

CS320 Web and Internet Programming SQL and MySQL

Chengyu Sun
California State University, Los Angeles

Web and Databases

- ◆ E-commerce sites
 - Products, order, customers
- ◆ News sites
 - Subscribers, articles
- ◆ Web boards
 - Users, postings
- ◆ ... anywhere where a large amount of information needs to be managed safely and efficiently

Database vs. File

- ◆ More efficient search
- ◆ ACID
 - Atomicity
 - Consistency
 - Isolation
 - Durability

Relational Model

- ◆ Proposed by Edgar F. Codd in early 1970's
- ◆ All major DBMS are relational (and the good ones are *object-relational*)

A Relational DB Example

orders

OID	CID	ODATE	SDATE
1	1	4/29/2005	NULL
2	2	3/20/2005	3/37/2005

customers

CID	FNAME	LNAME	ADDRESS
1	Chengyu	Sun	Street #215
2	Steve	Sun	Street #711

products

PID	Description	Price
1	Intel P4	\$200
2	Intel P3	\$49
3	AthlonXP	\$100
4	ASUS	\$128
5	TYAN	\$400

order_details

OID	PID	Quantity
1	1	2
1	5	2
2	2	1

Terminology

- ◆ Database Management System (DBMS)
- ◆ Database
- ◆ Table, relation
- ◆ Attribute, field
 - Type
- ◆ Record, tuple, row
- ◆ Column
- ◆ Schema

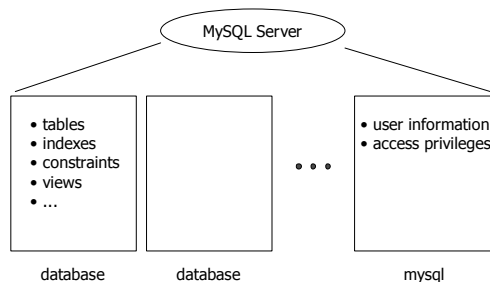
SQL

- ◆ Standard query language of relational databases
- ◆ Supported by all major relational databases with some variations

MySQL

- ◆ Not a good DBMS in the traditional sense
- ◆ Very popular in web development
 - Very fast search
 - Full text indexing and search
 - Many small things
 - drop if exists
 - insert into values
 - /* */
 - ...

Databases in MySQL



MySQL on the CS Server

- ◆ Version 4.1.10a
- ◆ One database per user
 - DB name is the same as the server account user name. E.g. cs320stu31
 - Username and password are the same as the ones for the server account
- ◆ Connect to the database
 - `mysql -p`

mysql Command Line Options

- ◆ `mysql [database]`
- ◆ `-?`
- ◆ `-u username`
 - default: current user
- ◆ `-p`
 - required if the password for the account is not empty
- ◆ `-h hostname`
 - default: localhost

Some MySQL Commands

- ◆ Status
 - `status;`
- ◆ Help
 - `\h` or `help;`
- ◆ Quiet MySQL client
 - `\q` or `quit;` or `exit;`
- ◆ Change password
 - `set password = password ('something');`
 - `set password for 'user'@'host' = password('something');`

More about MySQL Passwords

- ◆ Each <user,hostname> pair has a corresponding password
 - E.g. the password for <cs320stu31, localhost> is *different* from the one for <cs320stu31, localhost.localdomain>
- ◆ On Redhat/Fedora, localhost has three names:
 - localhost – default for `mysql`
 - localhost.localdomain – default for `tomcat`
 - cs.calstatela.edu

More MySQL Commands ...

- ◆ Show databases
 - `show databases;`
- ◆ Use database
 - `use dbname;`
- ◆ Show tables
 - `show tables;`
- ◆ Show table schema
 - `describe tablename;`

... More MySQL Commands

- ◆ Run a script
 - `\. demo.sql` or `source demo.sql;`
- ◆ Run a script at command prompt
 - `mysql < demo.sql`

Create a Table

```
create table table_name (  
  field_name field_type [NOT NULL] [UNIQUE] [DEFAULT value],  
  field_name field_type [NOT NULL] [UNIQUE] [DEFAULT value],  
  ...  
  [PRIMARY KEY(field_name, ...)]  
);  
  
create table products (  
  prod_id char(8) not null, -- product id  
  description text, -- product description  
  price decimal(12,2), -- price  
  primary key (prod_id)  
);
```

Field Types

- ◆ Numerical types
 - int, float, double, decimal(m,n)
- ◆ String types
 - char(n), varchar(n)
- ◆ Date and time
 - date, time, datetime, timestamp
 - ♦ 'yyyy-mm-dd hh:mm:ss'

Auto Increment Field

```
create table users (  
  id int auto_increment primary key,  
  username varchar(64) not null unique,  
  password char(16)  
);  
  
insert into users (username,password) values ('cysun','abcd');  
insert into users (username,password) values ('csun','xyz');
```

Populate Tables

- ◆ Insert a record
 - insert into orders values (1000, 1, '2004-04-29', '2004-05-01');
 - insert into orders values (1001, 2, '2004-05-01', NULL);
- ◆ Load a data file
 - load data local infile 'orders.txt' into table orders;
- ◆ Import a data file (at command prompt)
 - mysqlimport -u cs320stu31 -p orders.txt
 - ◆ \N for NULL

Search for Records

- select field(s) from table(s) where condition(s);
- ◆ select description, price from products;
 - ◆ select * from products;
 - ◆ select * from products where price < 300;
 - ◆ select * from products where prod_id = 'cpu-0001';

Pattern Matching

- ◆ LIKE, REGEXP
 - % -- any zero or more characters
 - . -- any single character
 - [abc], [a-z], [0-9] -- range
 - * -- zero or more instances of the preceding character
 - ^ -- beginning of a string
 - \$ -- end of a string
- ◆ select * from products where description like '%intel%';

Update Records

- update table set field=value [, ...] where condition(s);
- ◆ update products set price=320 where prod_id = 'cpu-0001';
 - ◆ update products set price=200, description='Intel Pentium M 1.7GHz' where prod_id = 'cpu-0001';

Delete Records

- delete from table where condition(s);
- ◆ Examples:
 - delete from orders;
 - delete from orders where order_date < '2003-12-31' and ship_date is not null;
 - ◆ Drop a database
 - drop database *cs320stu31*; -- Don't do this!
 - ◆ Drop a table
 - drop table *products*;

Schema Design Example ...

◆ Customer, Product, Order

```
public class Customer {
    int id;
    String lastName;
    String firstName;
    String address;
}

public class Product {
    int id;
    String description;
    double price;
}
```

... Schema Design Example

```
public class Order {  
    int id;  
    Date dateOrdered;  
    Date dateShipped;  
  
    Customer customer;  
    Map<Product, int> products;  
}
```

Simple Schema Design Rules

<u>OO</u>		<u>Relational</u>
Class		Table
Class variables		Attributes
Java types	⇒	SQL types
References		ID
Collection		New Table

Exercises

- ◆ Read MySQL Reference Manual
 - String functions
 - Date and time functions