CS 207: Applied Database Practicum
Week 3

Varun Dutt

School of Computing and Electrical Engineering
School of Humanities and Social Sciences
Indian Institute of Technology Mandi, India

Scaling the Heights

Assignment 2 code Snippet

```php
<?php
$email_to = "receiver_mail@gmail.com";
$email_subject = "My subject";
$email_message = "Hello world!";
$email_from = "Sender_mail@gmail.com"

$headers = 'From: '.$email_from."\r\n".
'Reply-To: '.$email_from."\r\n" .
'X-Mailer: PHP/' . phpversion();
@mail($email_to, $email_subject, $email_message, $headers);
?>
```

Parameters for the mail()

Parameter	Description
to	Required. Specifies the receiver / receivers of the email
subject	Required. Specifies the subject of the email. **Note:** This parameter cannot contain any newline characters
message	Required. Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters. **Windows note:** If a full stop is found on the beginning of a line in the message, it might be removed. To solve this problem, replace the full stop with a double dot: <?php $txt = str_replace("\n.", "\n..", $txt); ?>
headers	Optional. Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\r\n). **Note:** When sending an email, it must contain a From header. This can be set with this parameter or in the php.ini file.
parameters	Optional. Specifies an additional parameter to the sendmail program (the one defined in the sendmail_path configuration setting). (i.e. this can be used to set the envelope sender address when using sendmail with the -f sendmail option)

DATABASE

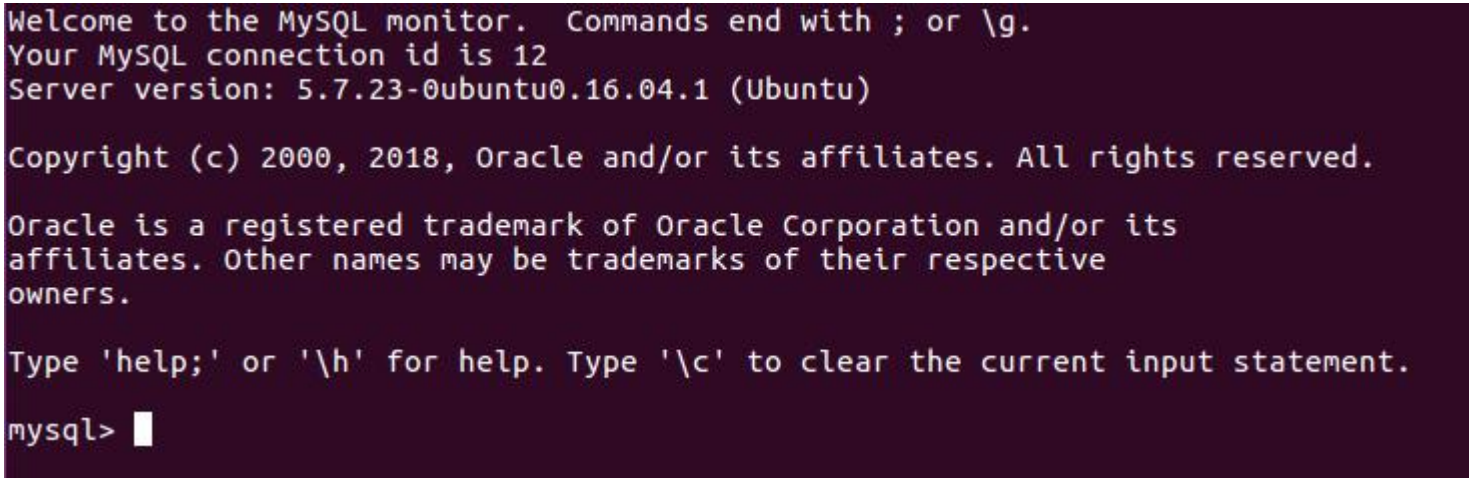
- A database is a kind of repository that stores a collection of data
- Each of the database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds
- Database are managed by Relational Database Management system(RDBMS)

RDBMS

- RDBMS stands for Relational Database Management System
- It is a software used to store and manage huge amount of data
- It stores data in separate tables rather than putting all data in one big storeroom
- It is termed as relational database because all the data is stored into different tables and relations are established using Primary keys or other keys known as Foreign Keys

MYSQL

- MYSQL is a Relational Database Management System
- MySQL uses a standard form of the well-known SQL data language
- SQL lets you access and manipulate databases



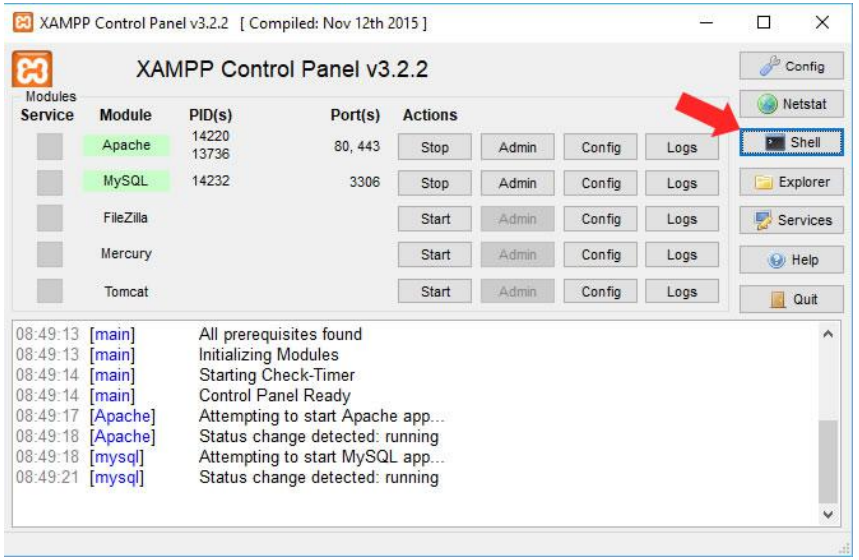
Why MYSQL is so popular?

- MySQL is released under an open-source license. So you have nothing to pay to use it
- MySQL works very quickly and works well even with large data sets
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc
- MySQL is very friendly to PHP, the most appreciated language for web development
- It provides a powerful mechanism for ensuring access to only authorized users hence providing data protection

Running Mysql in XAMPP

1. Start the Mysql in the XAMPP Control Panel

- Click on the shell icon as shown in the image and type
- *mysql -h localhost -u root*



2. If not using the Shell:

- For Linux users:

  sudo /opt/lampp/lampp startmysql

  /opt/lampp/bin/mysql -uroot –p

- For Windows users:

  cd c:\xampp\mysql\bin

  mysql.exe -u root --password

Install and run Mysql

- Run Mysql → mysql -u root -p
- Install mysql :

Commands used (Linux) -

sudo apt update
sudo apt install mysql-server
sudo mysql_install_db
sudo /usr/bin/mysql_secure_installation
(You'll be prompted to create a root password during the installation. Choose a secure one and make sure you remember it, because you'll need it later.)

- For reference - https://www.linode.com/docs/databases/mysql/install-mysql-on-ubuntu-14-04/

Common Terminologies

- Database: Database is a systematic collection of data. Databases support storage and manipulation of data. Databases contain tables
- Table : The part of a database that stores the data. A table has columns or attributes, and the data stored in rows
- Attributes : The columns in a table. All rows in table entities have the same attributes. Each attribute has a data type such as string, integer, or date
- Rows : The data entries in a table. Rows contain values for each attribute. Rows are also knows as records or tuples

Database creation in mysql

- The CREATE DATABASE statement is used to create a new SQL database.

  Syntax:

  CREATE DATABASE <databasename>

  Eg:

  CREATE DATABASE PROJECT;
- It will create a database called ‘PROJECT’.
- Once a database is created, you can check it in the list of databases with the following SQL command:

E.g.

SHOW DATABASES;

Database creation example

```
mysql> CREATE DATABASE PROJECT;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| PROJECT            |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

Dropping a Database in mysql

- The DROP DATABASE statement is used to drop an existing SQL database

  Syntax:

  DROP DATABASE <databasename>;

  Eg:

  DROP DATABASE PROJECT;

  - It will drop the database named as ‘PROJECT’

```
mysql> DROP DATABASE PROJECT;
Query OK, 0 rows affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```

Mysql Datatypes

Data type	Description
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. **Note:** If you put a greater value than 255 it will be converted to a TEXT type
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT	Holds a string with a maximum length of 65,535 characters
BLOB	For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data
ENUM(x,y,z,etc.)	Let you enter a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. **Note:** The values are sorted in the order you enter them. You enter the possible values in this format: ENUM('X','Y','Z')
SET	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice

Table creation in Mysql

- The CREATE TABLE statement is used to create a new table in a database

  Syntax:

  ```
  CREATE TABLE table_name (
      column1 datatype,
      column2 datatype,
     … );
  ```

  - Eg:CREATE TABLE Persons (

    ```
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
    );
    ```

Table creation

- The PersonID column is of type int and will hold an integer.
- The LastName, FirstName, Address, and City columns are of type varchar and will hold characters, and the maximum length for these fields is 255 characters.

Table creation example

```
mysql> create database  cs207
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> use cs207;
Database changed
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> CREATE TABLE Persons (
    ->      PersonID int,
    ->      LastName varchar(255),
    ->      FirstName varchar(255),
    ->      Address varchar(255),
    ->      City varchar(255)
    ->     );
Query OK, 0 rows affected (0.01 sec)

mysql> show tables
    -> ;
+-----------------+
| Tables_in_cs207 |
+-----------------+
| Persons         |
+-----------------+
1 row in set (0.00 sec)
```

Table insertion

- To insert into a new table you have to use INSERT INTO Statement
  - Syntax :
    - INSERT INTO table_name (column1, column2, column3, ...)
    - VALUES (value1, value2, value3, ...);
    - E.g.:

```
mysql> insert into Persons(PersonID,LastName,FirstName,Address,City) values (1,"Dutt","varun","IIT Mandi","Mandi");
Query OK, 1 row affected (0.01 sec)

mysql> select * from Persons;
+----------+----------+-----------+-----------+-------+
| PersonID | LastName | FirstName | Address   | City  |
+----------+----------+-----------+-----------+-------+
|        1 | Dutt     | varun     | IIT Mandi | Mandi |
+----------+----------+-----------+-----------+-------+
1 row in set (0.00 sec)
```

SELECT statement

- SELECT is used to select data from a database. The data returned is stored in a result table, called the result-set.
  Syntax:
  SELECT column1, column2, ...
  FROM table_name;
  - Eg:
  SELECT * FROM table_name;

```
mysql> select * from Persons;
+----------+----------+-----------+-----------+-------+
| PersonID | LastName | FirstName | Address   | City  |
+----------+----------+-----------+-----------+-------+
|        1 | Dutt     | varun     | IIT Mandi | Mandi |
+----------+----------+-----------+-----------+-------+
1 row in set (0.00 sec)
```

Creating Table Using Another Table

- A copy of an existing table can be created using a combination of the CREATE TABLE statement and the SELECT statement.
- The new table gets the same column definitions. All columns or specific columns can be selected.
- If you create a new table using an existing table, the new table will be filled with the existing values from the old table.

  Syntax :

```
CREATE TABLE new_table_name AS
  SELECT column1, column2,...
   FROM existing_table_name
   WHERE ....;
```

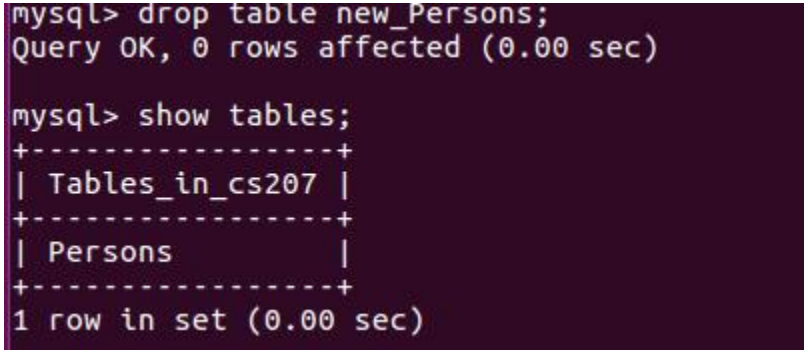
Creating Table Using Another Table example

```
mysql> create table new_Persons as select PersonID,LastName,FirstName,City from Persons;
Query OK, 1 row affected (0.02 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> show tables;
+-----------------+
| Tables_in_cs207 |
+-----------------+
| Persons         |
| new_Persons     |
+-----------------+
2 rows in set (0.00 sec)

mysql> select * from new_Persons;
+----------+----------+-----------+-------+
| PersonID | LastName | FirstName | City  |
+----------+----------+-----------+-------+
|        1 | Dutt     | varun     | Mandi |
+----------+----------+-----------+-------+
1 row in set (0.00 sec)
```

DROP Table

- The DROP TABLE statement is used to drop an existing table in a database.
- Deleting a table will result in loss of complete information stored in the table

  Syntax :
  - DROP TABLE table_name;
  - Eg. DROP TABLE Records;



ALTER Table

- The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.
- The ALTER TABLE statement is also used to add and drop various constraints on an existing table.
- To add a column in a table, use the following syntax:

  Syntax :

  ```
  ALTER TABLE table_name
  ADD column_name datatype;
  ```

- To change the data type of a column in a table, use the following syntax:

Syntax :

```
ALTER TABLE table_name
MODIFY COLUMN column_name datatype;
```

ALTER Table Example

```
mysql> ALTER TABLE Persons add Designation varchar(200) NULL ;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from Persons;
+----------+----------+-----------+-----------+-------+-------------+
| PersonID | LastName | FirstName | Address   | City  | Designation |
+----------+----------+-----------+-----------+-------+-------------+
|        1 | Dutt     | varun     | IIT Mandi | Mandi | NULL        |
+----------+----------+-----------+-----------+-------+-------------+
1 row in set (0.00 sec)
```

SQL Constraints

- SQL constraints are used to specify rules for data in a table.
- Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

Syntax :

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
    ....
);
```

Some common Constraints

The following constraints are commonly used in SQL:

- NOT NULL - Ensures that a column cannot have a NULL value.
- UNIQUE - Ensures that all values in a column are different.
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table.
- FOREIGN KEY - Uniquely identifies a row/record in another table.
- CHECK - Ensures that all values in a column satisfies a specific condition.
- DEFAULT - Sets a default value for a column when no value is specified.

SQL Constraints

```
mysql> CREATE TABLE new_table (ID int NOT NULL PRIMARY KEY,LastName varchar(255) NOT NULL,FirstName varchar(255),Age int);
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----------------+
| Tables_in_cs207 |
+-----------------+
| Persons         |
| new_table       |
+-----------------+
2 rows in set (0.00 sec)
```

```
mysql> describe new_table;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| ID        | int(11)      | NO   | PRI | NULL    |       |
| LastName  | varchar(255) | NO   |     | NULL    |       |
| FirstName | varchar(255) | YES  |     | NULL    |       |
| Age       | int(11)      | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

WHERE clause

- WHERE is used to filter records.It is used to extract only those records that fulfill a specified condition.
  Syntax:
  SELECT column1, column2, ...
  FROM table_name
  WHERE condition;

  - Eg:

  SELECT * FROM Customers WHERE Country='Mexico';

WHERE clause Example

```
mysql> select * from Persons;
+----------+-------------+---------------+-----------+-------+---------------------+
| PersonID | LastName    | FirstName     | Address   | City  | Designation         |
+----------+-------------+---------------+-----------+-------+---------------------+
|        1 | Dutt        | varun         | IIT Mandi | Mandi | Assistant Professor |
|        2 | choudhury   | abhinav       | IIT Mandi | Mandi | TA                  |
|        3 | Ladda       | aj            | IIT Mandi | Mandi | TA                  |
|        3 | gupta       | akul          | IIT Mandi | Mandi | TA                  |
|        4 | Suryavanshi | Virendrasingh | IIT Mandi | Mandi | TA                  |
|        5 | gupta       | akul          | IIT Mandi | Mandi | TA                  |
|        6 | Khandelwal  | aman          | IIT Mandi | Mandi | TA                  |
|        7 | Garg        | Shashwat      | IIT Mandi | Mandi | TA                  |
+----------+-------------+---------------+-----------+-------+---------------------+
8 rows in set (0.00 sec)

mysql> select * from Persons where Designation="TA";
+----------+-------------+---------------+-----------+-------+-------------+
| PersonID | LastName    | FirstName     | Address   | City  | Designation |
+----------+-------------+---------------+-----------+-------+-------------+
|        2 | choudhury   | abhinav       | IIT Mandi | Mandi | TA          |
|        3 | Ladda       | aj            | IIT Mandi | Mandi | TA          |
|        3 | gupta       | akul          | IIT Mandi | Mandi | TA          |
|        4 | Suryavanshi | Virendrasingh | IIT Mandi | Mandi | TA          |
|        5 | gupta       | akul          | IIT Mandi | Mandi | TA          |
|        6 | Khandelwal  | aman          | IIT Mandi | Mandi | TA          |
|        7 | Garg        | Shashwat      | IIT Mandi | Mandi | TA          |
+----------+-------------+---------------+-----------+-------+-------------+
7 rows in set (0.00 sec)
```

UPDATE statement

- The UPDATE statement is used to modify the existing records in a table.
  Syntax:
  UPDATE table_name
  SET column1 = value1, column2 = value2, ...
  WHERE condition;

  - Eg:

  UPDATE Customers SET ContactName = 'Alfred Schmidt',
  City= 'Frankfurt' WHERE CustomerID = 1;

UPDATE clause Example

```
mysql> select * from Persons;
+----------+-------------+---------------+-----------+-------+-------------+
| PersonID | LastName    | FirstName     | Address   | City  | Designation |
+----------+-------------+---------------+-----------+-------+-------------+
|        1 | Dutt        | varun         | IIT Mandi | Mandi | NULL        |
|        2 | choudhury   | abhinav       | IIT Mandi | Mandi | TA          |
|        3 | Ladda       | aj            | IIT Mandi | Mandi | TA          |
|        3 | gupta       | akul          | IIT Mandi | Mandi | TA          |
|        4 | Suryavanshi | Virendrasingh | IIT Mandi | Mandi | TA          |
|        5 | gupta       | akul          | IIT Mandi | Mandi | TA          |
|        6 | Khandelwal  | aman          | IIT Mandi | Mandi | TA          |
|        7 | Garg        | Shashwat      | IIT Mandi | Mandi | TA          |
+----------+-------------+---------------+-----------+-------+-------------+
8 rows in set (0.00 sec)

mysql> update Persons set Designation="Assistant Professor" where PersonID=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Persons;
+----------+-------------+---------------+-----------+-------+----------------------+
| PersonID | LastName    | FirstName     | Address   | City  | Designation          |
+----------+-------------+---------------+-----------+-------+----------------------+
|        1 | Dutt        | varun         | IIT Mandi | Mandi | Assistant Professor  |
|        2 | choudhury   | abhinav       | IIT Mandi | Mandi | TA                   |
|        3 | Ladda       | aj            | IIT Mandi | Mandi | TA                   |
|        3 | gupta       | akul          | IIT Mandi | Mandi | TA                   |
|        4 | Suryavanshi | Virendrasingh | IIT Mandi | Mandi | TA                   |
|        5 | gupta       | akul          | IIT Mandi | Mandi | TA                   |
|        6 | Khandelwal  | aman          | IIT Mandi | Mandi | TA                   |
|        7 | Garg        | Shashwat      | IIT Mandi | Mandi | TA                   |
+----------+-------------+---------------+-----------+-------+----------------------+
8 rows in set (0.00 sec)
```

DELETE **statement**

- The DELETE statement is used to delete existing records in a table.
  Syntax:
  DELETE FROM table_name
  WHERE condition;

  - Eg:

  DELETE FROM Customers WHERE CustomerName = 'Alfreds Futterkiste';

DELETE clause Example

```
mysql> select * from Persons;
+----------+-------------+---------------+-----------+-------+----------------------+
| PersonID | LastName    | FirstName     | Address   | City  | Designation          |
+----------+-------------+---------------+-----------+-------+----------------------+
|        1 | Dutt        | varun         | IIT Mandi | Mandi | Assistant Professor  |
|        2 | choudhury   | abhinav       | IIT Mandi | Mandi | TA                   |
|        3 | Ladda       | aj            | IIT Mandi | Mandi | TA                   |
|        3 | gupta       | akul          | IIT Mandi | Mandi | TA                   |
|        4 | Suryavanshi | Virendrasingh | IIT Mandi | Mandi | TA                   |
|        5 | gupta       | akul          | IIT Mandi | Mandi | TA                   |
|        6 | Khandelwal  | aman          | IIT Mandi | Mandi | TA                   |
|        7 | Garg        | Shashwat      | IIT Mandi | Mandi | TA                   |
+----------+-------------+---------------+-----------+-------+----------------------+
8 rows in set (0.00 sec)

mysql> DELETE FROM Persons  WHERE PersonID=3 and FirstName="akul" ;
Query OK, 1 row affected (0.01 sec)

mysql> select * from Persons;
+----------+-------------+---------------+-----------+-------+----------------------+
| PersonID | LastName    | FirstName     | Address   | City  | Designation          |
+----------+-------------+---------------+-----------+-------+----------------------+
|        1 | Dutt        | varun         | IIT Mandi | Mandi | Assistant Professor  |
|        2 | choudhury   | abhinav       | IIT Mandi | Mandi | TA                   |
|        3 | Ladda       | aj            | IIT Mandi | Mandi | TA                   |
|        4 | Suryavanshi | Virendrasingh | IIT Mandi | Mandi | TA                   |
|        5 | gupta       | akul          | IIT Mandi | Mandi | TA                   |
|        6 | Khandelwal  | aman          | IIT Mandi | Mandi | TA                   |
|        7 | Garg        | Shashwat      | IIT Mandi | Mandi | TA                   |
+----------+-------------+---------------+-----------+-------+----------------------+
7 rows in set (0.00 sec)
```

LIKE clause

- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

- There are two wildcards used in conjunction with the LIKE operator:

% - The percent sign represents zero, one, or multiple characters

_ - The underscore represents a single character

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

LIKE clause

LIKE Operator	Description
LIKE 'a%'	Finds any values that start with "a"
LIKE '%a'	Finds any values that end with "a"
LIKE 'a%o'	Finds any values that start with "a" and ends with "o"
LIKE '_r%'	Finds any values that have "r" in the second position

- Eg:

```
SELECT * FROM Customers WHERE CustomerName LIKE 'a%';
```

LIKE clause Example

```
mysql> select * from Persons;
+----------+-------------+---------------+-----------+-------+---------------------+
| PersonID | LastName    | FirstName     | Address   | City  | Designation         |
+----------+-------------+---------------+-----------+-------+---------------------+
|        1 | Dutt        | varun         | IIT Mandi | Mandi | Assistant Professor |
|        2 | choudhury   | abhinav       | IIT Mandi | Mandi | TA                  |
|        3 | Ladda       | aj            | IIT Mandi | Mandi | TA                  |
|        4 | Suryavanshi | Virendrasingh | IIT Mandi | Mandi | TA                  |
|        5 | gupta       | akul          | IIT Mandi | Mandi | TA                  |
|        6 | Khandelwal  | aman          | IIT Mandi | Mandi | TA                  |
|        7 | Garg        | Shashwat      | IIT Mandi | Mandi | TA                  |
+----------+-------------+---------------+-----------+-------+---------------------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM Persons WHERE FirstName LIKE 'a%';
+----------+------------+-----------+-----------+-------+-------------+
| PersonID | LastName   | FirstName | Address   | City  | Designation |
+----------+------------+-----------+-----------+-------+-------------+
|        2 | choudhury  | abhinav   | IIT Mandi | Mandi | TA          |
|        3 | Ladda      | aj        | IIT Mandi | Mandi | TA          |
|        5 | gupta      | akul      | IIT Mandi | Mandi | TA          |
|        6 | Khandelwal | aman      | IIT Mandi | Mandi | TA          |
+----------+------------+-----------+-----------+-------+-------------+
4 rows in set (0.00 sec)
```

MySql Functions

MIN(): MIN() function returns the smallest value of the selected column

MAX(): MAX() function returns the largest value of the selected column

COUNT(): COUNT() function returns the number of rows that matches a specified criteria

AVG(): AVG() function returns the average value of a numeric column

SUM(): SUM() function returns the total sum of a numeric column

MySql Functions Example

```
+---------+--------------+-----------+----------+
| OrderID | Product_name | ProductID | Quantity |
+---------+--------------+-----------+----------+
|       1 | Computer     |       201 |      210 |
|       2 | Printer      |       202 |      430 |
|       3 | Projector    |       203 |      137 |
|       4 | Harddisk     |       204 |      330 |
|       5 | Pendrive     |       205 |      540 |
|       6 | Mouse        |       206 |      450 |
+---------+--------------+-----------+----------+
6 rows in set (0.00 sec)
```

```
mysql> select max(Quantity) from order_table;
+---------------+
| max(Quantity) |
+---------------+
|           540 |
+---------------+
1 row in set (0.00 sec)

mysql> select min(Quantity) from order_table;
+---------------+
| min(Quantity) |
+---------------+
|           137 |
+---------------+
1 row in set (0.00 sec)

mysql> select avg(Quantity) from order_table;
+---------------+
| avg(Quantity) |
+---------------+
|      349.5000 |
+---------------+
1 row in set (0.00 sec)

mysql> select sum(Quantity) from order_table;
+---------------+
| sum(Quantity) |
+---------------+
|          2097 |
+---------------+
1 row in set (0.00 sec)

mysql> select count(Quantity) from order_table;
+-----------------+
| count(Quantity) |
+-----------------+
|               6 |
+-----------------+
1 row in set (0.00 sec)
```

Connecting mysql with php

- MySQL works very well in combination of various programming languages like PERL, C, C++, JAVA and PHP. Out of these languages, PHP is the most popular one because of its web application development capabilities.
- The PHP functions for use with MySQL have the following general format-

mysqli_function(value,value,…);

- For eg

```
$conn=mysqli_connect($connect);
mysqli_query($conn,"SQL statement");
```

Connection

- The mysqli_connect() function opens a new connection to the MySQL server.
- Syntax: mysqli_connect(host,username,password,dbname);

Parameter	Description
Host	Specifies a host name or an IP address
Username	Specifies the MySQL username
Password	Specifies the MySQL password
Dbname	Specifies the default database to be used

Connecting to Mysql using PHP

```php
<?php

define('user','root');
define('password','your_password');
define('host','localhost');
define('name','my_database');

$conn = mysqli_connect(host,user,password,name)
OR die('could not connect to mysql'.mysqli_connect_error());
echo "connection established";
?>
```

connection established

Mysql Dump

- Creating mysqldump to backup and restore mySQL databses/tables.
- To create dump file, write in shell (terminal) :
- To dump a single database:

  mysqldump db_name > dump.sql
- To dump some specific databases:

mysqldump --databases db1 db2 > dump.sql
- To dump all databases:

mysqldump --all-databases > dump.sql

```
root@Jarvis:/home/tonystark# mysql -u -p mysqldump cs207 > cs207.sql
```

Restoring from dump

- For restoring from dump file, write in shell (terminal) :

mysql -u <user> -p < db_backup.dump

- If the dump is of a single database you may have to add a line at the top of the dump file:

  USE <database-name-here>;

Some important SQL commands

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table

Keep in Mind : SQL keywords are NOT case sensitive:

select is the same as SELECT

References

- https://dev.mysql.com/doc/refman/5.7/en/mysqldump-sql-format.html
- https://www.w3schools.com/php/func_mysqli_connect.asp
- https://www.tutorialspoint.com/mysql/mysql-database-export.htm