FAIZAN CHOUDHARY

20BCS021

DBMS LAB

18th April 2022

Creation:

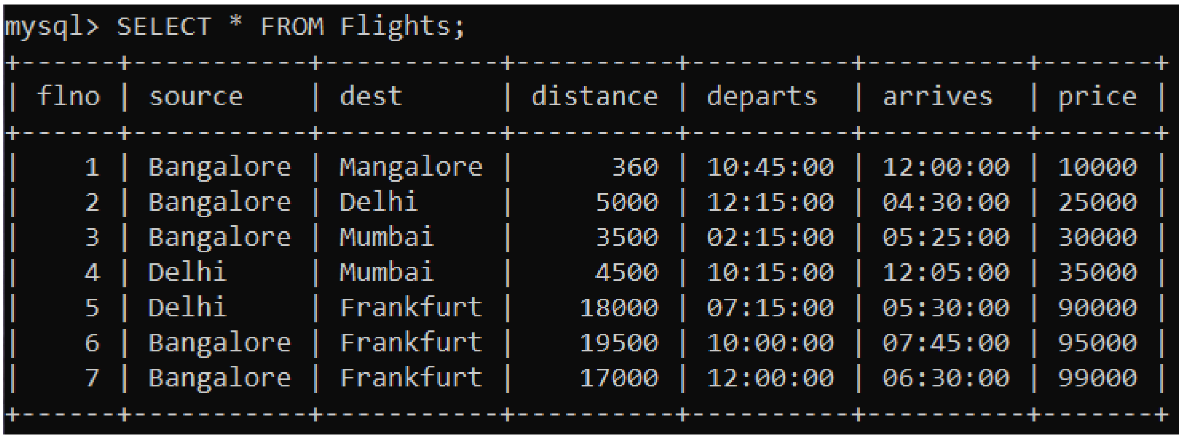
- `mysql> use 20bcs021_faizan;`
- `mysql> CREATE TABLE Flights`

```
    -> (
    -> flno     INT,
    -> source   VARCHAR(10),
    -> dest     VARCHAR(10),
    -> distance INT,
    -> departs  TIME,
    -> arrives  TIME,
    -> price    INT
    -> );
```

- `mysql> INSERT INTO Flights VALUES`

```
-> (1,'Bangalore','Mangalore',360,'10:45:00','12:00:00',10000),
-> (2,'Bangalore','Delhi',5000,'12:15:00','04:30:00',25000),
-> (3,'Bangalore','Mumbai',3500,'02:15:00','05:25:00',30000),
-> (4,'Delhi','Mumbai',4500,'10:15:00','12:05:00',35000),
-> (5,'Delhi','Frankfurt',18000,'07:15:00','05:30:00',90000),
-> (6,'Bangalore','Frankfurt',19500,'10:00:00','07:45:00',95000),
-> (7,'Bangalore','Frankfurt',17000,'12:00:00','06:30:00',99000);
```

- `mysql> SELECT * FROM Flights;`

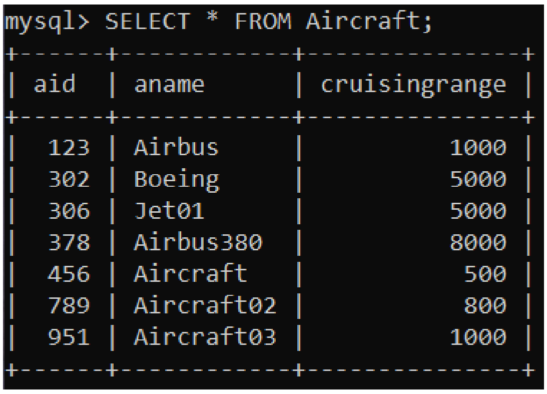


```
mysql> SELECT * FROM Flights;
+------+-----------+-----------+----------+----------+----------+-------+
| flno | source    | dest      | distance | departs  | arrives  | price |
+------+-----------+-----------+----------+----------+----------+-------+
|    1 | Bangalore | Mangalore |      360 | 10:45:00 | 12:00:00 | 10000 |
|    2 | Bangalore | Delhi     |     5000 | 12:15:00 | 04:30:00 | 25000 |
|    3 | Bangalore | Mumbai    |     3500 | 02:15:00 | 05:25:00 | 30000 |
|    4 | Delhi     | Mumbai    |     4500 | 10:15:00 | 12:05:00 | 35000 |
|    5 | Delhi     | Frankfurt |    18000 | 07:15:00 | 05:30:00 | 90000 |
|    6 | Bangalore | Frankfurt |    19500 | 10:00:00 | 07:45:00 | 95000 |
|    7 | Bangalore | Frankfurt |    17000 | 12:00:00 | 06:30:00 | 99000 |
+------+-----------+-----------+----------+----------+----------+-------+
```

-
  ```
  mysql> CREATE TABLE Aircraft
      -> (
      -> aid      INT,
      -> aname    VARCHAR(15),
      -> cruisingrange    INT
      -> );
  ```

-
  ```
  mysql> INSERT INTO Aircraft VALUES
      -> (123,'Airbus',1000),
      -> (302,'Boeing',5000),
      -> (306,'Jet01',5000),
      -> (378,'Airbus380',8000),
      -> (456,'Aircraft',500),
      -> (789,'Aircraft02',800),
      -> (951,'Aircraft03',1000);
  ```

-
  ```
  mysql> SELECT * FROM Aircraft;
  ```

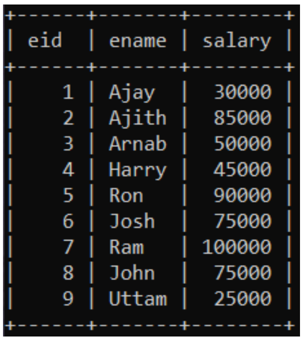


```
mysql> SELECT * FROM Aircraft;
+------+------------+---------------+
| aid  | aname      | cruisingrange |
+------+------------+---------------+
|  123 | Airbus     |          1000 |
|  302 | Boeing     |          5000 |
|  306 | Jet01      |          5000 |
|  378 | Airbus380  |          8000 |
|  456 | Aircraft   |           500 |
|  789 | Aircraft02 |           800 |
|  951 | Aircraft03 |          1000 |
+------+------------+---------------+
```

-
  ```
  mysql> CREATE TABLE Employees
      -> (
      -> eid      INT,
      -> ename    VARCHAR(20),
      -> salary   INT
      -> );
  ```

-
  ```
  mysql> INSERT INTO Employees VALUES
      -> (1,'Ajay',30000),
      -> (2,'Ajith',85000),
      -> (3,'Arnab',50000),
      -> (4,'Harry',45000),
      -> (5,'Ron',90000),
      -> (6,'Josh',75000),
      -> (7,'Ram',100000),
      -> (8, 'John', 75000),
      -> (9, 'Uttam', 25000);
  ```

-
  ```
  mysql> SELECT * FROM Employees;
  ```

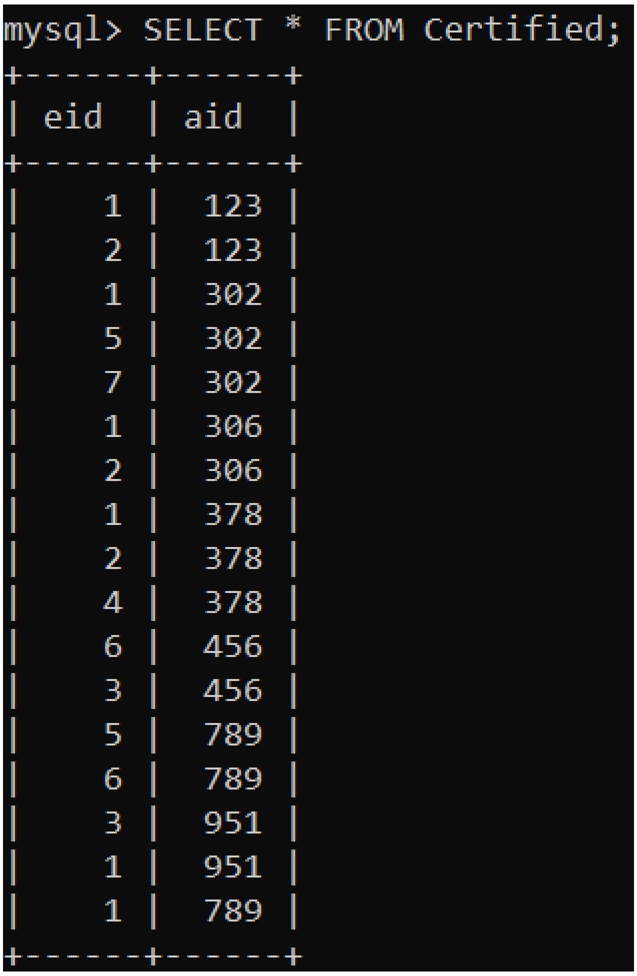


```
+------+-------+--------+
| eid  | ename | salary |
+------+-------+--------+
|    1 | Ajay  |  30000 |
|    2 | Ajith |  85000 |
|    3 | Arnab |  50000 |
|    4 | Harry |  45000 |
|    5 | Ron   |  90000 |
|    6 | Josh  |  75000 |
|    7 | Ram   | 100000 |
|    8 | John  |  75000 |
|    9 | Uttam |  25000 |
+------+-------+--------+
```

-
  ```
  mysql> CREATE TABLE Certified
      -> (
      -> eid      INT,
      -> aid      INT
      -> );
  ```

-
  ```
  mysql> INSERT INTO Certified VALUES
      -> (1,123),
      -> (2,123),
      -> (1,302),
      -> (5,302),
      -> (7,302),
      -> (1,306),
      -> (2,306),
      -> (1,378),
      -> (2,378),
      -> (4,378),
      -> (6,456),
      -> (3,456),
      -> (5,789),
      -> (6,789),
      -> (3,951),
      -> (1,951),
      -> (1,789);
  ```

-
  ```
  mysql> SELECT * FROM Certified;
  ```



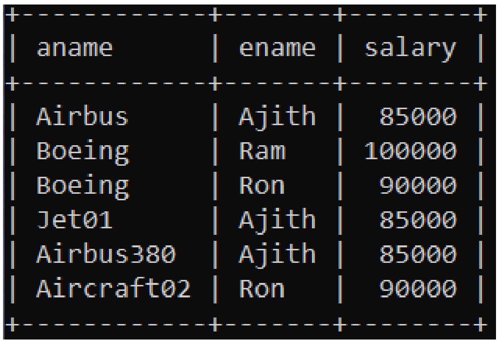
```
mysql> SELECT * FROM Certified;
+------+------+
| eid  | aid  |
+------+------+
|    1 |  123 |
|    2 |  123 |
|    1 |  302 |
|    5 |  302 |
|    7 |  302 |
|    1 |  306 |
|    2 |  306 |
|    1 |  378 |
|    2 |  378 |
|    4 |  378 |
|    6 |  456 |
|    3 |  456 |
|    5 |  789 |
|    6 |  789 |
|    3 |  951 |
|    1 |  951 |
|    1 |  789 |
+------+------+
```

Queries:

1. **Find the names of aircraft such that all pilots certified to operate them earn more than $80,000.**

```
mysql> SELECT A.aname, E.ename, E.salary
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid
    -> WHERE E.salary > 80000;
```

OUTPUT:

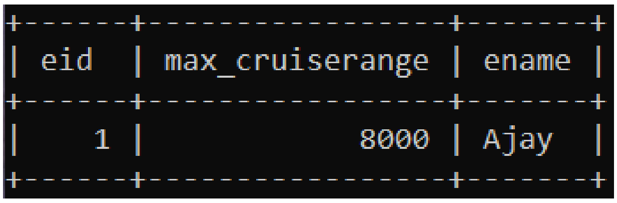


```
+------------+-------+--------+
| aname      | ename | salary |
+------------+-------+--------+
| Airbus     | Ajith |  85000 |
| Boeing     | Ram   | 100000 |
| Boeing     | Ron   |  90000 |
| Jet01      | Ajith |  85000 |
| Airbus380  | Ajith |  85000 |
| Aircraft02 | Ron   |  90000 |
+------------+-------+--------+
```

2. **For each pilot who is certified for more than three aircraft, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.**

```
mysql> SELECT E.eid, MAX(A.cruisingrange) AS max_cruiserange,
E.ename
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid
    -> GROUP BY E.eid
    -> HAVING COUNT(*) > 3;
```

OUTPUT:

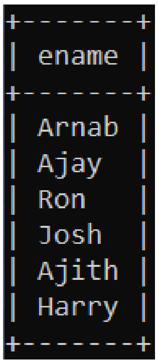


```
+------+-----------------+-------+
| eid  | max_cruiserange | ename |
+------+-----------------+-------+
|    1 |            8000 | Ajay  |
+------+-----------------+-------+
```

3. **Find the names of pilots whose salary is less than the price of the cheapest route from Bangalore to Frankfurt.**

```
mysql> SELECT E.ename
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid INNER JOIN Flights F
    -> WHERE E.salary < (SELECT MIN(price) FROM Flights WHERE
source='Bangalore' AND dest='Frankfurt')
    -> GROUP BY E.ename;
```

OUTPUT:

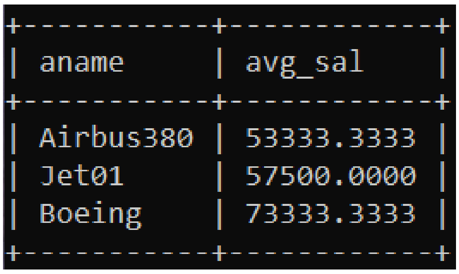


```
+-------+
| ename |
+-------+
| Arnab |
| Ajay  |
| Ron   |
| Josh  |
| Ajith |
| Harry |
+-------+
```

4. **For all aircraft with cruisingrange over 1000 miles, find the name of the aircraft and the average salary of all pilots certified for this aircraft.**

```
mysql> SELECT A.aname, AVG(E.salary) AS avg_sal
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid
    -> WHERE A.cruisingrange > 1000
    -> GROUP BY A.aid;
```

OUTPUT:

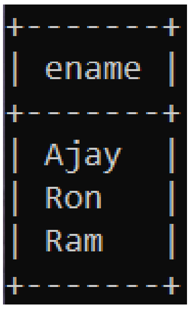


```
+-----------+------------+
| aname     | avg_sal    |
+-----------+------------+
| Airbus380 | 53333.3333 |
| Jet01     | 57500.0000 |
| Boeing    | 73333.3333 |
+-----------+------------+
```

5. **Find the names of pilots certified for some Boeing aircraft.**

```
mysql> SELECT E.ename
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid
    -> WHERE A.aname LIKE 'Boeing%';
```

OUTPUT:

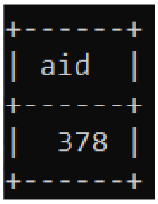


```
+-------+
| ename |
+-------+
| Ajay  |
| Ron   |
| Ram   |
+-------+
```

6. **Find the aid’s of all aircraft that can be used on routes from Bangalore to Delhi.**

```
mysql> SELECT A.aid
    -> FROM Aircraft A INNER JOIN Flights F
    -> WHERE A.cruisingrange > (SELECT MIN(distance) FROM Flights
WHERE source='Bangalore' AND dest='Delhi')
    -> GROUP BY A.aid;
```

OUTPUT:

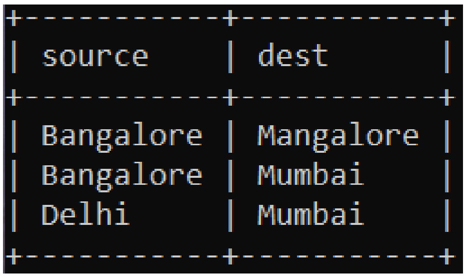


```
+-----+
| aid |
+-----+
| 378 |
+-----+
```

7. **Identify the routes that can be piloted by every pilot who makes more than $80,000.**

```
mysql> SELECT F.source, F.dest
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid INNER JOIN Flights F
    -> WHERE E.salary > 90000 AND A.cruisingrange > F.distance;
```

OUTPUT:

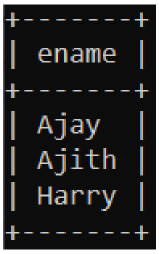


```
+-----------+-----------+
| source    | dest      |
+-----------+-----------+
| Bangalore | Mangalore |
| Bangalore | Mumbai    |
| Delhi     | Mumbai    |
+-----------+-----------+
```

8. **Print the enames of pilots who can operate planes with cruisingrange greater than 3000 miles but are not certified on any Boeing aircraft.**

```
mysql> SELECT DISTINCT E.ename
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid
    -> WHERE A.cruisingrange > 3000 AND A.aname NOT LIKE 'Boeing%';
```

OUTPUT:

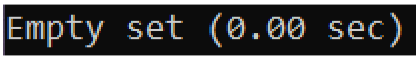


```
+-------+
| ename |
+-------+
| Ajay  |
| Ajith |
| Harry |
+-------+
```

9. **A customer wants to travel from Madison to New York with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in New York by 6 p.m.**

```
mysql> (SELECT F.departs
    -> FROM Flights F
    -> WHERE F.source='Madison' AND F.dest='New York' AND
TIME(F.arrives) < '18:00:00')
    -> UNION
    -> (SELECT F1.departs
    -> FROM Flights F1 INNER JOIN Flights F2 ON F1.dest = F2.source
    -> WHERE F1.source='Madison' AND F2.dest='New York'
    -> AND TIME(F1.arrives) < TIME(F2.departs) AND TIME(F2.arrives)
< '18:00:00')
    -> UNION
    -> (SELECT F1.departs
    -> FROM Flights F1 INNER JOIN Flights F2 ON F1.dest = F2.source
INNER JOIN Flights F3 ON F2.dest = F3.source
    -> WHERE F1.source='Madison' AND F3.dest='New York'
    -> AND TIME(F1.arrives) < TIME(F2.departs) AND TIME(F2.arrives)
< TIME(F3.departs) AND TIME(F3.arrives) < '18:00:00');
```

OUTPUT:



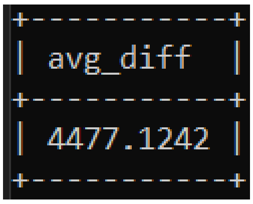
```
Empty set (0.00 sec)
```

(No such entry was added, due to complexity of this query)

10. Compute the difference between the average salary of a pilot and the average salary of all employees (including pilots).

```
mysql> SELECT
    -> ((SELECT AVG(salary) FROM employees) -
    -> (SELECT  AVG(salary)  FROM  employees  INNER  JOIN
certified  ON employees.eid=certified.eid))
    -> AS avg_diff;
```

OUTPUT:

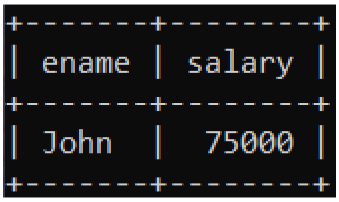


```
+-----------+
| avg_diff  |
+-----------+
| 4477.1242 |
+-----------+
```

11. Print the name and salary of every nonpilot whose salary is more than the average salary for pilots.

```
mysql> SELECT E.ename, E.salary
    -> FROM Employees E
    -> WHERE E.eid NOT IN (SELECT DISTINCT eid FROM Certified)
    -> AND E.salary > (SELECT AVG(salary) FROM Employees E1 INNER
JOIN Certified C1 ON E1.eid=C1.eid);
```

OUTPUT:

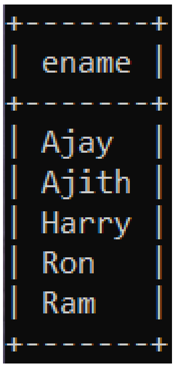


```
+-------+--------+
| ename | salary |
+-------+--------+
| John  |  75000 |
+-------+--------+
```

12. Print the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles.

```
mysql> SELECT DISTINCT E.ename
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid
    -> WHERE A.cruisingrange > 1000;
```

OUTPUT:

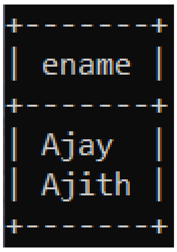


```
+-------+
| ename |
+-------+
| Ajay  |
| Ajith |
| Harry |
| Ron   |
| Ram   |
+-------+
```

13. **Print the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles, but on at least two such aircrafts.**

```
mysql> SELECT E.ename
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid
    -> WHERE A.cruisingrange > 1000
    -> GROUP BY E.ename
    -> HAVING COUNT(*) >= 2;
```

OUTPUT:

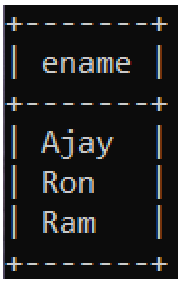


```
+-------+
| ename |
+-------+
| Ajay  |
| Ajith |
+-------+
```

14. **Print the names of employees who are certified only on aircrafts with cruisingrange longer than 1000 miles and who are certified on some Boeing aircraft.**

```
mysql> SELECT E.ename
    -> FROM Aircraft A INNER JOIN (Certified C INNER JOIN Employees
E ON C.eid=E.eid) ON A.aid=C.aid
    -> WHERE A.cruisingrange > 1000 AND A.aname LIKE 'Boeing%';
```

OUTPUT:



```
+-------+
| ename |
+-------+
| Ajay  |
| Ron   |
| Ram   |
+-------+
```