



# Databases

CSCI 201

Principles of Software Development

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



# Outline

- Databases
- SQL
- Try It!

# Databases



- Database systems store data and provide a means of accessing, updating, manipulating, and analyzing data
- Databases are stored in files on a file system, but they are arranged in a manner that allows for fast queries and updates
- A Database Management System (DBMS) is designed for programmers, not casual users

## SQL-Based

MySQL  
PostgreSQL  
Oracle  
Microsoft SQL Server  
IBM DB2

## NoSQL

MongoDB  
Firebase  
Apache CouchDB  
Apache Cassandra



# 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

OverallGrades

prefix	num	fname	lname	letterGrade
CSCI	103	Sheldon	Cooper	A
CSCI	104	Howard	Wolowitz	A-
CSCI	201	Leonard	Hofstadter	A
CSCI	201	Howard	Wolowitz	B
EE	101	Howard	Wolowitz	B-

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

# Relational Databases



OverallGrades

prefix	num	fname	lname	letterGrade
CSCI	103	Sheldon	Cooper	A
CSCI	104	Howard	Wolowitz	A-
CSCI	201	Leonard	Hofstadter	A
CSCI	201	Howard	Wolowitz	B
EE	101	Howard	Wolowitz	B-

Non-normalized

Class

classID	prefix	num
1	CSCI	103
2	CSCI	104
3	CSCI	201
4	EE	101

Normalized

Grades

gradeID	classID	studentID	letterGrade
1	1	1	A
2	2	3	A-
3	3	2	A
4	3	3	B
5	4	3	B-

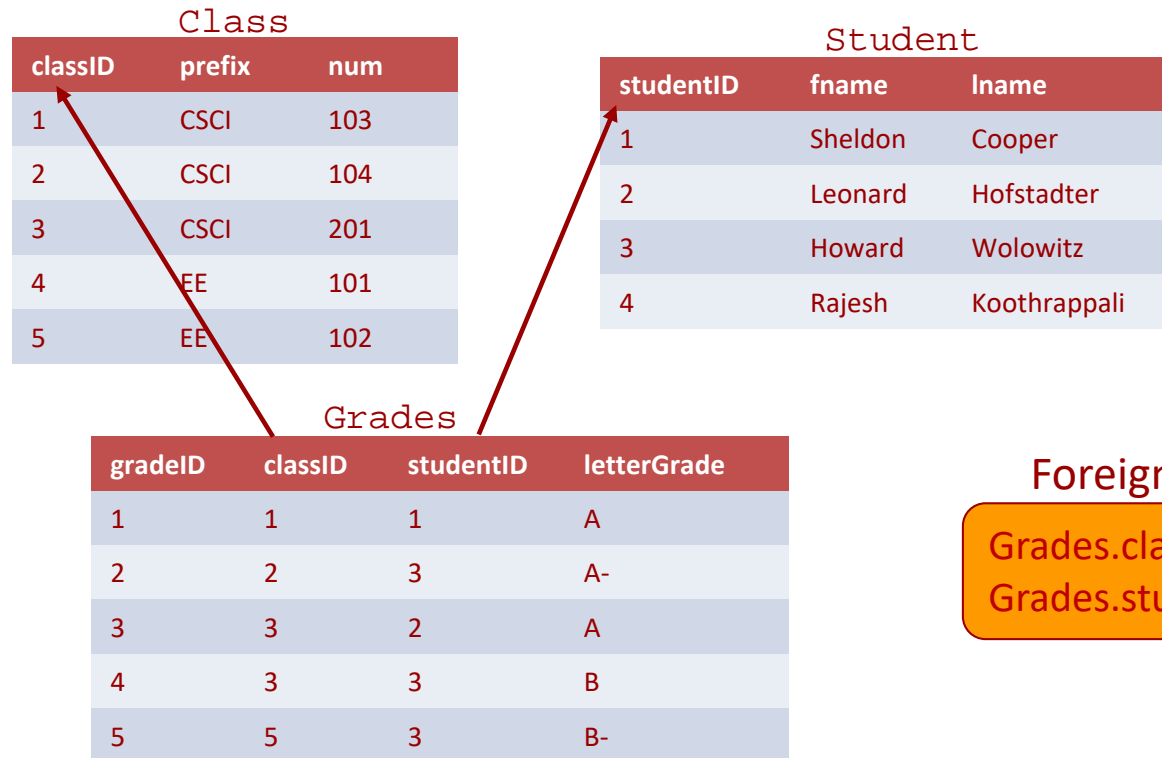
Student

studentID	fname	lname
1	Sheldon	Cooper
2	Leonard	Hofstadter
3	Howard	Wolowitz
4	Rajesh	Koothrappali

# Primary and Foreign Keys



- A **primary key** is a column (or a combination of multiple columns) in a table that provides a unique reference to a row in the table
- A **foreign key** is a link between two tables that uniquely identifies a row in another table



## Primary Keys

Class.classID  
Student.studentID  
Grades.gradeID

## Foreign Keys

Grades.classID  
Grades.studentID

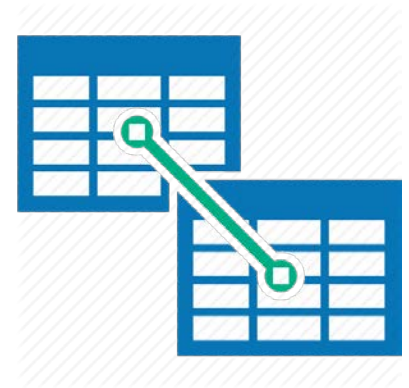


# Outline

- Databases
- SQL
- Try It!



- 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



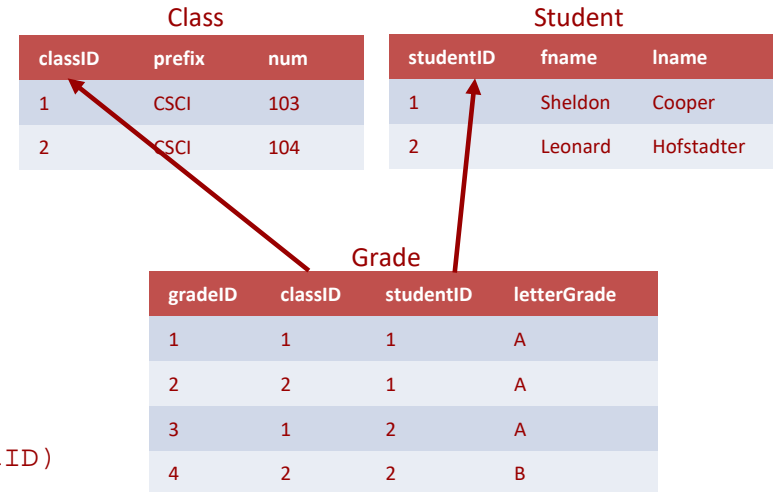
- After installing a DBMS, you will be able to run SQL scripts
- SQL scripts are files that contain a collection of SQL statements that can be used for recreating databases and populating them with initial or testing data
  - › This is often used in the testing phase of software engineering to test different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



# Sample Script



```
1 DROP DATABASE IF EXISTS StudentGrades;
2 CREATE DATABASE StudentGrades;
3 USE StudentGrades;
4 CREATE TABLE Student (
5     studentID INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
6     fname VARCHAR(50) NOT NULL,
7     lname VARCHAR(50) NOT NULL
8 );
9 INSERT INTO Student (fname, lname) VALUES ('Sheldon', 'Cooper');
10 INSERT INTO Student (fname, lname) VALUES ('Leonard', 'Hofstadter');
11 CREATE TABLE Class (
12     classID INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
13     prefix VARCHAR(5) NOT NULL,
14     num INT(4) NOT NULL
15 );
16 INSERT INTO Class (prefix, num) VALUES ('CSCI', 103);
17 INSERT INTO Class (prefix, num) VALUES ('CSCI', 104);
18 CREATE TABLE Grade (
19     gradeID INT(11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
20     classID INT(11) NOT NULL,
21     studentID INT(11) NOT NULL,
22     letterGrade VARCHAR(2) NOT NULL,
23     FOREIGN KEY fk1(classID) REFERENCES class(classID),
24     FOREIGN KEY fk2(studentID) REFERENCES student(studentID)
25 );
26 INSERT INTO Grade (studentID, classID, letterGrade) VALUES (1, 1, 'A');
27 INSERT INTO Grade (studentID, classID, letterGrade) VALUES (1, 2, 'A');
28 INSERT INTO Grade (studentID, classID, letterGrade) VALUES (2, 1, 'A');
29 INSERT INTO Grade (studentID, classID, letterGrade) VALUES (2, 2, 'B');
```



# SQL Visualization Tools



- A visualization tool connects to a database and allows execution of SQL statements
  - › The tool then will graphically display the results
- Free MySQL Clients
  - › Command line client (Windows, Mac, Linux)
  - › MySQL Workbench (Windows, Mac, Linux)
  - › Sequel Pro (Mac)
  - › Toad for MySQL (Windows)

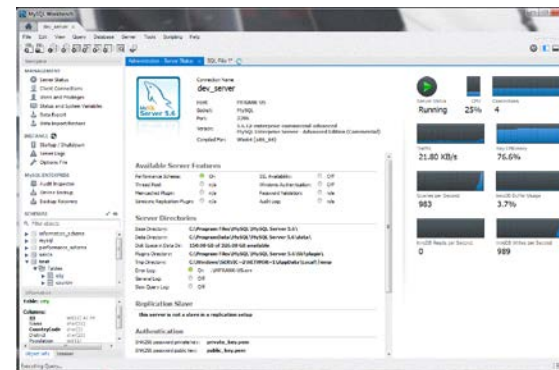
```
MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.0.37-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> CREATE DATABASE testdb;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| testdb |
+-----+
3 rows in set (0.02 sec)

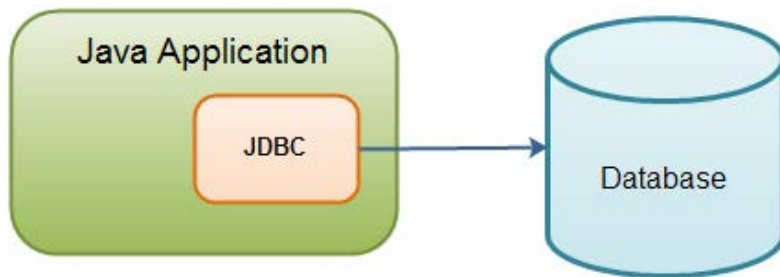
mysql>
```



# Executing SQL



- There are a few ways to execute SQL statements, which typically depend on the DBMS
- 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!



- Write SQL code to create the following database

