  - Relation defined as a set of tuples
  - Elements 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
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 $\text{dom}(A_i)$

- **Use the first definition of relation**
  - Attributes and the values within tuples are ordered
  - Simpler notation
Characteristics of Relations (cont’d.)

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

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

- Values and NULLs in tuples
  - Each value in a tuple is atomic
- 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.)

- Interpretation (meaning) of a relation
  - **Assertion**
    - Each tuple in the relation is a **fact** or a particular instance of the assertion
  - **Predicate**
    - Values in each tuple interpreted as values that satisfy predicate
Relational Model Notation

- Relation schema $R$ of degree $n$
  - Denoted by $R(A_1, A_2, ..., A_n)$
- 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
Relational Model Notation

- **n-tuple** $t$ in a relation $r(R)$
  - Denoted by $t = \langle v_1, v_2, \ldots, v_n \rangle$
  - $v_i$ is the value corresponding to attribute $A_i$

- Component values of tuples:
  - $t[A_i]$ and $t.A_i$ refer to the value $v_i$ in $t$ for attribute $A_i$
  - $t[A_u, A_w, \ldots, A_z]$ and $t.(A_u, A_w, \ldots, A_z)$ refer to the subtuple of values $\langle v_u, v_w, \ldots, v_z \rangle$ from $t$ corresponding to the attributes specified in the list
Relational Model Constraints

- Constraints
  - Restrictions on the actual 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
Relational Model Constraints (cont’d.)

- **Schema-based constraints or explicit constraints**
  - Can be directly expressed in schemas of the data model

- **Application-based or semantic constraints or business rules**
  - Cannot be directly expressed in schemas
  - Expressed and enforced by application program
Domain Constraints

- Typically include:
  - 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

- Key
  - Superkey of \( R \)
  - Removing any attribute \( A \) from \( K \) leaves a set of attributes \( K \) that is not a superkey of \( R \) any more
Key Constraints and Constraints on NULL Values (cont’d.)

- Key satisfies two properties:
  - 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 (cont’d.)

- **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**
Key Constraints and Constraints on NULL Values (cont’d.)

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

<table>
<thead>
<tr>
<th>License_number</th>
<th>Engine_serial_number</th>
<th>Make</th>
<th>Model</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texas ABC-739</td>
<td>A69352</td>
<td>Ford</td>
<td>Mustang</td>
<td>02</td>
</tr>
<tr>
<td>Florida TVP-347</td>
<td>B43696</td>
<td>Oldsmobile</td>
<td>Cutlass</td>
<td>05</td>
</tr>
<tr>
<td>New York MPO-22</td>
<td>X83554</td>
<td>Oldsmobile</td>
<td>Delta</td>
<td>01</td>
</tr>
<tr>
<td>California 432-TFY</td>
<td>C43742</td>
<td>Mercedes</td>
<td>190-D</td>
<td>99</td>
</tr>
<tr>
<td>California RSK-629</td>
<td>Y82935</td>
<td>Toyota</td>
<td>Camry</td>
<td>04</td>
</tr>
<tr>
<td>Texas RSK-629</td>
<td>U028365</td>
<td>Jaguar</td>
<td>XJS</td>
<td>04</td>
</tr>
</tbody>
</table>
Relational Databases and Relational Database Schemas

- **Relational database schema** $S$
  - Set of relation schemas $S = \{R_1, R_2, \ldots, R_m\}$
  - Set of integrity constraints $IC$

- **Relational database state**
  - Set of relation states $DB = \{r_1, r_2, \ldots, r_m\}$
  - Each $r_i$ is a state of $R_i$ and such that the $r_i$ relation states satisfy integrity constraints specified in $IC$
Relational Databases and Relational Database Schemas (cont’d.)

- **Invalid state**
  - Does not obey all the integrity constraints

- **Valid state**
  - Satisfies all the constraints in the defined set of integrity constraints IC
Integrity, Referential Integrity, and Foreign Keys

- **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, and Foreign Keys (cont’d.)

- **Foreign key rules:**
  - The attributes in FK have the same domain(s) as the primary key attributes PK.
  - Value of FK in a tuple $t_1$ of the current state $r_1(R_1)$ either occurs as a value of PK for some tuple $t_2$ in the current state $r_2(R_2)$ or is NULL.
Integrity, Referential Integrity, and Foreign Keys (cont’d.)

- 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
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
Update Operations, Transactions, and Dealing with Constraint Violations

- Operations of the relational model can be categorized into retrievals and updates
- Basic operations that change the states of relations in the database:
  - Insert
  - Delete
  - Update (or Modify)
Figure 3.6
One possible database state for the COMPANY relational database schema.

<table>
<thead>
<tr>
<th>Employee</th>
<th>Ssn</th>
<th>Bdate</th>
<th>Address</th>
<th>Sex</th>
<th>Salary</th>
<th>Super_ssn</th>
<th>Dno</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>123456789</td>
<td>1965-01-09</td>
<td>731 Fondren, Houston, TX</td>
<td>M</td>
<td>30000</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>Franklin</td>
<td>333445555</td>
<td>1955-12-09</td>
<td>638 Voss, Houston, TX</td>
<td>M</td>
<td>40000</td>
<td>888666555</td>
<td>5</td>
</tr>
<tr>
<td>Alicia</td>
<td>999887777</td>
<td>1969-01-19</td>
<td>3321 Castle, Spring, TX</td>
<td>F</td>
<td>25000</td>
<td>987654321</td>
<td>4</td>
</tr>
<tr>
<td>Jennifer</td>
<td>997654321</td>
<td>1941-06-20</td>
<td>291 Berry, Bellaire, TX</td>
<td>F</td>
<td>43000</td>
<td>888666555</td>
<td>4</td>
</tr>
<tr>
<td>Ramesh</td>
<td>668884444</td>
<td>1982-09-15</td>
<td>975 Fire Oak, Humble, TX</td>
<td>M</td>
<td>38000</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>Joyce</td>
<td>453453453</td>
<td>1972-07-31</td>
<td>5631 Rice, Houston, TX</td>
<td>F</td>
<td>25000</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>Ahmad</td>
<td>987987987</td>
<td>1989-03-29</td>
<td>980 Dallas, Houston, TX</td>
<td>M</td>
<td>25000</td>
<td>987654321</td>
<td>4</td>
</tr>
<tr>
<td>James</td>
<td>888666555</td>
<td>1937-11-10</td>
<td>450 Stone, Houston, TX</td>
<td>M</td>
<td>55000</td>
<td>NULL</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Department</th>
<th>Dnumber</th>
<th>Mgr_ssn</th>
<th>Mgr_start_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research</td>
<td>5</td>
<td>333445555</td>
<td>1988-05-22</td>
</tr>
<tr>
<td>Administration</td>
<td>4</td>
<td>987654321</td>
<td>1995-01-01</td>
</tr>
<tr>
<td>Headquarters</td>
<td>1</td>
<td>888666555</td>
<td>1981-06-19</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dept_locations</th>
<th>Dnumber</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Houston</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Stafford</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Bellaire</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Sugarland</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Houston</td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.6
One possible database state for the COMPANY relational database schema.

<table>
<thead>
<tr>
<th>WORKS_ON</th>
<th>PROJECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Essn</td>
<td>Pname</td>
</tr>
<tr>
<td>Pno</td>
<td>Pnumber</td>
</tr>
<tr>
<td>Hours</td>
<td>Dnum</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>123456789</td>
<td>ProductX</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>123456789</td>
<td>ProductY</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>666884444</td>
<td>ProductZ</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>453453453</td>
<td>Computerization</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>453453453</td>
<td>Reorganization</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>333445555</td>
<td>Newbenefits</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPENDENT</th>
<th>DEPENDENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Essn</td>
<td>Dependent_name</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
</tr>
<tr>
<td>333445555</td>
<td>Alice</td>
</tr>
<tr>
<td>333445555</td>
<td>Theodore</td>
</tr>
<tr>
<td>333445555</td>
<td>Joy</td>
</tr>
<tr>
<td>987654321</td>
<td>Abner</td>
</tr>
<tr>
<td>123456789</td>
<td>Michael</td>
</tr>
<tr>
<td>123456789</td>
<td>Alice</td>
</tr>
<tr>
<td>123456789</td>
<td>Elizabeth</td>
</tr>
</tbody>
</table>
Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.
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
- If an insertion violates one or more constraints
  - Default option is to reject the insertion
The Delete Operation

- Can violate only referential integrity
  - 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 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
The Transaction Concept

- **Transaction**
  - Executing program
  - Includes some database operations
  - Must leave the database in a valid or consistent state

- **Online transaction processing (OLTP) systems**
  - Execute transactions at rates that reach several hundred per second
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