USING DATABASES OVER THE WEB

Chapter 13

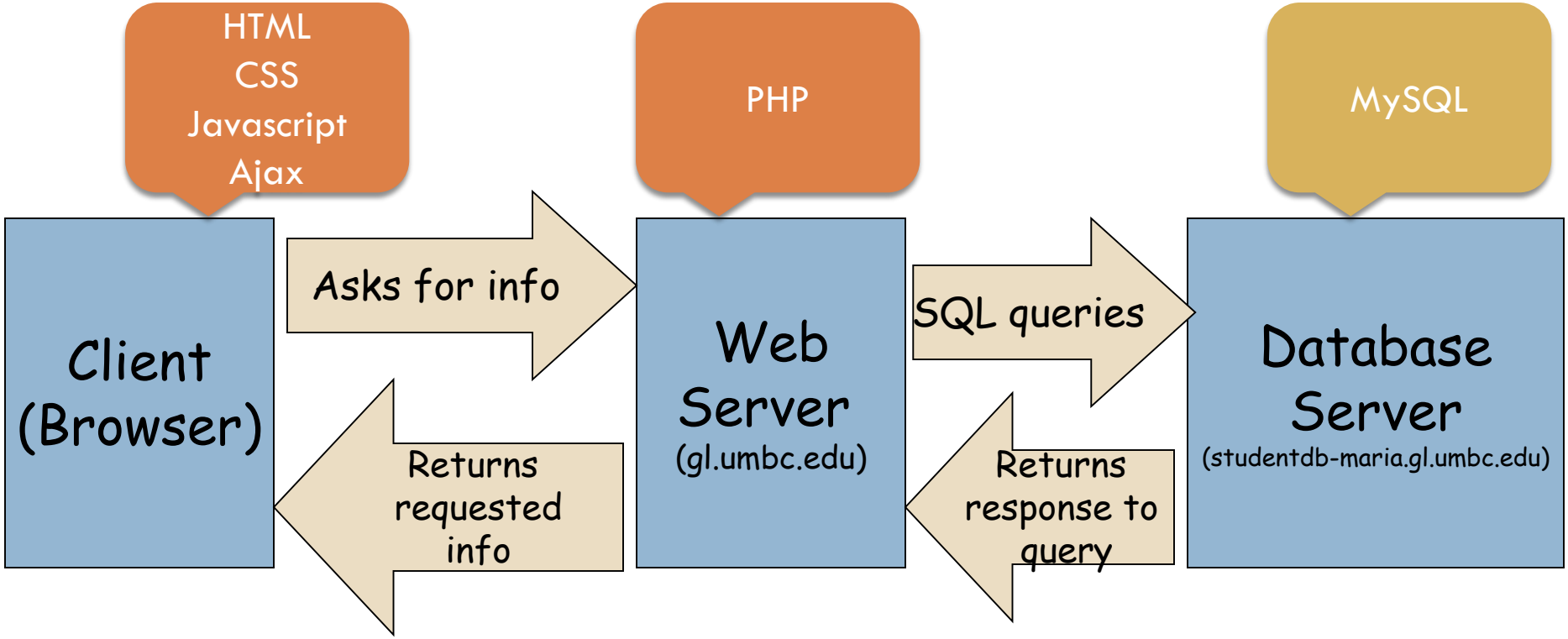
Note

- Examples for this chapter are at

https://swe.umbc.edu/~zzaidi1/is448/chap13-examples/

All PHP programs used in this chapter are zipped up as mysql1-php5.zip in the above examples folder

Client-Server-Database Architecture



13.3 Client-Server-Database Architecture

- Two-tier architecture
  - Client connects to the database to get information
  - Server or client performs computations and user interactions
- Problems with two-tier
  - Servers getting smaller so client software getting more complex
  - Keeping clients up to date difficult
- Three-tier architecture
  - Web server with applications sits between a browser and the database system
  - The web server accesses the database and carries out computations and deals with user interaction

13.1 Relational Databases

- A database stores data in a way allowing
  - Efficient changes
  - Efficient searching

- The relational model is currently the most popular model
  - Data is stored in tables
  - Columns are named
  - Each row contains values for each column, though some values may be missing
  - Rows are referred to as entities
  - The *primary key* is one or more columns in a table whose value(s) uniquely identify each row

- Example, cars table
  - Primary key is an id_number
  - Each row represents a different vehicle
  - Columns with description of the vehicles

car_id	car_name	car_description
1	Corvette	2-door
2	Accord	4-door
3	Altima	4-door

13.2 Structured Query Language

- SQL is a standardized language for manipulating and querying relational databases
- Although relational databases support SQL there may be some minor and some significant differences in the implementations

13.2 Structured Query Language

- SQL reserved words are not case sensitive
  - However, some systems may treat names such as column names as case sensitive
- SQL commands may have extra white space, including new lines, added to improve readability
- Single quotes ‘ are used to enclose literal strings

13.4 The MySQL Database System

- Popular free database system
- Most programming language libraries have some interface to MySQL

Connecting to the MySQL database

- Through the phpMyAdmin interface (see next slide)

Using phpMyAdmin Interface Access the MySQL Database on GL

- UMBC's OIT has also made available the phpMyAdmin interface to view and operate the database in a browser:

https://mysql-admin.umbc.edu/phpMyAdmin/

- Instructions to log on are here (see web browser section): https://wiki.umbc.edu/pages/viewpage.action?pageId=63800481
- You can change passwords, create tables, add rows to tables etc. through the web interface

Access the MySQL Database on GL

Access credentials

- **Host name: studentdb-maria.gl.umbc.edu**
- **Username: *your UMBC user name***
  - e.g., zzaidi1
- **Password: *your UMBC user name***
  - This is your default password to the MySQL database on GL
- Each of you has access to a database by the name: *your UMBC user name,* e.g., zzaidi1
- Can create tables inside this database.
- Cannot create new databases!

Basic Commands in MySQL

- CREATE: use to create new tables
- INSERT: use to enter a row into a table
- SELECT: use to retrieve a particular row from a table
- UPDATE: use to change the values in an existing row in a table
- DELETE: use to delete a single row from table
- DROP: use to delete a table altogether
  - Be cautious with the use of DROP
- Note: all commands in MySQL should end with a semi-colon

13.2 The CREATE TABLE Command

- Create a table with specified columns, each column having a specified type of data and satisfying certain constraints
- Syntax

CREATE TABLE *table_name(*
column_name_1 data_type constraints,
...
column_name_n data_type constraints);

- Most system support many data types
- Common types: INTEGER, REAL, DOUBLE, CHAR (*length)*

```
CREATE TABLE cars
(car_id INTEGER PRIMARY KEY NOT NULL,
car_name CHAR(20),
car_description CHAR(255));
```

13.2 Create Table Constraints

- The constraint NOT NULL causes an error to be raised if a row is inserted in which the corresponding column does not have a value
- The PRIMARY KEY constraint causes an error to be raised if a row is inserted in which the corresponding column has a value that equals the value in another row
  - This can be applied to a group of several columns if the primary key is multi-column

13.2 The SELECT Command

- Used to query databases
- The command returns a result, a virtual table

SELECT *column-names* FROM *table-name(s)* [WHERE *condition*];

  - The result table has columns as named
  - Rows are derived from the table named (also see the Join discussion on next slide about multiple tables)
  - The WHERE clause is optional
  - The WHERE clause specifies constraints on the rows selected
  - If * is used for the column names, all columns are selected

Example

```
SELECT car_description FROM cars WHERE car_name='Corvette';
```

13.2 Joins

- Example database tables
  - cars: car_id, car_name, car_description
  - equipment: equip_id, equipment_name, car_id
- Example query: list cars that have CD players
  - This involves two tables: cars, equipment
  - A virtual table is constructed with combinations of rows from the two tables cars and equipment: a *join* of the two tables
  - The WHERE clause selects which rows of the join are to be retained in the result

cars table

car_id	car_name	car_description
1	Corvette	…. Sports car ….
2	Accord	… Mid-range …..
3	Altima	.. Mid-size car….

equipment table

equip_id	equipment_name	car_id
1	CD player	1
2	CD Player	2
3	CD Player	3
4	Heated seats	1

13.2 A Query Using a Join

- Select and display all the equipment that a Corvette has

```
SELECT equipment.equipment_name, cars.car_name
FROM cars, equipment
WHERE equipment.car_id = cars.car_id
        AND cars.car_name = 'Corvette';
```

13.2 The INSERT Command

- Inserts a new row into a table
- Syntax

  INSERT INTO *table_name*
  (*column_name_1, ..., column_name_n*)
  VALUES (*value_1, value_2, ..., value_n*);
- The values provided will be placed into the corresponding columns
- Columns not named will receive no value
  - This will cause an error if the column was created with a NOT NULL constraint

```
INSERT INTO cars(car_id, car_name, car_description)
VALUES (1, 'Corvette', 'The Chevrolet Corvette is a sports car that has
                                    been manufactured by General Motors since 1953.');
```

13.2 The UPDATE Command

- Changes values in an existing row
- Syntax

  UPDATE *table_name*

  SET *column_name_1 = value_1,*

  …

  column_name_n = value_n

  WHERE *column_name = value*

- The WHERE clause identifies the row to be updated, probably by its primary key

```
UPDATE cars
SET car_description='Need new description'
WHERE car_name = 'Corvette';
```

13.2 The DELETE Command

- Removes one or more rows
- Syntax

```
DELETE FROM table_name
WHERE column_name = value;
```

- The WHERE clause determines which rows are deleted
- The sample syntax would probably be specifying a primary key value to identify one row
- However, the clause could be more general

```
DELETE FROM cars WHERE car_id = 3;
```

13.2 The DROP Command

- Remove a table or database from the system
  - A database system usually has several databases operating within it, essentially, named collections of tables
- Syntax

```
DROP (TABLE|DATABASE)[IF EXISTS] name;
```

- The IF EXISTS clause may be included to prevent an error indication if the table or database doesn't exist

```
DROP TABLE IF EXISTS equipment;
```

13.4 MySQL Commands

- MySQL supports a large subset of standard SQL
- Other commands
  - SHOW TABLES;
    - shows all the tables in your database
  - DESCRIBE *table_name;*
    - displays table's schema

Creating tables in the MySQL DB using phpMyAdmin interface

1. Login to phpMyAdmin and click on the database name and then click on create table by giving a name and number of columns and then follow the instructions on the website

Lab

- Log in to the database server **studentdb-maria.gl.umbc.edu using phyMyAdmin** website
- Create a table for storing guestbook comments, call this table guestbook
  - The table must have the columns
    - comment_id: Set comment_id to be 'AUTO_INCREMENT'
      - Refer to: http://dev.mysql.com/doc/refman/5.0/en/example-auto-increment.html
    - username
    - comment_text
    - phone_number
  - Decide on what types of values these columns must accept

    Refer to:

    http://dev.mysql.com/doc/refman/5.0/en/data-types.html

    http://bytes.com/serversidescripting/mysql/tutorials/introductiontomysql/page1.html
  - Insert two rows into the table, using the INSERT INTO command
  - View the rows you inserted using the SELECT * command
- Use cars.sql as an example as you work on this lab

PHP and Database Access

- There are modules available in PHP to access numerous different database systems

13.6 DB Access with PHP and MySQL

- Typically uses two documents
  - one where user can make a request for some data from a database
  - one to host the PHP code to process the request and generate the return HTML document

13.6 Connecting to MySQL

- The mysqli_connect function
  - First parameter is MySQL server host name
  - Second parameter is the MySQL username
  - Third parameter is the password
  - Fourth parameter is the database name
  - Returns false if it fails
- The mysqli_close function
- If you need to change the database from the one selected in mysqli_connect, then, can use the function mysqli_select

13.6 Requesting MySQL Operations

- The mysqli_query function
  - First argument: A link identifier returned by mysqli_connect()
  - Second argument: Takes a string parameter with a SQL query
  - Returns a *data structure* that is used to identify data that resulted from query

- Functions that can be applied to this *data structure*
  - **`mysqli_num_rows`** gives the number of rows returned for query
  - **`mysqli_num_fields`** gives the number of fields (columns) returned for query
  - **`mysqli_fetch_array`** returns an array with the next row of results
    - Each array is indexed with the database table column name. This index is used to get the value in that column returned by query

13.6 PHP/MySQL Example

- See cars.php, carsdata.html

Lab

- Download guestbook.html and change the form action to point to a new PHP page
- Use is448_guestbook.sql to create your database, if you didn't complete previous lab
- In the new PHP page,
  - take the user's name, comments and phone number from the form, and
  - insert them into the guestbook table
    - Think about how your SQL query should look if you want to 'insert' data into the database
- Check your database table to see if the values have been entered
- Use carsdata.html and cars.php as examples