SQL
CSCI 201L

Jeffrey Miller, Ph.D.
jeffrey.miller@usc.edu

HTTP://WWW-SCF.USC.EDU/~CSCI201
Outline

- SQL
- Try It!
Relational Databases

- Write the SQL code to create the following database, named StudentGrades

<table>
<thead>
<tr>
<th>classID</th>
<th>prefix</th>
<th>number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CSCI</td>
<td>103</td>
</tr>
<tr>
<td>2</td>
<td>CSCI</td>
<td>104</td>
</tr>
<tr>
<td>3</td>
<td>CSCI</td>
<td>201</td>
</tr>
<tr>
<td>4</td>
<td>EE</td>
<td>101</td>
</tr>
<tr>
<td>5</td>
<td>EE</td>
<td>102</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>studentID</th>
<th>fname</th>
<th>lname</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sheldon</td>
<td>Cooper</td>
</tr>
<tr>
<td>2</td>
<td>Leonard</td>
<td>Hofstadter</td>
</tr>
<tr>
<td>3</td>
<td>Howard</td>
<td>Wolowitz</td>
</tr>
<tr>
<td>4</td>
<td>Rajesh</td>
<td>Koothrappali</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gradeID</th>
<th>classID</th>
<th>studentID</th>
<th>letterGrade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>A-</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3</td>
<td>B-</td>
</tr>
</tbody>
</table>

Tables are called relations
Columns are called attributes
Rows are called tuples
Connections are called relationships
DROP DATABASE if exists StudentGrades;
CREATE DATABASE StudentGrades;
USE StudentGrades;
CREATE TABLE Student (studentID int(11) primary key not null auto_increment, fname varchar(50) not null, lname varchar(50) not null);
INSERT INTO Student (fname, lname) VALUES ('Sheldon', 'Cooper');
INSERT INTO Student (fname, lname) VALUES ('Leonard', 'Hofstadter');
CREATE TABLE Class (classID int(11) primary key not null auto_increment, prefix varchar(5) not null, number int(4) not null);
INSERT INTO Class (prefix, number) VALUES ('CSCI', 103);
INSERT INTO Class (prefix, number) VALUES ('CSCI', 104);
CREATE TABLE Grade (gradeID int(11) primary key not null auto_increment, classID int(11) not null, studentID int(11) not null, letterGrade varchar(2) not null, FOREIGN KEY fk1(classID) REFERENCES class(classID), FOREIGN KEY fk2(studentID) REFERENCES student(studentID)));
INSERT INTO Grade (studentID, classID, letterGrade) VALUES (1, 1, 'A');
INSERT INTO Grade (studentID, classID, letterGrade) VALUES (1, 2, 'A');
INSERT INTO Grade (studentID, classID, letterGrade) VALUES (2, 1, 'A');
INSERT INTO Grade (studentID, classID, letterGrade) VALUES (2, 2, 'A');
Executing SQL

- There are a few ways to execute SQL statements, which typically depend on the DBMS.
- MySQL has the MySQL Workbench and command line options.
- We often want to be able to access databases from within programs though.
  - This can be to insert, update, delete, or query the data.
- The Java Database Connectivity (JDBC) drivers allow us to embed SQL in our Java code and execute those statements on a database programmatically.
SQL Statements

- Write SQL code to do the following
  - Insert CSCI 170 into the database
  - Change Leonard’s grade in EE 101 to a C+
  - Add Amy Farrah Fowler into the Student table
  - Give Amy a grade of A for CSCI 103
  - Give Amy a grade of B+ for CSCI 170
Write SQL code to do the following

- Insert CSCI 170 into the database
  - INSERT INTO class (prefix, number) VALUES ('CSCI', 170);

- Change Leonard’s grade in EE 101 to a C+
  - UPDATE grade g, student s, class c SET g.letterGrade='C+' WHERE s.studentID=g.studentID AND s.fname='Leonard' AND s.lname='Hofstadter' AND g.classID=c.classID AND c.prefix='EE' AND c.number='101';

- Add Amy Farrah Fowler into the Student table
  - INSERT INTO student (fname, lname) VALUES ('Amy', 'Farrah Fowler');

- Give Amy a grade of A for CSCI 103
  - INSERT INTO grade (classID, studentID, letterGrade) VALUES ((SELECT classID FROM class WHERE prefix='CSCI' AND number=103), (SELECT studentID FROM student WHERE fname='Amy' AND lname='Farrah Fowler'), 'A');

- Give Amy a grade of B+ for CSCI 170
  - INSERT INTO grade (classID, studentID, letterGrade) VALUES ((SELECT classID FROM class WHERE prefix='CSCI' AND number=170), (SELECT studentID FROM student WHERE fname='Amy' AND lname='Farrah Fowler'), 'B+');
SQL Queries

- Write SQL code to do the following
  - Print out all of the classes offered
  - Print out all of the students without the studentID value
  - Print out the name and grade for each student in CSCI 103
  - Print out the class and grades for Sheldon Cooper
  - Print out the name, grade, and class for all students in all of the classes
SQL Queries Solution

- Write SQL code to do the following
  - Print out all of the classes offered
    - SELECT * FROM class;
  - Print out all of the students without the studentID value
    - SELECT fname, lname FROM student;
  - Print out the name and grade for each student in CSCI 103
    - SELECT s.fname, s.lname, g.letterGrade FROM student s, grade g, class c WHERE s.studentID=g.studentID AND g.classID=c.classID AND c.prefix='CSCI' AND c.number=201;
  - Print out the class and grades for Sheldon Cooper
    - SELECT c.prefix, c.number, g.letterGrade FROM student s, grade g, class c WHERE c.classID=g.classID AND s.studentID=g.studentID AND s.fname='Sheldon' AND s.lname='Cooper';
  - Print out the name, grade, and class for all students in all of the classes
    - SELECT s.fname, s.lname, c.prefix, c.number, g.letterGrade FROM student s, grade g, class c WHERE c.classID=g.classID AND s.studentID=g.studentID ORDER BY s.fname, s.lname, c.prefix, c.number;