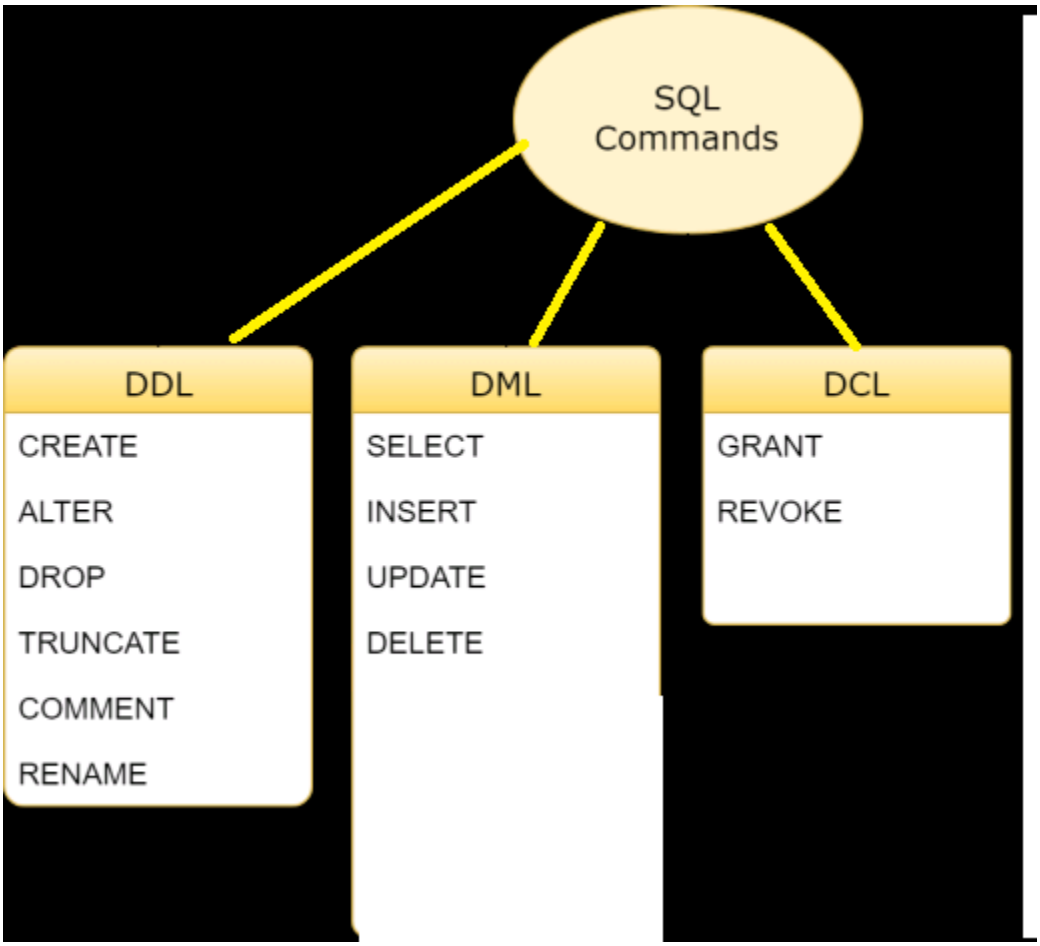
MYSQL DATABSE TUTORIAL



DDL:

DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

- CREATE – to create database and its objects like (table, index, views, store procedure, function, and triggers)
- ALTER – alters the structure of the existing database

- DROP – delete objects from the database
- TRUNCATE – remove all records from a table, including all spaces allocated for the records are removed
- COMMENT – add comments to the data dictionary
- RENAME – rename an object

DML:

DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE etc, and it is used to store, modify, retrieve, delete and update data in a database.

- SELECT – retrieve data from a database
- INSERT – insert data into a table
- UPDATE – updates existing data within a table
- DELETE – Delete all records from a database table

DCL:

DCL is short name of Data Control Language which includes commands such as GRANT and mostly concerned with rights, permissions and other controls of the database system.

- GRANT – allow users access privileges to database
- REVOKE – withdraw users access privileges given by using the GRANT command

run your xampp & open any browser & type "Localhost/phpmyadmin" :

(1)to create database use command:

SQL> create database practice;

(2)to create table use command:

```
SQL>create table  myrecord
       (     id   int,
             name varchar(20)
        );
```

First of all understand Data types in MYSQL:

Int Type:

THE FIVE INTEGERS TYPES SUPPORTED IN MYSQL ARE TINYINT,SMALLINT,MEDIUMINT,INT,BIGINT.

CREATE table data EXAMPLE:

```
Create  table  data
(fti  TINYINT,
fsi  SMALLINT,
fmi  MEDIUMINT,
fi   INT,
fbi   BIGINT);
```

INSERT EXAMPLE :

```
Insert into
data  values(123456789,123456789,123456789,123456789,123456787654321);
```

Insert into data values(-123456789,-123456789,-123456789,-123456789,-123456787654321);

SELECT EXAMPLE:

SELECT * FROM DATA;

Float data type:

Create example:

Create table data1
(price FLOAT(5,2));

INSERT EXAMPLE:

Insert into data1 values(13.6);
Insert into data1 values(876.90);
Insert into data1 values(-5.2);
Insert into data1 values(-12345.678);

SELECT EXAMPLE:

SELECT pirce from data1;

Char data type:

Char type which is used to fixed –length strings and must be declared with a size specifier in paretheses.

Create example:

Create table data3 (alphabet char(10));

Insert example:

Insert into data values('abcedefghijklmno');

Select example:

Select alphabet from data;

Date TYPE::
Create table data4(birthday DATE);

INSERT EXAMPLE:
 INSERT into data4 values(‘2003-03-04’);
Select example:
Select birthday from data4;

Time:
Crate table data5(t   TIME);

INSERT EXAMPLE:
Insert into data5 values(‘12:30:56’);
Select example:
Select t from data5;

Date & Time:
Create example:
Create table data6(event   Datetime);
Insert example:
Insert into data6 values(NOW());
SELECT EXAMPLE:
Select event from data6;

run your xampp & open any browser & type "Localhost/phpmyadmin" :

(1)to create database use command:

SQL> create database practice;

(2)to create table use command:

```
SQL>create table  myrecord
        (     id   int,
              name varchar(20)
         );
```

(3)to insert record into table:
there is two type:

type 1:

```
insert into  myrecord(id,name)values('1','om');
```

type 2:

```
insert into  myrecord values('2','som');
```

(4)Display record from table:

(a) how to display all record from table:

```
SQL>select  * from  myrecord;
```

(b)how to display particular column record:

```
SQL>select  name from myrecord;  //for single column

SQL>select id,name from myrecord; //for more than one column
```

(c)how to display particular rows record:

```
SQL>select  *  from myrecord where id='1';      // for single row

SQL>select * from myrecord  where id in(1,2);    //for more than one row
```

(5)how to update record of a table:

```
SQL>update  myrecord   set id='3' ,
```

name='omanjali' where id='1';   //update by id

SQL>update myrecord set id='3' ,
name='omanjali' where name='som'; //update by name

(6)how to delete record from table:

SQL>delete from myrecord ; //delete all record

SQL>delete from myrecord where id='1';   //delete by id

SQL>delete from myrecord where name='om';   //delete by name

(7)how to add new column into a table :

SQL>alter table myrecord
        add city varchar(200);

(8)how to modify size or type of of a column:

SQL>alter table myrecord
  modify city varchar(20);

(9)how to rename a table :

SQL>rename table myrecord to mynewrecord;

(10)how to rename a column:

syntax:

alter
table table_name change old_column_name new_column_name data_type
;

example:

SQL>ALTER TABLE
mynewrecord CHANGE name myname VARCHAR(20);

(11)how to change position of column into table:

SYNTAX:

ALTER TABLE
tablename MODIFY COLUMN columnname datatype AFTER column;

example:

SQL>ALTER
TABLE mynewrecord MODIFY COLUMN name VARCHAR(50) A FTER id;

(12)how to drop a particular column :

SQL>alter table mynewrecord drop name;

==========
==========
==========

create table lefttable
(
id int primary key auto_increment,
name varchar(200),
city varchar(200)
);

inset some values into lefttable...

(1)Order by :

select * from lefttable order by id;

(2)Group by:

select * from lefttable group by id;

(3)having clause:

select * from lefttable having id<3;

(4)Like :

select * from lefttable where name like 'o%';

(5)Avg:

select avg(id) from lefttable ;
ALTER TABLE mynewrecord DROP name
(6)Sum :

select sum(id) from lefttable;

(7)Count :

select count(id) from lefttable;

(8)Min :

select min(id) from lefttable;

(9)Max :

select max(id) from lefttable;

(10)Lcase :

SELECT LCASE('Tech on the Net');

(11)Ucase :

SELECT UCASE('myteststring');

==========
==========

Primary key & Foreign key concept :

==========
==========

(1)CREATE TABLE categories(
cat_id int not null auto_increment primary key, //creating a primary key
cat_name varchar(255) not null,
cat_description VARCHAR(200)
) ;

(2)CREATE TABLE products(
prd_id int not null auto_increment primary key,
prd_name varchar(355) not null,
prd_price decimal,
cat_id int not null,
FOREIGN KEY fk_cat(cat_id) //creating a foreign key
REFERENCES categories(cat_id)
ON UPDATE CASCADE
ON DELETE RESTRICT
);

how to add foreign key to a column:

(a)just create another table name is myorder:

create table myorder
(order_id int not null primary key auto_increment,
order_name varchar(200)
);

(b)now add a column order_id to products table:
alter table products add order_id int not null;

(c)now we add foreign key to column order_id in products table:

```
ALTER TABLE products

ADD FOREIGN KEY   fk_order(order_id)

REFERENCES  myorder(order_id)

ON DELETE NO ACTION

ON UPDATE CASCADE;
```

Conclusion :

a table contain only one primary and more than foreign key..

==========

JOIN EXAMPLE

==========

Step 1:

Create table lefttable:

```
create table lefttable
(id  int(10),
name  varchar(20),
city   varchar(20));
```

Create table righttable:

```
create table righttable
```

(id int(10),
fname varchar(20),
fcity   varchar(20));

Note: enter some values & some same id value into both tables ..

Step 2:

(1)simple join example :

select l.name,r.fname from lefttable l , righttable r;

(2)Left Join example:

select l.name,r.fname from lefttable l LEFT JOIN righttable r ON l.id=r.id;

(3)Right Join example:

select l.name,r.fname from lefttable l RIGHT JOIN righttable
r ON l.id=r.id;

(4)Inner Join example:

select l.name,r.fname from lefttable l INNER JOIN righttable
r ON l.id=r.id;

For practice on join let see following example:-

step 1:-

create table postjobs
(jobid int,

jobloc varchar(200),
jobtitle varchar(200),
jobdes compname);

step 2:-

create table companyregister
(comp_id   int,
compname varchar(200),
email varchar(200),
password varchar(200),
companydes varchar(200)

);

Now how to display jobtitle ,jobdesc,jobloc and company name :-

(lets see using left join and where clause)

select postjobs.jobloc, postjobs.
jobtitle,postjobs.jobdes,companyregister.compname from postjobs left join companyregister on postjobs.session=companyregister.email where postjobs.jobtitle='web developer' or postjobs.jobloc='mumbai,pune';

Subqueries :

A subquery is a SQL query nested inside a larger query.

A subquery may occur in:
- A SELECT clause
- A FROM clause
- A WHERE clause

A subquery is usually added within the WHERE Clause of another SQL SELECT statement.

You can use the comparison operators, such as >, <, or =. The comparison operator can also be a multiple-row operator, such as IN, ANY, SOME, or ALL.
A subquery can be treated as an inner query, which is a SQL query placed as a part of another query called as outer query.
The inner query executes first before its parent query so that the results of the inner query can be passed to the outer query.

MySQL Subqueries: Using Comparisons:

A subquery can be used before or after any of the comparison operators. The subquery can return at most one value. The value can be the result of an arithmetic expression or a column function. SQL then compares the value that results
from the subquery with the value on the other side of the comparison operator.
You can use the following comparison operators:

Operator	**Description**
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
!=	Not equal to
<>	Not equal to
<=>	NULL-safe equal to operator

(1)create table employee
(employee_id int primary key auto_increment,
first_name varchar(200),
last_name varchar(200),
salary decimal);

(2)insert record inside table as given below :

```
employee_id | first_name  | last_name  | salary   |
+-------------+-------------+------------+----------+
|         100 | Steven      | King       | 24000.00 |
|         101 | Neena       | Kochhar    | 17000.00 |
|         102 | Lex         | De Haan    | 17000.00 |
|         103 | Alexander   | Hunold     |  9000.00 |
|         108 | Nancy       | Greenberg  | 12000.00 |
|         109 | Daniel      | Faviet     |  9000.00 |
|         120 | Matthew     | Weiss      |  8000.00 |
|         121 | Adam        | Fripp      |  8200.00 |
|         122 | Payam       | Kaufling   |  7900.00 |
|- - - - - - - - - - - - - - - - - - - - - - - - - -
```

For example, suppose you want to
find the employee id, first_name, last_name, and
salaries for employees whose average salary is
higher than the average salary throughout the company.

Then you will use sub query example:

```
SQL>SELECT employee_id,first_name,last_name,salary
      FROM employees WHERE salary >
       (SELECT AVG(SALARY) FROM employees);
```