CSLR-51 DBMS - Session 5

1. Relational Database Design – Airlines Travel Schema
Execute the following Queries in SQL over the Flight Schema given below.

Flights(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)
Aircraft(aid: integer, aname: string, cruising range: integer)
Certified(eid: integer, aid: integer)
Employees(eid: integer, ename: string, salary: integer)

Note: Every pilot is certified for some aircraft, and only pilots are certified to fly. Cruising range
means the maximum distance an aircraft can fly without landing, say, 10000 miles. Aircraft
Id(aid) is the company id of the aircraft e.g. Aircraft(101, Boeing, 1000). Employees include pilots along with Airlines(Aircraft) staff.

Identify the Primary key for each table. Before inserting values, please go through the questions
below which shall facilitate you to choose appropriate values for the fields in the table.

Airlines Table Creation

```
CREATE TABLE flights(
   flno INTEGER,
   src VARCHAR(50),
   dest VARCHAR(50),
   distance INTEGER,
   departs TIME,
   arrives TIME,
   price INTEGER,
   PRIMARY KEY(flno,src,dest)
);

CREATE TABLE aircraft(
   aid INTEGER PRIMARY KEY,
   aname VARCHAR(50),
   crusingrange INTEGER
);

CREATE TABLE certified(
   eid INTEGER ,
```

```
    aid INTEGER ,
    PRIMARY KEY (eid,aid)
);

CREATE TABLE employees(
    eid INTEGER PRIMARY KEY,
    ename VARCHAR(50),
    salary INTEGER
);
```

Table Values:

```
INSERT INTO flights VALUES
( 1001,'Chennai','Mumbai',1000,'18:00:00','23:30:00',15000 ),
( 1002,'Trichy','Agartala',2500,'19:30:00','22:20:00',50000 ),
( 1005,'Trichy','Agartala',2500,'13:30:00','17:20:00',30000 ),
( 1007,'Trichy','Agartala',2500,'10:30:00','15:20:00',15000 ),
( 1003,'Ladhak','NY',500,'15:00:00','23:15:00',1000 ),
( 1002,'Ladhak','London',1800,'01:30:00','19:30:00',450000 ),
( 1001,'Ladhak','Trichy',2000,'12:00:00','20:15:00',1200 ),
( 1004,'Ladhak','Tokyo',3470,'05:30:00','20:30:00',23400 ),
( 1005,'Chandigarh','Surat',2450,'20:00:00','24:00:00',45000 ),
( 1007,'Chandigarh','Surat',2450,'14:30:00','20:00:00',12000 ),
( 1006,'Chandigarh','Surat',2450,'04:30:00','22:30:00',50000 ),
( 1003,'Chandigarh','Surat',2450,'10:30:00','17:45:00',10000 ),
( 1003,'Kolkata','Chennai',235,'12:15:00','23:00:00',30000 ),
( 1002,'Kolkata','Mumbai',500,'00:00:00','10:30:00',15000 ),
( 1001,'Mumbai','Hyderabad',645,'13:45:00','15:00:00',20000 ),
( 1005,'Hyderabad','Bangalore',350,'19:15:00','23:15:00',50000 ),
( 1004,'Banglore','Chennai',100,'14:30:00','19:00:00',10000 ),
( 1005,'Chennai','Dublin',12000,'08:10:00','12:30:00',2000 ),
( 1004,'Trichy','Chennai',1000,'09:20:00','11:20:00',50000 ),
( 1008,'Chennai','Delhi',5000,'16:00:00','23:30:00',80000 ),
( 1009,'Delhi','Dubai',12500,'10:40:00','20:00:00',100000 ),
( 1010,'Dubai','Frankfurt',50000,'22:00:00','23:00:00',75000 ),
( 1002,'Dublin','Pairs',15000,'22:00:00','23:55:00',50000 ),
( 1005,'Pairs','Dubai',25000,'19:15:00','22:10:00',25000 ),
( 1008,'Dubai','Kolkata',7500,'12:10:00','16:15:00',10000 ),
( 1009,'Kolkata','Chennai',1500,'05:45:00','08:30:00',23500 ),
( 1001,'Chennai','Banglore',1500,'08:45:00','13:45:00',50000 ),
( 1010,'Banglore','Dubai',3000,'12:15:00','17:30:00',10000 ),
( 1004,'Dubai','Berlin',15000,'16:00:00','22:00:00',25000 ),
( 1002,'Berlin','Dublin',8500,'18:30:00','23:50:00',75000 );

INSERT INTO aircraft VALUES
( 1001,'Boeing',100 ),
( 1002,'Antanov',5000 ),
( 1003,'Antanov',1500 ),
```

```
( 1004,'Boeing',5000 ),
( 1005,'Airbus',500 ),
( 1006,'Boeing',15000 ),
( 1007,'Boeing',2000 ),
( 1008,'Airbus',750 ),
( 1009,'Antanov',2500 ),
( 1010,'Airbus',15000 );

INSERT INTO certified VALUES
( 101,1001 ),
( 101,1002 ),
( 101,1004 ),
( 102,1005 ),
( 102,1006 ),
( 102,1003 ),
( 103,1005 ),
( 103,1006 ),
( 103,1004 ),
( 104,1001 ),
( 104,1002 ),
( 104,1005 ),
( 104,1006 ),
( 105,1008 ),
( 105,1010 ),
( 105,1009 ),
( 105,1005 ),
( 105,1006 ),
( 106,1003 ),
( 107,1002 ),
( 107,1001 ),
( 108,1004 ),
( 109,1006 ),
( 109,1007 ),
( 110,1008 ),
( 110,1001 );


INSERT INTO employees VALUES
( 101,'Albert',10000 ),
( 102,'Bob',250000 ),
( 103,'Clair',1500 ),
( 104,'Douglas',450000 ),
( 105,'Einstein',30000 ),
( 106,'Franklin',1500000 ),
( 107,'George',500000 ),
( 108,'Harry',100000 ),
( 109,'Jack',250000 ),
( 110,'Lincon',75000 ),
( 201,'Morris',50000 ),
```

(202,'Nick',100000),
(203,'Parker',1000000),
(204,'Robert',150000),
(205,'Sam',500000);

1. **Find the names of aircraft such that all pilots certified to operate them earn more than Rs.50,000.**
   Query :
   SELECT DISTINCT aname FROM (employees NATURAL JOIN certified) NATURAL JOIN aircraft WHERE salary > 50000;

   Output :

```
mysql> SELECT DISTINCT aname FROM (employees NATURAL JOIN certified) NATURAL JOI
N aircraft WHERE salary > 50000;
+----------+
| aname    |
+----------+
| Boeing   |
| Antanov  |
| Airbus   |
+----------+
3 rows in set (0.00 sec)
```

2. **For each pilot who is certified for more than three aircraft, find the eid and the maximum cruising range of the aircraft for which she/he is certified.**
   Query :
   SELECT eid,MAX(crusingrange) FROM certified NATURAL JOIN aircraft GROUP BY eid HAVING COUNT(*) > 3;

   Output :

```
mysql> SELECT eid,MAX(crusingrange) FROM certified NATURAL JOIN aircraft GROUP B
Y eid HAVING COUNT(*) > 3;
+-----+-------------------+
| eid | MAX(crusingrange) |
+-----+-------------------+
| 104 |             15000 |
| 105 |             15000 |
+-----+-------------------+
2 rows in set (0.00 sec)
```

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

   Query :

   SELECT DISTINCT ename FROM employees WHERE salary < (SELECT min(price) FROM flights WHERE src = 'Trichy' AND dest = 'Agartala');

   Output :

```
mysql> SELECT DISTINCT ename FROM employees WHERE salary < (SELECT min(price) FR
OM flights WHERE src = 'Trichy' AND dest = 'Agartala');
+--------+
| ename  |
+--------+
| Albert |
| Clair  |
+--------+
2 rows in set (0.00 sec)
```

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.**

   Query :

   SELECT DISTINCT aname,AVG(e.salary) FROM (aircraft as a JOIN certified as c ON a.aid = c.aid) JOIN employees as e ON c.eid = e.eid WHERE a.crusingrange > 1000 GROUP BY aname;

   Output :

```
mysql> SELECT DISTINCT aname,AVG(e.salary) FROM (aircraft as a JOIN certified as
 c ON a.aid = c.aid) JOIN employees as e ON c.eid = e.eid WHERE a.crusingrange >
 1000 GROUP BY aname;
+---------+---------------+
| aname   | AVG(e.salary) |
+---------+---------------+
| Antanov |   456666.6667 |
| Boeing  |   149222.2222 |
| Airbus  |    30000.0000 |
+---------+---------------+
3 rows in set (0.00 sec)
```

5. **Find the names of pilot/s certified for some Boeing aircraft who drove the maximum distance on all flights departing from Ladakh.**

   Query :

   SELECT DISTINCT e.ename FROM ((employees as e NATURAL JOIN certified as c) NATURAL JOIN aircraft as a) JOIN flights as f ON a.crusingrange > f.distance WHERE a.aname = 'Boeing' AND f.src = 'Ladhak' AND f.distance = (SELECT MAX(distance) FROM flights WHERE src = 'Ladhak');

Output :

```
mysql> SELECT DISTINCT e.ename FROM ((employees as e NATURAL JOIN certified as c
) NATURAL JOIN aircraft as a) JOIN flights as f ON a.crusingrange > f.distance W
HERE a.aname = 'Boeing' AND f.src = 'Ladhak' AND f.distance = (SELECT MAX(distan
ce) FROM flights WHERE src = 'Ladhak');
+-----------+
| ename     |
+-----------+
| Albert    |
| Bob       |
| Clair     |
| Douglas   |
| Einstein  |
| Harry     |
| Jack      |
+-----------+
7 rows in set (0.00 sec)
```

6. **Find the aids of all aircraft that can be used on routes from Chandigarh to Surat.**

   Query :

   SELECT DISTINCT aid FROM flights JOIN aircraft ON crusingrange >= distanc
   e WHERE src = 'Chandigarh' AND dest = 'Surat';

   Output :

```
mysql> SELECT DISTINCT aid FROM flights JOIN aircraft ON crusingrange >= distanc
e WHERE src = 'Chandigarh' AND dest = 'Surat';
+-------+
| aid   |
+-------+
| 1002  |
| 1004  |
| 1006  |
| 1009  |
| 1010  |
+-------+
5 rows in set (0.00 sec)
```

7. **Identify the routes that can be piloted by every pilot who makes more than 100,000.**

   Query :

   SELECT DISTINCT src, dest FROM employees NATURAL JOIN certified NATURAL JOIN aircraft JOIN flights ON crusingrange >= distance WHERE salary > 100000;

   Output :

```
mysql> SELECT DISTINCT src, dest FROM employees NATURAL JOIN certified NATURAL J
OIN aircraft JOIN flights ON crusingrange >= distance WHERE salary > 100000;
+------------+-----------+
| src        | dest      |
+------------+-----------+
| Chennai    | Banglore  |
| Chennai    | Mumbai    |
| Ladhak     | Trichy    |
| Mumbai     | Hyderabad |
| Berlin     | Dublin    |
| Dublin     | Pairs     |
| Kolkata    | Mumbai    |
| Ladhak     | London    |
| Trichy     | Agartala  |
| Chandigarh | Surat     |
| Kolkata    | Chennai   |
| Ladhak     | NY        |
| Banglore   | Chennai   |
| Dubai      | Berlin    |
| Ladhak     | Tokyo     |
| Trichy     | Chennai   |
| Chennai    | Dublin    |
| Hyderabad  | Bangalore |
| Chennai    | Delhi     |
| Dubai      | Kolkata   |
| Delhi      | Dubai     |
| Banglore   | Dubai     |
+------------+-----------+
22 rows in set (0.00 sec)
```

8. **Print the enames of pilots who can operate planes with cruising range greater than 3000 miles but are not certified on any Boeing aircraft.**
   Query :
   SELECT ename FROM employees NATURAL JOIN certified NATURAL JOIN aircraft WHERE crusingrange > 3000 AND aname <> 'Boeing';

   Output :

```
mysql> SELECT ename FROM employees NATURAL JOIN certified NATURAL JOIN aircraft
WHERE crusingrange > 3000 AND aname <> 'Boeing';
+----------+
| ename    |
+----------+
| Albert   |
| Douglas  |
| Einstein |
| George   |
+----------+
4 rows in set (0.00 sec)
```

9. **Compute the difference between the average salary of a pilot and the average salary of all employees (including pilots).**
   Query :
   SELECT (SELECT AVG(a.salary) FROM (SELECT DISTINCT ename, salary FROM employees NATURAL JOIN certified) as a) - (SELECT AVG(salary) FROM employees) AS 'diff between avg salaries';

   Output :

```
mysql> SELECT (SELECT AVG(a.salary) FROM (SELECT DISTINCT ename, salary FROM emp
loyees NATURAL JOIN certified) as a) - (SELECT AVG(salary) FROM employees) AS 'd
iff between avg salaries';
+--------------------------+
| diff between avg salaries |
+--------------------------+
|              -14450.0000 |
+--------------------------+
1 row in set (0.01 sec)
```

10. **Print the name and salary of every nonpilot whose salary is more than the average salary for pilots.**
    Query :
    SELECT ename, salary FROM employees WHERE eid NOT IN (SELECT DISTINCT eid FROM certified) AND salary > (SELECT AVG(salary) FROM employees WHERE eid IN (SELECT DISTINCT eid FROM certified));

Output :

```
mysql> SELECT ename, salary FROM employees WHERE eid NOT IN (SELECT DISTINCT eid
 FROM certified) AND salary > (SELECT AVG(salary) FROM employees WHERE eid IN (S
ELECT DISTINCT eid FROM certified));
+--------+---------+
| ename  | salary  |
+--------+---------+
| Parker | 1000000 |
| Sam    |  500000 |
+--------+---------+
2 rows in set (0.00 sec)
```

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

    Query :

    SELECT DISTINCT ename FROM employees NATURAL JOIN certified NATURAL JOIN aircraft WHERE crusingrange > 1000;

    Output :

```
mysql> SELECT DISTINCT ename FROM employees NATURAL JOIN certified NATURAL JOIN
aircraft WHERE crusingrange > 1000;
+----------+
| ename    |
+----------+
| Albert   |
| Bob      |
| Clair    |
| Douglas  |
| Einstein |
| Franklin |
| George   |
| Harry    |
| Jack     |
+----------+
9 rows in set (0.00 sec)
```

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

    Query :

    SELECT ename FROM employees NATURAL JOIN certified NATURAL JOIN aircraft WHERE crusingrange < 1000 GROUP BY ename HAVING COUNT(*) >= 2;

    Output :

```
mysql> SELECT ename FROM employees NATURAL JOIN certified NATURAL JOIN aircraft
WHERE crusingrange < 1000 GROUP BY ename HAVING COUNT(*) >= 2;
+----------+
| ename    |
+----------+
| Douglas  |
| Einstein |
| Lincon   |
+----------+
3 rows in set (0.00 sec)
```

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

    Query :

    SELECT DISTINCT ename FROM employees NATURAL JOIN certified NATURAL JOIN aircraft WHERE crusingrange > 1000 AND aname = 'Boeing';

    Output :

```
mysql> SELECT DISTINCT ename FROM employees NATURAL JOIN certified NATURAL JOIN
aircraft WHERE crusingrange > 1000 AND aname = 'Boeing';
+----------+
| ename    |
+----------+
| Albert   |
| Bob      |
| Clair    |
| Douglas  |
| Einstein |
| Harry    |
| Jack     |
+----------+
7 rows in set (0.00 sec)
```

14. **Find the eids of pilots certified for some Boeing aircraft.**

    Query :

    SELECT DISTINCT eid from certified NATURAL JOIN aircraft WHERE aname = 'Boeing';

    Output :

```
mysql> SELECT DISTINCT eid from certified NATURAL JOIN aircraft WHERE aname = 'B
oeing';
+-----+
| eid |
+-----+
| 101 |
| 102 |
| 103 |
| 104 |
| 105 |
| 107 |
| 108 |
| 109 |
| 110 |
+-----+
9 rows in set (0.00 sec)
```

15. **Retrieve the names of pilots certified for some Boeing aircraft.**

    Query :

    SELECT DISTINCT ename from employees NATURAL JOIN certified NATURAL JOIN aircraft WHERE aname = 'Boeing';

Output :

```
mysql> SELECT DISTINCT ename from employees NATURAL JOIN certified NATURAL JOIN
aircraft WHERE aname = 'Boeing';
+----------+
| ename    |
+----------+
| Albert   |
| Bob      |
| Clair    |
| Douglas  |
| Einstein |
| George   |
| Harry    |
| Jack     |
| Lincon   |
+----------+
9 rows in set (0.00 sec)
```

16. **Find the aids of all aircraft that can be used on non-stop flights from Kolkata to Madras.**

    Query :

    SELECT DISTINCT aid FROM flights JOIN aircraft ON crusingrange >= distance WHERE src = 'Kolkata' AND dest = 'Chennai';

    Output :

```
mysql> SELECT DISTINCT aid FROM flights JOIN aircraft ON crusingrange >= distanc
e WHERE src = 'Kolkata' AND dest = 'Chennai';
+------+
| aid  |
+------+
| 1002 |
| 1003 |
| 1004 |
| 1005 |
| 1006 |
| 1007 |
| 1008 |
| 1009 |
| 1010 |
+------+
9 rows in set (0.00 sec)
```

17. **Identify the flights that can be piloted by every pilot whose salary is more than 70,000.**

    Query :

    SELECT DISTINCT aid FROM aircraft NATURAL JOIN certified NATURAL JOIN employees WHERE salary > 70000 ORDER BY aid;

    Output :

```
mysql> SELECT DISTINCT aid FROM aircraft NATURAL JOIN certified NATURAL JOIN emp
loyees WHERE salary > 70000 ORDER BY aid;
+------+
| aid  |
+------+
| 1001 |
| 1002 |
| 1003 |
| 1004 |
| 1005 |
| 1006 |
| 1007 |
| 1008 |
+------+
8 rows in set (0.00 sec)
```

18. **Find the names of pilots who can operate planes with a range greater than 3,000 miles but are not certified on any Boeing aircraft.**

    Query :

    SELECT DISTINCT ename FROM employees NATURAL JOIN certified NATURAL JOIN aircraft WHERE crusingrange < 3000 AND aname <> 'Boeing';

    Output :

```
mysql> SELECT DISTINCT ename FROM employees NATURAL JOIN certified NATURAL JOIN
aircraft WHERE crusingrange < 3000 AND aname <> 'Boeing';
+----------+
| ename    |
+----------+
| Bob      |
| Clair    |
| Douglas  |
| Einstein |
| Franklin |
| Lincon   |
+----------+
6 rows in set (0.01 sec)
```

19. **Find the eids of employees who make the highest salary in every airline.**

    Query :

    SELECT eid,aname,salary FROM employees NATURAL JOIN certified NATURAL JOIN aircraft WHERE (aname,salary) IN (SELECT aname,MAX(salary) FROM employees NATURAL JOIN certified NATURAL JOIN aircraft GROUP BY aname);

    Output :

```
mysql> SELECT eid,aname,salary FROM employees NATURAL JOIN certified NATURAL JOI
N aircraft WHERE (aname,salary) IN ( SELECT aname,MAX(salary) FROM employees NAT
URAL JOIN certified NATURAL JOIN aircraft GROUP BY aname);
+-----+---------+---------+
| eid | aname   | salary  |
+-----+---------+---------+
| 104 | Airbus  |  450000 |
| 106 | Antanov | 1500000 |
| 107 | Boeing  |  500000 |
+-----+---------+---------+
3 rows in set (0.00 sec)
```

20. **Retrieve the eids of employees who make the second highest salary.**

    Query :

    SELECT eid FROM employees WHERE salary = (SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary) FROM employees));

    Output :

```
mysql> SELECT eid FROM employees WHERE salary = (SELECT MAX(salary) FROM employe
es WHERE salary < (SELECT MAX(salary) FROM employees));
+-----+
| eid |
+-----+
| 203 |
+-----+
1 row in set (0.00 sec)
```

21. **Find the eids of employees who are certified for the largest number of aircraft.**

Query :

SELECT eid FROM certified GROUP BY eid HAVING COUNT(*) = (SELECT COUNT(*) FROM certified GROUP BY eid ORDER BY COUNT(*) DESC LIMIT 1);

Output :

```
mysql> SELECT eid FROM certified GROUP BY eid HAVING COUNT(*) = (SELECT COUNT(*)
 FROM certified GROUP BY eid ORDER BY COUNT(*) DESC LIMIT 1);
+-----+
| eid |
+-----+
| 105 |
+-----+
1 row in set (0.00 sec)
```

22. **Find the eids of employees who are certified for exactly three aircrafts.**

Query :

SELECT eid FROM certified GROUP BY eid HAVING COUNT(*) = 3;

Output :

```
mysql> SELECT eid FROM certified GROUP BY eid HAVING COUNT(*) = 3;
+-----+
| eid |
+-----+
| 101 |
| 102 |
| 103 |
+-----+
3 rows in set (0.00 sec)
```

23. **Find the total amount paid to pilots who drove greater than 500,000 miles together across all their journeys on the routes from Chennai to Dublin and return routes also. You need to consider all direct flights along with the connecting flights as well.**

Query :

SELECT (SELECT price FROM flights WHERE src = 'Chennai' AND dest = 'Dublin') + (SELECT price FROM flights WHERE src = 'Dublin' AND dest = 'Chennai') AS Tot_price;

Output :

```
mysql> SELECT (SELECT price FROM flights WHERE src = 'Chennai' AND dest = 'Dublin') +
 (SELECT price FROM flights WHERE src = 'Dublin' AND dest = 'Chennai') AS Tot_price;
+-----------+
| Tot_price |
+-----------+
|     77000 |
+-----------+
1 row in set (0.00 sec)
```

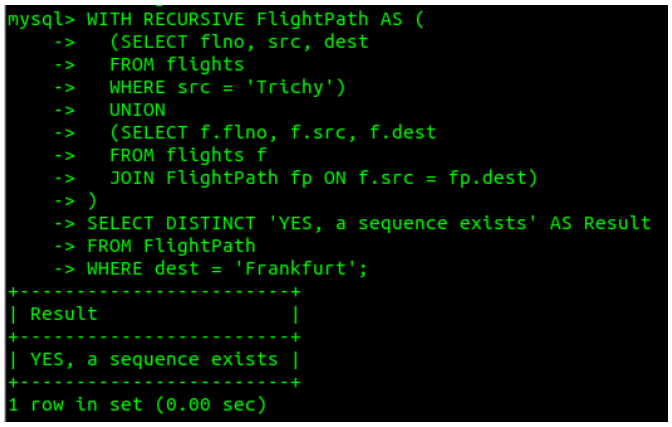
24. **Is there a sequence of flights from Tiruchirappalli to Frankfurt? Each flight in the sequence is required to depart from the city that is the destination of the previous flight; the first flight must leave Tiruchirappalli, the last flight must reach Frankfurt, and there is no restriction on the number of intermediate flights. Your**

query must determine whether a sequence of flights from Tiruchirappalli to Frankfurt exists for any input Flights relation instance.

Query :

```
WITH RECURSIVE FlightPath AS (  (SELECT flno, src, dest
  FROM flights
  WHERE src = 'Trichy')
  UNION
  (SELECT f.flno, f.src, f.dest
  FROM flights f
  JOIN FlightPath fp ON f.src = fp.dest) ) SELECT DISTINCT 'YES, a sequence exists
AS Result
FROM FlightPath
WHERE dest = 'Frankfurt';
```

Output :



```
mysql> WITH RECURSIVE FlightPath AS (
    ->   (SELECT flno, src, dest
    ->   FROM flights
    ->   WHERE src = 'Trichy')
    ->   UNION
    ->   (SELECT f.flno, f.src, f.dest
    ->   FROM flights f
    ->   JOIN FlightPath fp ON f.src = fp.dest)
    -> )
    -> SELECT DISTINCT 'YES, a sequence exists' AS Result
    -> FROM FlightPath
    -> WHERE dest = 'Frankfurt';
+------------------------+
| Result                 |
+------------------------+
| YES, a sequence exists |
+------------------------+
1 row in set (0.00 sec)
```

25. **Create your own query: define what you want to do in English, then write the query in SQL. Make it as difficult as you wish, the harder the better.**

Statement :

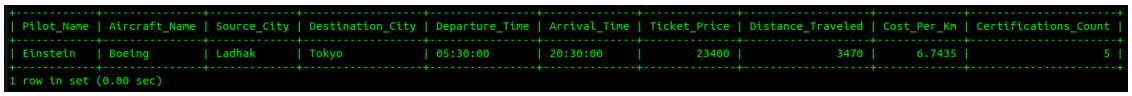
Find the most cost-effective flight route operated by the most certified pilot, using the aircraft with the longest cruising range, departing before noon and arriving after 6 PM, and return the pilots name, aircraft name, and route details.

Query :

```
SELECT
        e.ename AS Pilot_Name,
        a.aname AS Aircraft_Name,
        f.src AS Source_City,
        f.dest AS Destination_City,
        f.departs AS Departure_Time,
        f.arrives AS Arrival_Time,
```

```
        f.price AS Ticket_Price,
        f.distance AS Distance_Traveled,
        (f.price / f.distance) AS Cost_Per_Km,
        (SELECT COUNT(*) FROM certified c1 WHERE c1.eid = e.eid) AS
Certifications_Count
FROM
        flights f NATURAL JOIN employees e
        NATURAL JOIN certified c
        JOIN aircraft a ON f.distance <= (SELECT MAX(a1.crusingrange) FROM aircraft
a1 WHERE a1.aid = c.aid)
WHERE f.departs < '12:00:00' AND f.arrives > '18:00:00' AND
e.eid = ( SELECT e2.eid FROM certified c2 NATURAL JOIN employees e2 GROUP BY
e2.eid ORDER BY COUNT(*) DESC LIMIT 1) ORDER BY Cost_Per_Km ASC LIMIT 1;
```

Output :



Pilot_Name	Aircraft_Name	Source_City	Destination_City	Departure_Time	Arrival_Time	Ticket_Price	Distance_Traveled	Cost_Per_Km	Certifications_Count
Einstein	Boeing	Ladhak	Tokyo	05:30:00	20:30:00	23400	3470	6.7435	5

1 row in set (0.00 sec)

2. With continuation to Session 04 exercise, execute all the example queries provided in Subsections 7.1.1 to 7.4.2 in text book by Navathe et al. pertaining to keywords 'TRIGGER', 'VIEW', 'EXCEPT' and 'CONTAINS'.

Query :

SELECT Fname, Lname FROM Employee WHERE NOT EXISTS ((SELECT Pnumber FROM Project WHERE Dnum = 5) EXCEPT (SELECT Pno FROM Works_on WHERE Ssn = Essn));

Output :

```
mysql> SELECT Fname, Lname FROM Employee WHERE  NOT EXISTS ( ( SELECT Pnumber FR
OM Project WHERE Dnum = 5) EXCEPT (SELECT Pno FROM Works_on WHERE Ssn = Essn));
Empty set (0.00 sec)
```

Query :

CREATE TRIGGER Salary_Violation BEFORE SELECT OR UPDATE OF salary,Super_Ssn ON Employee FOR EACH ROW WHEN (NEW.salary > (SELECT salary FROM Employee WHERE Ssn = new.Super_Ssn)) INFORM_SUPERVISOR(new.super_ssn,new.ssn);

Query :

CREATE VIEW works_on1 AS SELECT Fname,Lname,Pname,hours FROM Employee,Project,Works_on WHERE Ssn = Essn AND Pno = Pnumber;

Output :

```
mysql> table works_on1;
+----------+---------+-----------------+-------+
| Fname    | Lname   | Pname           | hours |
+----------+---------+-----------------+-------+
| Franklin | Wlong   | Computerization |  10.0 |
| Ahmed    | Jabbar  | Computerization |  35.0 |
| Alicia   | Zelaya  | Computerization |  10.0 |
| Jennifer | Wallace | Newbenefits     |  20.0 |
| Ahmed    | Jabbar  | Newbenefits     |   5.0 |
| Alicia   | Zelaya  | Newbenefits     |  30.0 |
| Jhon     | Smith   | ProductX        |  32.5 |
| Joyce    | English | ProductX        |  20.0 |
| Jhon     | Smith   | ProductY        |   7.5 |
| Franklin | Wlong   | ProductY        |  10.0 |
| Joyce    | English | ProductY        |  20.0 |
| Franklin | Wlong   | ProductZ        |  10.0 |
| Ramesh   | Narayan | ProductZ        |  40.0 |
| Franklin | Wlong   | Reorganization  |  10.0 |
| James    | Borg    | Reorganization  |  15.0 |
| Jennifer | Wallace | Reorganization  |  15.0 |
+----------+---------+-----------------+-------+
16 rows in set (0.00 sec)
```

Query :

CREATE VIEW dept_info(dept_name,no_of_emps,tot_sal) AS SELECT Dname,COUNT(*),SUM(salary) FROM Department,Employee WHERE Dnumber = Dno GROUP BY Dname;

Output :

```
mysql> table dept_info;
+----------------+------------+-----------+
| dept_name      | no_of_emps | tot_sal   |
+----------------+------------+-----------+
| Research       |          4 | 133000.00 |
| Headquarters   |          1 |  55000.00 |
| Administration |          3 |  93000.00 |
+----------------+------------+-----------+
3 rows in set (0.00 sec)
```

Query :

SELECT Fname,Lname FROM works_on1 WHERE Pname = 'ProductX';

Output :

```
mysql> SELECT Fname,Lname FROM works_on1 WHERE Pname = 'ProductX';
+-------+---------+
| Fname | Lname   |
+-------+---------+
| Jhon  | Smith   |
| Joyce | English |
+-------+---------+
2 rows in set (0.00 sec)
```

Query :

DROP VIEW works_on1;

Output :

```
mysql> DROP VIEW works_on1;
Query OK, 0 rows affected (0.00 sec)
```

Query :

CREATE VIEW dept5emp AS SELECT * FROM Employee WHERE Dno = 5;

Output :

```
mysql> table dept5emp;
+----------+-------+---------+-----------+------------+-------------------------+--------+----------+-----------+-----+
| 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 |
| 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 |
+----------+-------+---------+-----------+------------+-------------------------+--------+----------+-----------+-----+
4 rows in set (0.00 sec)
```

Query :

CREATE VIEW basic_emp_data AS SELECT Fname,Lname,Addr FROM Employee;

Output :

```
mysql> table basic_emp_data;
+----------+---------+--------------------------+
| Fname    | Lname   | Addr                     |
+----------+---------+--------------------------+
| Jhon     | Smith   | 731 Fondren, Houston, TX |
| Franklin | Wlong   | 638 Voss, Houston, TX    |
| Joyce    | English | 5631 Rice, Houston, TX   |
| Ramesh   | Narayan | 975 Fire Oak, Humble, TX |
| James    | Borg    | 450 Stone, Houston, TX   |
| Jennifer | Wallace | 291 Berry, Bellaire, TX  |
| Ahmed    | Jabbar  | 980 Dallas, Houston, TX  |
| Alicia   | Zelaya  | 3321 Castle, Spring, TX  |
+----------+---------+--------------------------+
8 rows in set (0.00 sec)
```

3. Write the following as triggers on the EMPLOYEE Schema which you have already created. 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.

1. **Assure that deleting details of an employee deletes his dependent records also.**

   Query:

   Delimiter $$
   CREATE TRIGGER Delete_dependent BEFORE DELETE ON Employee FOR EACH ROW BEGIN DELETE FROM Dependent WHERE Essn = OLD.Ssn; END$$
   Delimiter ;

   Output:

```
mysql> table Employee;
+---------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| 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      | 50000.00 | 987654321 |   4 |
| 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)
```

```
mysql> table Dependent;
+-----------+----------------+--------+------------+--------------+
| Essn      | Dependent_name | Gender | Bdate      | Relationship |
+-----------+----------------+--------+------------+--------------+
| 123456789 | Alice          | F      | 1988-12-30 | Daughter     |
| 123456789 | Elizabeth      | F      | 1967-05-05 | Spouse       |
| 123456789 | Michae         | M      | 1988-01-04 | Son          |
| 333445555 | Alice          | F      | 1986-04-05 | Daughter     |
| 333445555 | Joy            | F      | 1958-05-03 | Spouse       |
| 333445555 | Theodore       | M      | 1983-10-25 | Son          |
| 391391391 | George         | M      | 1950-04-15 | Son          |
| 987654321 | Abner          | M      | 1942-02-28 | Spouse       |
+-----------+----------------+--------+------------+--------------+
8 rows in set (0.00 sec)
```

   Deleting Kristin will delete George from Dependent

   DELETE FROM Employee WHERE Fname = ‘Kristin’

   New Output:

```
mysql> table Employee;
+---------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| 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 |
| 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 |
+---------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
8 rows in set (0.01 sec)
```

```
mysql> table Dependent;
+-----------+----------------+--------+------------+--------------+
| Essn      | Dependent_name | Gender | Bdate      | Relationship |
+-----------+----------------+--------+------------+--------------+
| 123456789 | Alice          | F      | 1988-12-30 | Daughter     |
| 123456789 | Elizabeth      | F      | 1967-05-05 | Spouse       |
| 123456789 | Michae         | M      | 1988-01-04 | Son          |
| 333445555 | Alice          | F      | 1986-04-05 | Daughter     |
| 333445555 | Joy            | F      | 1958-05-03 | Spouse       |
| 333445555 | Theodore       | M      | 1983-10-25 | Son          |
| 987654321 | Abner          | M      | 1942-02-28 | Spouse       |
+-----------+----------------+--------+------------+--------------+
7 rows in set (0.00 sec)
```

2. **Whenever a department with exactly one project is shifted to a new location, ensure that the project is also shifted to the new location.**

Query:

```
DELIMITER $$
CREATE TRIGGER Shift_project_location AFTER INSERT OR UPDATE OF Dlocation
ON Dept_lcoations FOR EACH ROW BEGIN IF (SELECT COUNT(*) FROM Project
WHERE Dnum = NEW.Dnumber) = 1 THEN UPDATE Project SET Plocation =
NEW.Dlocation WHERE Dnum = NEW.Dnumber; END IF; END$$
DELIMITER ;
```

Output:

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

```
mysql> table Project;
+-----------------+---------+-----------+------+
| Pname           | Pnumber | Plocation | Dnum |
+-----------------+---------+-----------+------+
| ProductX        |       1 | Bellaire  |    5 |
| ProductY        |       2 | Sugarland |    5 |
| ProductZ        |       3 | Houston   |    5 |
| Computerization |      10 | Stafford  |    4 |
| Reorganization  |      20 | Houston   |    1 |
| Newbenefits     |      30 | Stafford  |    4 |
+-----------------+---------+-----------+------+
6 rows in set (0.00 sec)
```

Query : UPDATE Dept_locations SET Dlocation = 'New York' WHERE Dnumber = 1;

New Output :

```
mysql> table Dept_locations;
+---------+-----------+
| Dnumber | Dlocation |
+---------+-----------+
|       1 | New York  |
|       4 | Stafford  |
|       5 | Bellaire  |
|       5 | Houston   |
|       5 | Sugarland |
+---------+-----------+
5 rows in set (0.00 sec)
```

```
mysql> table Project;
+-----------------+---------+-----------+------+
| Pname           | Pnumber | Plocation | Dnum |
+-----------------+---------+-----------+------+
| ProductX        |       1 | Bellaire  |    5 |
| ProductY        |       2 | Sugarland |    5 |
| ProductZ        |       3 | Houston   |    5 |
| Computerization |      10 | Stafford  |    4 |
| Reorganization  |      20 | New York  |    1 |
| Newbenefits     |      30 | Stafford  |    4 |
+-----------------+---------+-----------+------+
6 rows in set (0.00 sec)
```

Query : UPDATE Dept_locations SET Dlocation = 'Las Vagas' WHERE Dnumber = 5;

New Output :

```
mysql> table Dept_locations;
+---------+-----------+
| Dnumber | Dlocation |
+---------+-----------+
|       1 | New York  |
|       4 | Las Vagas |
|       5 | Bellaire  |
|       5 | Houston   |
|       5 | Sugarland |
+---------+-----------+
5 rows in set (0.01 sec)
```

```
mysql> table Project;
+-----------------+---------+-----------+------+
| Pname           | Pnumber | Plocation | Dnum |
+-----------------+---------+-----------+------+
| ProductX        |       1 | Bellaire  |    5 |
| ProductY        |       2 | Sugarland |    5 |
| ProductZ        |       3 | Houston   |    5 |
| Computerization |      10 | Stafford  |    4 |
| Reorganization  |      20 | New York  |    1 |
| Newbenefits     |      30 | Stafford  |    4 |
+-----------------+---------+-----------+------+
6 rows in set (0.00 sec)
```

3. **Assure at all times that there are no departments with more than 3 projects.**
   Query:
   DELIMITER $$
   CREATE TRIGGER Check_project_cnt BEFORE INSERT ON Project FOR EACH ROW BEGIN IF (SELECT COUNT(*) FROM Project WHERE Dnum = NEW.Dnum) >= 3 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Department cannot have more than 3 projects'; END IF; END$$
   DELIMITER ;

Output:

```
mysql> table Project;
+-----------------+---------+-----------+------+
| Pname           | Pnumber | Plocation | Dnum |
+-----------------+---------+-----------+------+
| ProductX        |       1 | Bellaire  |    5 |
| ProductY        |       2 | Sugarland |    5 |
| ProductZ        |       3 | Houston   |    5 |
| Computerization |      10 | Stafford  |    4 |
| Reorganization  |      20 | New York  |    1 |
| Newbenefits     |      30 | Stafford  |    4 |
+-----------------+---------+-----------+------+
6 rows in set (0.00 sec)

mysql> INSERT INTO Project VALUES ('ProductW',10,'Houstan',5);
ERROR 1644 (45000): Department cannot have more than 3 projects
```

4. **Assure that no employees work for more than one department.**

   Query:

   DELIMITER $$
   CREATE TRIGGER Emp_dept_cnt BEFORE INSERT ON Employee FOR EACH ROW BEGIN IF (SELECT COUNT(*) FROM Employee WHERE Ssn = NEW.Ssn) > 1 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Employee cannot work in more than one department'; END IF; END$$
   DELIMITER ;

   Output:

```
mysql> table Employee;
+----------+-------+---------+-----------+------------+-------------------------+--------+----------+-----------+-----+
| 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 |
| 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 |
+----------+-------+---------+-----------+------------+-------------------------+--------+----------+-----------+-----+
8 rows in set (0.00 sec)

mysql> INSERT INTO Employee VALUES ('Kristin','M','Joe','391391391','1925-08-09','193 Hawk Houstan, TX','F',50000,'9876543
21',4);
ERROR 1644 (45000): Employee cannot have work in more than 1 departments
```

5. **Whenever a project is dropped, dissociate all the employees from the particular project.**

   Query:

   DELIMITER $$
   CREATE TRIGGER Drop_emps_on_proj BEFORE DELETE ON Project FOR EACH ROW BEGIN DELETE FROM Works_on WHERE Pno = OLD.Pnumber; END$$
   DELIMITER ;

Output:

```
mysql> table Project;
+-----------------+---------+-----------+------+
| Pname           | Pnumber | Plocation | Dnum |
+-----------------+---------+-----------+------+
| ProductX        |       1 | Bellaire  |    5 |
| ProductY        |       2 | Sugarland |    5 |
| ProductZ        |       3 | Houston   |    5 |
| Computerization |      10 | Stafford  |    4 |
| Reorganization  |      20 | New York  |    1 |
| Newbenefits     |      30 | Stafford  |    4 |
+-----------------+---------+-----------+------+
6 rows in set (0.00 sec)
```

```
mysql> table Works_on;
+-----------+-----+-------+
| Essn      | Pno | Hours |
+-----------+-----+-------+
| 123456789 |   1 |  32.5 |
| 123456789 |   2 |   7.5 |
| 333445555 |   2 |  10.0 |
| 333445555 |   3 |  10.0 |
| 333445555 |  10 |  10.0 |
| 333445555 |  20 |  10.0 |
| 453453453 |   1 |  20.0 |
| 453453453 |   2 |  20.0 |
| 666884444 |   3 |  40.0 |
| 888665555 |  20 |  15.0 |
| 987654321 |  20 |  15.0 |
| 987654321 |  30 |  20.0 |
| 987987987 |  10 |  35.0 |
| 987987987 |  30 |   5.0 |
| 999887777 |  10 |  10.0 |
| 999887777 |  30 |  30.0 |
+-----------+-----+-------+
16 rows in set (0.00 sec)
```

Query : DELETE FROM Project WHERE Pname = ‘ProductZ’;

```
mysql> table Project;
+-----------------+---------+-----------+------+
| Pname           | Pnumber | Plocation | Dnum |
+-----------------+---------+-----------+------+
| ProductX        |       1 | Bellaire  |    5 |
| ProductY        |       2 | Sugarland |    5 |
| Computerization |      10 | Stafford  |    4 |
| Reorganization  |      20 | New York  |    1 |
| Newbenefits     |      30 | Stafford  |    4 |
+-----------------+---------+-----------+------+
5 rows in set (0.00 sec)
```

```
mysql> table Works_on;
+-----------+-----+-------+
| Essn      | Pno | Hours |
+-----------+-----+-------+
| 123456789 |   1 |  32.5 |
| 123456789 |   2 |   7.5 |
| 333445555 |   2 |  10.0 |
| 333445555 |  10 |  10.0 |
| 333445555 |  20 |  10.0 |
| 453453453 |   1 |  20.0 |
| 453453453 |   2 |  20.0 |
| 888665555 |  20 |  15.0 |
| 987654321 |  20 |  15.0 |
| 987654321 |  30 |  20.0 |
| 987987987 |  10 |  35.0 |
| 987987987 |  30 |   5.0 |
| 999887777 |  10 |  10.0 |
| 999887777 |  30 |  30.0 |
+-----------+-----+-------+
14 rows in set (0.00 sec)
```

6. **When a new department is inaugurated, ensure that it is not co-located with any other departments.**

   Query:

   DELIMITER $$
   CREATE TRIGGER Dept_loc_check BEFORE INSERT ON Dept_locations FOR EACH ROW BEGIN IF EXISTS (SELECT 1 FROM Dept_locations WHERE Dlocation = NEW.Dlocation) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = '2 Departments cant be in same location'; END IF; END$$
   DELIMITER ;

   Output:

```
mysql> table Dept_locations;
+---------+-----------+
| Dnumber | Dlocation |
+---------+-----------+
|       1 | New York  |
|       4 | Las Vagas |
|       5 | Bellaire  |
|       5 | Houston   |
|       5 | Sugarland |
+---------+-----------+
5 rows in set (0.00 sec)

mysql> INSERT INTO Dept_locations VALUES (3,'New York');
ERROR 1644 (45000): 2 Departments cant be in same location
```

7. **For every employee, ensure that his dependent Birthdate is less than his Birthdate.**

   Query:

   DELIMITER $$
   CREATE TRIGGER Bday_check BEFORE INSERT ON Dependent FOR EACH ROW BEGIN IF EXISTS (SELECT 1 FROM Employee WHERE Ssn = NEW.Essn AND Bdate >= NEW.Bdate) THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Employee birthday cant be lower than dependent birthday'; END IF; END$$
   DELIMITER ;

   Output:

```
mysql> table Employee;
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| 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      | 50000.00 | 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)

mysql> table Dependent;
+-----------+----------------+--------+------------+--------------+
| Essn      | Dependent_name | Gender | Bdate      | Relationship |
+-----------+----------------+--------+------------+--------------+
| 123456789 | Alice          | F      | 1988-12-30 | Daughter     |
| 123456789 | Elizabeth      | F      | 1967-05-05 | Spouse       |
| 123456789 | Michae         | M      | 1988-01-04 | Son          |
| 333445555 | Alice          | F      | 1986-04-05 | Daughter     |
| 333445555 | Joy            | F      | 1958-05-03 | Spouse       |
| 333445555 | Theodore       | M      | 1983-10-25 | Son          |
| 987654321 | Abner          | M      | 1942-02-28 | Spouse       |
+-----------+----------------+--------+------------+--------------+
7 rows in set (0.00 sec)

mysql> INSERT INTO Dependent VALUES ('391391391','George','M','1905-04-15','Son');
ERROR 1644 (45000): Employee birthday cant be lower than dependent birthday
```

8. **Increment 1000 rupees to the salary for those employees if any of his/her dependents expire.**

Query:

```
DELIMITER $$
CREATE TRIGGER Give_comp AFTER DELETE ON Dependent FOR EACH ROW
BEGIN UPDATE Employee SET salary = salary + 1000 WHERE Ssn = Old.Essn;
END$$
DELIMITER ;
```

Output:

```
mysql> table Employee;
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| 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      | 50000.00 | 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)

mysql> table Dependent;
+-----------+----------------+--------+------------+--------------+
| Essn      | Dependent_name | Gender | Bdate      | Relationship |
+-----------+----------------+--------+------------+--------------+
| 123456789 | Alice          | F      | 1988-12-30 | Daughter     |
| 123456789 | Elizabeth      | F      | 1967-05-05 | Spouse       |
| 123456789 | Michae         | M      | 1988-01-04 | Son          |
| 333445555 | Alice          | F      | 1986-04-05 | Daughter     |
| 333445555 | Joy            | F      | 1958-05-03 | Spouse       |
| 333445555 | Theodore       | M      | 1983-10-25 | Son          |
| 391391391 | George         | M      | 1905-04-15 | Son          |
| 987654321 | Abner          | M      | 1942-02-28 | Spouse       |
+-----------+----------------+--------+------------+--------------+
8 rows in set (0.00 sec)

mysql> DELETE FROM Dependent WHERE Dependent_name = 'George';
Query OK, 1 row affected (0.01 sec)

mysql> table Employee;
+----------+-------+---------+-----------+------------+--------------------------+--------+----------+-----------+-----+
| 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)
```