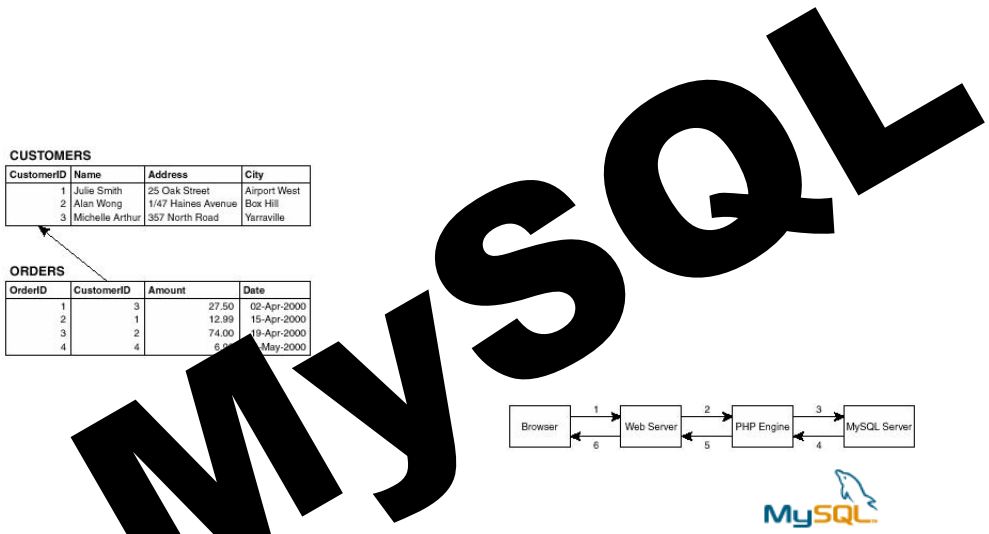


09 MySQL I

MySQL I

- Designing Your Web Database
  - Why Database?
  - Relational Database Concepts
  - How to Design Your Web Database
  - Web Database Architecture
- Creating Your Web Database
- Workshop

Designing Your Web Database

- Why Database?
  - RDBMSs (Relational Database Management Systems) can provide faster access to data than flat files.
  - RDBMSs can be easily queried to extract sets of data that fit certain criteria.
  - RDBMSs have built-in mechanisms for dealing with concurrent access so that you as a programmer don ’t have to worry about it.
  - RDBMSs provide random access to your data.
  - RDBMSs have built-in privilege systems.

Designing Your Web Database

- Relational Database Concepts
  - Relational databases are, by far, the most commonly used type of database.
  - Table
    - Relational databases are made up of relations, more commonly called **tables**.

CUSTOMERS

CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Yarraville

Figure 7.1 Book-O-Rama’s customer details are stored in a table.

- The table has a name (CUSTOMERS), a number of columns, each corresponding to a different piece of data, and rows that correspond to individual customers.

Relational Database Concepts (cont.)

- Column

CUSTOMERS

CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Yarraville

Figure 7.1 Book-O-Rama's customer details are stored in a table.

- Each column in the table has a unique name and contains different data.
- Each column has an associated data type.
  - For instance, in the CUSTOMERS table in Figure 7.1, you can see that CustomerID is an integer and the other three columns are strings.
- Columns are sometimes called **fields** or **attributes**.

Relational Database Concepts (cont.)

- Row

CUSTOMERS

CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Yarraville

Figure 7.1 Book-O-Rama's customer details are stored in a table.

- Each row in the table represents a different customer.
- Because of the tabular format, they all have the same attributes.
- Rows are also called records or tuples.

- Values
  - Each row consists of a set of individual values that correspond to columns.
  - Each value must have the data type specified by its column.

Relational Database Concepts (cont.)

- Keys
  - We need to have a way of identifying each specific customer by assigning a unique CustomerID e.g., bank account number or club membership number.
  - Databases usually consist of multiple tables and use a key as a reference from one table to another.
    - Each row in the ORDERS table represents a single order, placed by single customer.
    - We know who the customer is because we store their CustomerID.
    - The relational database term for this relationship is **foreign key**.
      - » CustomerID is the **primary key** in CUSTOMERS, but when it appears in another table, such as ORDERS, it is referred to as a **foreign key**.

CUSTOMERS

CustomerID	Name	Address	City
1	Julie Smith	25 Oak Street	Airport West
2	Alan Wong	1/47 Haines Avenue	Box Hill
3	Michelle Arthur	357 North Road	Yarraville

ORDERS

OrderID	CustomerID	Amount	Date
1	3	27.50	02-Apr-2000
2	1	12.99	15-Apr-2000
3	2	74.00	19-Apr-2000
4	4	6.99	01-May-2000

Figure 7.2 Each order in the Orders table refers to a customer from the Customers table.

Relational Database Concepts (cont.)

- Schemas
  - The complete set of the table designs for a database is called the database *schema.*
    - Akin to a blueprint for the database.
    - Show the tables along with their columns, the data types of the columns and indicate the **primary key** of each table and any **foreign keys**.
      - » CUSTOMERS(CustomerID, Name, Address, City)
      - » ORDERS(OrderID, CustomerID, Amount, Date)
- Relationship
  - Foreign keys represent a relationship between data in two tables.
  - Three basic kinds of relationships exist in a relational database.
  - Relationships can be either one-to-one, one-to-many, or many-to-many.
    - **one-to-one** relationship: there is one of each thing in the relationship.
    - **one-to-many** relationship: one row in one table is linked to many rows in another table.
    - **many-to-many** relationship: many rows in one table are linked to many rows in another table.

How to Design Your Web Database

- Think About the Real World Objects You Are Modeling
  - For example: In the Book-O-Rama
    - We want to store information about our customers, the books that we sell, and details of the orders.
      - » The customers all have name and address.
      - » The books have an ISBN, an author, a title, and a price.
      - » The orders have date, total amount, and a set of books that were ordered.
- Avoid Storing Redundant Data
  - Waste of space and Update anomalies
    - Modification anomaly
      - » Occur when we are trying to modify the database.
    - Insertion anomaly
      - » Occur when we are trying to insert information into a database
      - » For example, one row might tell us that Julie lives in **Airport West,** and another might tell us she lives in **Airport**.
    - Deletion anomaly
      - » Occur when we are deleting rows from the database.
      - » For example, when an order has been shipped, we delete it from the database.

How to Design Your Web Database (cont.)

- Use Atomic Column Values
  - Each attribute in each row, we store only one thing.
    - For example, we need to know what books make up each order. There are several ways we could do this.
      - » We could add a column to the Orders table which lists all the books that have been ordered, as shown in Figure 7.5, but **this is not a good idea for a few reasons.**

ORDERS

OrderID	CustomerID	Amount	Date	Books Ordered
1	3	27.50	02-Apr-2000	0-672-31697-8
2	1	12.99	15-Apr-2000	0-672-31745-1, 0-672-31509-2
3	2	74.00	19-Apr-2000	0-672-31697-8
4	4	6.99	01-May-2000	0-672-31745-1, 0-672-31509-2, 0-672-31697-8

4

Figure 7.5 With this design, the Books Ordered attribute in each row has multiple values.

ORDERS

OrderID	CustomerID	Amount	Date
1	3	27.50	02-Apr-2000
2	1	12.99	15-Apr-2000
3	2	74.00	19-Apr-2000
4	4	6.99	01-May-2000

ORDER_ITEMS

OrderID	ISBN	Quantity
1	0-672-31697-8	1
2	0-672-31745-1	2
2	0-672-31509-2	1
3	0-672-31697-8	1
4	0-672-31745-1	1
4	0-672-31509-2	2
4	0-672-31697-8	1

How to Design Your Web Database (cont.)

- Choose Sensible Keys
  - Make sure that the keys you choose are unique, e.g., CustomerID, etc.
- Think About the Questions You Want to Ask the Database
  - Make sure that the database contains all the data required, and that the appropriate links exist between tables to answer the questions you have.
- Avoid Designs with Many Empty Attributes

May have a null value

BOOKS

ISBN	Author	Title	Price	Review
0-672-31687-8	Michael Morgan	Java 2 for Professional Developers	34.99	
0-672-31745-1	Thomas Down	Installing Debian GNU/Linux	24.99	
0-672-31509-2	Pruitt, et al.	Teach Yourself GIMP in 24 Hours	24.99	

Have no null value

BOOK_REVIEWS

ISBN	Review

Figure 7.7 To add reviews, we can either add a Reviews column to the Books table, or add a table specifically for reviews.

Web Database Architecture

- Basic Operation of the Web Server
  - Delivering a static page

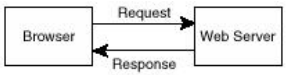


Figure 7.8 The client/server relationship between a Web browser and Web server requires communication.

  - Delivering a dynamic page

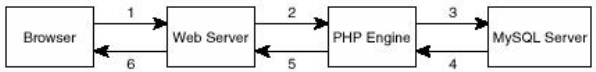


Figure 7.9 The basic Web database architecture consists of the Web browser, Web server, scripting engine, and database server.

Creating Your Web Database

- A Note on Using the MySQL Monitor (XAMPP)
- Introduction to MySQL's Privilege System
- Setting Up a User for the Web
- Using the Right Database
- Creating Database Tables
- Getting into MySQL Prompt

Creating Your Web Database

- A Note on Using the MySQL Monitor (XAMPP)
  - SQL statements end each command with a semicolon (;).
  - If you leave off the semicolon, nothing will happen.
    - For example:

```
MariaDB [(none)]> grant select
->
```

      - This means MySQL is expecting more input.
  - SQL statements are not case sensitive, but database and table names are case sensitive in Linux.

Creating Your Web Database

- Introduction to MySQL's Privilege System
  - Principle of Least Privilege
    - A *privilege* is the right to perform a particular action on a particular object, and is associated with a particular user.
    - The concept is very similar to file permissions.
    - When you create a user within MySQL, you grant her a set of privileges to specify what she can and cannot do within the system.
    - A user (or process) should have the lowest level of privilege required in order to perform his assigned task.

Creating Your Web Database

- Introduction to MySQL's Privilege System (cont.)
  - The GRANT command is used to create users and give them privileges.
  - The general form of the GRANT command is as follows:

```
GRANT privileges [columns]ON item
TO user_name [IDENTIFIED BY 'password']
[WITH GRANT OPTION]
```

Note: WITH GRANT OPTION allows the specified user to grant her own privileges to others

  - Types and Levels of Privilege
    - Three basic types of privileges exist in MySQL.
      - Privileges suitable for granting to regular users.
      - Privileges suitable for administrators.
      - A couple of special privileges.

Introduction to MySQL's Privilege System (cont.)

- Privileges suitable for granting to regular users.

Privilege	Applies To	Description
SELECT	tables, columns	Allows users to select rows (records) from tables.
INSERT	tables, columns	Allows users to insert new rows into tables.
UPDATE	tables, columns	Allows users to modify values in existing table rows.
DELETE	tables	Allows users to delete existing table rows.
INDEX	tables	Allows users to create and drop indexes on particular tables.
ALTER	tables	Allows users to alter the structure of existing tables by, for example, adding columns, renaming columns or tables, and changing data types of columns.
CREATE	databases, tables	Allows users to create new databases or tables. If particular database or table is specified in the GRANT they can only CREATE that database or table, which means they will have to DROP it first.
DROP	databases, tables	Allows users to drop (delete) databases or tables.

Introduction to MySQL's Privilege System (cont.)

- Privileges suitable for administrators.

Privilege	Description
RELOAD	Allows an administrator to reload grant tables and flush privileges, hosts, logs, and tables.
SHUTDOWN	Allows an administrator to shut down the MySQL server.
PROCESS	Allows an administrator to view server processes and kill them.
FILE	Allows data to be read into tables from files and vice versa.

- A couple of special privileges.

Privilege	Description
ALL	Grants all the privileges listed in Tables 8.1 and 8.2.You can also write ALL PRIVILEGES instead of ALL.
USAGE	Grants no privileges. This will create a user and allow her to log on, but it won 't allow her to do anything. Usually you will go on to add more privileges later.

Introduction to MySQL's Privilege System (cont.)

- The REVOKE Command
  - It is used to take privileges away from user. It is very similar to GRANT in syntax:

```
REVOKE privileges [(columns)]
ON item
FROM user_name
```

- Example Using GRANT and REVOKE

```
MariaDB [(none)]> grant all
    -> on *
    -> to fred identified by 'mnb123'
    -> with grant option;
```

```
MariaDB [(none)]> revoke all
    -> on *
    -> from fred;
```


Setting Up a User for the Web

- In most cases they'll only need to SELECT INSERT DELETE and UPDATE rows from tables. You can set this up as follows:

```
MariaDB [(none)]> grant select,insert,delete,update
    -> on books.*
    -> to bookorama identified by 'bookorama123';
```

- Grant select, insert, delete, update on all tables in a database called "books" to user "bookorama" with password "bookorama123".

- Logging Out as root
  - You can log out of the MySQL monitor by typing **q**uit or exit.

Creating Your Web Database

Using the Right Database

- The first thing you'll need to do when you log in is to specify which database you want to use. You can do this by typing:

```
MariaDB [(none)]> use dbname;
```

  - where *dbname* is the name of your database.
  - In this example, we'll use the books database:

```
MariaDB [(none)]> use books;
```

  - When you type this command, MySQL should give you a response such as

```
Database changed
```

  - If you don't select a database before starting work, MySQL will give you an error message such as

```
ERROR 1046: No Database Selected
```

  - Alternatively, you can avoid the *use* command by specifying the database when you log in at the command line, as follows:

```
C:\xampp\mysql\bin> mysql.exe -D dbname -u username -p
```


Creating Your Web Database

Creating Database Tables

- Here 's the Book-O-Rama schema:

```
Customers(CustomerID,Name,Address,City)
Orders(OrderID,CustomerID,Amount,Date)
Books(ISBN,Author,Title,Price)
Order_Items(OrderID,ISBN,Quantity)
Book_Reviews(ISBN,Review)
```

```
create table Customers
( CustomerID int unsigned not null auto_increment primary key,
  Name char(30) not null,
  Address char(40) not null,
  City char(20) not null);

create table Orders
( OrderID int unsigned not null auto_increment primary key,
  CustomerID int unsigned not null,
  Amount float(6,2),
  Date date not null);
```

```
Format of data type "date" is YYYY-MM-DD
```


Creating Your Web Database

Creating Database Tables (cont.)

- Here 's the Book-O-Rama schema:

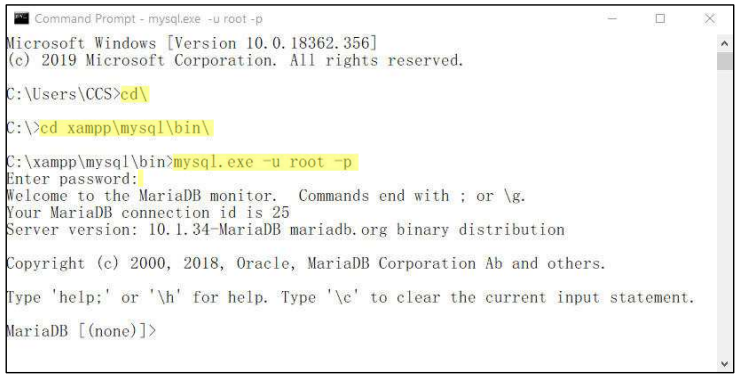
```
create table Books
( ISBN char(13) not null primary key,
  Author char(30),
  Title char(60),
  Price float(4,2));

create table Order_Items
( OrderID int unsigned not null,
  ISBN char(13) not null,
  Quantity tinyint unsigned,
  Primary key (OrderID, ISBN));

create table Book_Reviews
( ISBN char(13) not null primary key,
  Review text);
```

```
tinyint unsigned holds an integer 0 – 255
```


Creating Your Web Database

Getting into MySQL Prompt

- Going to MySQL Console (XAMPP for Windows)



Creating Your Web Database

- Getting int MySQL Prompt (cont.)
  - Creating a database and display all

```
Command Prompt - mysql.exe -u root -p

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| db_ndt             |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| test               |
| webapp621          |
+--------------------+
7 rows in set (0.00 sec)

MariaDB [(none)]>
```


Creating Your Web Database

- Getting int MySQL Prompt (cont.)
  - Creating a database and display all

```
Command Prompt - mysql.exe -u root -p

MariaDB [(none)]> create database webapp;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| db_ndt             |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| test               |
| webapp             |
| webapp621          |
+--------------------+
8 rows in set (0.00 sec)

MariaDB [(none)]>
```


Creating Your Web Database

- Getting int MySQL Prompt (cont.)
  - Using a database and create tables

Step 1

```
Command Prompt - mysql.exe -u root -p

MariaDB [(none)]> use webapp;
Database changed
```

Step 2

```
MariaDB [webapp]> create table student
    -> (studentid char(4) not null primary key,
    -> name char(30) not null,
    -> age tinyint not null,
    -> earn tinyint not null);
Query OK, 0 rows affected (0.07 sec)
```

Step 3

```
MariaDB [webapp]> create table register
    -> (no tinyint not null auto_increment primary key,
    -> studentid char(4) not null,
    -> register tinyint not null,
    -> date date not null,
    -> paid bool not null);
Query OK, 0 rows affected (0.10 sec)

MariaDB [webapp]>
```


Creating Your Web Database

- Getting int MySQL Prompt (cont.)
  - Displaying all tables and describe each table

```
Select Command Prompt - mysql.exe -u root -p

MariaDB [webapp]> show tables;
+-----------------+
| Tables_in_webapp |
+-----------------+
| register        |
| student         |
+-----------------+
2 rows in set (0.00 sec)

MariaDB [webapp]>
```

```
Command Prompt - mysql.exe -u root -p

MariaDB [webapp]> describe register;
+-----------+---------+------+-----+---------+----------------+
| Field     | Type    | Null | Key | Default | Extra          |
+-----------+---------+------+-----+---------+----------------+
| no        | tinyint(4) | NO | PRI | NULL | auto_increment |
| studentid | char(4) | NO   |     | NULL    |                |
| register  | tinyint(4) | NO |     | NULL    |                |
| date      | date    | NO   |     | NULL    |                |
| paid      | tinyint(1) | NO |     | NULL    |                |
+-----------+---------+------+-----+---------+----------------+
5 rows in set (0.02 sec)

MariaDB [webapp]>
```

```
Command Prompt - mysql.exe -u root -p

MariaDB [webapp]> describe student;
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| studentid | char(4) | NO   | PRI | NULL    |       |
| name      | char(30) | NO  |     | NULL    |       |
| age       | tinyint(4) | NO |     | NULL    |       |
| earn      | tinyint(4) | NO |     | NULL    |       |
+-----------+---------+------+-----+---------+-------+
4 rows in set (0.01 sec)

MariaDB [webapp]>
```

Creating Your Web Database

- Getting int MySQL Prompt (cont.)
  - Insert data into a table

```
MariaDB [abc]> show tables;
+---------------+
| Tables_in_abc |
+---------------+
| customers     |
+---------------+
1 row in set (0.001 sec)

MariaDB [abc]> describe customers;
+------------+------------------+------+-----+---------+----------------+
| Field      | Type             | Null | Key | Default | Extra          |
+------------+------------------+------+-----+---------+----------------+
| CustomerID | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| Name       | char(30)         | NO   |     | NULL    |                |
| Address    | char(40)         | NO   |     | NULL    |                |
| City       | char(20)         | NO   |     | NULL    |                |
+------------+------------------+------+-----+---------+----------------+
4 rows in set (0.024 sec)

MariaDB [abc]> insert into customers values
    ->     (NULL, "Julie Smith", "25 Oak Street", "Airport West");
Query OK, 1 row affected (0.037 sec)

MariaDB [abc]>
```


Workshop

- MySQL I: Introduction to MySQL
  - Page 235 (1-10)