MySQL Tutorial by Mayur Purushvani

What is MySQL ?

- **MySQL** is a database management system.

- **MySQL** is a relational **database** management system based on SQL – Structured Query Language. The application is **used for** a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for **MySQL** however, is for the purpose of a web **database**.

Quering Data :

- SELECT :
  - Select * from [table_name];

Sorting Data :

- ORDER BY:
  - Select * from [tale_name] ORDERED BY [Col_1];

Filtering Data :

- WHERE :
  - Select * from [table_name] where [field_name];
- SELECT DISTINCT : [Unique value]
  - Select DISTINCT from [table_name];
- AND :
  - Select * from [table_name] where [condition1] AND [condition2]
- OR :
  - Select * from [table_name] where [condition1] OR [condition2]
- BETWEEN :
  - Select * from [table_name] where [field_name] BETWEEN [value1] AND [value2];
  - Select * from [employee] where [age] BETWEEN [18] AND [30];

- IN :
    - Select * from [table_name] where [field_name] IN [arg1,arg2,…]
    - Select * from [employee] where [country] IN [‘india’,’china’,’america]

- LIKE :
    - Select * from [table_name] where [field_name] LIKE [value];
    - Select * from [employee] where [first_name] LIKE ‘%a’;
- LIMIT : [Get first 5 records]
    - Select * from [table_name] where [condition] LIMIT 5;
- IS NULL / IS NOT NULL :
    - Select * from [table_name] where [condition] IS NULL
    - Select * from [table_name] where [condition] IS NOT NULL
    - E.G : Select * from [employee] where [sales] IS NULL [get the employees who don’t have a sales representative]
    - E.G : Select * from [employee] where [sales] IS NOT NULL [get the employees who have a sales representative]

Joining table :

- Table/column aliases :
    - Select [column_name] AS [Aliase_name] from [table_name];
    - Select [salary_increment] AS [incr_sal] from [employee];
- Inner join :
    - Select name, salary from employee e1 INNER JOIN employee_list e2 ON e1.salary = e2.salary;
- Outer join :
    -
- Left join : (Find all employees whose salary is in employee table)
    - Select name, salary from employee e1 LEFT JOIN employee_list e2 ON e1.salary = e2.salary;
- Right join :
    - Select name, salary from employee e1 RIGHT JOIN employee_list e2 ON e1.salary = e2.salary;
- Cross join :
    - Select name,salary from employee AS e1 CROSS JOIN employee_list AS e2;
- Self-join :
    - Select e1.name, e2.name, e1.salary from employee e1 INNER JOIN employee e2 ON e1.name=e2.name;

Grouping Data :

- Group By :
  - Select * from employee GROUP BY salary;
- Having :
  - Select * from employee HAVING salary > 30000;

- Rollup : [generate multiple grouping sets considering a hierarchy between columns specified in the GROUP BY clause.]
  - Select * from employee GROUP BY salary WITH ROLLUP;

SubQueries :

- Subquery :
  - Select * from employee Where salary = (Select max(salary) from employee);
- Derived Table : [introduce you to the derived table concept and show you how to use it to simplify complex queries.]
  - Select * from employee INNER JOIN employee_list USING (salary);
- Exists :
  - Select * from employee Where EXISTS(Select * from employee_list where employee_list.name = employee.name);

Common Table Expression :

- CTE :
  - WITH salary_gt_30k AS (SELECT name,salary from employee where salary > 30000);
- Recursive CTE :
  - WITH RECURSIVE salary_gt_30k AS (SELECT name,salary from employee where salary > 30000);

SET Operators :

- UNION and UNION ALL :
  - [combine two or more result sets from multiple SELECT statements into a single result set]
  - Select id from employee UNION select id from employee_list;

- INTERSECT :
    - [show you how to emulate the MySQL INTERSECT operator]
    - Select DISTINCT id from employee INNER JOIN employee_list USING(id);
- MINUS :
    - [explain to you the SQL MINUS operator and show you how to simulate it.]
    - Select name from employee LEFT JOIN employee_list on name where employee_list.salary IS NULL;

Modifyig data in MySQL :

- Insert :
    - Insert into employee (id,name,salary) values('1','mayur','30000');
- Insert multiple rows :
    - Insert into employee (id,name,salary) values('1','mayur','30000'),('2','manubhai','50000');
- Insert into select :
    - Insert into employee(phone,city)Select name,salary from employee where salary > 30000;
- Insert ignore :
    - Insert IGNORE into employee(email) values ('mayur@gmail.com');
- Update :
    - Update employee SET email = mp@gmail.com where id = 2;
- Update Join :
    - Update employee INNER JOIN salary ON employee.name=employee_list.name SET salary = salary + salary;
- Delete :
    - Delete from employee where id = 2;
- Delete Join :
    - Delete employee from employee LEFT JOIN employee_list ON employee.id = employee.id;
- Replace :
    - Replace into employee(id,salary) values (2,40000);
- Prepared Statement :
    - Prepare stmt From 'select name from employee where id=?';

MySQL Transaction :

- Transaction :
    - It includes START, TASACTION, COMMIT, ROLLBACK.
    - START TRANSACTION;
    - ROLLBACK;
    - COMMIT;
- Tale Locking :
    - LOCK TABLE EMPLOYEE READ;

Managing MySQL databases and tables :

- Select a MySQL db :
    - Use [db_name];
- Managing db :
    - Mysql –u root -p
    - Create Database [db_name];
    - Show databases;
- Create db :
    - Create database [db_name];
- Drop db :
    - Drop database ‘employee’;
- MySQL Storage engines :
    - [see the pdf]
- Create table :
    - Create table employee(id INT Auto_increment primary key, Name varchar(50), salary varchar(20));
- ALERT table :
    - ALTER TABLE employee ADD phone varchar(30);
- Renaming table :
    - Rename table employee TO employee_list;
- Removing a column from table :
    - ALTER TABLE employee DROP phone;
- Adding a column to a table :
    - ALTER TABLE employee ADD COLUMN phone INT;
- Drop table :
    - Drop table employee;
- Temporary tables :
    - CREATE TEMPORERY TABLE employee(id INT, name Varchar(30));
- Truncate table : [Remove all data from table.]
    - Truncate table employee;

- Generated columns :
    - Alert table employee ADD Column annual_salary DOUBLE GENERATED ALWAYS AS (salary*12) STORED;

MySQL Data Types :

- INT
- DECIMAL
- BIT
- BOOLEAN
- CHAR
- VARCHAR
- TEXT
- DATE
- TIME
- DATETIME
- TIMESTAMP
- JSON
- ENUM

MySQL Constraints :

- Not Null Constraints :
    - INSERT INTO tasks(title ,start_date, end_date) VALUES('Learn MySQL NOT NULL constraint', '2017-02-01','2017-02-02'),('Check and update NOT NULL constraint to your database', '2017-02-01',NULL);
- Primary Key Constraints :
    - Alter table employee ADD Primary KEY(ID);
- Foreign Key Constraints :
    - Alter table employee Drop Foreign key [name];
- Disable Foreign key checks :
    - SET foreign_key_checks = 0;
- UNIQUE Constraints :
    - Create table employee (id INT, name varchar(20), phone varchar(20) UNIQUE);
- Check Constraints :
    - Create table employee (id INT, name varchar(20), salary DECIMAL(10,2) NOT NULL CHECK (salary>=0));

MySQL globalization :

- Character set :
  - SET @str = CONVERT (“Hello this is mayur” USING utf8);
  - SELECT LENGTH(@str), CHAR_LENGTH(@str);
- Collection :
  - Create DATABASE employee CHARACTER SET utf8 COLLATE utf8_unicode_ci;

REFERENCE :

MySQL Tutorial - Learn MySQL Fast, Easy and Fun.

MySQL Cheat Sheet (github.com)