SQLTutorial

PlayList Link: https://www.youtube.com/playlist?list=PLUDwpEzHYYLvWEwDxZViN1shP-pGyZdtT

Tutorial 1: Overview on Database, DBMS/RDBMS, SQL | Database Components

```
SQL Session-1
******************
What is DBMS & RDBMS
What is Database
What is SQL
Database Components - Client & Server
Database Vs Schema
Installing MySQL
MySQL Workbench
Connecting to the database & Querying Tables
SQL Commands
******************


DBMS--> DataBase Management System
        Dbase,Foxpro MS-access etc..

RDBMS --> Relational DataBase Management System
        Oracle,MS-SQL Server, DB2, MySQL ,MS-Access etc...


What is Database ---> Data Storage area

SQL (Structure Query Language)--> used to communicate to the database to perform different kinds of operations.
```

```
Data is actually stored in the Databse server ( Remote machines)
We can send SQL Commands using client software.

2 Types client
        --> Graphical mode(GUI)
        --> CLI (Command Line Interface)


Oracle Database ----> SQLDeveloper (GUI)
                      SQLPLus  (CLI)
                      Toad
                      Squirell
                      Aquadata studio
                      etc..

MySQL ------> MySQLWorkbech (GUI)
              MySQL CommandLinetool (CLI)
                Toad
```

Tutorial 2: Installing MySQL, Working with SQL Workbench & Command Line Clients

To download MySQL software

https://dev.mysql.com/downloads/windows/installer/8.0.html

CLI commnads

Show databases;

use <table-name>;

select * from <table-name>;

Tutorial 3: Create Database,Table, Inserting Records and Retrieving Data

DESCRIBE command gives the structure of the table

```
create database if not exists mydb;
use mydb;
create table student(
sno int(3),
sname varchar(10),
marks int(3)
);
describe student;

insert into student values(101, 'vamsi',30);
insert into student(sname,sno,marks) values('krishna',102,35);
insert into student values(103, 'pallapolu',null);

select * from student;
```

Tutorial 4: Filtering Rows(Where Clause), Operators & Pattern Matching

Where clause

```
USE hr;
SELECT * FROM employees;
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY FROM EMPLOYEES;
SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY, SALARY+300 as SAL FROM EMPLOYEES;
select * from employees where salary > 15000;
select * from employees where salary <= 5000;
select * from employees where department_id=30;
select * from employees where commission_pct is null;
select * from employees where first_name='Jennifer';
select department_id from employees;
select distinct department_id from employees;
select distinct * from employees;
```

Logical Operators

```sql
use hr;
/* Logical Operators */
select * from employees where salary > 15000 and job_id='Ad_vp';
select * from employees where salary > 15000 or job_id='Ad_vp';
select * from employees where not first_name='david';

/* Between & IN Operators */
select * from employees where salary between 10000 and 12000;
select * from employees where salary not between 10000 and 12000;

select * from employees where salary=3400 or salary=2500 or salary=3000;
select * from employees where salary in(3400, 2500, 3000);
select * from employees where salary not in(3400, 2500, 3000);
```

Wild Card Operators

```sql
/* Pattern Matching Operators */
select * from employees where first_name like 's%';
select * from employees where first_name like '%r';
select * from employees where first_name like 's%r';
select * from employees where first_name like '%m%';
select * from employees where first_name not like 's%';
select * from employees where first_name like '%e_';
select * from employees where first_name like '___';
```

Tutorial 5: DDL (Data Definition Language) Commands

DDL Commands

- create
- alter
    - add column
    - drop column
    - modify column
    - rename column <old-col-name> to <new-col-name>
- drop
- truncate
- rename

If we want to reduce the size of the column, no data should be present in the column

Drop

To drop all the data bases object (both table defination and data will be deleted)

Truncate

Table struture will be present but data will be deleted

Data will be deleted permanently, we cannot roll back the data

Delete

Table struture will be present

Data will be deleted but we can roll back the data

Tutorial 6: MySQL Built-in Functions in SQL

```sql
-- string functions --
select lower('VAMSI');
select upper('vamsi') from dual;
use hr;
select length(first_name) from employees;
select * from employees where length(first_name)=4;
select trim('  vamsi ') from dual;
select trim('z' from 'zzoraclez') from dual;
select instr('oracle','a');
select substr('oracle',2,3);
select substring('oracle',2,3);

-- print only first 3 charcters of first_name
select substring(first_name,1,3) from employees;

select concat('vamsi', 'krishna');
select first_name, last_name, concat(first_name, last_name) fullname from employees;
```

```sql
-- Numeric functions --
select abs(-40);
select abs(40);

select sqrt(25);
select pow(2,3);
select mod(4,2);

select truncate(12345.6789,3);
select truncate(12345.6789,2);
select truncate(12345.6789,1);
select truncate(1239,-1);
select truncate(1239,-2);
select truncate(1239,-3);

select greatest(10,20,30,40.5,61.2);
select least(10,20,30,40.5,61.2);
```

```sql
-- date functions --
select curdate();
select current_date();
select curtime();
select current_time();
select sysdate();
select now();
select year('2021-07-19');
select month('2021-07-19');
select monthname('2021-07-19');
select day('2021-07-19');

use hr;
select * from employees;
select * from employees where year(hire_date)="1987";
select * from employees where monthname(hire_date)='June';
```

```sql
-- Aggregate functions --
select max(salary) from employees;
select min(salary) from employees;
select sum(salary) from employees;
select avg(salary) from employees;
select count(*) from employees;
```

Tutorial 7: Group By, Having & ,Order By Clauses in SQL

```sql
use hr;
select * from employees;

select department_id,  max(salary) from employees group by department_id;
select job_id, count(*) from employees group by job_id;
select job_id, department_id, count(*) from employees group by job_id, department_id;

select department_id, sum(salary) from employees group by department_id having sum(salary)>20000;

select department_id, sum(salary) from employees group by department_id having sum(salary)>20000 order by sum(salary);

select department_id, sum(salary) from employees where department_id <> 50 group by department_id having sum(salary)>20000 order by sum(salary);

select * from employees order by department_id desc;

select * from employees order by salary desc;

select first_name, last_name, salary, max(salary) from employees;
select first_name, last_name, salary, max(salary) from employees where salary not in(select max(salary) from employees);
```

Tutorial 8: Union & Union All in SQL

Set operators

- Union
- Union All
- Intersect
- Minus

To retrieve the data from multiple tables we can use:

- Set operators
- Joins
- Sub queries

UNION

- It retrieve the data from both the tables but it eliminate the duplictes

UNION ALL

- It retrieve the data from both the tables but it does not eliminate the duplictes

INTERSECT

- It retrieve the common data among the tables

MINUS

- T1 – T2
- It retrieves the data which is present in T1 but not T2

MySQL doesn’t support intersect and minus operators

```
CREATE TABLE A(SNAME varchar(10), NUM INT(2));
CREATE TABLE B(NUM INT(2),GRADE VARCHAR(3));
INSERT INTO A VALUES('ABC',10);
INSERT INTO A VALUES('XYZ',11);
INSERT INTO A VALUES('PQR',12);
INSERT INTO A VALUES('MNO',14);
COMMIT;
INSERT INTO B VALUES(11,'A');
INSERT INTO B VALUES(12,'B');
INSERT INTO B VALUES(13,'C');
INSERT INTO B VALUES(15,'B');
COMMIT;

select num from A union all select num from B;
```

Tutorial 9: SQL Joins and Sub Queries

```
use db;
CREATE TABLE TAB1(NUMID INT(3));
CREATE TABLE TAB2(NUMID INT(3));
INSERT INTO TAB1 VALUES(10);
INSERT INTO TAB1 VALUES(11);
INSERT INTO TAB1 VALUES(12);
INSERT INTO TAB1 VALUES(14);
INSERT INTO TAB2 VALUES(11);
INSERT INTO TAB2 VALUES(12);
INSERT INTO TAB2 VALUES(13);
INSERT INTO TAB2 VALUES(15);

select * from tab1;
select * from tab2;

select tab1.numid, tab2.numid from tab1 inner join tab2 on tab1.numid=tab2.numid;
select tab1.numid, tab2.numid from tab1 left outer join tab2 on tab1.numid=tab2.numid;
select tab1.numid, tab2.numid from tab1 right outer join tab2 on tab1.numid=tab2.numid;
describe tab1;
describe tab2;
```

```
use hr;
select * from employees;
select * from departments;
select * from employees inner join departments on employees.department_id<>departments.department_id;
select first_name, salary, employees.department_id from employees
inner join departments on
employees.department_id=departments.department_id;

select first_name, salary, emp.department_id from employees emp
left join departments dep on
emp.department_id=dep.department_id;

-- Self Join --
select * from employees;
select  E.EMPLOYEE_ID, E.FIRST_NAME, M.manager_id, M.FIRST_NAME from employees e, employees m where e.manager_id=m.manager_id;
```

Sub Queries

```
/* sub queries
Contains two parts:
-> Outer Query
-> Inner Querey
Two Types
-> Single row sub query (<=, >=, !=)
-> Multiple row sub query (in, any, all)
*/
-- Single Row Sub Queries
-- display employees name whose salary is less than ellen's
use hr;
select salary from employees where first_name='ELLEN';
select * from employees where salary<11000;
select * from employees where salary<(select salary from employees where first_name='ELLEN');

-- 2nd max salary
select max(salary) from employees where salary < (select max(salary) from employees);

-- 3rd max salary from employees
select max(salary) from employees where salary <
(select max(salary) from employees where salary <
(select max(salary) from employees)
);
```

```
-- multile row sub queries
select salary from employees where department_id=30;
select first_name, department_id, salary from employees where salary in(select salary from employees where department_id=30);

select first_name, department_id, salary from employees where salary > any(select salary from employees where department_id=30);
select first_name, department_id, salary from employees where salary < any(select salary from employees where department_id=30);

select first_name, department_id, salary from employees where salary > all(select salary from employees where department_id=30);
select first_name, department_id, salary from employees where salary < all(select salary from employees where department_id=30);

select * from departments;
select first_name, last_name , d.department_name from employees e, departments d where e.department_id=d.department_id;
```

```
-- List out the employees who are having salary less than the maximum salary
-- and
-- also having the hire date greater than the hire date of the employee who is having maximum salary

select * from employees where
salary < (select max(salary) from employees)
and
hire_date > (select hire_date from employees where salary=(select max(salary) from employees))
order by hire_date;
```

```
-- nth highest salry
-- 3rd highest salry
use hr;
select first_name, salary from employees e1 where 4=
(select count(distinct salary) from employees e2 where e2.salary > e1.salary);
```

Tutorial 10: Integrity Constraints

Primary key can be applied only any single column for a table

Composite primary key can be applied on multiple columns at table level only

When you insert a record into a child table, first it will verify in parent table wheather record is present or not

If you try to delete a record from parent record directly without deleteing records from child table , it's not allowed

```
use db;
drop table if exists school;
create table school(
sno int(3),
sname varchar(15),
marks int(3),
primary key(sno)
);
insert into school values(101, 'arun', 90);
insert into school values(102, 'kiran', 70);
insert into school values(103, 'amit', 80);
select * from school;

drop table if exists library;
create table library(
sno int(3) ,
book_name varchar(10),
foreign key(sno) references school(sno) on delete cascade
);
insert into library values(102, 'Java');
insert into library values(103, 'C');
insert into library values(null, 'dot net');

select * from library;
```

```
drop table if exists student1;
create table student1(
sno int(3) unique,
sname varchar(10),
marks int(5) check(marks between 1 and 10)
);

insert into student1 values(1,'a', 1);
insert into student1 values(2,'b', 11);
select * from student1;

create table location(
city varchar(10) check(city in('hyd','blor','chen')),
country varchar(5),
pin int(5)
);

insert into location values('hyd','india',1);
insert into location values('pune','india',1);
select * from location;
```

Tutorial 11: Auto Increment & Limit

```sql
use db;
create table student2(
sno int(3) primary key auto_increment,
sname varchar(10),
marks int(3)
);

insert into student2(sname, marks) values('vamsi', 92);
insert into student2(sname, marks) values('krishna', 94);
insert into student2(sname, marks) values('pallapolu', 96);

select * from student2;

set sql_safe_updates=0;
delete from student2;
alter table student2 auto_increment=100;
```

```sql
-- limit keyword
use hr;
select * from employees limit 10;
select * from employees limit 5, 10;

-- 3rd highest salry
select distinct salary from employees order by salary desc limit 2,1;

-- 4th highest salry
select distinct salary from employees order by salary desc limit 3,1;
```

Tutorial 12: SQL Views & Index

```
use hr;
select * from employees;
create view emp_v1 as select * from employees limit 10;
select * from emp_v1;
drop view emp_v1;

create index idx_employees on employees(first_name);
```

Tutorial 13: TCL Commands | Commit | Rollback

```
set autocommit=0;
use db;
select * from student;
describe student;
drop table if exists student;
create table student(
sno int(2),
sname varchar(10),
marks int(3)
);
insert into student values(1, 'a', 1);
insert into student values(1, 'a', 1);
insert into student values(1, 'a', 1);
commit;
select * from student;
rollback;
```

Tutorial 14: JDBC | Java Database Connectivity with MySQL

Create a connection

Create a statement Query

Execute the statement

Store the results in result set

Close the connection

```
JDBC
-----------

1) Java(JDK)
2) Eclipse
3) Mysql JDBC Driver

https://dev.mysql.com/downloads/connector/j/


insert, update delete -- DML commands


1) Create a connection
2) Create statement/Query
3) Execute statment/Query
4) close connection

select  -- Data Retrieval/Data Query language

1) Create a connection
2) Create statement/Query
3) Execute statment/Query
4) Store the results in result set
5) close connection
```

```java
package mysql;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class DMLCommands {

    public static void main(String[] args) throws SQLException {
        //Create  a connection
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/db", "root","root");
        //Create a statement
        Statement st=con.createStatement();

        //String query="insert into student values(4, 'John', 97)";
        //String query="update student set sname='smith' where sid=4";
        String query="delete from student where sid=4";

        //Execute the query
        st.execute(query);
        //Close the connection
        con.close();
        System.out.println("Query executed");
    }

}
```

```java
//Create  a connection
Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/hr", "root","root");
//Create a statement
Statement st=con.createStatement();

String query="select employee_id, first_name, last_name from employees limit 5";

//Execute the query
ResultSet rset=st.executeQuery(query);

while(rset.next()) {
    int id=rset.getInt("employee_id");
    String fname=rset.getString("first_name");
    String lname=rset.getString("last_name");
    System.out.println(id+"|"+fname+"|"+lname);

}

//Close the connection
con.close();
System.out.println("Query executed");
```

```
Console ×
<terminated> SelectCommand [Java Application] C:\Program Files\Amazon Corre
100|Steven|King
101|Neena|Kochhar
102|Lex|De Haan
103|Alexander|Hunold
104|Bruce|Ernst
Query executed
```

Tutorial 15: ODBC | Open Database Connectivity with MySQL

Client Tool

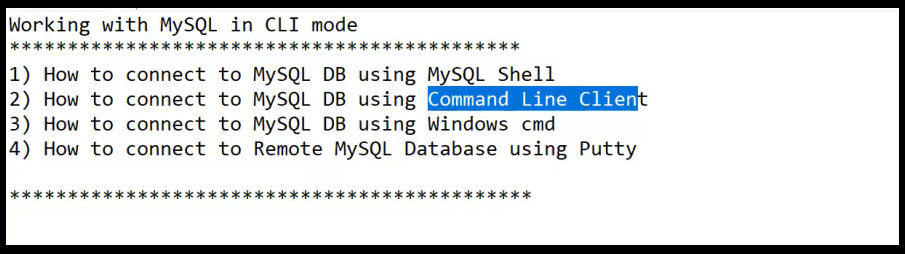
https://download.cnet.com/SQL-Query-Tool-Using-ODBC-x64-Edition/3000-10254_4-10836848.html

```
MySQL ODBC
------------
Download & Install MySQL ODBC Drivers
Creating DataSource(DSN)
Download SQL ODBC  Query Tool
Connecting to MySQL DB using ODBC

MySQL ODBC Driver/Connector
-------
https://www.mysql.com/products/connector/

DBC Client GUI tool:
https://download.cnet.com/SQL-Query-Tool-Using-ODBC-x64-Edition/3000-10254_4-10836848.html
```

Tutorial 16: MySQL Command Line Tools/Clients | Connecting to Remote MySQL Database Server



MySQL Shell commands link:

https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-commands.html

MySQL Shell Commands

- \sql
- \connect root@localhost:3306/db

To connect to my sql database using cmd:

Open cmd

cd C:\Program Files\MySQL\MySQL Server 8.0\bin

- mysql -u root -p