CSLR51 – Database Management Systems Laboratory
#Session: 07 || Date: 12/09/2024
Viva Due: 1. Queries i to p and 3. a to s (12/09/2024)
Moodle Due: 17/09/2024 at 11 PM

1. Write the following as triggers on the corresponding schema mentioned which you have already developed. In each case, disallow if it does not satisfy the stated constraint. You may assume that the desired condition holds before any change to the database is attempted. Also, prefer to modify the database, even if it means inserting tuples with NULL or default values, rather than rejecting the attempted modification.

Flight Schema
i. Create a trigger that handles an update command to find the total salary of all pilots. Check the condition such that the new tuples inserted should not be null and salary should be more than 50,000.

```
mysql> select * from EMPLOYEES;
+-----+--------+--------+
| EID | ENAME  | SALARY |
+-----+--------+--------+
|  10 | Mac    | 150000 |
|  11 | Kick   |  49000 |
|  12 | Just   | 151000 |
|  13 | Jessy  |  25000 |
|  14 | Hiran  |  10000 |
|  15 | Hannon |  19000 |
+-----+--------+--------+
6 rows in set (0.00 sec)

mysql> CREATE TABLE TOTAL_SALARY(Total_salary int);
Query OK, 0 rows affected (0.06 sec)

mysql> INSERT INTO TOTAL_SALARY(SELECT SUM(SALARY) FROM EMPLOYEES);
Query OK, 1 row affected (0.01 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> CREATE TRIGGER Q1_2 BEFORE INSERT ON EMPLOYEES FOR EACH ROW
BEGIN DECLARE msg VARCHAR(20);  IF(NEW.SALARY IS NULL) THEN SET
msg="Error_sal"; signal sqlstate '45000' set message_text = msg;
ELSEIF(NEW.SALARY<=50000) THEN SET msg="Error_sal1"; signal sqlstate '45000' set
message_text=msg; END IF; END/
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER ;
```

```
mysql> DELETE FROM EMPLOYEES WHERE EID=34;
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO EMPLOYEES VALUES(34,'Sharv',NULL);
ERROR 1644 (45000): Error_sal
mysql> INSERT INTO EMPLOYEES VALUES(34,'Sharv',36000);
ERROR 1644 (45000): Error_sal1

mysql> SELECT * FROM TOTAL_SALARY/
+--------------+
| Total_salary |
+--------------+
|       475000 |
+--------------+
1 row in set (0.00 sec)

mysql> UPDATE EMPLOYEES SET SALARY=60000 WHERE EID=11/
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM TOTAL_SALARY/
+--------------+
| Total_salary |
+--------------+
|       485000 |
+--------------+
1 row in set (0.00 sec)
```

j. Create a trigger to set salary as 30,000 if there is a NULL present in it. Also check whether a salary of a pilot is greater than the salary of a non pilot.

```
mysql> DELIMITER /
mysql> CREATE TRIGGER Q2
    -> BEFORE INSERT ON EMPLOYEES
    -> FOR EACH ROW
    -> BEGIN
    -> IF(NEW.SALARY IS NULL) THEN
    -> SET NEW.SALARY=30000;
    -> END IF;
    -> END/
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO EMPLOYEES VALUES(36,'Shaji',NULL);
Query OK, 1 row affected (0.03 sec)
```

```
mysql> SELECT * FROM EMPLOYEES;
+-----+--------+--------+
| EID | ENAME  | SALARY |
+-----+--------+--------+
|  10 | Mac    | 150000 |
|  11 | Kick   |  60000 |
|  12 | Just   | 151000 |
|  13 | Jessy  |  25000 |
|  14 | Hiran  |  10000 |
|  15 | Hannon |  19000 |
|  34 | Sharv  |  70000 |
|  36 | Shaji  |  30000 |
+-----+--------+--------+
8 rows in set (0.01 sec)

mysql> select Eid from employees where Eid not in (select distinct(Eid) from certified natural join employees)and Salary>( select min(salary) from (select distinct(Eid),Salary from certified natural join employees) B );
+-----+
| Eid |
+-----+
|  15 |
+-----+
1 row in set (0.01 sec)
```

k. Create a trigger to foil any attempt to lower the salary of an employee.

```
mysql> CREATE TRIGGER Q3 BEFORE UPDATE ON EMPLOYEES  FOR EACH ROW BEGIN DECLARE msg VARCHAR(20);IF(NEW.SALARY<(SELECT SALARY FROM EMPLOYEES WHERE NEW.EID=EMPLOYEES.EID)) THEN SET msg="Error_update"; signal sqlstate '45000' set message_text = msg; END IF; END/
Query OK, 0 rows affected (0.03 sec)

mysql> update EMPLOYEES SET SALARY=200 WHERE EID=11/
ERROR 1644 (45000): Error_update
```

l. When inserting a new certification for an employee, check that the aircraft id exists in the Aircraft.

```
mysql> INSERT INTO CERTIFIED VALUES(11,106);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`Lab2`.`CERTIFIED`, CONSTRAINT `CERTIFIED_ibfk_2` FOREIGN KEY (`AID`) REFERENCES `AIRCRAFT` (`AID`))
```

m. When making any modifications to the Aircraft table, check that the cruisingrange is greater than or equal to distance of flights.

```
mysql> CREATE TRIGGER Qm
   -> BEFORE UPDATE ON AIRCRAFT
   -> FOR EACH ROW
   -> BEGIN
   -> DECLARE msg VARCHAR(20);
   -> IF(NEW.Cruisingrange<ANY(SELECT Distance FROM flights WHERE aid=new.Aid))
THEN

   -> SET msg="_error_";
   -> signal sqlstate '45000' set message_text=msg;
   -> END IF;
   -> END/
Query OK, 0 rows affected (0.03 sec)

mysql> update aircraft set Cruisingrange=3000 where Aid=101;
ERROR 1644 (45000): _error_
```

n. When a new certification is inserted into Certified, also insert an employee with the id of that employee and a NULL salary.

```
mysql> CREATE TRIGGER Qn
   -> AFTER INSERT ON certified
   -> FOR EACH ROW
   -> BEGIN
   -> insert into employees values(NEW.Eid,NULL,NULL);
   -> END/
Query OK, 0 rows affected (0.01 sec)

mysql> insert into certified value(1,106);
Query OK, 1 row affected (0.01 sec)
```

o. Terminate pilots and their certification when the pilot retires.

```
mysql> CREATE TRIGGER Delete_certified
   -> AFTER DELETE ON EMPLOYEES
   -> FOR EACH ROW
   -> BEGIN
   -> DELETE FROM CERTIFIED WHERE OLD.EID=CERTIFIED.EID;
   -> END/
Query OK, 0 rows affected (0.03 sec)

mysql> delete from employees where Eid=11;
Query OK, 1 row affected (0.01 sec)
```

p. Write a trigger for the condition mentioned: Suppose we want to prevent the average salary of an employee from dropping below Rs. 80,000. This constraint could be violated by an insertion, a deletion, or an update to the salary column of Employee Table.

```
mysql> CREATE TRIGGER Qp
   -> AFTER UPDATE ON employees
   -> FOR EACH ROW
   -> BEGIN
   -> DECLARE mesg VARCHAR(20);
   -> IF((select avg(Salary) FROM employees)<80000) THEN
   -> SET mesg="sal_violate";
   -> SIGNAL SQLSTATE '45000' set message_text=mesg;
   -> END IF;
   -> END/
Query OK, 0 rows affected (0.01 sec)

mysql> update employees  set Salary=0 where Eid=12;
ERROR 1644 (45000): sal_violate

mysql> CREATE TRIGGER Qp_1 AFTER INSERT ON employees FOR EACH ROW BEGIN
DECLARE mesg VARCHAR(20); IF((select avg(Salary) FROM employees)<80000) THEN SET
mesg="sal_violate1"; SIGNAL SQLSTATE '45000' set message_text=mesg; END IF; END/
Query OK, 0 rows affected (0.01 sec)

mysql> insert into employees value(1,'K',0);
ERROR 1644 (45000): sal_violate1

mysql> CREATE TRIGGER Qp_2 AFTER DELETE ON employees FOR EACH ROW BEGIN
DECLARE mesg VARCHAR(20); IF((select avg(Salary) FROM employees)<80000) THEN SET
mesg="sal_violate2"; SIGNAL SQLSTATE '45000' set message_text=mesg; END IF; END/
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> delete from Employees where Eid=10;
ERROR 1644 (45000): sal_violate2
```

2. Write the following as Cursors on the corresponding Schema.
Employee Schema
q. Develop a stored procedure to insert a new attribute ‘address’ in DEPENDENT and update the same as that of the employee’s address.

```
mysql> ALTER TABLE DEPENDENT ADD COLUMN Address VARCHAR(255);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> DELIMITER /
mysql> CREATE PROCEDURE UpdateDependentAddress()
   -> BEGIN
   -> DECLARE done INT DEFAULT FALSE;
   -> DECLARE emp_id INT;
   -> DECLARE emp_address VARCHAR(255);
   -> DECLARE emp_cursor CURSOR FOR SELECT Ssn,Address FROM employee;
   -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET done=TRUE;
   -> OPEN emp_cursor;
   -> read_loop:LOOP
   -> FETCH emp_cursor INTO emp_id,emp_address;
   -> IF done THEN
   -> LEAVE read_loop;
   -> END IF;
   -> UPDATE DEPENDENT SET Address=emp_address WHERE Essn=emp_id;
   -> END LOOP;
   -> CLOSE emp_cursor;
   -> END/
Query OK, 0 rows affected (0.03 sec)

mysql> CALL UpdateDependentAddress();
Query OK, 0 rows affected (0.03 sec)
```

r. Develop a stored procedure to display the fname, ssn and salary, grade of an employee. Handle the condition such that if salary of an employee is 1 - 10000, assign grade3, grade2 if salary in between 10000 and 50000 and grade1 if salary > 50000. Handle exception with an error message when an invalid case occurs.

```
mysql> CREATE PROCEDURE DisplayDetails1(IN emp_ssn VARCHAR(15))
   -> BEGIN
   -> DECLARE e_fname VARCHAR(25);
   -> DECLARE e_salary INT;
   -> DECLARE e_grade VARCHAR(10);
   -> DECLARE emp_exist BOOLEAN DEFAULT TRUE;
   -> DECLARE emp_cursor CURSOR FOR SELECT Fname,Salary FROM employee WHERE
Ssn=emp_ssn;
   -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET emp_exist=FALSE;
   -> OPEN emp_cursor;
   -> FETCH emp_cursor INTO e_fname,e_salary;
   -> IF emp_exist THEN
   -> IF e_salary BETWEEN 1 AND 10000 THEN
   -> SET e_grade='Grade3';
   -> ELSEIF e_salary BETWEEN 10001 AND 50000 THEN
   -> SET e_grade='Grade2';
   -> ELSEIF e_salary>50000 THEN
```

```
   -> SET e_grade='Grade1';
   -> ELSE
   -> SIGNAL SQLSTATE '45000'
   -> SET message_text='Invalid salary detected';
   -> END IF;
   -> SELECT e_fname AS Fname,emp_ssn AS Ssn,e_salary AS Salary,e_grade AS Grade;
   -> ELSE
   -> SIGNAL SQLSTATE '45000'
   -> SET message_text='Employee not found';
   -> END IF;
   -> CLOSE emp_cursor;
   -> END/
Query OK, 0 rows affected (0.01 sec)

mysql> CALL DisplayDetails1('653298655');
+---------+-----------+--------+--------+
| Fname   | Ssn       | Salary | Grade  |
+---------+-----------+--------+--------+
| Richson | 653298655 |  39000 | Grade2 |
+---------+-----------+--------+--------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL DisplayDetails1('653298000');
ERROR 1644 (45000): Employee not found
```

s. Create a stored procedure to display deptno, avgsalary and #employees in each department. Handle exceptions with an error message when invalid deptno is given.

```
mysql> CREATE PROCEDURE Display_dept_info(IN d_n INT)
   -> BEGIN
   -> DECLARE total_employees INT;
   -> DECLARE avg_salary DECIMAL(10,2);
   -> DECLARE dept_exist BOOLEAN DEFAULT FALSE;
   -> SELECT COUNT(*) INTO total_employees FROM employee WHERE Dno=d_n;
   -> IF total_employees>0 THEN
   -> SET dept_exist=TRUE;
   -> SELECT AVG(salary),COUNT(*) INTO avg_salary,total_employees FROM empl
oyee WHERE Dno=d_n GROUP BY Dno;
   -> SELECT d_n,avg_salary,total_employees;
   -> ELSE
   -> SIGNAL SQLSTATE '45000'
   -> SET message_text="Invalid dept";
   -> END IF;
```

```
   -> END/
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> CALL Display_dept_info(4);
+------+------------+-----------------+
| d_n  | avg_salary | total_employees |
+------+------------+-----------------+
|    4 |   38166.67 |               6 |
+------+------------+-----------------+
1 row in set (0.00 sec)
```

Flight Schema

t. Develop a stored procedure to update an employee record given the employee id. Print a message after the update is successfully done with an exception handling of a invalid employee id.

```
mysql> CREATE PROCEDURE UpdateEmployee(IN e_id INT,IN e_name VARCHAR(20),IN
e_sal INT)
   -> BEGIN
   -> DECLARE done INT DEFAULT 0;
   -> DECLARE p_id INT;
   -> DECLARE e_cursor CURSOR FOR SELECT Eid FROM employees WHERE Eid=e_id;

   -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET done=1;
   -> OPEN e_cursor;
   -> FETCH e_cursor INTO p_id;
   -> IF done THEN
   -> SIGNAL SQLSTATE '45000'
   -> SET message_text="Invalid eid";
   -> ELSE
   -> UPDATE employees SET Ename=e_name,Salary=e_sal WHERE Eid=p_id;
   -> END IF;
   -> END/
Query OK, 0 rows affected (0.03 sec)

mysql> CALL UpdateEmployee(1,"lo",20);
ERROR 1644 (45000): Invalid eid
```

u. Develop a stored procedure to display the name, salary of each employee from employee table. Handle the condition such that if salary of an employee is above 50,000 rank them as Grade ‘A’ else as Grade ‘B’.

```
mysql> CREATE PROCEDURE Display_employee()
   -> begin
```

```
    -> DECLARE e_id INT;
    -> DECLARE e_name VARCHAR(25);
    -> DECLARE e_sal INT;
    -> DECLARE e_grade CHAR(1);
    -> DECLARE done INT DEFAULT FALSE;
    -> DECLARE e_cursor CURSOR FOR SELECT Eid,Ename,Salary FROM employees;
    -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET done=TRUE;
    -> OPEN e_cursor;
    -> read_loop:LOOP
    -> FETCH e_cursor INTO e_id,e_name,e_sal;
    -> IF done THEN
    -> LEAVE read_loop;
    -> END IF;
    -> IF e_sal>50000 THEN
    -> SET e_grade='A';
    -> ELSEIF e_sal<=50000 THEN
    -> SET e_grade='B';
    -> END IF;
    -> SELECT e_id AS ID,e_name AS NAME,e_sal AS SALARY,e_grade AS GRADE;
    -> END LOOP;
    -> CLOSE e_cursor;
    -> END/
Query OK, 0 rows affected (0.01 sec)

mysql> CALL Display_employee();
    -> /
+------+-------+--------+-------+
| ID   | NAME  | SALARY | GRADE |
+------+-------+--------+-------+
|    9 | Larry |  20000 | B     |
+------+-------+--------+-------+
1 row in set (0.00 sec)

+------+------+--------+-------+
| ID   | NAME | SALARY | GRADE |
+------+------+--------+-------+
|   10 | Mac  | 150000 | A     |
+------+------+--------+-------+
1 row in set (0.01 sec)

+------+------+--------+-------+
| ID   | NAME | SALARY | GRADE |
+------+------+--------+-------+
|   12 | Just | 151000 | A     |
```

```
+------+------+--------+-------+
1 row in set (0.01 sec)

+------+-------+--------+-------+
| ID   | NAME  | SALARY | GRADE |
+------+-------+--------+-------+
|   13 | Jessy |  25000 | B     |
+------+-------+--------+-------+
1 row in set (0.01 sec)

+------+-------+--------+-------+
| ID   | NAME  | SALARY | GRADE |
+------+-------+--------+-------+
|   14 | Hiran |  10000 | B     |
+------+-------+--------+-------+
1 row in set (0.02 sec)

+------+--------+--------+-------+
| ID   | NAME   | SALARY | GRADE |
+------+--------+--------+-------+
|   15 | Hanoon | 150000 | A     |
+------+--------+--------+-------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.03 sec)
```

v. Develop a stored procedure that builds a name list of all employees who are certified for a Boeing aircraft and handle an exception with an error message.

```
mysql> CREATE PROCEDURE Boeing()
    -> BEGIN
    -> DECLARE e_id INT;
    -> DECLARE e_name VARCHAR(25);
    -> DECLARE done INT DEFAULT FALSE;
    -> DECLARE e_cursor CURSOR FOR SELECT Eid,Ename FROM ((employees natural join
certified) natural join aircraft) WHERE Aname='Boeing';
    -> DECLARE CONTINUE HANDLER FOR NOT FOUND SET done=TRUE;
    -> OPEN e_cursor;
    -> read_loop:LOOP
    -> FETCH e_cursor INTO e_id,e_name;
    -> IF done THEN
    -> LEAVE read_loop;
    -> END IF;
    -> SELECT e_id AS EID,e_name AS ENAME;
    -> END LOOP;
```

```
    -> CLOSE e_cursor;
    -> END/
Query OK, 0 rows affected (0.01 sec)

mysql> CALL Boeing();

+------+-------+
| EID  | ENAME |
+------+-------+
|   10 | Mac   |
+------+-------+
1 row in set (0.00 sec)

+------+-------+
| EID  | ENAME |
+------+-------+
|   10 | Mac   |
+------+-------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

Some other keyword based Queries in MySQL

3. On the Company Relational Schema, execute the following queries.

a. Display all odd numbered alternate records from 'Employee' table.

```
mysql> WITH NumberedEmployee AS (SELECT *,ROW_NUMBER() OVER( ORDER BY
(SELECT NULL)) AS rn FROM employee)
    -> SELECT * FROM NumberedEmployee WHERE (rn%4) IN (1,3);
+--------+-------+---------+-----------+-----------+----------------------+------+----------+-----------+-----+----
+
| Fname  | Minit | Lname   | Ssn       | Bdate      | Address              | Sex  | Salary   | Super_ssn |
Dno | rn |
+--------+-------+---------+-----------+-----------+----------------------+------+----------+-----------+-----+----
+
| Hiran  | L     | Farook  | 653298100 | 1962-12-30 | 21 Oak Forest,Katy,TX  | M    | 90000.00 |
65329869 |   9 |  1 |
| Allen  | C     | Mar     | 653298654 | 1962-12-30 | 99 Oak Forest,Katy,TX  | M    | 37000.00 |
653298655 |   4 |  3 |
| Rich   | G     | Mario   | 653298656 | 1962-12-30 | 101 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   4 |  5 |
| Teju   | G     | Chouhan | 653298660 | 1962-12-30 | 111 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   3 |  7 |
```

```
| Kiran  | P     | Yadav   | 653298662 | 1962-12-30 | 114 Oak Forest,Katy,TX | M    | 30000.00 |
653298661 |   1 |  9 |
| Andrea | G     | Khan    | 653298665 | 1962-12-30 | 192 Oak Forest,Katy,TX | F    | 60000.00 |
653298660 |   5 | 11 |
| Rocky  | H     | Stone   | 653298698 | 1962-12-30 | 201 Oak Forest,Katy,TX | M    | 51000.00 |
653298654 |   9 | 13 |
+--------+-------+---------+-----------+------------+-----------------------+------+----------+-----------+-----+----
+
7 rows in set (0.00 sec)
```

b. Display all even numbered alternate records from 'Employee' table.

```
mysql> WITH NumberedEmployee AS (SELECT *,ROW_NUMBER() OVER(ORDER BY
(SELECT NULL)) AS rn FROM employee)
   -> SELECT * FROM NumberedEmployee WHERE (rn%4) NOT IN (1,3);
+----------+-------+--------+-----------+------------+-----------------------+------+----------+-----------+-----+----
+
| Fname    | Minit | Lname  | Ssn       | Bdate      | Address               | Sex  | Salary   | Super_ssn |
Dno | rn |
+----------+-------+--------+-----------+------------+-----------------------+------+----------+-----------+-----+----
+
| Alam XYZ | K     | Marini | 653298653 | 1962-12-30 | 98 Oak Forest,Katy,TX  | F    | 37000.00 |
653298663 |   4 |  2 |
| Richson  | K     | Mario  | 653298655 | 1962-12-30 | 100 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   4 |  4 |
| Richton  | G     | Mario  | 653298657 | 1962-12-30 | 102 Oak Forest,Katy,TX | F    | 39000.00 |
653298653 |   4 |  6 |
| Tejas    | G     | Khana  | 653298661 | 1962-12-30 | 112 Oak Forest,Katy,TX | M    | 41000.00 |
653298660 |   2 |  8 |
| Mukesh   | H     | Ragav  | 653298663 | 1962-12-30 | 115 Oak Forest,Katy,TX | F    | 70000.00 |
653298653 |   5 | 10 |
| Ramsay   | K     | Marini | 65329869  | 1962-12-30 | 98 Oak Forest,Katy,TX  | M    | 38000.00 |
653298653 |   4 | 12 |
+----------+-------+--------+-----------+------------+-----------------------+------+----------+-----------+-----+----
+
6 rows in set (0.00 sec)
```

c. Find year from birth date when the date is a VARCHAR column instead of the proper DATE data type.

```
mysql> SELECT Bdate,YEAR(STR_TO_DATE(Bdate,'%Y-%m-%d')) AS BIRTH_YEAR FROM e
mployee;
+------------+------------+
| Bdate      | BIRTH_YEAR |
+------------+------------+
| 1962-12-30 |       1962 |
```

```
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
| 1962-12-30 |       1962 |
+------------+------------+
13 rows in set (0.00 sec)
```

d. Select first 3 characters of first name.

```
mysql> SELECT SUBSTRING(Fname,1,3) AS first_3_char FROM employee;
+--------------+
| first_3_char |
+--------------+
| Hir          |
| Ala          |
| All          |
| Ric          |
| Ric          |
| Ric          |
| Tej          |
| Tej          |
| Kir          |
| Muk          |
| And          |
| Ram          |
| Roc          |
+--------------+
13 rows in set (0.00 sec)
```

e. Find duplicate rows in a table of your choice.

```
mysql> SELECT Fname,Lname,count(*) FROM employee GROUP BY Fname,Lname HAVING
COUNT(*)>1;
+-------+-------+----------+
| Fname | Lname | count(*) |
+-------+-------+----------+
| Rich  | Mario |        2 |
+-------+-------+----------+
```

1 row in set (0.00 sec)

f. Delete the duplicate records retrieved using the above query without using a temporary table.

```
mysql> DELETE FROM employee WHERE Ssn NOT IN (SELECT MAX(A.Ssn)  AS M FROM e
mployee A GROUP BY A.Fname,A.Lname);
ERROR 1093 (HY000): You can't specify target table 'employee' for update in FROM clause
```

g. Delete the duplicate records retrieved using the above query using a temporary table.

```
mysql> CREATE TEMPORARY TABLE temp_emp AS
    -> SELECT MAX(Ssn) AS e_id FROM employee GROUP BY Fname,Lname;
Query OK, 13 rows affected (0.01 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql> DELETE FROM employee WHERE Ssn NOT IN (table temp_emp);
Query OK, 1 row affected (0.01 sec)
```

h. Extract the 3rd maximum salary. Also find nth max salary.

```
mysql> SELECT DISTINCT Salary FROM employee ORDER BY Salary DESC LIMIT 1 OFF
SET 2;
+----------+
| Salary   |
+----------+
| 60000.00 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT DISTINCT Salary FROM employee ORDER BY Salary DESC LIMIT 1 OFF
SET n-1;
```

i. Get first 3 max salaries. Also find first n max salaries.

```
mysql> SELECT DISTINCT Salary FROM employee ORDER BY Salary DESC LIMIT 3;
+----------+
| Salary   |
+----------+
| 90000.00 |
| 70000.00 |
| 60000.00 |
+----------+
3 rows in set (0.00 sec)

mysql> WITH RankedSalary AS (
    -> SELECT Salary,ROW_NUMBER() OVER(ORDER BY Salary desc) as rn FROM empl
oyee)
```

```
   -> SELECT Salary FROM RankedSalary WHERE rn<=3;
+----------+
| Salary   |
+----------+
| 90000.00 |
| 70000.00 |
| 60000.00 |
+----------+
3 rows in set (0.00 sec)

mysql> SELECT DISTINCT Salary FROM employee ORDER BY Salary DESC LIMIT n;
```

j. Display year, month, day as separate attributes from employee's date of birth.

```
mysql> SELECT DAY(Bdate) AS DATE,MONTH(Bdate) AS MONTH,YEAR(Bdate) AS YEAR f
rom employee;
+------+-------+------+
| DATE | MONTH | YEAR |
+------+-------+------+
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
|   30 |    12 | 1962 |
+------+-------+------+
13 rows in set (0.00 sec)
```

k. Retrieve the date part of the date or datetime expression.

```
mysql> SELECT distinct DAY(Bdate) AS DAY from employee;
+------+
| YEAR |
+------+
|   30 |
+------+
1 row in set (0.00 sec)

mysql> SELECT distinct DATE(Bdate) AS DATE from employee;
```

```
+------------+
| YEAR       |
+------------+
| 1962-12-30 |
+------------+
1 row in set (0.00 sec)
```

l. Get position of 'a' in name 'Sundar Pitchai' from employee table.

```
mysql> SELECT LOCATE('a',Fname) AS position_of_a FROM employee WHERE Fname='
Sundar';
+---------------+
| position_of_a |
+---------------+
|             5 |
+---------------+
1 row in set (0.00 sec)
```

m. Get fname from employee table after removing white spaces from left side.

```
mysql> SELECT LTRIM(Fname) AS TRIMMED_NAME FROM employee;
+--------------+
| TRIMMED_NAME |
+--------------+
| Sundar       |
| Hiran        |
| Alam XYZ     |
| Allen        |
| Richson      |
| Rich         |
| Richton      |
| Teju         |
| Tejas        |
| Kiran        |
| Mukesh       |
| Andrea       |
| Ramsay       |
| Rocky        |
+--------------+
14 rows in set (0.00 sec)
```

n. Get length of fname from employee table.

```
mysql> SELECT Fname,LENGTH(Fname) AS LENGTH FROM employee;
+----------+--------+
| Fname    | LENGTH |
+----------+--------+
```

Sundar	6
Hiran	5
Alam XYZ	8
Allen	5
Richson	7
Rich	4
Richton	7
Teju	4
Tejas	5
Kiran	5
Mukesh	6
Andrea	6
Ramsay	6
Rocky	5
+----------+--------+
14 rows in set (0.00 sec)

o. Get fname from employee table after replacing 'o' with '*'.

```
mysql> SELECT Fname,REPLACE(Fname,'o','*') AS UPDATED_NAME FROM employee;
+----------+--------------+
| Fname    | UPDATED_NAME |
+----------+--------------+
| Sundar   | Sundar       |
| Hiran    | Hiran        |
| Alam XYZ | Alam XYZ     |
| Allen    | Allen        |
| Richson  | Richs*n      |
| Rich     | Rich         |
| Richton  | Richt*n      |
| Teju     | Teju         |
| Tejas    | Tejas        |
| Kiran    | Kiran        |
| Mukesh   | Mukesh       |
| Andrea   | Andrea       |
| Ramsay   | Ramsay       |
| Rocky    | R*cky        |
+----------+--------------+
14 rows in set (0.00 sec)
```

p. Get fname and lname as a single attribute from employee table separated by a '_'.

```
mysql> SELECT CONCAT(Fname,'_',Lname) from employee;
+------------------------+
| CONCAT(Fname,'_',Lname) |
+------------------------+
```

| Sundar_Pitchai |
| Hiran_Farook |
| Alam XYZ_Marini |
| Allen_Mar |
| Richson_Mario |
| Rich_Mario |
| Richton_Mario |
| Teju_Chouhan |
| Tejas_Khana |
| Kiran_Yadav |
| Mukesh_Ragav |
| Andrea_Khan |
| Ramsay_Marini |
| Rocky_Stone |
+------------------------+
14 rows in set (0.00 sec)

q. Find all employee records containing the word "Jai", regardless of whether it was stored as JAI, Jai, or jai.

```
mysql> SELECT * FROM employee WHERE LOWER(Fname) LIKE '%jai%';
+-------+-------+---------+-----------+------------+---------------------+------+----------+-----------+-----+
| Fname | Minit | Lname   | Ssn       | Bdate      | Address             | Sex  | Salary   | Super_ssn | Dno |
+-------+-------+---------+-----------+------------+---------------------+------+----------+-----------+-----+
| VIJai | K     | Pitchai | 653298000 | 1962-12-30 | 1 Oak Forest,Katy,TX | M    | 39000.00 | 653298653 |   4 |
+-------+-------+---------+-----------+------------+---------------------+------+----------+-----------+-----+
1 row in set (0.01 sec)
```

r. Find the number of employees according to the gender whose DOB is between 05/01/1980 to 31/12/2024.

```
mysql> SELECT Sex,COUNT(*) FROM employee WHERE Bdate BETWEEN '1980-01-05' AN
D '2024-12-31' GROUP  BY Sex;
+------+----------+
| Sex  | COUNT(*) |
+------+----------+
| M    |        2 |
| F    |        1 |
+------+----------+
2 rows in set (0.00 sec)
```

s. Retrieve the mysql username and password.

```
mysql> SELECT USER();
```

```
+----------------+
| USER()         |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)
```

Password Hashing: MySQL stores user passwords as hashed values in the mysql.user table. For MySQL versions starting from 5.7, the authentication_string column in the mysql.user table contains these hashes. Hashing is a one-way function, which means it's computationally infeasible to revert the hash back to the original password.

t. Find all the employee first name/s whose name consists of three or more words.

```
mysql> select Fname from employee WHERE (LENGTH(Fname)-LENGTH(REPLACE(Fname,
' ',''))+1)>=3;
+------------+
| Fname      |
+------------+
| Nan Yu Chi |
+------------+
1 row in set (0.00 sec)
```

u. Get employee details from employee table whose first name ends with 'n' and name contains 4 letters.

```
mysql> select * from employee WHERE Fname like '___n';
+-------+-------+---------+-----------+------------+---------------------+------+----------+-----------+-----+
| Fname | Minit | Lname   | Ssn       | Bdate      | Address             | Sex  | Salary   | Super_ssn |
Dno |
+-------+-------+---------+-----------+------------+---------------------+------+----------+-----------+-----+
| Phin  | K     | Pitchai | 653298000 | 1962-12-30 | 1 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   4 |
+-------+-------+---------+-----------+------------+---------------------+------+----------+-----------+-----+
1 row in set (0.00 sec)
```

v. Get employee details from employee table whose joining month is “January”.

```
mysql> SELECT * FROM employee WHERE MONTH(Bdate)=1;
+-------+-------+---------+-----------+------------+---------------------+------+----------+-----------+-----+
| Fname | Minit | Lname   | Ssn       | Bdate      | Address             | Sex  | Salary   | Super_ssn |
Dno |
+-------+-------+---------+-----------+------------+---------------------+------+----------+-----------+-----+
| Phin  | K     | Pitchai | 653298000 | 1962-01-30 | 1 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   4 |
+-------+-------+---------+-----------+------------+---------------------+------+----------+-----------+-----+
1 row in set (0.00 sec)
```

w. Fetch data that are common in two query results.

```
mysql> SELECT * FROM ( select * from employee WHERE Salary>38000)  T1 NATURAL JOIN (select * from employee WHERE Fname like '%n') T2;
+---------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
| Fname   | Minit | Lname   | Ssn       | Bdate      | Address              | Sex  | Salary   | Super_ssn | Dno |
+---------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
| Phin    | K     | Pitchai | 653298000 | 1962-01-30 | 1 Oak Forest,Katy,TX   | M    | 39000.00 | 653298653 |   4 |
| Hiran   | L     | Farook  | 653298100 | 1962-12-30 | 21 Oak Forest,Katy,TX  | M    | 90000.00 | 65329869  |   9 |
| Richson | K     | Mario   | 653298655 | 1984-12-03 | 100 Oak Forest,Katy,TX | M    | 39000.00 | 653298653 |   4 |
| Richton | G     | Mario   | 653298657 | 1984-12-03 | 102 Oak Forest,Katy,TX | F    | 39000.00 | 653298653 |   4 |
+---------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
4 rows in set (0.00 sec)
```

x. Get first names of employees who has '*' in last_name.

```
mysql> SELECT Fname FROM employee WHERE Lname LIKE '%*%';
+-------+
| Fname |
+-------+
| Phin  |
+-------+
1 row in set (0.00 sec)
```

y. Find department from dept table after replacing special character with a white space.

```
mysql> SELECT Dname FROM department;
+----------------+
| Dname          |
+----------------+
| Accounts       |
| Administration |
| C#S            |
| Clinic         |
| Emergency      |
| Headquarters   |
| I*T            |
| Research       |
| Unassigned     |
+----------------+
9 rows in set (0.00 sec)
```

```
mysql> SELECT REPLACE(
    -> REPLACE(
    -> Dname,'#',' '),
    -> '*',' ') AS
    -> CLEANED_DEPARTMENT FROM department;
+--------------------+
| CLEANED_DEPARTMENT |
+--------------------+
| Accounts           |
| Administration     |
| C S                |
| Clinic             |
| Emergency          |
| Headquarters       |
| I T                |
| Research           |
| Unassigned         |
+--------------------+
9 rows in set (0.00 sec)

mysql> SELECT REGEXP_REPLACE(Dname,'[^a-zA-Z0-9]',' ') AS CLEANED_DEPARTMENT
 FROM department;
+--------------------+
| CLEANED_DEPARTMENT |
+--------------------+
| Accounts           |
| Administration     |
| C S                |
| Clinic             |
| Emergency          |
| Headquarters       |
| I T                |
| Research           |
| Unassigned         |
+--------------------+
9 rows in set (0.02 sec)
```

z. Retrieve the number of employees joined with respect to a particular year and a particular month from employee table.

```
mysql> SELECT YEAR(Bdate),MONTH(Bdate),COUNT(*) FROM employee GROUP BY YEAR(
Bdate),MONTH(Bdate);
+-------------+--------------+----------+
| YEAR(Bdate) | MONTH(Bdate) | COUNT(*) |
```

```
+-------------+--------------+----------+
|        1962 |            1 |        1 |
|        1962 |           12 |       10 |
|        1984 |           12 |        3 |
+-------------+--------------+----------+
3 rows in set (0.00 sec)
```

aa. Extract characters within a specified range of length from department field.

```
mysql> SELECT SUBSTR(Dname,2,5) FROM DEPARTMENT;
+------------------+
| SUBSTR(Dname,2,5) |
+------------------+
| ccoun            |
| dmini            |
| #S               |
| linic            |
| merge            |
| eadqu            |
| *T               |
| esear            |
| nassi            |
+------------------+
9 rows in set (0.00 sec)
```

bb. Convert the name of the employee to lowercase and then as uppercase.

```
mysql> SELECT LOWER(Fname),UPPER(Fname) FROM employee;
+--------------+--------------+
| LOWER(Fname) | UPPER(Fname) |
+--------------+--------------+
| phin         | PHIN         |
| hiran        | HIRAN        |
| alam xyz     | ALAM XYZ     |
| allen        | ALLEN        |
| richson      | RICHSON      |
| rich         | RICH         |
| richton      | RICHTON      |
| teju         | TEJU         |
| tejas        | TEJAS        |
| kiran        | KIRAN        |
| mukesh       | MUKESH       |
| andrea       | ANDREA       |
| ramsay       | RAMSAY       |
| rocky        | ROCKY        |
+--------------+--------------+
```

14 rows in set (0.00 sec)

cc. Select FIRST n records from a department table.

```
mysql> SELECT * FROM DEPARTMENT LIMIT 3;
+----------------+---------+-----------+----------------+
| Dname          | Dnumber | Mgr_ssn   | Mgr_start_date |
+----------------+---------+-----------+----------------+
| Unassigned     |       0 | 653298660 | 1980-09-07     |
| Research       |       1 | 653298653 | 1988-05-22     |
| Administration |       2 | 653298654 | 1988-05-22     |
+----------------+---------+-----------+----------------+
3 rows in set (0.00 sec)
```

(Replace 3 by n)

dd. Select LAST n records from a department table.

```
mysql> SELECT * FROM DEPARTMENT ORDER BY  Dnumber DESC LIMIT 3;
+-----------+---------+-----------+----------------+
| Dname     | Dnumber | Mgr_ssn   | Mgr_start_date |
+-----------+---------+-----------+----------------+
| Emergency |      10 | 653298660 | 1988-07-23     |
| C#S       |       9 | 653298662 | 1988-06-22     |
| Accounts  |       6 | 65329869  | 1988-06-22     |
+-----------+---------+-----------+----------------+
3 rows in set (0.00 sec)
```

(Replace 3 by n)

ee. Select first name from employee table which contain only numbers.

```
mysql> SELECT Fname FROM employee WHERE Fname REGEXP '^[0-9]+$';
+-------+
| Fname |
+-------+
| 5678  |
+-------+
1 row in set (0.00 sec)
```

ff. Get fname, lname from employee table as separate rows.

```
mysql> SELECT Fname AS Name,'First_name' AS type FROM employee
    -> UNION ALL
    -> SELECT Lname AS Name,'Last_name' AS type FROM employee
    -> ;
+----------+------------+
| Name     | type       |
+----------+------------+
| 5678     | First_name |
```

Hiran	First_name
Alam XYZ	First_name
Allen	First_name
Richson	First_name
Rich	First_name
Richton	First_name
Teju	First_name
Tejas	First_name
Kiran	First_name
Mukesh	First_name
Andrea	First_name
Ramsay	First_name
Rocky	First_name
S*GH	Last_name
Farook	Last_name
Marini	Last_name
Mar	Last_name
Mario	Last_name
Mario	Last_name
Mario	Last_name
Chouhan	Last_name
Khana	Last_name
Yadav	Last_name
Ragav	Last_name
Khan	Last_name
Marini	Last_name
Stone	Last_name
+----------+------------+
28 rows in set (0.00 sec)

gg. Create an empty table emptem with the same structure as emp.
mysql> CREATE TABLE emptem LIKE employee;
Query OK, 0 rows affected (0.07 sec)

hh. If there are two tables emp1 and emp2, and both have common records, fetch all the records, but common records only once.
mysql> SELECT * FROM (select * from employee WHERE Salary>38000) T UNION SELECT * FROM (select * from employee WHERE Fname like '%n') V;
+---------+-------+---------+-----------+------------+-----------------------+------+----------+-----------+-----+
| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
+---------+-------+---------+-----------+------------+-----------------------+------+----------+-----------+-----+
| 5678 | K | S*GH | 653298000 | 1962-01-30 | 1 Oak Forest,Katy,TX | M | 39000.00 | 653298653 | 4 |

Hiran	L	Farook	653298100	1962-12-30	21 Oak Forest,Katy,TX	M	90000.00	65329869	9
Richson	K	Mario	653298655	1984-12-03	100 Oak Forest,Katy,TX	M	39000.00	653298653	4
Rich	G	Mario	653298656	1984-12-03	101 Oak Forest,Katy,TX	M	39000.00	653298653	4
Richton	G	Mario	653298657	1984-12-03	102 Oak Forest,Katy,TX	F	39000.00	653298653	4
Teju	G	Chouhan	653298660	1962-12-30	111 Oak Forest,Katy,TX	M	39000.00	653298653	3
Tejas	G	Khana	653298661	1962-12-30	112 Oak Forest,Katy,TX	M	41000.00	653298660	2
Mukesh	H	Ragav	653298663	1962-12-30	115 Oak Forest,Katy,TX	F	70000.00	653298653	5
Andrea	G	Khan	653298665	1962-12-30	192 Oak Forest,Katy,TX	F	60000.00	653298660	5
Rocky	H	Stone	653298698	1962-12-30	201 Oak Forest,Katy,TX	M	51000.00	653298654	9
Allen	C	Mar	653298654	1962-12-30	99 Oak Forest,Katy,TX	M	37000.00	653298655	4
Kiran	P	Yadav	653298662	1962-12-30	114 Oak Forest,Katy,TX	M	30000.00	653298661	1
+---------+-------+---------+-----------+------------+-----------------------+------+----------+-----------+-----+
12 rows in set (0.00 sec)

ii. Extract only common records from two tables emp1 and emp2.

```
mysql> SELECT * FROM ( select * from employee WHERE Salary>38000)  T1 NATURAL JOIN (select * from employee WHERE Fname like '%n') T2;
```

+---------+-------+--------+-----------+------------+-----------------------+------+----------+-----------+-----+
| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
+---------+-------+--------+-----------+------------+-----------------------+------+----------+-----------+-----+
Hiran	L	Farook	653298100	1962-12-30	21 Oak Forest,Katy,TX	M	90000.00	65329869	9
Richson	K	Mario	653298655	1984-12-03	100 Oak Forest,Katy,TX	M	39000.00	653298653	4
Richton	G	Mario	653298657	1984-12-03	102 Oak Forest,Katy,TX	F	39000.00	653298653	4
+---------+-------+--------+-----------+------------+-----------------------+------+----------+-----------+-----+
3 rows in set (0.00 sec)

jj. Retrieve all records of emp1 those should not present in emp2?

```
mysql> SELECT * FROM ( select * from employee WHERE Salary>38000) T1 EXCEPT SELECT * FROM (select * from employee WHERE Fname like '%n') T2;
```

```
+--------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
| Fname  | Minit | Lname   | Ssn       | Bdate      | Address              | Sex  | Salary   | Super_ssn | Dno |
+--------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
| 5678   | K     | S*GH    | 653298000 | 1962-01-30 | 1 Oak Forest,Katy,TX   | M    | 39000.00 | 653298653 |   4 |
| Rich   | G     | Mario   | 653298656 | 1984-12-03 | 101 Oak Forest,Katy,TX | M    | 39000.00 | 653298653 |   4 |
| Teju   | G     | Chouhan | 653298660 | 1962-12-30 | 111 Oak Forest,Katy,TX | M    | 39000.00 | 653298653 |   3 |
| Tejas  | G     | Khana   | 653298661 | 1962-12-30 | 112 Oak Forest,Katy,TX | M    | 41000.00 | 653298660 |   2 |
| Mukesh | H     | Ragav   | 653298663 | 1962-12-30 | 115 Oak Forest,Katy,TX | F    | 70000.00 | 653298653 |   5 |
| Andrea | G     | Khan    | 653298665 | 1962-12-30 | 192 Oak Forest,Katy,TX | F    | 60000.00 | 653298660 |   5 |
| Rocky  | H     | Stone   | 653298698 | 1962-12-30 | 201 Oak Forest,Katy,TX | M    | 51000.00 | 653298654 |   9 |
+--------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
7 rows in set (0.00 sec)

mysql> SELECT T.* FROM (SELECT * FROM employee WHERE Salary>38000) T LEFT JOIN
(SELECT * FROM employee WHERE Fname LIKE "%n") R ON T.Ssn=R.Ssn WHERE R.Ssn IS
NULL;
+--------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
| Fname  | Minit | Lname   | Ssn       | Bdate      | Address              | Sex  | Salary   | Super_ssn | Dno |
+--------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
| 5678   | K     | S*GH    | 653298000 | 1962-01-30 | 1 Oak Forest,Katy,TX   | M    | 39000.00 | 653298653 |   4 |
| Rich   | G     | Mario   | 653298656 | 1984-12-03 | 101 Oak Forest,Katy,TX | M    | 39000.00 | 653298653 |   4 |
| Teju   | G     | Chouhan | 653298660 | 1962-12-30 | 111 Oak Forest,Katy,TX | M    | 39000.00 | 653298653 |   3 |
| Tejas  | G     | Khana   | 653298661 | 1962-12-30 | 112 Oak Forest,Katy,TX | M    | 41000.00 | 653298660 |   2 |
| Mukesh | H     | Ragav   | 653298663 | 1962-12-30 | 115 Oak Forest,Katy,TX | F    | 70000.00 | 653298653 |   5 |
| Andrea | G     | Khan    | 653298665 | 1962-12-30 | 192 Oak Forest,Katy,TX | F    | 60000.00 | 653298660 |   5 |
| Rocky  | H     | Stone   | 653298698 | 1962-12-30 | 201 Oak Forest,Katy,TX | M    | 51000.00 | 653298654 |   9 |
+--------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
7 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM employee E WHERE Salary>38000 AND NOT EXISTS(SELECT 9
FROM employee E2 WHERE E2.Fname LIKE '%n' AND E.Ssn=E2.Ssn);
+--------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
| Fname  | Minit | Lname   | Ssn       | Bdate      | Address              | Sex  | Salary   | Super_ssn |
Dno |
+--------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
| 5678   | K     | S*GH    | 653298000 | 1962-01-30 | 1 Oak Forest,Katy,TX   | M    | 39000.00 |
653298653 |   4 |
| Rich   | G     | Mario   | 653298656 | 1984-12-03 | 101 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   4 |
| Teju   | G     | Chouhan | 653298660 | 1962-12-30 | 111 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   3 |
| Tejas  | G     | Khana   | 653298661 | 1962-12-30 | 112 Oak Forest,Katy,TX | M    | 41000.00 |
653298660 |   2 |
| Mukesh | H     | Ragav   | 653298663 | 1962-12-30 | 115 Oak Forest,Katy,TX | F    | 70000.00 |
653298653 |   5 |
| Andrea | G     | Khan    | 653298665 | 1962-12-30 | 192 Oak Forest,Katy,TX | F    | 60000.00 |
653298660 |   5 |
| Rocky  | H     | Stone   | 653298698 | 1962-12-30 | 201 Oak Forest,Katy,TX | M    | 51000.00 |
653298654 |   9 |
+--------+-------+---------+-----------+------------+----------------------+------+----------+-----------+-----+
7 rows in set (0.00 sec)
```

kk. Find rows that contain at least one of the two words ‘mysql’, ‘oracle’.

```
mysql> SELECT * FROM employee WHERE Fname LIKE '%mysql%' OR Fname LIKE
'%oracle%';
+-------+-------+-------+-----------+------------+--------------------+------+----------+-----------+-----+
| Fname | Minit | Lname | Ssn       | Bdate      | Address            | Sex  | Salary   | Super_ssn |
Dno |
+-------+-------+-------+-----------+------------+--------------------+------+----------+-----------+-----+
| mysql | K     | S*GH  | 653298000 | 1962-01-30 | 1 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   4 |
+-------+-------+-------+-----------+------------+--------------------+------+----------+-----------+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM employee WHERE MATCH(Fname) AGAINST('mysql oracle');
+-------+-------+-------+-----------+------------+--------------------+------+----------+-----------+-----+
| Fname | Minit | Lname | Ssn       | Bdate      | Address            | Sex  | Salary   | Super_ssn |
Dno |
+-------+-------+-------+-----------+------------+--------------------+------+----------+-----------+-----+
| mysql | K     | S*GH  | 653298000 | 1962-01-30 | 1 Oak Forest,Katy,TX | M    | 39000.00 |
653298653 |   4 |
+-------+-------+-------+-----------+------------+--------------------+------+----------+-----------+-----+
```

1 row in set (0.00 sec)

II. In a string attribute of the company schema, match the following using regular expression.

i) Beginning of the string.

```
mysql> SELECT Fname FROM employee WHERE Fname REGEXP '^m';
+--------+
| Fname  |
+--------+
| mysql  |
| Mukesh |
+--------+
2 rows in set (0.00 sec)
```

ii) Match any character (including carriage return and newline).

```
mysql> SELECT Fname FROM employee WHERE Fname REGEXP '^[\\s\\S]*';
+----------+
| Fname    |
+----------+
| mysql    |
| Hiran    |
| Alam XYZ |
| Allen    |
| Richson  |
| Rich     |
| Richton  |
| Teju     |
| Tejas    |
| Kiran    |
| Mukesh   |
| Andrea   |
| Ramsay   |
| Rocky    |
+----------+
14 rows in set (0.00 sec)
```

iii) Match the end of a string.

```
mysql> SELECT Fname FROM employee WHERE Fname REGEXP 'n$';
+---------+
| Fname   |
+---------+
| Hiran   |
| Allen   |
| Richson |
```

| Richton |
| Kiran |
+---------+
5 rows in set (0.00 sec)

iv) Any sequence of zero or more characters.

```
mysql> SELECT Fname FROM employee WHERE Fname REGEXP '.*';
+----------+
| Fname    |
+----------+
| mysql    |
| Hiran    |
| Alam XYZ |
| Allen    |
| Richson  |
| Rich     |
| Richton  |
| Teju     |
| Tejas    |
| Kiran    |
| Mukesh   |
| Andrea   |
| Ramsay   |
| Rocky    |
+----------+
14 rows in set (0.00 sec)
```

v) Either of the sequences xy or abc.

```
mysql> SELECT Fname FROM employee WHERE Fname REGEXP 'xy|abc';
+----------+
| Fname    |
+----------+
| Alam XYZ |
+----------+
1 row in set (0.00 sec)
```