#### Databases

### **An Introducation**

#### What is a database?

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated.

This is *NOT* a database class, so we will ignore most things, but from an Systems Administration perspective, knowing how to install, configure, and manage a database is important. Rather, being able to do these to a DBMS is important.

#### What is a DBMS?

- Database Management System.
  - Allows us to manage a collection of databases
  - Also consists of access controls, user information, and database metadata

#### **DBMS Notes**

- Users are independent of system users
- DBMS has own set of users and passwords
- Users are identified by username *AND* hostname • joe@localhost and joe@www.thegummibear.com are two different users
- Access controls are given on a per-database / per-user instance
- User and access control management are administrative actions (the kind we care about)

#### A few random terms

- Database
  - A collection of tables
- Table
  - A collection of rows and columns (in db lingo, these are records and fields respectively)

#### **DBMS** Actions we care about

- User management
  - add, modify, delete
- Database management
  add, delete databases
- Permissions or user account controls for a database

#### Non-administrative actions (we only moderately care about)

- Create, Modify, Delete tables
- Add, update, delete records of a table
- Retrieve information from a table
- With RDBMS, these are usually performed using SQL.

#### DBMS commands we really care about

- To add a user
  - create user 'fred'@'localhost' identified with mysql\_native\_password by 'SecretPass123?'.
- To remove a user:
  - o drop user 'fred'@'localhost';
- Grant permissions
  - grant all privileges on sampledb.\* to 'fred'@'localhost';
  - this is the most common grant statement, but instead of all, you could grant one or all of select, insert, update, delete (and more).
- Revoke permissions
  - o revoke all privileges on sampledb.\* from 'fred'@'localhost';

## DBMS commands we kind of care about

- Select a database to perform operations on
  - use mysql;
  - use sampledb;
- See permissions for a user (after you have selected the mysql db)
  select User, Host, Db from db;
- See users that you have created
  - select User,Host from user;

# DBMS commands we don't really care about but we need to be able to do in order to test our permissions

- Create a database
  - ° create database testdb
- Use it
  - ° use testdb
- Show tables in the db
- o show tables;
- Create a new table (this can be very complex, take the database class for this)
  - create table numbers (x INT);
  - $\circ$  Creates a table called numbers with a single field named  ${\bf x}$  which is designed to store an integer.
  - create table garbage (name VARCHAR(10), age INT);
  - Creates a table named garbage with 2 fields, name and age. Other than that, it is beyond our course.

# DBMS commands we don't really care about but we need to be able to do in order to test our permissions

- Delete a table
  - drop table numbers
- Insert record (row) into table
  - insert into numbers values (10);
  - o insert into garbage VALUES ('tommy', 30);
- Delete a record
  - o delete from garbage where name='tommy';
  - You probably don't really need to know this one.
- Get info from a table
  - Select \* from numbers
    - $\circ~$  Shows all the records from the numbers table

# **RDBMS Install (Server)**

- Install mysql-server package
- Install mysql-client package
- Remove unwanted users and database
- Enable remote connections
- Manage users and databases

## **RDBMS Install (Client)**

A client may want to access a database. I.e. a web server may want to load dynamic content from a database onto a website.

Steps:

- connect to db server socket (3306?)
- authenticate with user/pass
- choose db to use
- manipulate tables of db (or read data)
- I usually install the mysql-tools package to test connectivity