Web Programming

YJ – Aug 2015

Course Outline

There're 16 lectures (2.5 hrs each) in this course. We'll mainly focus on web programming using LAMP model.

During the 24 hours tutorials in each week, our tutors will help you go through all the contents in the lecture notes and help you with the assignments

All scheduled lectures are learner-paced which means they could be altered depends on the situation. We help you gain rather than cram!

Course Outline

- L1 ~ Web programming intro
- L2 ~ HTML5
- L3 ~ CSS3
- L4 ~ JavaScripts
- L5 ~ PHP 1 Basic
- L6 ~ PHP 2 OO && Functions
- L7 ~ PHP 3 OO && Sessions
- L8 ~ PHP 4 Advanced features
- L9 ~ Mysql
- L10 ~ Mysql 2
- L11 ~ XML/JSON
- L12 ~ Ajax
- L13 ~ Framework: Bootstrap && Laravel
- L14 ~ CMS: Wordpress, Joomla, etc.
- L15 ~ Final Projects
- L16 ~ Final Projects

SQL

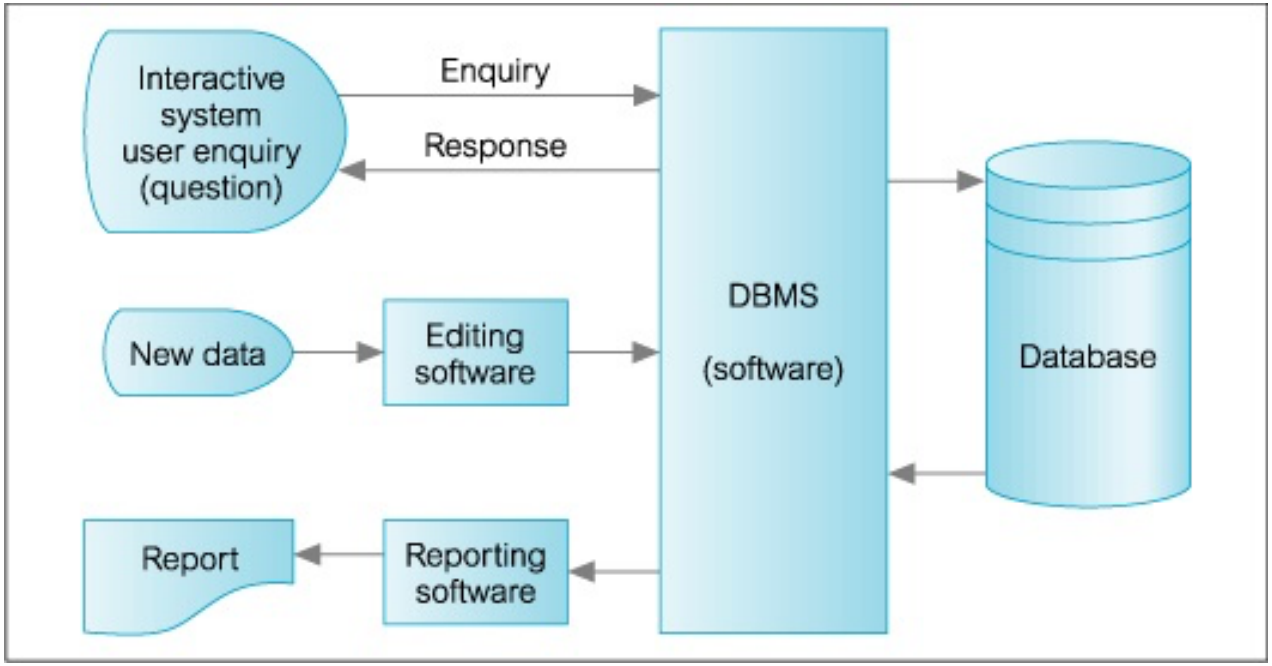
SQL is a database computer language designed for the retrieval and management of data in relational database. SQL stands for Structured Query Language.

SQL is the standard language for Relation Database System. All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix, postgres and SQL Server use SQL as standard database language.

- Allows users to access data in relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures, and views

RDBMS

- RDBMS stands for **R**elational **D**atabase **M**anagement **S**ystem. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.



Popular RDBMS

- MySQL

  MySQL is an open source SQL database, which is developed by Swedish company MySQL AB. MySQL comes with a very fast, multi-threaded, multi-user, and robust SQL database server.

- MS SQL Server

  MS SQL Server is a Relational Database Management System developed by Microsoft Inc.

- ORACLE

  It is a very large and multi-user database management system. Oracle is a relational database management system developed by 'Oracle Corporation'.

  Oracle works to efficiently manage its resource, a database of information, among the multiple clients requesting and sending data in the network.

  It is an excellent database server choice for client/server computing. Oracle supports all major operating systems for both clients and servers, including MSDOS, NetWare, UnixWare, OS/2 and most UNIX flavors.

- MS ACCESS

  Microsoft Access is an entry-level database management software. MS Access database is not only an inexpensive but also powerful database for small-scale projects.

Some Concepts

- Table
- Field
- Row
- Column
- Constraints
  - NOT NULL Constraint: Ensures that a column cannot have NULL value.
  - DEFAULT Constraint: Provides a default value for a column when none is specified.
  - UNIQUE Constraint: Ensures that all values in a column are different.
  - PRIMARY Key: Uniquely identified each rows/records in a database table.
  - FOREIGN Key: Uniquely identified a rows/records in any another database table.
  - CHECK Constraint: The CHECK constraint ensures that all values in a column satisfy certain conditions.
  - INDEX: Use to create and retrieve data from the database very quickly.

SQL Syntax

```
SELECT SUM(column_name)
FROM   table_name
WHERE  CONDITION
GROUP BY column_name
HAVING (ARITHMETIC FUNCTION CONDITION)
ORDER BY COLUMN1, COLUMN2

UNION

SELECT SUM(column_name)
FROM   table_name
WHERE  CONDITION
GROUP BY column_name
HAVING (arithmetic function condition)
ORDER BY COLUMN1, COLUMN2
;
```

Command

Command	Description
CREATE	Creates a new table, a view of a table, or other object in database
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other object in the database.

```
CREATE DATABASE DatabaseName;
```

```
CREATE TABLE CUSTOMERS(
   ID   INT              NOT NULL,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

```
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| ID      | int(11)      | NO   | PRI |         |       |
| NAME    | varchar(20)  | NO   |     |         |       |
| AGE     | int(11)      | NO   |     |         |       |
| ADDRESS | char(25)     | YES  |     | NULL    |       |
| SALARY  | decimal(18,2)| YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
```

Command

Command	Description
CREATE	Creates a new table, a view of a table, or other object in database
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other object in the database.

DROP DATABASE DatabaseName;

DROP TABLE table_name;

ALTER TABLE table_name **ADD** column_name datatype;

ALTER TABLE table_name **DROP COLUMN** column_name;

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

Command

Command	Description
INSERT	Creates a record
SELECT	Retrieves certain records from one or more tables
UPDATE	Modifies records
DELETE	Deletes records

Insert

```
INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)]
VALUES (value1, value2, value3,...valueN);

INSERT INTO TABLE_NAME VALUES (value1,value2,value3,...valueN);

INSERT INTO `City` (`ID`, `Name`, `CountryCode`, `District`, `Population`)
VALUES
  (1893, 'Tianjin', 'CHN', 'Tianjin', 5286800);
```

Update & Delete

```
UPDATE table_name
SET column1 = value1, column2 = value2...., columnN = valueN
WHERE [condition];

UPDATE CITY SET POPULATION = '6286800' WHERE ID='1893';


DELETE FROM table_name WHERE [condition];

DELETE FROM CITY WHERE ID='1893';
```

Select, Where, AND&OR

```
SELECT column1, column2, columnN
FROM table_name
WHERE [condition1] AND [condition2]...AND [conditionN];
```

```
SELECT column1, column2, columnN
FROM table_name
WHERE [condition1] OR [condition2]...OR [conditionN];
```

```
SELECT * FROM CITY WHERE COUNTRYCODE = 'CHN';
```

```
SELECT Population FROM CITY WHERE Name= 'Tianjin';
```

```
SELECT * FROM CITY WHERE COUNTRYCODE = 'CHN' AND NAME='TIANJIN';
```

```
SELECT * FROM CITY WHERE NAME='TIANJIN' OR NAME='WUHAN';
```

Select … As

```
SELECT column1, column2....
FROM table_name AS alias_name
WHERE [condition];
```

```
SELECT column_name AS alias_name
FROM table_name
WHERE [condition];
```

```
SELECT C.ID, C.NAME, C.AGE, O.AMOUNT
      FROM CUSTOMERS AS C, ORDERS AS O
      WHERE  C.ID = O.CUSTOMER_ID;
```

Distinct

The SQL **DISTINCT** keyword is used in conjunction with SELECT statement to eliminate all the duplicate records and fetching only unique records.

There may be a situation when you have multiple duplicate records in a table. While fetching such records, it makes more sense to fetch only unique records instead of fetching duplicate records.

SELECT DISTINCT column1, column2,.....columnN
FROM table_name
WHERE [condition];

SELECT DISTINCT COUNTRYCODE **FROM** CITY;

Like

Statement	Description
WHERE SALARY LIKE '200%'	Finds any values that start with 200
WHERE SALARY LIKE '%200%'	Finds any values that have 200 in any position
WHERE SALARY LIKE '_00%'	Finds any values that have 00 in the second and third positions
WHERE SALARY LIKE '2_%_%'	Finds any values that start with 2 and are at least 3 characters in length
WHERE SALARY LIKE '%2'	Finds any values that end with 2
WHERE SALARY LIKE '_2%3'	Finds any values that have a 2 in the second position and end with a 3
WHERE SALARY LIKE '2___3'	Finds any values in a five-digit number that start with 2 and end with 3

SELECT * **FROM** CITY **WHERE NAME like 'TIANJI_'**;

Order By

The SQL **ORDER BY** clause is used to sort the data in ascending or descending order, based on one or more columns. Some database sorts query results in ascending order by default.

```
SELECT column_list
FROM table_name
[WHERE condition]
[ORDER BY column1, column2, .. columnN] [ASC | DESC];
```

```
SELECT NAME FROM CITY WHERE COUNTRYCODE = 'CHN' ORDER BY NAME;
```

Group By

The SQL **GROUP BY** clause is used in collaboration with the SELECT statement to arrange identical data into groups.

```
SELECT column1, column2
FROM table_name
WHERE [ conditions ]
GROUP BY column1, column2
ORDER BY column1, column2;
```

```
SELECT COUNT(*), CONTINENT FROM COUNTRY GROUP BY CONTINENT;
```

Having

The HAVING clause enables you to specify conditions that filter which group results appear in the final results.

```
SELECT column_name, aggregate_function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name
HAVING aggregate_function(column_name) operator value;
```

```
SELECT COUNT(*), CONTINENT
FROM COUNTRY
GROUP BY CONTINENT
HAVING COUNT(*) > 30;
```

Limit

The LIMIT clause is used to specify the number of records to return.
The LIMIT clause can be very useful on large tables with thousands of records.
Returning a large number of records can impact on performance.

```
SELECT column_name FROM table_name
WHERE [condition]
LIMIT 2
ORDER BY column_name DESC;

SELECT * FROM CITY LIMIT 10 OFFSET 15;
```

Functions

- SQL COUNT Function

  The SQL COUNT aggregate function is used to count the number of rows in a database table.

  SELECT *COUNT*(*) **FROM** COUNTRY **WHERE** CONTINENT=**'ASIA'**;

- SQL MAX Function

  The SQL MAX aggregate function allows us to select the highest (maximum) value for a certain column.

  SELECT NAME, POPULATION **FROM** CITY
  WHERE POPULATION = (**SELECT** *MAX*(POPULATION) **FROM** CITY);

- SQL MIN Function

  The SQL MIN aggregate function allows us to select the lowest (minimum) value for a certain column.

  SELECT NAME, POPULATION **FROM** CITY
  WHERE POPULATION = (**SELECT** *MIN*(POPULATION) **FROM** CITY);

- SQL AVG Function

  The SQL AVG aggregate function selects the average value for certain table column.

  SELECT *AVG*(POPULATION), COUNTRYCODE **FROM** CITY **GROUP BY** COUNTRYCODE;

Functions

- SQL SUM Function

  The SQL SUM aggregate function allows selecting the total for a numeric column.

  SELECT *SUM*(POPULATION), COUNTRYCODE **FROM** CITY
  GROUP BY COUNTRYCODE
  ORDER BY *SUM*(POPULATION) **DESC**;

- SQL SQRT Functions

  This is used to generate a square root of a given number.

  SELECT *SQRT*(16);

- SQL RAND Function

  This is used to generate a random number using SQL command.

  SELECT RAND( );

  ORDER BY RAND();