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

<table>
<thead>
<tr>
<th>orders</th>
<th>CID</th>
<th>ODATE</th>
<th>SDATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4/26/2005</td>
<td>NULL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>products</th>
<th>PID</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intel PI</td>
<td>$200</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>AthlonXP</td>
<td>$99</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ASUS</td>
<td>$128</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TYAN</td>
<td>$400</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>customers</th>
<th>CID</th>
<th>FName</th>
<th>LNAME</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chengyu</td>
<td>Sun</td>
<td>Street #215</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Steve</td>
<td>Sun</td>
<td>Street #211</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>order_details</th>
<th>OID</th>
<th>PID</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

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 Server

- `tables`
- `indexes`
- `constraints`
- `views`
- `...`

- `user information`
- `access privileges`

- `database`
- `database`
- `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;`
- `Quite 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

```sql
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, ...)]
```
```sql
create table products (      
    prod_id char(8) not null, -- product id      
    description text, -- product description      
    price decimal(12,2), -- price      
    primary key (prod_id)  
) ;
```
```sql
insert into users (username,password) values ('cysun','abcd');
insert into users (username,password) values ('csun','xyz');
```

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

```sql
create table users (  
    id int auto_increment primary key,  
    username varchar(64) not null unique,  
    password char(16)  
) ;
```
```sql
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

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

  public class Product {
      int id;
      String description;
      double price;
  }
  ```
... Schema Design Example

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

Simple Schema Design Rules

<table>
<thead>
<tr>
<th>OO</th>
<th>Relational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Table</td>
</tr>
<tr>
<td>Class variables</td>
<td>Attributes</td>
</tr>
<tr>
<td>Java types</td>
<td>SQL types</td>
</tr>
<tr>
<td>References</td>
<td>ID</td>
</tr>
<tr>
<td>Collection</td>
<td>New Table</td>
</tr>
</tbody>
</table>

Exercises

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