Databases and SQL
CSCI 201L

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

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

- Databases
- SQL
- Try It!
Databases

- Database systems store data and provide a means of accessing, updating, manipulating, and analyzing data.

- A Database Management Systems (DBMS) is designed for programmers, not casual users.
  - MySQL
  - Oracle
  - Microsoft SQL Server
  - IBM DB2
  - PostgreSQL
Relational Databases

- Relational database management systems (RDBMS) provide three things
  - Structure – the representation of the data
  - Integrity – constraints on the data
  - Language – means for accessing and manipulating data

Tables are called relations
Columns are called attributes
Rows are called tuples
Connections are called relationships
Outline

- Databases
- SQL
- Try It!
SQL

- The Structured Query Language (SQL) is the primary language supported by DBMSs for accessing and manipulating data

- Some MySQL SQL statements you should know
  - SHOW
  - CREATE DATABASE
  - USE
  - CREATE TABLE
  - INSERT
  - UPDATE
  - SELECT
  - DELETE
  - DROP

- If you are not familiar with SQL, there are many good tutorials online and our textbook has a good chapter reference as well
SQL Scripts

- After installing a DBMS, you will be able to run SQL scripts
- SQL scripts are files that contain a collection of SQL statements that can be used for recreating databases and populating them with initial or testing data
  - This is often used in the testing phase of software engineering to test different scenarios without requiring the QA engineers to insert all of the data manually
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.
Outline

- Databases
- SQL
- Try it!
SQL Queries

- Write SQL code based on the database schema provided on slide 7 to do the following
  - 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
  - Change Leonard’s grade in EE 101 to a C+
  - Add Amy Farrah Fowler into the Student table
  - Give Amy grades of A for CSCI 103
SQL Queries Solution

- Write SQL code to do the following
  - Print out the name and grade for each student in CSCI 103
    ```sql
    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
    ```sql
    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
    ```sql
    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;
    ```
  - Change Leonard’s grade in EE 101 to a C+
    ```sql
    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
    ```sql
    INSERT INTO student (fname, lname) VALUES ('Amy', 'Farrah Fowler');
    ```
  - Give Amy a grade of A for CSCI 103
    ```sql
    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');
    ```