CS275 – Intro to Databases

Relations and Constraints
Chap. 3.1 – 3.4

Relational Model

• Data representation and integrity constraints
• Table creation and modification using
• Data creation and modification, and retrieval
• Relational database from ER diagrams
• Views

Data Representation

• Relational model
  – Database is a collection of relations (tables)
  – Each table consists of rows (data) and columns (properties)
Data Representation

- Relational model
  - Simple
  - (Relatively) easy to construct complex queries

<table>
<thead>
<tr>
<th>ENAME</th>
<th>SNO</th>
<th>ICSTATE</th>
<th>ADDRESS</th>
<th>DNNUMBER</th>
<th>ENAME</th>
<th>DNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Doe</td>
<td>12345</td>
<td>CA</td>
<td>123 Main St, TX</td>
<td>5</td>
<td>Jones</td>
<td>Mathematics</td>
</tr>
<tr>
<td>Jane Smith</td>
<td>67890</td>
<td>NY</td>
<td>456 Downtown, NY</td>
<td>4</td>
<td>Smith</td>
<td>Psychology</td>
</tr>
<tr>
<td>Mary Johnson</td>
<td>10111</td>
<td>CA</td>
<td>789 University Ave, CA</td>
<td>3</td>
<td>Johnson</td>
<td>Biology</td>
</tr>
</tbody>
</table>

Data Representation

- A relation
  - Relation schema
  - Relation instance

Data Representation

- A relation
  - Relation schema
  - Columns of the table (also called fields, attributes)
    - Domain of each field
      - Domain name
        - Associated values
    - Student(sid: string, name: string, login: string, age: integer, gpa: real)
Data Representation

- A relation
  - Relation instance
    - Records (tuples, rows)

<table>
<thead>
<tr>
<th>EMPLOYEE</th>
<th>Ssn</th>
<th>BDATE</th>
<th>ADDRESS</th>
<th>DEPARTMENT</th>
<th>PHONE_NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith, John K.</td>
<td>123456789</td>
<td>1965-01-01</td>
<td>405 1st Avenue, NY</td>
<td>5</td>
<td>555-123-4567</td>
</tr>
<tr>
<td>Davis, Alice J.</td>
<td>987654321</td>
<td>1960-07-15</td>
<td>902 2nd Street, LA</td>
<td>4</td>
<td>444-567-8901</td>
</tr>
<tr>
<td>Johnson, Michael K.</td>
<td>654321098</td>
<td>1970-08-09</td>
<td>345 3rd Avenue, TX</td>
<td>5</td>
<td>777-890-1234</td>
</tr>
<tr>
<td>Mitchell, Linda A.</td>
<td>876543210</td>
<td>1955-12-31</td>
<td>654 4th Avenue, CA</td>
<td>4</td>
<td>666-789-0123</td>
</tr>
<tr>
<td>Brown, James E.</td>
<td>543210987</td>
<td>1975-10-10</td>
<td>234 5th Avenue, NJ</td>
<td>1</td>
<td>111-222-3333</td>
</tr>
</tbody>
</table>

Some (interesting) properties
- No duplicates of records (sets)
- Order of records does not matter
- Order of columns in general does not matter

Domain constraints
- SSN = ‘123456789’
- BDate = ‘Twenty-Years Ago’
- GPA = ‘So So’
- Salary = ‘Above Average’
- Phone_Number = ‘7-8599’
Data Representation

• Degree
  – # of fields
• Cardinality
  – # of tuples

Data Representation

• What do we remember?
  – Relation
  – Relation schema
  – Relation instance
  – Tuples
  – Fields
  – Domain constraints
  – Cardinality
  – Degree

Data Creation

CREATE TABLE Students (sid CHAR(20),
           name CHAR(30),
           login CHAR(20),
           age INTEGER,
           gpa REAL
         );
Table Update

CREATE TABLE Students (sid CHAR(20),
name CHAR(30),
login CHAR(20),
age INTEGER,
gpa REAL);

INSERT INTO Students (sid, name, login, age, gpa) VALUES(53688, 'Smith', 'smith@ee', 18, 3.2)

DELETE FROM Students S WHERE S.name = 'Smith'

UPDATE Student S SET S.age = S.age+1, S.gpa = S.gpa -1 WHERE S.sid = 53688

Table Update

UPDATE Student S
SET S.gpa = S.gpa -1
WHERE S.gpa >= 3.3

Integrity Constraints

• An IC
  – is a condition specified on a database schema
  – restricts the data that can be stored in an instance
  – is enforced by DBMS
Integrity Constraints

• Why do we need them?

Integrity Constraints

• Key constraints
  – A certain minimal subset of the fields of a relation that is a unique identifier

• Candidate keys
  – No two tuples have the same identical values in all the fields of a key
  – No proper subset of the set of the fields in a key is a unique identifier

Integrity Constraints

• Every table has a key (guaranteed!)
• Why?
Integrity Constraints

• Primary key

CREATE TABLE Students (sid CHAR(20),
    name CHAR(30),
    login CHAR(20),
    age INTEGER,
    gpa REAL
    UNIQUE (name, age),
    CONSTRAINTS StudentKey PRIMARY KEY (sid)
);

Integrity Constraints

• Foreign key

Student(sid: string, name: string, login: string, age: integer, gpa: real)
Enrolled(studentid: string, cid: string, grade: string)

• How do we ensure only certain students can enroll?
Integrity Constraints

- Foreign key on a table can refer to itself.
- Example?
  - Employee and Manager

Integrity Constraints

- Foreign key on a table can refer to itself.
- Example?
  - Employee and Manager

Integrity Constraints

- Not NULL constraint
- When do we need this?
Integrity Constraints

CREATE TABLE Enrolled (studentid CHAR(20),
cid CHAR(20),
grade CHAR(10)
PRIMARY KEY (studentid, cid),
FOREIGN KEY (studentid) REFERENCES students );

General Constraints

• Business rules
  – Only students with GPA > 2.5 are allowed to enroll
• Other examples
• Why are they not IC’s?

Enforcing Integrity Constraints

INSERT
INTO Students (sid, name, login, age, gpa)
VALUES(null, ‘Smith’, ‘smith@ee’, 18, 3.2)

What other constraints can be violated by insertion?
Enforcing Integrity Constraints

```
INSERT
INTO Students (sid, name, login, age, gpa)
VALUES(null, 'Smith', 'smith@ee', 18, 3.2)
```

What other constraints can be violated by insertion?
- domain
- unique
- primary key
- foreign key

---

Enforcing Integrity Constraints

```
INSERT
INTO Students (sid, name, login, age, gpa)
VALUES(null, 'Smith', 'smith@ee', 18, 3.2)
```

What other constraints can be violated by deletion?
- foreign key

---

Enforcing Integrity Constraints

```
INSERT
INTO Students (sid, name, login, age, gpa)
VALUES(null, 'Smith', 'smith@ee', 18, 3.2)
```

What other constraints can be violated by deletion?
- foreign key
Enforcing Integrity Constraints

```
INSERT
INTO Students (sid, name, login, age, gpa)
VALUES(null, 'Smith', 'smith@ee', 18, 3.2)
```

What other constraints can be violated by update?

Enforcing Foreign Key Constraints

- **Insertion**
  - What happens if the studentid is invalid when inserting a new enrollment record?

- **Deletion**
  - What happens when deleting a student record?
Enforcing Foreign Key Constraints

• Deletion
  – What happens when deleting a student record?
    • Do not delete.

CREATE TABLE Enrolled (studentid CHAR(20),
cid CHAR(20),
grade CHAR(10)
PRIMARY KEY (studentid, cid),
FOREIGN KEY (studentid) REFERENCES students
  ON DELETE NO ACTION
);

• Deletion
  – What happens when deleting a student record?
    • Remove the record in the Student table
    • Remove all records in the Enrollment table with the given StudentID.

CREATE TABLE Enrolled (studentid CHAR(20),
cid CHAR(20),
grade CHAR(10)
PRIMARY KEY (studentid, cid),
FOREIGN KEY (studentid) REFERENCES students
  ON DELETE CASCADE
);

• Deletion
  – What happens when deleting a student record?
    • Remove the record in the Student table
    • Set the StudentID to NULL for all records in Enrollment with the given StudentID.

CREATE TABLE Enrolled (studentid CHAR(20),
cid CHAR(20),
grade CHAR(10)
PRIMARY KEY (studentid, cid),
FOREIGN KEY (studentid) REFERENCES students
  ON DELETE SET NULL
);
Enforcing Foreign Key Constraints

• Deletion
  – What happens when deleting a student record?
    • Remove the record in the Student table
    • Set the StudentID to the default value for all records in Enrollment
      with the given StudentID.

CREATE TABLE Enrolled (studentid CHAR(20) DEFAULT '53666',
                        cid CHAR(20),
                        grade CHAR(10)
                        PRIMARY KEY (studentid, cid),
                        FOREIGN KEY (studentid) REFERENCES students
                        ON DELETE SET DEFAULT
                       );

Enforcing Foreign Key Constraints

• Update
  – What happens when updating the ID of a student?

  • ON UPDATE NO ACTION
  • ON UPDATE CASCADE
  • ON UPDATE SET DEFAULT
  • ON UPDATE SET NULL
Enforcing Foreign Key Constraints

• Update
  – What happens when updating the ID of a student?

```sql
CREATE TABLE Enrolled (studentid CHAR(20),
cid CHAR(20),
grade CHAR(10)
PRIMARY KEY (studentid, cid),
FOREIGN KEY (studentid) REFERENCES students
  ON DELETE CASCADE
  ON UPDATE NO ACTION
);
```

Data Retrieval

• SELECT *
  FROM Students S
  WHERE S.age < 18

• SELECT S.name, S.login
  FROM Students S
  WHERE S.age < 18
Questions?