CSLR-51 DBMS SESSION 7

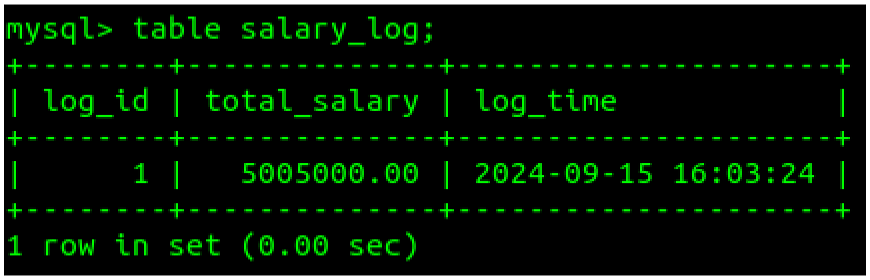
1. Write the following as triggers on the corresponding schema mentioned which you have already developed. In each case, disallow if it does not satisfy the stated constraint. You may assume that the desired condition holds before any change to the database is attempted. Also, prefer to modify the database, even if it means inserting tuples with NULL or default values, rather than rejecting the attempted modification. Employee Schema (already given as part of S5 So, no need to do this again if you have done already. In that case, proceed from 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.

Query:

CREATE TRIGGER find_tot AFTER UPDATE ON employees FOR EACH ROW BEGIN DECLARE tot_sal INT; IF (OLD.salary <> NEW.salary) THEN SELECT SUM(salary) INTO tot_sal FROM employees; INSERT INTO salary_log(total_salary) VALUES (tot_sal); END IF; END

Output:

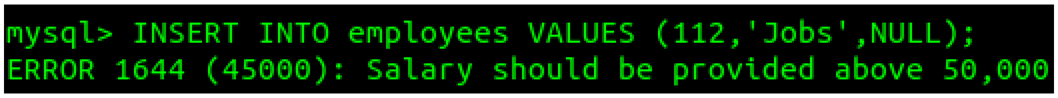


```
mysql> table salary_log;
+--------+--------------+---------------------+
| log_id | total_salary | log_time            |
+--------+--------------+---------------------+
|      1 |   5005000.00 | 2024-09-15 16:03:24 |
+--------+--------------+---------------------+
1 row in set (0.00 sec)
```

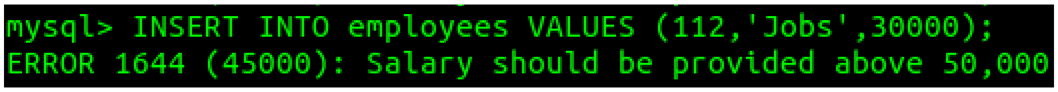
Query:

CREATE TRIGGER chk_sal BEFORE INSERT ON employees FOR EACH ROW BEGIN IF (NEW.salary IS NULL OR NEW.salary <= 50000) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Salary should be provided above 50,000"; END IF; END

Output:



```
mysql> INSERT INTO employees VALUES (112,'Jobs',NULL);
ERROR 1644 (45000): Salary should be provided above 50,000
```



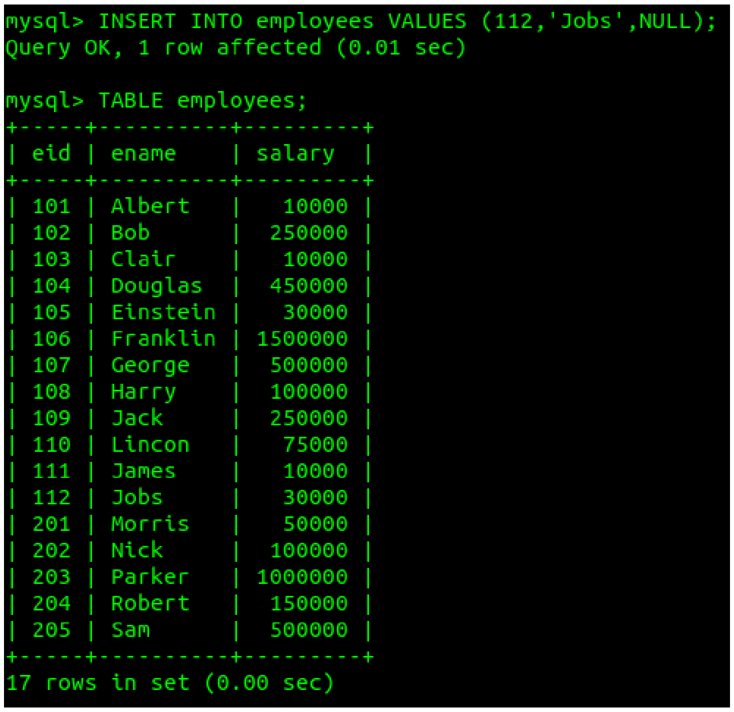
```
mysql> INSERT INTO employees VALUES (112,'Jobs',30000);
ERROR 1644 (45000): Salary should be provided above 50,000
```

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

Query:

CREATE TRIGGER set_sal BEFORE INSERT ON employees FOR EACH ROW BEGIN IF (NEW.salary IS NULL) THEN SET NEW.salary = 30000; END IF; END

Output:

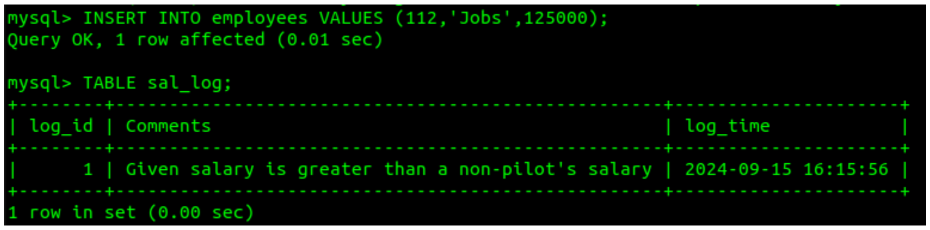


```
mysql> INSERT INTO employees VALUES (112,'Jobs',NULL);
Query OK, 1 row affected (0.01 sec)

mysql> TABLE employees;
+-----+----------+---------+
| eid | ename    | salary  |
+-----+----------+---------+
| 101 | Albert   |   10000 |
| 102 | Bob      |  250000 |
| 103 | Clair    |   10000 |
| 104 | Douglas  |  450000 |
| 105 | Einstein |   30000 |
| 106 | Franklin | 1500000 |
| 107 | George   |  500000 |
| 108 | Harry    |  100000 |
| 109 | Jack     |  250000 |
| 110 | Lincon   |   75000 |
| 111 | James    |   10000 |
| 112 | Jobs     |   30000 |
| 201 | Morris   |   50000 |
| 202 | Nick     |  100000 |
| 203 | Parker   | 1000000 |
| 204 | Robert   |  150000 |
| 205 | Sam      |  500000 |
+-----+----------+---------+
17 rows in set (0.00 sec)
```

Query:

CREATE TRIGGER pilot_sal BEFORE INSERT ON employees FOR EACH ROW BEGIN IF (NEW.salary > SOME (SELECT T3.salary FROM ((SELECT E1.eid,E1.salary FROM employees AS E1) EXCEPT (SELECT E2.eid,E2.salary FROM employees AS E2 NATURAL JOIN certified AS C2)) AS T3)) THEN INSERT INTO sal_log(Comments) VALUES ("Given salary is greater than a non-pilot's salary"); ELSE INSERT INTO sal_log(Comments) VALUES ("Given salary is lesser than all of non-pilot's salary"); END IF; END

Output:



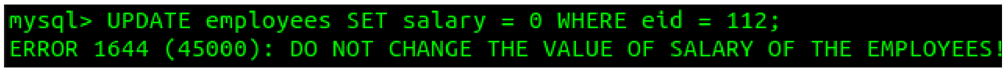
```
mysql> INSERT INTO employees VALUES (112,'Jobs',125000);
Query OK, 1 row affected (0.01 sec)

mysql> TABLE sal_log;
+--------+---------------------------------------------------+---------------------+
| log_id | Comments                                          | log_time            |
+--------+---------------------------------------------------+---------------------+
|      1 | Given salary is greater than a non-pilot's salary | 2024-09-15 16:15:56 |
+--------+---------------------------------------------------+---------------------+
1 row in set (0.00 sec)
```

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

Query:

CREATE TRIGGER sec_sal BEFORE UPDATE ON employees FOR EACH ROW BEGIN IF(OLD.salary <> NEW.salary) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "DO NOT CHANGE THE VALUE OF SALARY OF THE EMPLOYEES!"; END IF; END

Output:



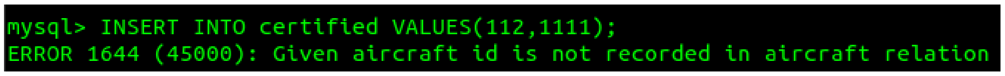
```
mysql> UPDATE employees SET salary = 0 WHERE eid = 112;
ERROR 1644 (45000): DO NOT CHANGE THE VALUE OF SALARY OF THE EMPLOYEES!
```

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

Query:

CREATE TRIGGER check_aid BEFORE INSERT ON certified FOR EACH ROW BEGIN IF(NEW.aid NOT IN (SELECT aid FROM aircraft)) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Given aircraft id is not recorded in aircraft relation"; END IF; END

Output:



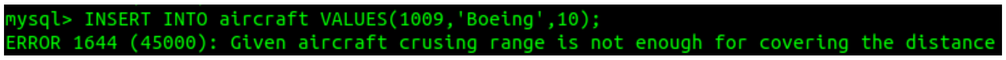
```
mysql> INSERT INTO certified VALUES(112,1111);
ERROR 1644 (45000): Given aircraft id is not recorded in aircraft relation
```

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

Query:

CREATE TRIGGER check_cr BEFORE INSERT ON aircraft FOR EACH ROW BEGIN IF (NEW.crusingrange < (SELECT MIN(distance) FROM flights)) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Given aircraft crusing range is not enough for covering the distance"; END IF; END

Output:



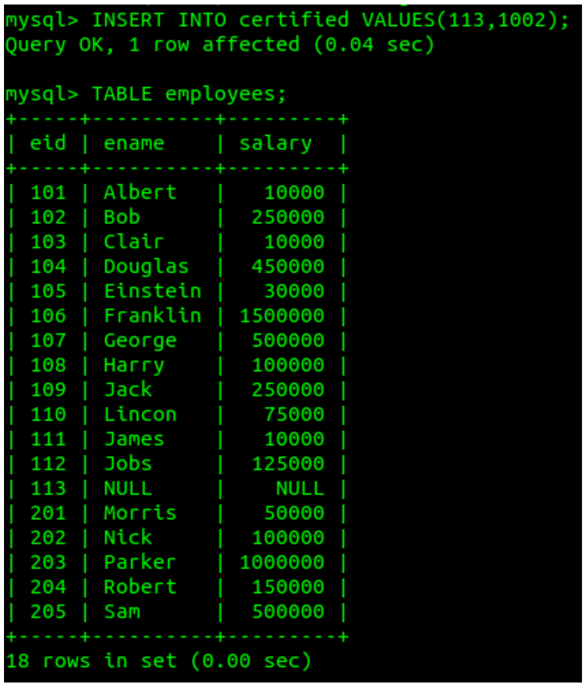
```
mysql> INSERT INTO aircraft VALUES(1009,'Boeing',10);
ERROR 1644 (45000): Given aircraft crusing range is not enough for covering the distance
```

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

Query:

CREATE TRIGGER ins_eid AFTER INSERT ON certified FOR EACH ROW BEGIN INSERT INTO employees VALUES (NEW.eid,NULL,NULL); END

Output:



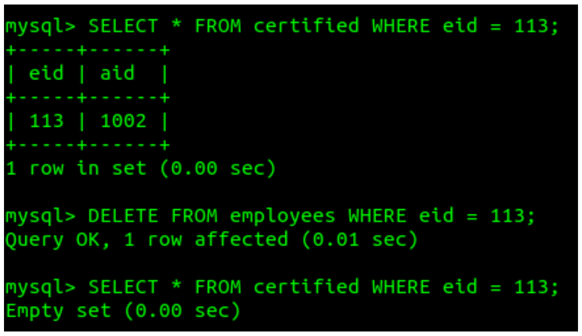
```
mysql> INSERT INTO certified VALUES(113,1002);
Query OK, 1 row affected (0.04 sec)

mysql> TABLE employees;
+-----+----------+---------+
| eid | ename    | salary  |
+-----+----------+---------+
| 101 | Albert   |   10000 |
| 102 | Bob      |  250000 |
| 103 | Clair    |   10000 |
| 104 | Douglas  |  450000 |
| 105 | Einstein |   30000 |
| 106 | Franklin | 1500000 |
| 107 | George   |  500000 |
| 108 | Harry    |  100000 |
| 109 | Jack     |  250000 |
| 110 | Lincon   |   75000 |
| 111 | James    |   10000 |
| 112 | Jobs     |  125000 |
| 113 | NULL     |    NULL |
| 201 | Morris   |   50000 |
| 202 | Nick     |  100000 |
| 203 | Parker   | 1000000 |
| 204 | Robert   |  150000 |
| 205 | Sam      |  500000 |
+-----+----------+---------+
18 rows in set (0.00 sec)
```

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

<u>Query:</u>

CREATE TRIGGER del_cert AFTER DELETE ON employees FOR EACH ROW BEGIN DELETE FROM certified WHERE eid = OLD.eid; END

<u>Output:</u>



```
mysql> SELECT * FROM certified WHERE eid = 113;
+-----+------+
| eid | aid  |
+-----+------+
| 113 | 1002 |
+-----+------+
1 row in set (0.00 sec)

mysql> DELETE FROM employees WHERE eid = 113;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM certified WHERE eid = 113;
Empty set (0.00 sec)
```

p. Write a trigger for the condition mentioned: Suppose we want to prevent the average salary of an employee from dropping below Rs. 50,000. This constraint could be violated by an insertion, a deletion, or an update to the salary column of the Employee Table.

<u>Query:</u>

CREATE TRIGGER chk_avg_ins AFTER INSERT ON employees FOR EACH ROW BEGIN IF ((SELECT AVG(salary) FROM employees) <= 50000) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "The average salary of the employees is lower than 50000!"; END IF; END

CREATE TRIGGER chk_avg_ins AFTER UPDATE ON employees FOR EACH ROW BEGIN IF ((SELECT AVG(salary) FROM employees) <= 50000) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "The average salary of the employees is lower than 50000!"; END IF; END

CREATE TRIGGER chk_avg_ins AFTER DELETE ON employees FOR EACH ROW BEGIN IF ((SELECT AVG(salary) FROM employees) <= 50000) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "The average salary of the employees is lower than 50000!"; END IF; END

2. Write the following as Cursors on the corresponding Schema.

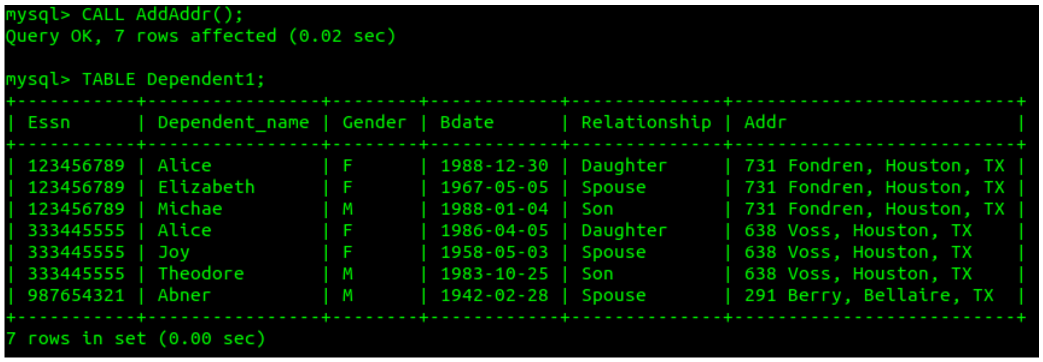
<u>Employee Schema</u>

q. Develop a stored procedure to insert a new attribute ‘address’ in DEPENDENT and update the same as that of the employee’s address.

<u>Query:</u>

CREATE PROCEDURE AddAddr() BEGIN ALTER TABLE Dependent1 ADD Addr VARCHAR(50) DEFAULT NULL; UPDATE Dependent1 AS D1 JOIN Employee AS E1 ON D1.Essn = E1.Ssn SET D1.Addr = E1.Addr; END

<u>Output:</u>

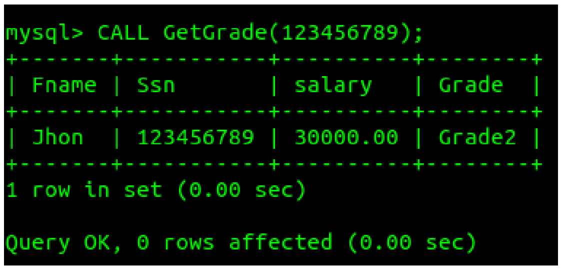

```
mysql> CALL AddAddr();
Query OK, 7 rows affected (0.02 sec)

mysql> TABLE Dependent1;
+-----------+----------------+--------+------------+--------------+-------------------------+
| Essn      | Dependent_name | Gender | Bdate      | Relationship | Addr                    |
+-----------+----------------+--------+------------+--------------+-------------------------+
| 123456789 | Alice          | F      | 1988-12-30 | Daughter     | 731 Fondren, Houston, TX |
| 123456789 | Elizabeth      | F      | 1967-05-05 | Spouse       | 731 Fondren, Houston, TX |
| 123456789 | Michae         | M      | 1988-01-04 | Son          | 731 Fondren, Houston, TX |
| 333445555 | Alice          | F      | 1986-04-05 | Daughter     | 638 Voss, Houston, TX   |
| 333445555 | Joy            | F      | 1958-05-03 | Spouse       | 638 Voss, Houston, TX   |
| 333445555 | Theodore       | M      | 1983-10-25 | Son          | 638 Voss, Houston, TX   |
| 987654321 | Abner          | M      | 1942-02-28 | Spouse       | 291 Berry, Bellaire, TX |
+-----------+----------------+--------+------------+--------------+-------------------------+
7 rows in set (0.00 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 exceptions with an error message when an invalid case occurs.

<u>Query:</u>

CREATE PROCEDURE GetGrade(IN emp_ssn CHAR(10)) BEGIN IF(EXISTS(SELECT * FROM Employee WHERE Ssn = emp_ssn)) THEN SELECT Fname,Ssn,salary,(CASE WHEN salary BETWEEN 1 AND 9999 THEN 'Grade3' WHEN salary BETWEEN 10000 AND 50000 THEN 'Grade2' WHEN salary >= 50000 THEN 'Grade1' ELSE 'Invalid salary range identified' END) AS Grade FROM Employee WHERE Ssn = emp_ssn; ELSE SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid SSN given'; END IF; END

Output:



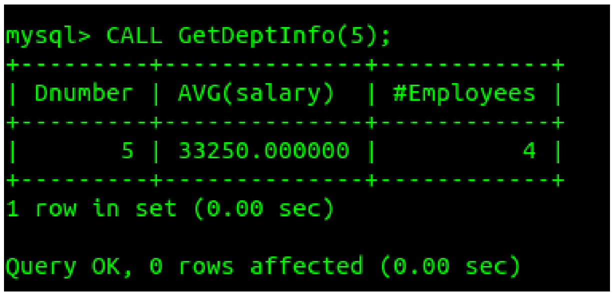
```
mysql> CALL GetGrade(123456789);
+-------+-----------+----------+--------+
| Fname | Ssn       | salary   | Grade  |
+-------+-----------+----------+--------+
| Jhon  | 123456789 | 30000.00 | Grade2 |
+-------+-----------+----------+--------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

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.

Query:

CREATE PROCEDURE GetDeptInfo(IN depno INT) BEGIN IF(EXISTS(SELECT * FROM Department WHERE Dnumber = depno)) THEN SELECT Dnumber,AVG(salary),COUNT(*) AS '#Employees' FROM Department JOIN Employee ON Dnumber = Dno WHERE Dnumber = depno GROUP BY Dnumber; ELSE SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid Depno given'; END IF; END

Output:



```
mysql> CALL GetDeptInfo(5);
+---------+--------------+------------+
| Dnumber | AVG(salary)  | #Employees |
+---------+--------------+------------+
|       5 | 33250.000000 |          4 |
+---------+--------------+------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

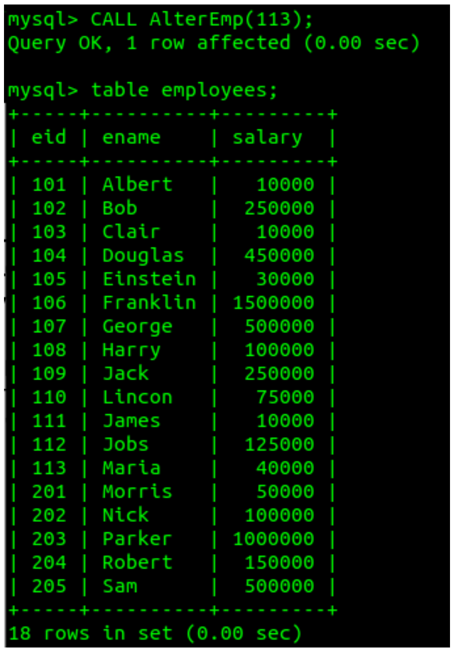
Flight Schema

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.

Query:

CREATE PROCEDURE AlterEmp(IN empid INT) BEGIN IF(EXISTS(SELECT * FROM employees WHERE eid = empid)) THEN UPDATE employees SET salary = 40000 WHERE eid = empid; ELSE SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid emp id given'; END IF; END

Output:



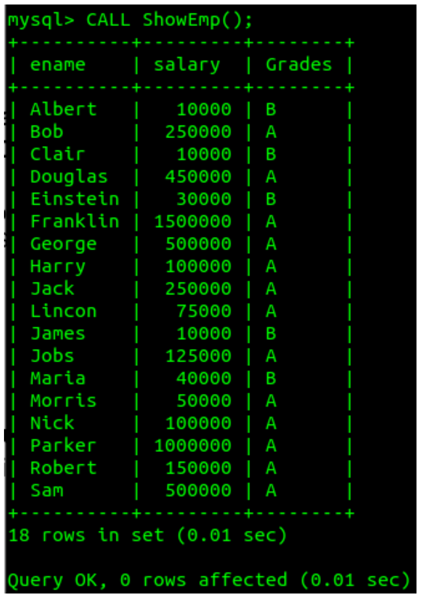
```
mysql> CALL AlterEmp(113);
Query OK, 1 row affected (0.00 sec)

mysql> table employees;
+-----+----------+---------+
| eid | ename    | salary  |
+-----+----------+---------+
| 101 | Albert   |   10000 |
| 102 | Bob      |  250000 |
| 103 | Clair    |   10000 |
| 104 | Douglas  |  450000 |
| 105 | Einstein |   30000 |
| 106 | Franklin | 1500000 |
| 107 | George   |  500000 |
| 108 | Harry    |  100000 |
| 109 | Jack     |  250000 |
| 110 | Lincon   |   75000 |
| 111 | James    |   10000 |
| 112 | Jobs     |  125000 |
| 113 | Maria    |   40000 |
| 201 | Morris   |   50000 |
| 202 | Nick     |  100000 |
| 203 | Parker   | 1000000 |
| 204 | Robert   |  150000 |
| 205 | Sam      |  500000 |
+-----+----------+---------+
18 rows in set (0.00 sec)
```

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’.

Query:

CREATE PROCEDURE ShowEmp() BEGIN SELECT ename,salary,(CASE WHEN salary >= 50000 THEN 'A' ELSE 'B' END) AS Grades FROM employees; END

Output:



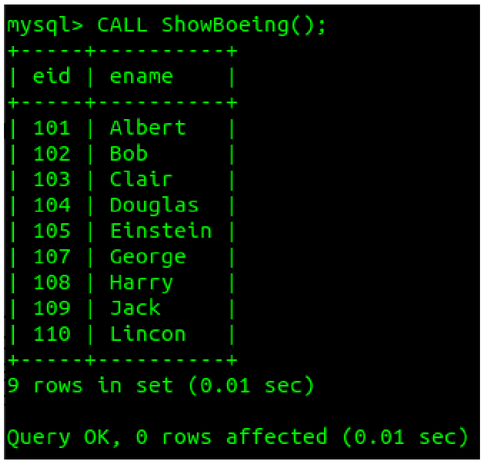
```
mysql> CALL ShowEmp();
+----------+---------+--------+
| ename    | salary  | Grades |
+----------+---------+--------+
| Albert   |   10000 | B      |
| Bob      |  250000 | A      |
| Clair    |   10000 | B      |
| Douglas  |  450000 | A      |
| Einstein |   30000 | B      |
| Franklin | 1500000 | A      |
| George   |  500000 | A      |
| Harry    |  100000 | A      |
| Jack     |  250000 | A      |
| Lincon   |   75000 | A      |
| James    |   10000 | B      |
| Jobs     |  125000 | A      |
| Maria    |   40000 | B      |
| Morris   |   50000 | A      |
| Nick     |  100000 | A      |
| Parker   | 1000000 | A      |
| Robert   |  150000 | A      |
| Sam      |  500000 | A      |
+----------+---------+--------+
18 rows in set (0.01 sec)

Query OK, 0 rows affected (0.01 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.

Query:

CREATE PROCEDURE ShowBoeing() BEGIN IF(EXISTS(SELECT * FROM employees NATURAL JOIN certified NATURAL JOIN aircraft WHERE aname = 'Boeing')) THEN SELECT DISTINCT eid,ename FROM employees NATURAL JOIN certified NATURAL JOIN aircraft WHERE aname = 'Boeing'; ELSE SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No pilots driving Boeing plane'; END IF; END

Output:



```
mysql> CALL ShowBoeing();
+-----+----------+
| eid | ename    |
+-----+----------+
| 101 | Albert   |
| 102 | Bob      |
| 103 | Clair    |
| 104 | Douglas  |
| 105 | Einstein |
| 107 | George   |
| 108 | Harry    |
| 109 | Jack     |
| 110 | Lincon   |
+-----+----------+
9 rows in set (0.01 sec)

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

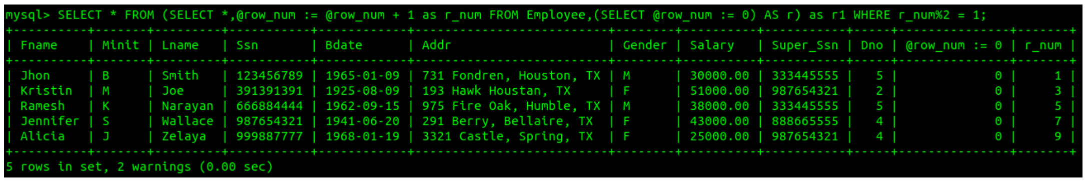
3. On the Company Relational Schema, execute the following queries.

a. Display all odd numbered alternate records from ‘Employee’ table.

Query:

SELECT * FROM (SELECT *,@row_num := @row_num + 1 as r_num FROM Employee,(SELECT @row_num := 0) AS r) as r1 WHERE r_num%2 = 1;

Output:



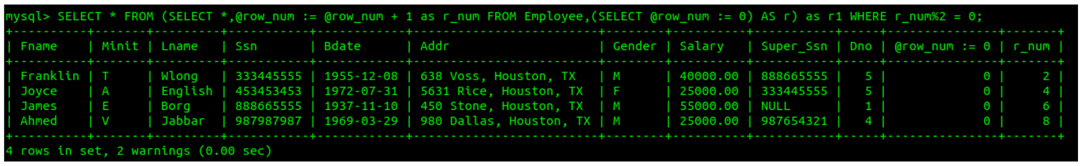
```
mysql> SELECT * FROM (SELECT *,@row_num := @row_num + 1 as r_num FROM Employee,(SELECT @row_num := 0) AS r) as r1 WHERE r_num%2 = 1;
+----------+-------+---------+-----------+------------+---------------------------+--------+----------+-----------+-----+-----------------+-------+
| Fname    | Minit | Lname   | Ssn       | Bdate      | Addr                      | Gender | Salary   | Super_Ssn | Dno | @row_num := 0 | r_num |
+----------+-------+---------+-----------+------------+---------------------------+--------+----------+-----------+-----+-----------------+-------+
| Jhon     | B     | Smith   | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M      | 30000.00 | 333445555 |   5 |               0 |     1 |
| Kristin  | M     | Joe     | 391391391 | 1925-08-09 | 193 Hawk Houstan, TX     | F      | 51000.00 | 987654321 |   2 |               0 |     3 |
| Ramesh   | K     | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M      | 38000.00 | 333445555 |   5 |               0 |     5 |
| Jennifer | S     | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX  | F      | 43000.00 | 888665555 |   4 |               0 |     7 |
| Alicia   | J     | Zelaya  | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX  | F      | 25000.00 | 987654321 |   4 |               0 |     9 |
+----------+-------+---------+-----------+------------+---------------------------+--------+----------+-----------+-----+-----------------+-------+
5 rows in set, 2 warnings (0.00 sec)
```

b. Display all even numbered alternate records from ‘Employee’ table.

Query:

SELECT * FROM (SELECT *,@row_num := @row_num + 1 as r_num FROM Employee,(SELECT @row_num := 0) AS r) as r1 WHERE r_num%2 = 0;

Output:



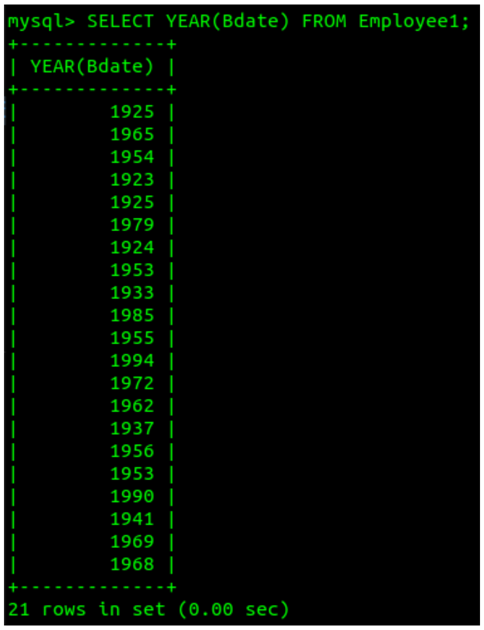
```
mysql> SELECT * FROM (SELECT *,@row_num := @row_num + 1 as r_num FROM Employee,(SELECT @row_num := 0) AS r) as r1 WHERE r_num%2 = 0;
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+---------------+-------+
| Fname    | Minit | Lname   | Ssn       | Bdate      | Addr                     | Gender | Salary   | Super_Ssn | Dno | @row_num := 0 | r_num |
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+---------------+-------+
| Franklin | T     | Wlong   | 333445555 | 1955-12-08 | 638 Voss, Houston, TX    | M      | 40000.00 | 888665555 |   5 |             0 |     2 |
| Joyce    | A     | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX   | F      | 25000.00 | 333445555 |   5 |             0 |     4 |
| James    | E     | Borg    | 888665555 | 1937-11-10 | 450 Stone, Houston, TX   | M      | 55000.00 | NULL      |   1 |             0 |     6 |
| Ahmed    | V     | Jabbar  | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX  | M      | 25000.00 | 987654321 |   4 |             0 |     8 |
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+---------------+-------+
4 rows in set, 2 warnings (0.00 sec)
```

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

Query:

SELECT YEAR(Bdate) FROM Employee1;

Output:



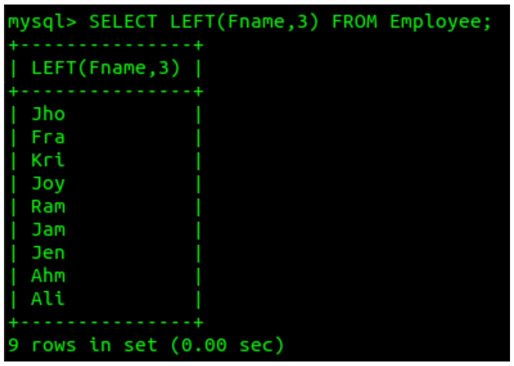
```
mysql> SELECT YEAR(Bdate) FROM Employee1;
+-------------+
| YEAR(Bdate) |
+-------------+
|        1925 |
|        1965 |
|        1954 |
|        1923 |
|        1925 |
|        1979 |
|        1924 |
|        1953 |
|        1933 |
|        1985 |
|        1955 |
|        1994 |
|        1972 |
|        1962 |
|        1937 |
|        1956 |
|        1953 |
|        1990 |
|        1941 |
|        1969 |
|        1968 |
+-------------+
21 rows in set (0.00 sec)
```

d. Select first 3 characters of first name.

Query:

SELECT LEFT(Fname,3) FROM Employee;

Output:



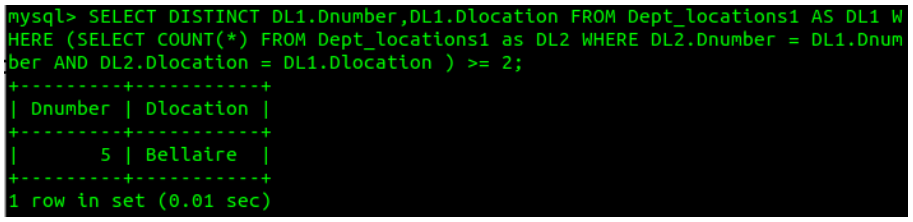
```
mysql> SELECT LEFT(Fname,3) FROM Employee;
+---------------+
| LEFT(Fname,3) |
+---------------+
| Jho           |
| Fra           |
| Kri           |
| Joy           |
| Ram           |
| Jam           |
| Jen           |
| Ahm           |
| Ali           |
+---------------+
9 rows in set (0.00 sec)
```

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

Query:

SELECT DISTINCT DL1.Dnumber,DL1.Dlocation FROM Dept_locations1 AS DL1 WHERE (SELECT COUNT(*) FROM Dept_locations1 as DL2 WHERE DL2.Dnumber = DL1.Dnumber AND DL2.Dlocation = DL1.Dlocation) >= 2;

Output:



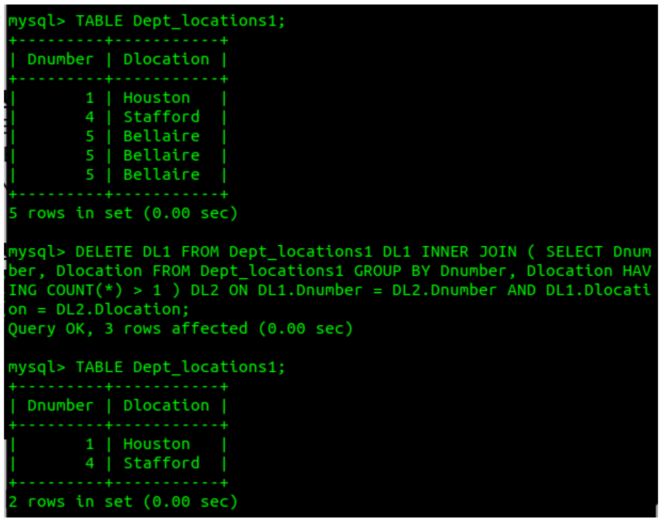
```
mysql> SELECT DISTINCT DL1.Dnumber,DL1.Dlocation FROM Dept_locations1 AS DL1 W
HERE (SELECT COUNT(*) FROM Dept_locations1 as DL2 WHERE DL2.Dnumber = DL1.Dnum
ber AND DL2.Dlocation = DL1.Dlocation ) >= 2;
+---------+-----------+
| Dnumber | Dlocation |
+---------+-----------+
|       5 | Bellaire  |
+---------+-----------+
1 row in set (0.01 sec)
```

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

Query:

DELETE DL1 FROM Dept_locations1 DL1 INNER JOIN (SELECT Dnumber, Dlocation FROM Dept_locations1 GROUP BY Dnumber, Dlocation HAVING COUNT(*) > 1) DL2 ON DL1.Dnumber = DL2.Dnumber AND DL1.Dlocation = DL2.Dlocation;

Output:



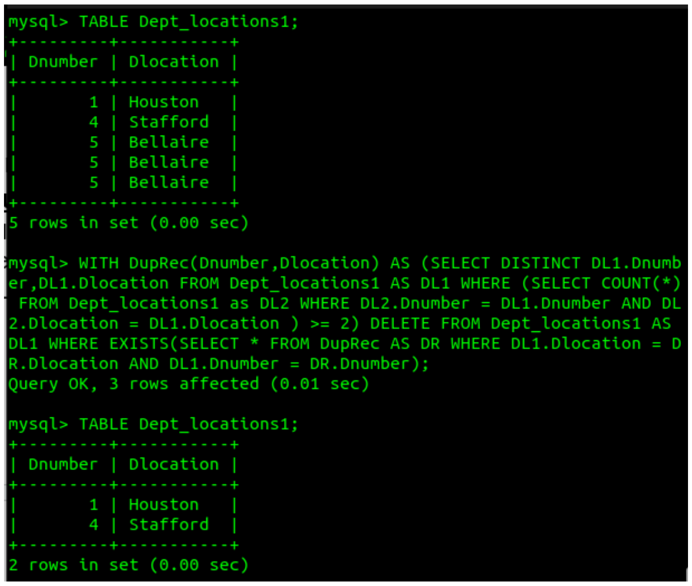
```
mysql> TABLE Dept_locations1;
+---------+-----------+
| Dnumber | Dlocation |
+---------+-----------+
|       1 | Houston   |
|       4 | Stafford  |
|       5 | Bellaire  |
|       5 | Bellaire  |
|       5 | Bellaire  |
+---------+-----------+
5 rows in set (0.00 sec)

mysql> DELETE DL1 FROM Dept_locations1 DL1 INNER JOIN ( SELECT Dnum
ber, Dlocation FROM Dept_locations1 GROUP BY Dnumber, Dlocation HAV
ING COUNT(*) > 1 ) DL2 ON DL1.Dnumber = DL2.Dnumber AND DL1.Dlocati
on = DL2.Dlocation;
Query OK, 3 rows affected (0.00 sec)

mysql> TABLE Dept_locations1;
+---------+-----------+
| Dnumber | Dlocation |
+---------+-----------+
|       1 | Houston   |
|       4 | Stafford  |
+---------+-----------+
2 rows in set (0.00 sec)
```

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

Query:

WITH DupRec(Dnumber,Dlocation) AS (SELECT DISTINCT DL1.Dnumber,DL1.Dlocation FROM Dept_locations1 AS DL1 WHERE (SELECT COUNT(*) FROM Dept_locations1 as DL2 WHERE DL2.Dnumber = DL1.Dnumber AND DL2.Dlocation = DL1.Dlocation) >= 2) DELETE FROM Dept_locations1 AS DL1 WHERE EXISTS(SELECT * FROM DupRec AS DR WHERE DL1.Dlocation = DR.Dlocation AND DL1.Dnumber = DR.Dnumber);

Output:



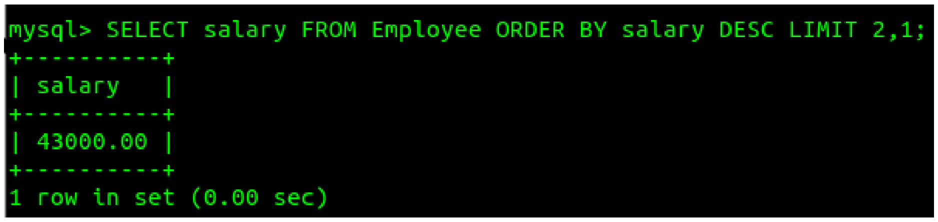
```
mysql> TABLE Dept_locations1;
+---------+-----------+
| Dnumber | Dlocation |
+---------+-----------+
|       1 | Houston   |
|       4 | Stafford  |
|       5 | Bellaire  |
|       5 | Bellaire  |
|       5 | Bellaire  |
+---------+-----------+
5 rows in set (0.00 sec)

mysql> WITH DupRec(Dnumber,Dlocation) AS (SELECT DISTINCT DL1.Dnumb
er,DL1.Dlocation FROM Dept_locations1 AS DL1 WHERE (SELECT COUNT(*)
 FROM Dept_locations1 as DL2 WHERE DL2.Dnumber = DL1.Dnumber AND DL
2.Dlocation = DL1.Dlocation ) >= 2) DELETE FROM Dept_locations1 AS
DL1 WHERE EXISTS(SELECT * FROM DupRec AS DR WHERE DL1.Dlocation = D
R.Dlocation AND DL1.Dnumber = DR.Dnumber);
Query OK, 3 rows affected (0.01 sec)

mysql> TABLE Dept_locations1;
+---------+-----------+
| Dnumber | Dlocation |
+---------+-----------+
|       1 | Houston   |
|       4 | Stafford  |
+---------+-----------+
2 rows in set (0.00 sec)
```

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

Query:

SELECT salary FROM Employee ORDER BY salary DESC LIMIT 2,1;

Output:



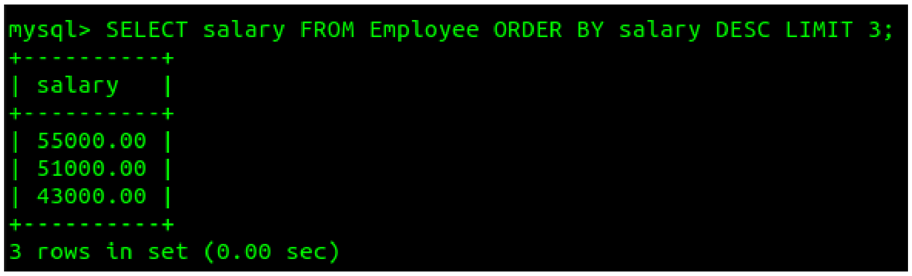
```
mysql> SELECT salary FROM Employee ORDER BY salary DESC LIMIT 2,1;
+----------+
| salary   |
+----------+
| 43000.00 |
+----------+
1 row in set (0.00 sec)
```

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

Query:

SELECT salary FROM Employee ORDER BY salary DESC LIMIT 3;

Output:



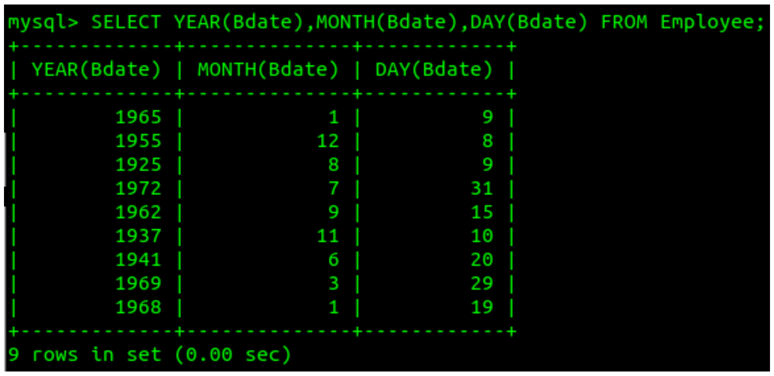
```
mysql> SELECT salary FROM Employee ORDER BY salary DESC LIMIT 3;
+----------+
| salary   |
+----------+
| 55000.00 |
| 51000.00 |
| 43000.00 |
+----------+
3 rows in set (0.00 sec)
```

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

Query:

SELECT YEAR(Bdate),MONTH(Bdate),DAY(Bdate) FROM Employee;

Output:



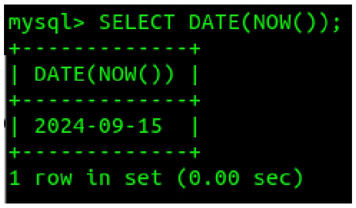
```
mysql> SELECT YEAR(Bdate),MONTH(Bdate),DAY(Bdate) FROM Employee;
+-------------+--------------+------------+
| YEAR(Bdate) | MONTH(Bdate) | DAY(Bdate) |
+-------------+--------------+------------+
|        1965 |            1 |          9 |
|        1955 |           12 |          8 |
|        1925 |            8 |          9 |
|        1972 |            7 |         31 |
|        1962 |            9 |         15 |
|        1937 |           11 |         10 |
|        1941 |            6 |         20 |
|        1969 |            3 |         29 |
|        1968 |            1 |         19 |
+-------------+--------------+------------+
9 rows in set (0.00 sec)
```

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

Query:

SELECT DATE(NOW());

Output:



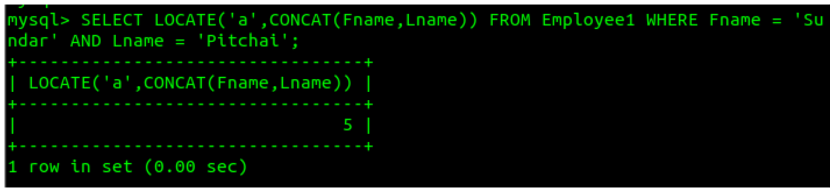
```
mysql> SELECT DATE(NOW());
+-------------+
| DATE(NOW()) |
+-------------+
| 2024-09-15  |
+-------------+
1 row in set (0.00 sec)
```

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

Query:

SELECT LOCATE('a',CONCAT(Fname,Lname)) FROM Employee1 WHERE Fname = 'Sundar' AND Lname = 'Pitchai';

Output:



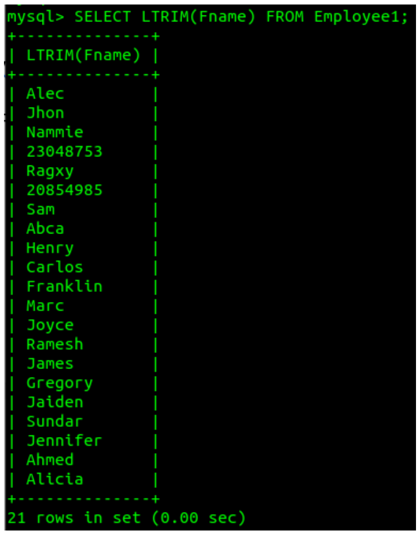
```
mysql> SELECT LOCATE('a',CONCAT(Fname,Lname)) FROM Employee1 WHERE Fname = 'Su
ndar' AND Lname = 'Pitchai';
+---------------------------------+
| LOCATE('a',CONCAT(Fname,Lname)) |
+---------------------------------+
|                               5 |
+---------------------------------+
1 row in set (0.00 sec)
```

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

Query:

SELECT LTRIM(Fname) FROM Employee1;

Output:



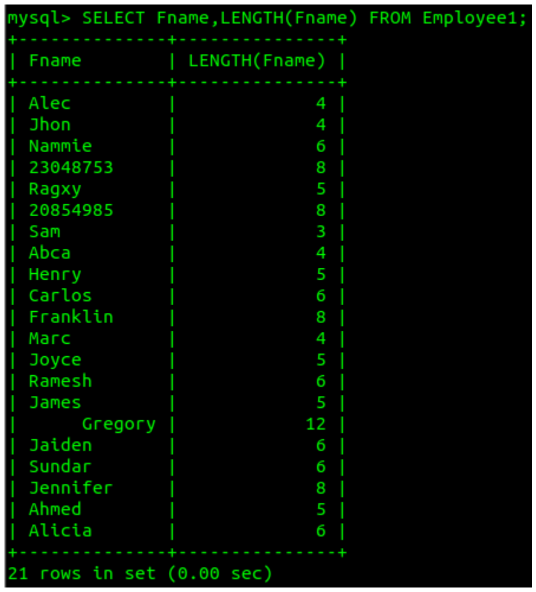
```
mysql> SELECT LTRIM(Fname) FROM Employee1;
+--------------+
| LTRIM(Fname) |
+--------------+
| Alec         |
| Jhon         |
| Nammie       |
| 23048753     |
| Ragxy        |
| 20854985     |
| Sam          |
| Abca         |
| Henry        |
| Carlos       |
| Franklin     |
| Marc         |
| Joyce        |
| Ramesh       |
| James        |
| Gregory      |
| Jaiden       |
| Sundar       |
| Jennifer     |
| Ahmed        |
| Alicia       |
+--------------+
21 rows in set (0.00 sec)
```

n. Get length of fname from employee table.

Query:

SELECT Fname,LENGTH(Fname) FROM Employee1;

Output:



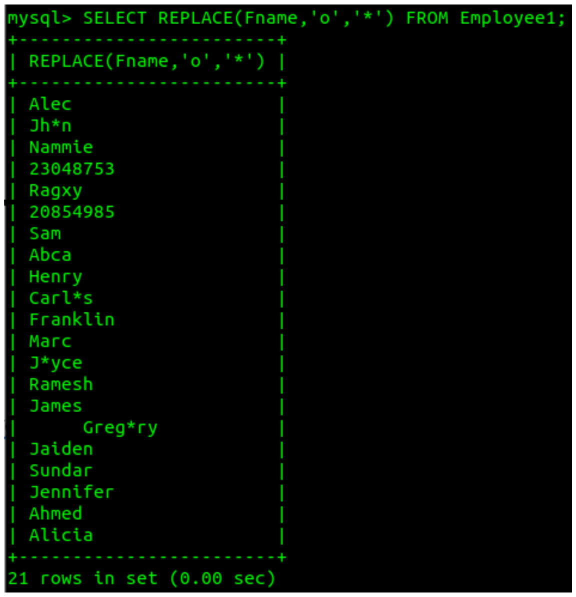
```
mysql> SELECT Fname,LENGTH(Fname) FROM Employee1;
+--------------+---------------+
| Fname        | LENGTH(Fname) |
+--------------+---------------+
| Alec         |             4 |
| Jhon         |             4 |
| Nammie       |             6 |
| 23048753     |             8 |
| Ragxy        |             5 |
| 20854985     |             8 |
| Sam          |             3 |
| Abca         |             4 |
| Henry        |             5 |
| Carlos       |             6 |
| Franklin     |             8 |
| Marc         |             4 |
| Joyce        |             5 |
| Ramesh       |             6 |
| James        |             5 |
|      Gregory |            12 |
| Jaiden       |             6 |
| Sundar       |             6 |
| Jennifer     |             8 |
| Ahmed        |             5 |
| Alicia       |             6 |
+--------------+---------------+
21 rows in set (0.00 sec)
```

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

Query:

SELECT REPLACE(Fname,'o','*') FROM Employee1;

Output:



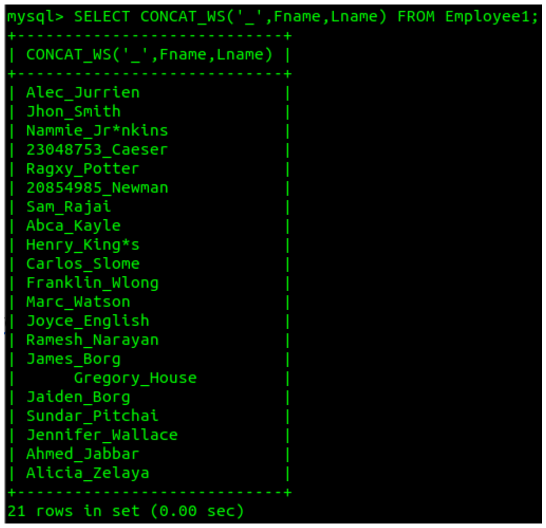
```
mysql> SELECT REPLACE(Fname,'o','*') FROM Employee1;
+------------------------+
| REPLACE(Fname,'o','*') |
+------------------------+
| Alec                   |
| Jh*n                   |
| Nammie                 |
| 23048753               |
| Ragxy                  |
| 20854985               |
| Sam                    |
| Abca                   |
| Henry                  |
| Carl*s                 |
| Franklin               |
| Marc                   |
| J*yce                  |
| Ramesh                 |
| James                  |
|       Greg*ry          |
| Jaiden                 |
| Sundar                 |
| Jennifer               |
| Ahmed                  |
| Alicia                 |
+------------------------+
21 rows in set (0.00 sec)
```

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

<u>Query:</u>

SELECT CONCAT_WS('_',Fname,Lname) FROM Employee1;

<u>Output:</u>



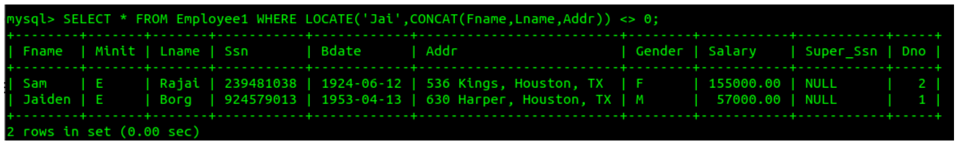
```
mysql> SELECT CONCAT_WS('_',Fname,Lname) FROM Employee1;
+----------------------------+
| CONCAT_WS('_',Fname,Lname) |
+----------------------------+
| Alec_Jurrien               |
| Jhon_Smith                 |
| Nammie_Jr*nkins            |
| 23048753_Caeser            |
| Ragxy_Potter               |
| 20854985_Newman            |
| Sam_Rajai                  |
| Abca_Kayle                 |
| Henry_King*s               |
| Carlos_Slome               |
| Franklin_Wlong             |
| Marc_Watson                |
| Joyce_English              |
| Ramesh_Narayan             |
| James_Borg                 |
|      Gregory_House         |
| Jaiden_Borg                |
| Sundar_Pitchai             |
| Jennifer_Wallace           |
| Ahmed_Jabbar               |
| Alicia_Zelaya              |
+----------------------------+
21 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.

<u>Query:</u>

SELECT * FROM Employee1 WHERE LOCATE('Jai',CONCAT(Fname,Lname,Addr)) <> 0;

<u>Output:</u>



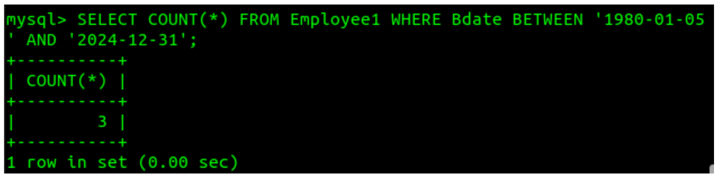
```
mysql> SELECT * FROM Employee1 WHERE LOCATE('Jai',CONCAT(Fname,Lname,Addr)) <> 0;
+--------+-------+-------+-----------+------------+------------------------+--------+-----------+-----------+-----+
| Fname  | Minit | Lname | Ssn       | Bdate      | Addr                   | Gender | Salary    | Super_Ssn | Dno |
+--------+-------+-------+-----------+------------+------------------------+--------+-----------+-----------+-----+
| Sam    | E     | Rajai | 239481038 | 1924-06-12 | 536 Kings, Houston, TX  | F      | 155000.00 | NULL      |   2 |
| Jaiden | E     | Borg  | 924579013 | 1953-04-13 | 630 Harper, Houston, TX | M      |  57000.00 | NULL      |   1 |
+--------+-------+-------+-----------+------------+------------------------+--------+-----------+-----------+-----+
2 rows in set (0.00 sec)
```

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

Query:

SELECT COUNT(*) FROM Employee1 WHERE Bdate BETWEEN '1980-01-05' AND '2024-12-31';

Output:



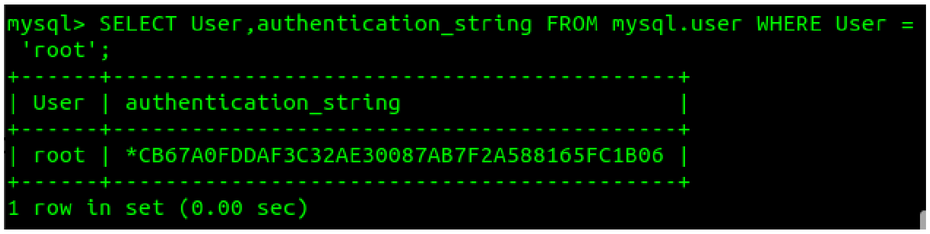
```
mysql> SELECT COUNT(*) FROM Employee1 WHERE Bdate BETWEEN '1980-01-05
' AND '2024-12-31';
+----------+
| COUNT(*) |
+----------+
|        3 |
+----------+
1 row in set (0.00 sec)
```

s. Retrieve the mysql username and password.

Query:

SELECT User,authentication_string FROM mysql.user WHERE User = 'root';

Output:



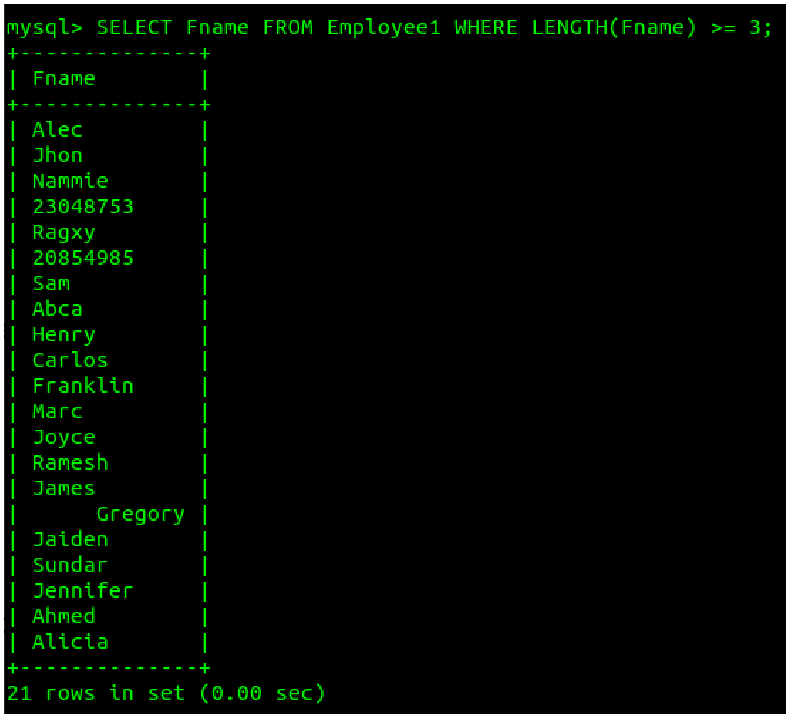
```
mysql> SELECT User,authentication_string FROM mysql.user WHERE User =
 'root';
+------+-------------------------------------------+
| User | authentication_string                     |
+------+-------------------------------------------+
| root | *CB67A0FDDAF3C32AE30087AB7F2A588165FC1B06 |
+------+-------------------------------------------+
1 row in set (0.00 sec)
```

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

Query:

SELECT Fname FROM Employee1 WHERE LENGTH(Fname) >= 3;

Output:



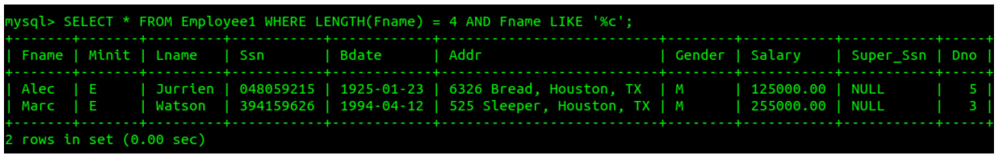
```
mysql> SELECT Fname FROM Employee1 WHERE LENGTH(Fname) >= 3;
+--------------+
| Fname        |
+--------------+
| Alec         |
| Jhon         |
| Nammie       |
| 23048753     |
| Ragxy        |
| 20854985     |
| Sam          |
| Abca         |
| Henry        |
| Carlos       |
| Franklin     |
| Marc         |
| Joyce        |
| Ramesh       |
| James        |
|      Gregory |
| Jaiden       |
| Sundar       |
| Jennifer     |
| Ahmed        |
| Alicia       |
+--------------+
21 rows in set (0.00 sec)
```

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

<u>Query:</u>

SELECT * FROM Employee1 WHERE LENGTH(Fname) = 4 AND Fname LIKE '%c';

<u>Output:</u>



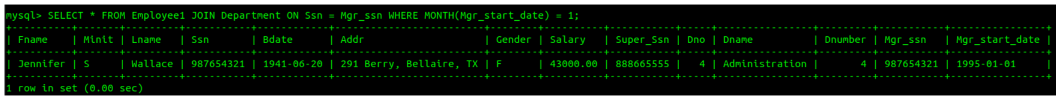
```
mysql> SELECT * FROM Employee1 WHERE LENGTH(Fname) = 4 AND Fname LIKE '%c';
+-------+-------+---------+-----------+------------+--------------------------+--------+-----------+-----------+-----+
| Fname | Minit | Lname   | Ssn       | Bdate      | Addr                     | Gender | Salary    | Super_Ssn | Dno |
+-------+-------+---------+-----------+------------+--------------------------+--------+-----------+-----------+-----+
| Alec  | E     | Jurrien | 048059215 | 1925-01-23 | 6326 Bread, Houston, TX  | M      | 125000.00 | NULL      |   5 |
| Marc  | E     | Watson  | 394159626 | 1994-04-12 | 525 Sleeper, Houston, TX | M      | 255000.00 | NULL      |   3 |
+-------+-------+---------+-----------+------------+--------------------------+--------+-----------+-----------+-----+
2 rows in set (0.00 sec)
```

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

<u>Query:</u>

SELECT * FROM Employee1 JOIN Department ON Ssn = Mgr_ssn WHERE MONTH(Mgr_start_date) = 1;

<u>Output:</u>



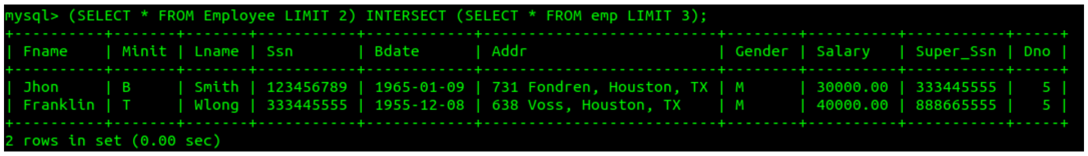
```
mysql> SELECT * FROM Employee1 JOIN Department ON Ssn = Mgr_ssn WHERE MONTH(Mgr_start_date) = 1;
+----------+-------+---------+-----------+------------+-------------------------+--------+-----------+-----------+-----+----------------+---------+-----------+----------------+
| Fname    | Minit | Lname   | Ssn       | Bdate      | Addr                    | Gender | Salary    | Super_Ssn | Dno | Dname          | Dnumber | Mgr_ssn   | Mgr_start_date |
+----------+-------+---------+-----------+------------+-------------------------+--------+-----------+-----------+-----+----------------+---------+-----------+----------------+
| Jennifer | S     | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX | F      | 43000.00  | 888665555 |   4 | Administration |       4 | 987654321 | 1995-01-01     |
+----------+-------+---------+-----------+------------+-------------------------+--------+-----------+-----------+-----+----------------+---------+-----------+----------------+
1 row in set (0.00 sec)
```

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

<u>Query:</u>

(SELECT * FROM Employee LIMIT 2) INTERSECT (SELECT * FROM emp LIMIT 3);

<u>Output:</u>



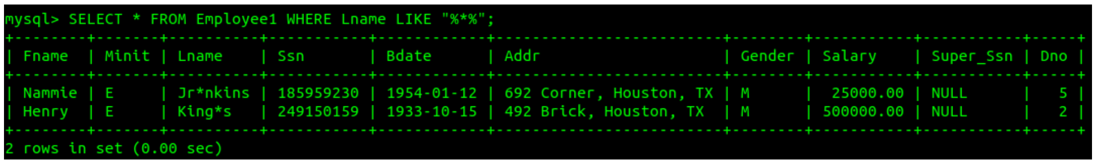
```
mysql> (SELECT * FROM Employee LIMIT 2) INTERSECT (SELECT * FROM emp LIMIT 3);
+----------+-------+-------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| Fname    | Minit | Lname | Ssn       | Bdate      | Addr                     | Gender | Salary   | Super_Ssn | Dno |
+----------+-------+-------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| Jhon     | B     | Smith | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M      | 30000.00 | 333445555 |   5 |
| Franklin | T     | Wlong | 333445555 | 1955-12-08 | 638 Voss, Houston, TX    | M      | 40000.00 | 888665555 |   5 |
+----------+-------+-------+-----------+------------+--------------------------+--------+----------+-----------+-----+
2 rows in set (0.00 sec)
```

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

<u>Query:</u>

SELECT * FROM Employee1 WHERE Lname LIKE "%*%";

<u>Output:</u>



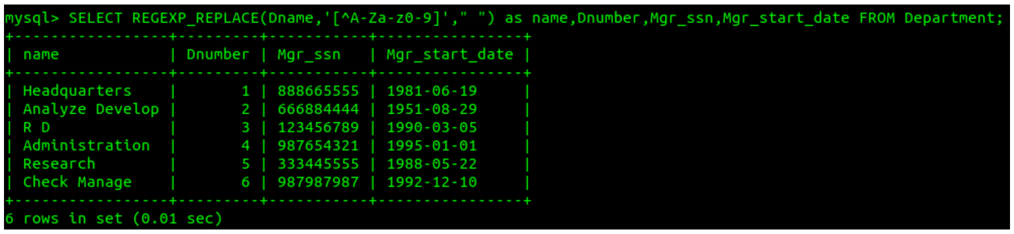
```
mysql> SELECT * FROM Employee1 WHERE Lname LIKE "%*%";
+--------+-------+----------+-----------+------------+-------------------------+--------+-----------+-----------+-----+
| Fname  | Minit | Lname    | Ssn       | Bdate      | Addr                    | Gender | Salary    | Super_Ssn | Dno |
+--------+-------+----------+-----------+------------+-------------------------+--------+-----------+-----------+-----+
| Nammie | E     | Jr*nkins | 185959230 | 1954-01-12 | 692 Corner, Houston, TX | M      |  25000.00 | NULL      |   5 |
| Henry  | E     | King*s   | 249150159 | 1933-10-15 | 492 Brick, Houston, TX  | M      | 500000.00 | NULL      |   2 |
+--------+-------+----------+-----------+------------+-------------------------+--------+-----------+-----------+-----+
2 rows in set (0.00 sec)
```

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

<u>Query:</u>

SELECT REGEXP_REPLACE(Dname,'[^A-Za-z0-9]'," ") as name, Dnumber, Mgr_ssn, Mgr_start_date FROM Department;

<u>Output:</u>



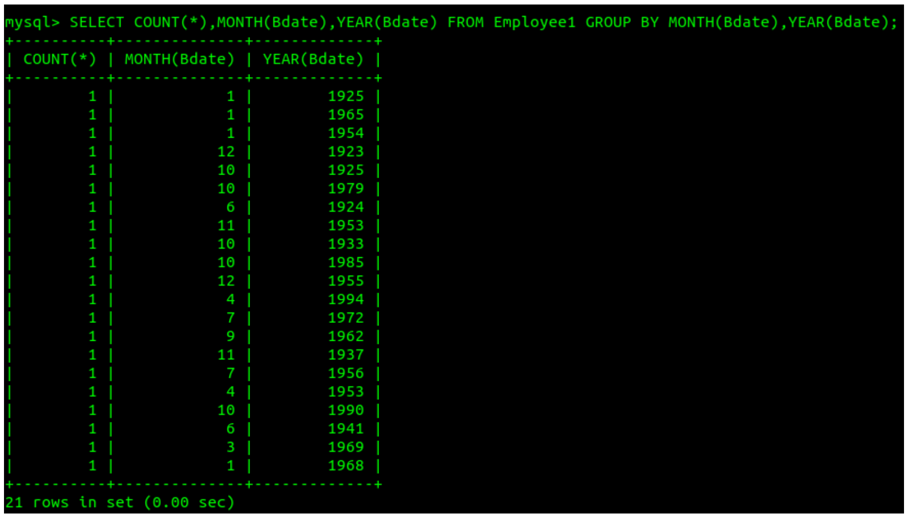
```
mysql> SELECT REGEXP_REPLACE(Dname,'[^A-Za-z0-9]'," ") as name,Dnumber,Mgr_ssn,Mgr_start_date FROM Department;
+-----------------+---------+-----------+----------------+
| name            | Dnumber | Mgr_ssn   | Mgr_start_date |
+-----------------+---------+-----------+----------------+
| Headquarters    |       1 | 888665555 | 1981-06-19     |
| Analyze Develop |       2 | 666884444 | 1951-08-29     |
| R D             |       3 | 123456789 | 1990-03-05     |
| Administration  |       4 | 987654321 | 1995-01-01     |
| Research        |       5 | 333445555 | 1988-05-22     |
| Check Manage    |       6 | 987987987 | 1992-12-10     |
+-----------------+---------+-----------+----------------+
6 rows in set (0.01 sec)
```

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

<u>Query:</u>

SELECT COUNT(*),MONTH(Bdate),YEAR(Bdate) FROM Employee1 GROUP BY MONTH(Bdate),YEAR(Bdate);

Output:



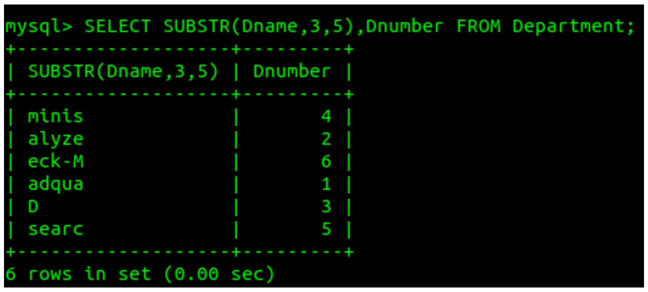
```
mysql> SELECT COUNT(*),MONTH(Bdate),YEAR(Bdate) FROM Employee1 GROUP BY MONTH(Bdate),YEAR(Bdate);
+----------+--------------+-------------+
| COUNT(*) | MONTH(Bdate) | YEAR(Bdate) |
+----------+--------------+-------------+
|        1 |            1 |        1925 |
|        1 |            1 |        1965 |
|        1 |            1 |        1954 |
|        1 |           12 |        1923 |
|        1 |           10 |        1925 |
|        1 |           10 |        1979 |
|        1 |            6 |        1924 |
|        1 |           11 |        1953 |
|        1 |           10 |        1933 |
|        1 |           10 |        1985 |
|        1 |           12 |        1955 |
|        1 |            4 |        1994 |
|        1 |            7 |        1972 |
|        1 |            9 |        1962 |
|        1 |           11 |        1937 |
|        1 |            7 |        1956 |
|        1 |            4 |        1953 |
|        1 |           10 |        1990 |
|        1 |            6 |        1941 |
|        1 |            3 |        1969 |
|        1 |            1 |        1968 |
+----------+--------------+-------------+
21 rows in set (0.00 sec)
```

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

Query:

SELECT SUBSTR(Dname,3,5),Dnumber FROM Department;

Output:



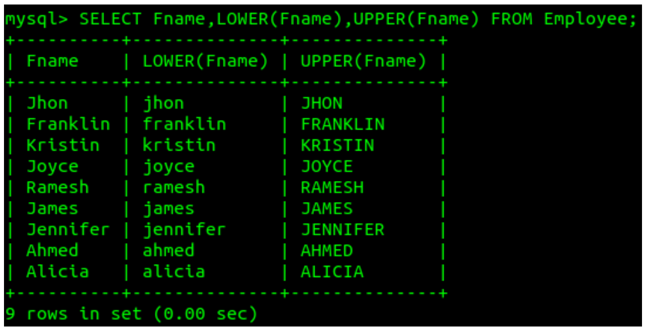
```
mysql> SELECT SUBSTR(Dname,3,5),Dnumber FROM Department;
+-------------------+---------+
| SUBSTR(Dname,3,5) | Dnumber |
+-------------------+---------+
| minis             |       4 |
| alyze             |       2 |
| eck-M             |       6 |
| adqua             |       1 |
| D                 |       3 |
| searc             |       5 |
+-------------------+---------+
6 rows in set (0.00 sec)
```

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

Query:

SELECT Fname,LOWER(Fname),UPPER(Fname) FROM Employee;

Output:



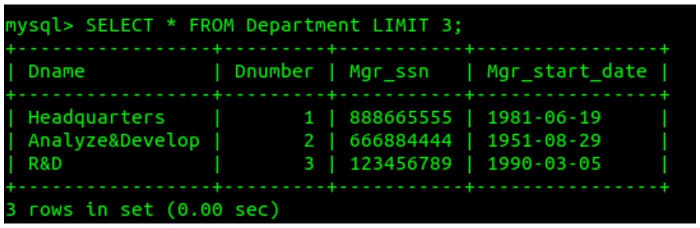
```
mysql> SELECT Fname,LOWER(Fname),UPPER(Fname) FROM Employee;
+----------+--------------+--------------+
| Fname    | LOWER(Fname) | UPPER(Fname) |
+----------+--------------+--------------+
| Jhon     | jhon         | JHON         |
| Franklin | franklin     | FRANKLIN     |
| Kristin  | kristin      | KRISTIN      |
| Joyce    | joyce        | JOYCE        |
| Ramesh   | ramesh       | RAMESH       |
| James    | james        | JAMES        |
| Jennifer | jennifer     | JENNIFER     |
| Ahmed    | ahmed        | AHMED        |
| Alicia   | alicia       | ALICIA       |
+----------+--------------+--------------+
9 rows in set (0.00 sec)
```

cc. Select FIRST n records from a department table.

Query:

SELECT * FROM Department LIMIT 3;

Output:



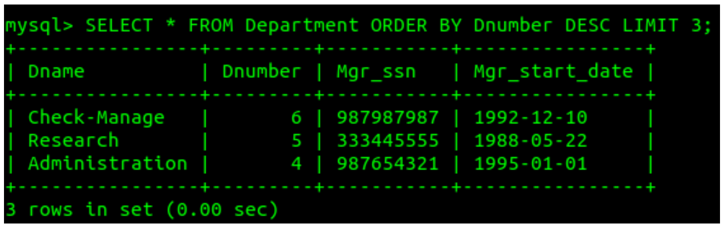
```
mysql> SELECT * FROM Department LIMIT 3;
+-----------------+---------+-----------+----------------+
| Dname           | Dnumber | Mgr_ssn   | Mgr_start_date |
+-----------------+---------+-----------+----------------+
| Headquarters    |       1 | 888665555 | 1981-06-19     |
| Analyze&Develop |       2 | 666884444 | 1951-08-29     |
| R&D             |       3 | 123456789 | 1990-03-05     |
+-----------------+---------+-----------+----------------+
3 rows in set (0.00 sec)
```

dd. Select LAST n records from a department table.

Query:

SELECT * FROM Department ORDER BY Dnumber DESC LIMIT 3;

Output:



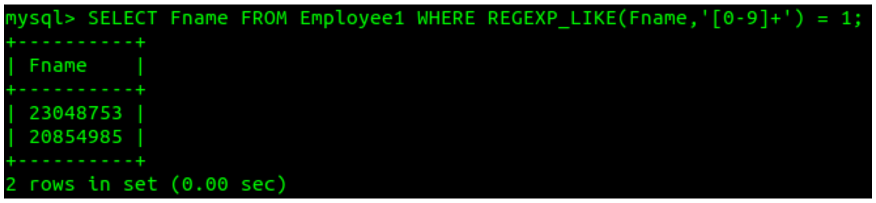
```
mysql> SELECT * FROM Department ORDER BY Dnumber DESC LIMIT 3;
+----------------+---------+-----------+----------------+
| Dname          | Dnumber | Mgr_ssn   | Mgr_start_date |
+----------------+---------+-----------+----------------+
| Check-Manage   |       6 | 987987987 | 1992-12-10     |
| Research       |       5 | 333445555 | 1988-05-22     |
| Administration |       4 | 987654321 | 1995-01-01     |
+----------------+---------+-----------+----------------+
3 rows in set (0.00 sec)
```

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

Query:

SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'[0-9]+') = 1;

Output:



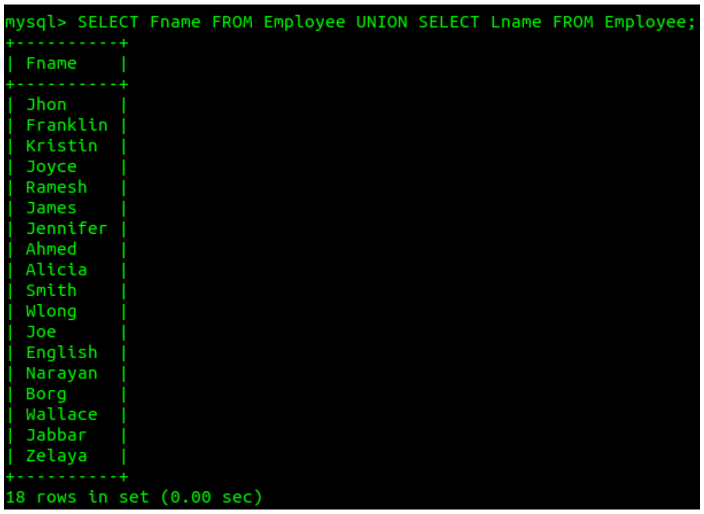
```
mysql> SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'[0-9]+') = 1;
+----------+
| Fname    |
+----------+
| 23048753 |
| 20854985 |
+----------+
2 rows in set (0.00 sec)
```

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

<u>Query:</u>

SELECT Fname FROM Employee UNION SELECT Lname FROM Employee;

<u>Output:</u>



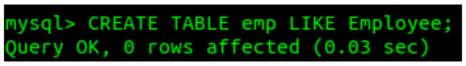
```
mysql> SELECT Fname FROM Employee UNION SELECT Lname FROM Employee;
+----------+
| Fname    |
+----------+
| Jhon     |
| Franklin |
| Kristin  |
| Joyce    |
| Ramesh   |
| James    |
| Jennifer |
| Ahmed    |
| Alicia   |
| Smith    |
| Wlong    |
| Joe      |
| English  |
| Narayan  |
| Borg     |
| Wallace  |
| Jabbar   |
| Zelaya   |
+----------+
18 rows in set (0.00 sec)
```

gg. Create an empty table emptem with the same structure as emp.

<u>Query:</u>

CREATE TABLE emp LIKE Employee;

<u>Output:</u>



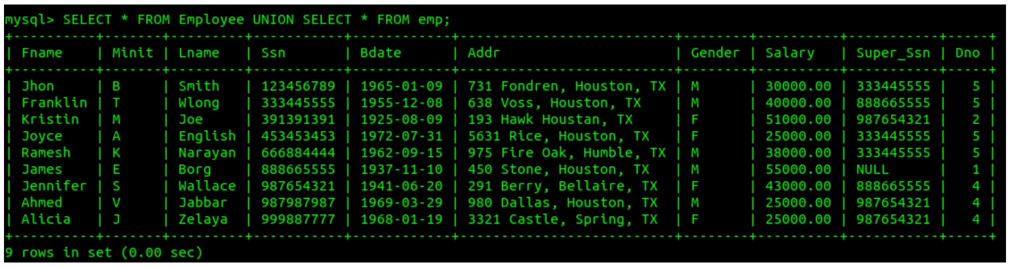
```
mysql> CREATE TABLE emp LIKE Employee;
Query OK, 0 rows affected (0.03 sec)
```

hh. If there are two tables emp1 and emp2, and both have common records, fetch all the records, but common records only once.

<u>Query:</u>

SELECT * FROM Employee UNION SELECT * FROM emp;

<u>Output:</u>



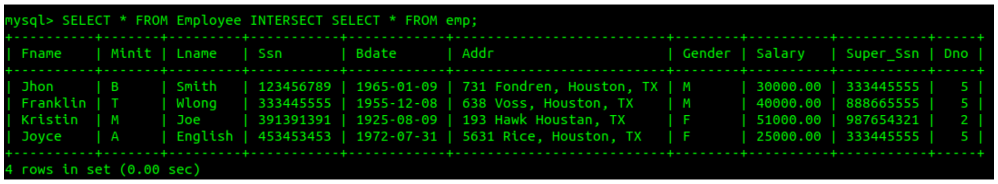
```
mysql> SELECT * FROM Employee UNION SELECT * FROM emp;
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| Fname    | Minit | Lname   | Ssn       | Bdate      | Addr                     | Gender | Salary   | Super_Ssn | Dno |
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| Jhon     | B     | Smith   | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M      | 30000.00 | 333445555 |   5 |
| Franklin | T     | Wlong   | 333445555 | 1955-12-08 | 638 Voss, Houston, TX    | M      | 40000.00 | 888665555 |   5 |
| Kristin  | M     | Joe     | 391391391 | 1925-08-09 | 193 Hawk Houstan, TX     | F      | 51000.00 | 987654321 |   2 |
| Joyce    | A     | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX   | F      | 25000.00 | 333445555 |   5 |
| Ramesh   | K     | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M      | 38000.00 | 333445555 |   5 |
| James    | E     | Borg    | 888665555 | 1937-11-10 | 450 Stone, Houston, TX   | M      | 55000.00 | NULL      |   1 |
| Jennifer | S     | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX  | F      | 43000.00 | 888665555 |   4 |
| Ahmed    | V     | Jabbar  | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX  | M      | 25000.00 | 987654321 |   4 |
| Alicia   | J     | Zelaya  | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX  | F      | 25000.00 | 987654321 |   4 |
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
9 rows in set (0.00 sec)
```

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

<u>Query:</u>

SELECT * FROM Employee INTERSECT SELECT * FROM emp;

<u>Output:</u>



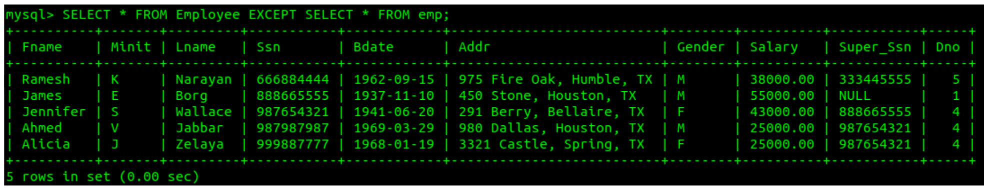
```
mysql> SELECT * FROM Employee INTERSECT SELECT * FROM emp;
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| Fname    | Minit | Lname   | Ssn       | Bdate      | Addr                     | Gender | Salary   | Super_Ssn | Dno |
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| Jhon     | B     | Smith   | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M      | 30000.00 | 333445555 |   5 |
| Franklin | T     | Wlong   | 333445555 | 1955-12-08 | 638 Voss, Houston, TX    | M      | 40000.00 | 888665555 |   5 |
| Kristin  | M     | Joe     | 391391391 | 1925-08-09 | 193 Hawk Houstan, TX     | F      | 51000.00 | 987654321 |   2 |
| Joyce    | A     | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX   | F      | 25000.00 | 333445555 |   5 |
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
4 rows in set (0.00 sec)
```

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

<u>Query:</u>

SELECT * FROM Employee EXCEPT SELECT * FROM emp;

<u>Output:</u>



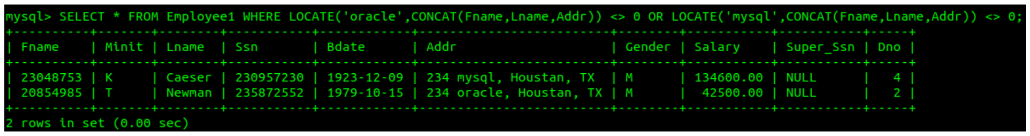
```
mysql> SELECT * FROM Employee EXCEPT SELECT * FROM emp;
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| Fname    | Minit | Lname   | Ssn       | Bdate      | Addr                     | Gender | Salary   | Super_Ssn | Dno |
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| Ramesh   | K     | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M      | 38000.00 | 333445555 |   5 |
| James    | E     | Borg    | 888665555 | 1937-11-10 | 450 Stone, Houston, TX   | M      | 55000.00 | NULL      |   1 |
| Jennifer | S     | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX  | F      | 43000.00 | 888665555 |   4 |
| Ahmed    | V     | Jabbar  | 987987987 | 1969-03-29 | 980 Dallas, Houston, TX  | M      | 25000.00 | 987654321 |   4 |
| Alicia   | J     | Zelaya  | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX  | F      | 25000.00 | 987654321 |   4 |
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
5 rows in set (0.00 sec)
```

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

<u>Query:</u>

SELECT * FROM Employee1 WHERE LOCATE('oracle',CONCAT(Fname,Lname,Addr)) <> 0 OR LOCATE('mysql',CONCAT(Fname,Lname,Addr)) <> 0;

<u>Output:</u>



```
mysql> SELECT * FROM Employee1 WHERE LOCATE('oracle',CONCAT(Fname,Lname,Addr)) <> 0 OR LOCATE('mysql',CONCAT(Fname,Lname,Addr)) <> 0;
+----------+-------+--------+-----------+------------+-------------------------+--------+-----------+-----------+-----+
| Fname    | Minit | Lname  | Ssn       | Bdate      | Addr                    | Gender | Salary    | Super_Ssn | Dno |
+----------+-------+--------+-----------+------------+-------------------------+--------+-----------+-----------+-----+
| 23048753 | K     | Caeser | 230957230 | 1923-12-09 | 234 mysql, Houstan, TX  | M      | 134600.00 | NULL      |   4 |
| 20854985 | T     | Newman | 235872552 | 1979-10-15 | 234 oracle, Houstan, TX | M      |  42500.00 | NULL      |   2 |
+----------+-------+--------+-----------+------------+-------------------------+--------+-----------+-----------+-----+
2 rows in set (0.00 sec)
```

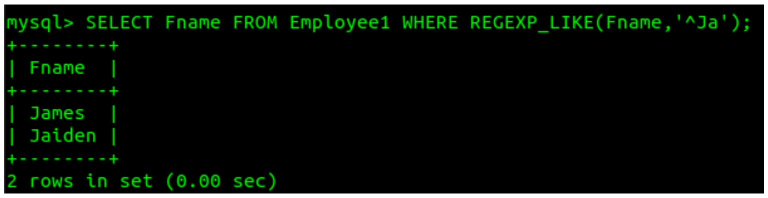
II. In a string attribute of the company schema, match the following using regular expression.

i) Beginning of the string.

Query:

SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'^Ja');

Output:



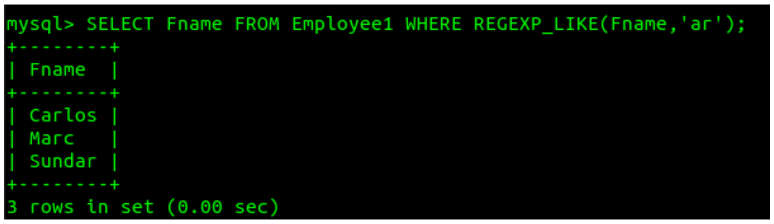
```
mysql> SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'^Ja');
+--------+
| Fname  |
+--------+
| James  |
| Jaiden |
+--------+
2 rows in set (0.00 sec)
```

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

Query:

SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'ar');

Output:



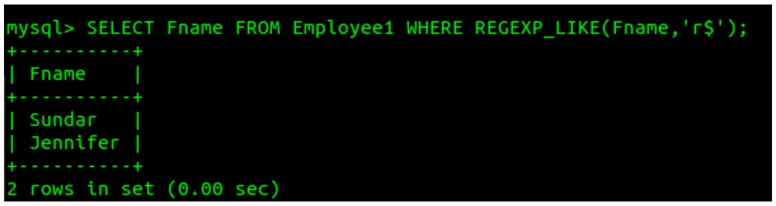
```
mysql> SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'ar');
+--------+
| Fname  |
+--------+
| Carlos |
| Marc   |
| Sundar |
+--------+
3 rows in set (0.00 sec)
```

iii) Match the end of a string.

Query:

SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'r$');

Output:



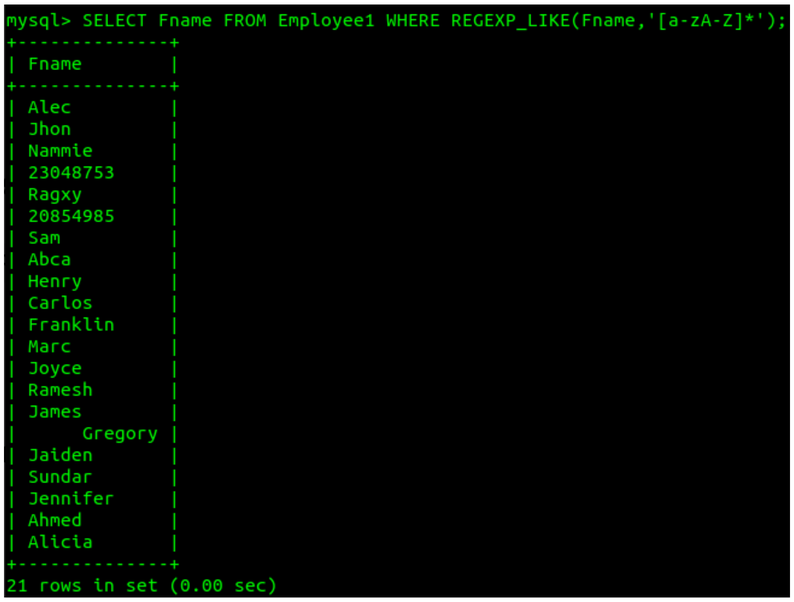
```
mysql> SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'r$');
+----------+
| Fname    |
+----------+
| Sundar   |
| Jennifer |
+----------+
2 rows in set (0.00 sec)
```

iv) Any sequence of zero or more characters.

Query:

SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'[a-zA-Z]*');

<u>Output:</u>



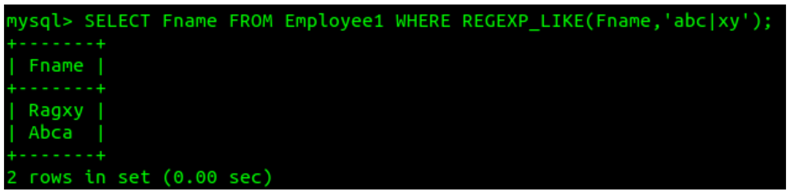
```
mysql> SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'[a-zA-Z]*');
+--------------+
| Fname        |
+--------------+
| Alec         |
| Jhon         |
| Nammie       |
| 23048753     |
| Ragxy        |
| 20854985     |
| Sam          |
| Abca         |
| Henry        |
| Carlos       |
| Franklin     |
| Marc         |
| Joyce        |
| Ramesh       |
| James        |
|      Gregory |
| Jaiden       |
| Sundar       |
| Jennifer     |
| Ahmed        |
| Alicia       |
+--------------+
21 rows in set (0.00 sec)
```

v) Either of the sequences xy or abc.

<u>Query:</u>

SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'abc|xy');

<u>Output:</u>



```
mysql> SELECT Fname FROM Employee1 WHERE REGEXP_LIKE(Fname,'abc|xy');
+-------+
| Fname |
+-------+
| Ragxy |
| Abca  |
+-------+
2 rows in set (0.00 sec)
```