**MySQL NOTES**

**By Devdatta supnekar**

-- 1. check all the data base present in the system ----------------------

SHOW DATABASES;

-- 2. Create data base using following commands -----------------------

CREATE DATABASE mydatabase1;

-- 3. Drop data base using following commands ---------------------------

DROP DATABASE mydatabase1;

-- 4. select the database which you want to work on it --------------------

USE devdb;

-- 5. check all the tables present in that databases ------------------------

SHOW TABLES;

-- 6. creating the table ---------------------------------------------------------

CREATE TABLE EMP

(EMPNO INT,

ENAME VARCHAR(20),

JOB VARCHAR(20),

SALARY INT) ;

-- 7. inserting records --------------------------------------------------------

INSERT INTO EMP

VALUES(10,'SURESH','MANAGER',4500) ;

INSERT INTO EMP

VALUES(11,'RAMESH','CLERK',2500) ;

-- 8. display all columns records from table -------------------------------------

SELECT \* FROM EMP;

-- 9. display ename and job column from table-------------------------------------

SELECT ENAME, JOB FROM EMP ;

-- 10. updating table column values ------------------------------------------------

-- ( change the setting is prefrencess and restart the sql )

UPDATE EMP

SET SALARY = SALARY + 500 ;

-- 11. deleting records from table -------------------------------------------------

DELETE FROM EMP;

-- 12. droping the entire table ----------------------------------------------------

DROP TABLE EMP;

-- 13. using where clause ----------------------------------------------------------

SELECT \* FROM EMP WHERE JOB = 'MANAGER' ;

SELECT \* FROM EMP WHERE EMPNO = 10 ;

SELECT \* FROM EMP WHERE CITY = ‘MUMBAI’ ;

SELECT \* FROM EMP WHERE SALARY > 5000 ;

-- 14 updating values using where clause -------------------------------------

UPDATE EMP

SET SALARY = SALARY + 500 WHERE JOB = 'MANAGER';

-- 15. deleting all manger jobs from table -------------------------------------------

DELETE FROM EMP WHERE JOB = 'MANAGER' ;

-- 16. to add new column in exsiting table --------------------------------------------

ALTER TABLE EMP

ADD BONUS INT;

-- 17. to delete column from table -----------------------------------------------------

ALTER TABLE EMP

DROP COLUMN BONUS ;

-- 18. math function -------------------------------------------------------------------

SELECT SQRT(25)'MATH';

SELECT ABS(-10)'MATH FUNCTION';

SELECT CEILING(3.45)'MATH FUNCTION' ;

SELECT FLOOR(3.45)'MATH FUNCTION' ;

SELECT POWER(3,4)'MATH FUNCTION' ;

-- 19. creating date/time and auto-increment column -------------------------------------

DROP TABLE EMPMASTER;

CREATE TABLE EMPMASTER

(EMPNO INT auto\_increment,

ENAME VARCHAR(20),

JOB VARCHAR(20),

JDATE DATE,

SALARY INT,

HRA INT, PF INT,

NETSALARY INT,

PRIMARY KEY (EMPNO)

);

-- 20. inserting date/time values ------------------------------------------------------

INSERT INTO EMPMASTER(ENAME,JOB,JDATE,SALARY)

VALUES('MANISH','MANAGER', '2011-01-10', 9500);

INSERT INTO EMPMASTER(ENAME,JOB,JDATE,SALARY)

VALUES('SAGAR','CLERK','2011-01-11', 4500);

INSERT INTO EMPMASTER(ENAME,JOB,JDATE,SALARY)

VALUES('VIJAY','OPERATOR','2012-03-13', 3500);

INSERT INTO EMPMASTER(ENAME,JOB,JDATE,SALARY)

VALUES('VIKAS','CLERK','2013-04-22',2500);

SELECT \* FROM EMPMASTER;

-- 21. updating HRA , PF NETSALARY -------------------------------------------------------

UPDATE EMPMASTER

SET HRA = SALARY \* 5/ 100 ;

UPDATE EMPMASTER

SET PF = SALARY \* 2/ 100 ;

UPDATE EMPMASTER

SET NETSALARY = SALARY + HRA - PF ;

INSERT INTO EMPMASTER(ENAME,JOB,JDATE,SALARY)

VALUES('DEV','SUP','2016-04-08',4500);

SELECT \* FROM EMPMASTER;

-- 22. using default keyword ---------------------------------------------------------

CREATE TABLE INFO

(SRNO INT AUTO\_INCREMENT,

PNAME VARCHAR(20),

CITY VARCHAR(20) DEFAULT 'MUMBAI',

PINCODE VARCHAR(10),

PRIMARY KEY (SRNO)

)

SELECT \* FROM INFO;

INSERT INTO INFO(PNAME,PINCODE)

VALUES('RAKESH','421005') ;

-- 23. aggregate functions --------------------------------------------------------------

SELECT \* FROM EMP;

SELECT SUM(SALARY)'TOTAL SALARY' FROM EMP ;

SELECT MIN(SALARY)'MINIMUM SALARY' FROM EMP ;

SELECT MAX(SALARY)'MAXIMUM SALARY' FROM EMP ;

SELECT AVG(SALARY)'AVERAGE SALARY' FROM EMP ;

-- 24. string functions ------------------------------------------------------------------

SELECT ASCII('A') ;

select lower('SQL SERVER') ;

SELECT UPPER('sql server') ;

SELECT REPLACE('STRING FUNCTIONS','STRING','SQL') ;

SELECT LEFT('SQL FUNCTIONS',3) ;

SELECT RIGHT('SQL FUNCTIONS',9) ;

SELECT RTRIM('SILICON ') ;

SELECT LTRIM(' SILICON') ;

SELECT REVERSE('SILICON') ;

SELECT 'SILICON' + SPACE(3) + 'COMPUTER' ;

SELECT SUBSTRING('STRING FUNCTIONS',1,6) ;

-- 25 unique constraint ------------------------------------------------------------

CREATE TABLE EMPDATA100

(EMPNO INT AUTO\_INCREMENT,

ENAME VARCHAR(20),

JOB VARCHAR(20),

SALARY INT,

EMAILID VARCHAR(50),

PRIMARY KEY (EMPNO)

);

SELECT \* FROM EMPDATA100 ;

ALTER TABLE EMPDATA100

ADD CONSTRAINT UNIQUE\_CON UNIQUE(EMAILID);

INSERT INTO EMPDATA100(ENAME,JOB,SALARY,EMAILID)

VALUES('MANISH','CLERK',2500,'MANISH@GMAIL.COM');

INSERT INTO EMPDATA100(ENAME,JOB,SALARY,EMAILID)

VALUES('DEV','MANGER',8500,'MANISH@GMAIL.COM');

SELECT \* FROM EMPDATA100 ;

-- 26 group by -------------------------------------------------------------------------

SELECT JOB,SUM(SALARY) FROM EMP GROUP BY JOB ;

SELECT JOB,SUM(SALARY) FROM EMP GROUP BY JOB HAVING JOB='CLERK';

-- 27. create and manage view ----------------------------------------------

CREATE VIEW BUSSINESS\_BOOKS

AS SELECT \* FROM EMPMASTER WHERE JOB = 'BUSINESS';

SELECT \* FROM BUSSINESS\_BOOKS ;

CREATE VIEW CLERK\_VIEW AS

SELECT \* FROM EMP WHERE JOB = 'CLERK' AND SALARY > 5000;

-- 27. droping the view ----------------------------------------------------------

DROP VIEW BUSSINESS\_BOOKS ;

SELECT \* FROM BUSSINESS\_BOOKS ;

-- 28. order by clause --------------------------------------------------------------

SELECT \* FROM EMPMASTER;

SELECT ENAME , JOB , SALARY FROM EMPMASTER ORDER BY SALARY ASC;

SELECT ENAME , JOB , SALARY FROM EMPMASTER ORDER BY SALARY DESC;

################################# JOINS ##############################################

-- create the first table

Create table tblEmployee

(

ID int ,

Name varchar(50),

Gender varchar(50),

Salary int,

PRIMARY KEY (ID),

DepartmentId int,

FOREIGN KEY (DepartmentId) REFERENCES tblDepartment(ID)

);

SELECT \* FROM tblEmployee;

Insert into tblEmployee

values (1, 'Tom', 'Male', 4000, 1);

Insert into tblEmployee

values (2, 'Pam', 'Female', 3000, 3);

Insert into tblEmployee

values (3, 'John', 'Male', 3500, 1);

Insert into tblEmployee

values (4, 'Sam', 'Male', 4500, 2);

Insert into tblEmployee

values (5, 'Todd', 'Male', 2800, 2);

Insert into tblEmployee

values (6, 'Ben', 'Male', 7000, 1);

Insert into tblEmployee

values (7, 'Sara', 'Female', 4800, 3);

Insert into tblEmployee

values (8, 'Valarie', 'Female', 5500, 1);

Insert into tblEmployee

values (9, 'James', 'Male', 6500, NULL);

Insert into tblEmployee

values (10, 'Russell', 'Male', 8800, NULL);

SELECT \* FROM tblEmployee;

-- create the second table -------------------------------------

Create table tblDepartment

(

ID int primary key,

DepartmentName varchar(50),

Location varchar(50),

DepartmentHead varchar(50)

);

Insert into tblDepartment values (1, 'IT', 'London', 'Rick');

Insert into tblDepartment values (2, 'Payroll', 'Delhi', 'Ron');

Insert into tblDepartment values (3, 'HR', 'New York', 'Christie');

Insert into tblDepartment values (4, 'Other Department', 'Sydney', 'Cindrella');

select \* from tblDepartment;

-- ############################# INNER JOIN ###################################### ---

SELECT Name, Gender, Salary, DepartmentName

FROM tblEmployee

INNER JOIN tblDepartment

ON tblEmployee.DepartmentId = tblDepartment.Id;

-- ############################# LEFT JOIN ###################################### ---

SELECT Name, Gender, Salary, DepartmentName

FROM tblEmployee

LEFT OUTER JOIN tblDepartment

ON tblEmployee.DepartmentId = tblDepartment.Id;

-- ############################# RIGHT JOIN ###################################### ---

SELECT Name, Gender, Salary, DepartmentName

FROM tblEmployee

RIGHT OUTER JOIN tblDepartment

ON tblEmployee.DepartmentId = tblDepartment.Id;

-- ############################# FULL JOIN ###################################### ---

SELECT Name, Gender, Salary, DepartmentName

FROM tblEmployee

FULL JOIN tblDepartment

ON tblEmployee.DepartmentId = tblDepartment.Id;

-- ############################# CROSS JOIN ###################################### ---

SELECT Name, Gender, Salary, DepartmentName

FROM tblEmployee

CROSS JOIN tblDepartment;

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* EXTRA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*88

CREATE TABLE IF NOT EXISTS products (

productID INT NOT NULL AUTO\_INCREMENT,

productCode CHAR(3) NOT NULL DEFAULT '',

name VARCHAR(30) NOT NULL DEFAULT '',

quantity INT NOT NULL DEFAULT 0,

price DECIMAL(7,2) NOT NULL DEFAULT 99999.99,

PRIMARY KEY (productID)

);

select \* from products;

describe products;

SHOW CREATE TABLE products;

INSERT INTO products VALUES (1001, 'PEN', 'Pen Red', 5000, 1.23);

INSERT INTO products VALUES (null, 'PEN', 'Pen Red', 5000, 1.23);