

AIRLINE FLIGHT DATABASE:

3.The following relations keep track of airline flight information:

Flights (flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)

Aircraft (aid: integer, aname: string, cruisingrange: integer)

Certified (eid: integer, aid: integer)

Employees (eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft, and only pilots are certified to fly.

For the above schema, perform the following.

- a) Create the above tables by specifying primary keys and foreign keys.
- b) Insert around 10 records in each of the tables.
- c) Find the names of aircraft such that all pilots certified to operate them earn more than 80,000.
- d) For each pilot who is certified for more than three aircraft, find the eid and the maximum cruising range of the aircraft that he (or she) is certified for.
- e). Find the names of pilots whose salary is less than the price of the cheapest route from Los Angeles to Honolulu.
- f) Find the second highest salary of an employee.
- g) Create a stored procedure that remove employees based on employee id.

Solution:

a)

```
SQL>create table flight(fno int primary key, ffrom varchar2(20), fto varchar2(20), distance number(8), departs varchar2(10), arrives varchar2(10), price float);
```

```
SQL>create table aircraft(aid int,aname varchar2(20),crange int,primary key(aid));
```

```
SQL>create table employees(eid int, ename varchar2(20),salary int, primary key(eid));
```

```
SQL>create table certified(eid int,aid int,primary key(eid,aid) , foreign key(eid) references  
employees(eid),foreign key(aid) references aircraft(aid));
```

```
SQL> desc flight;
```

Name	Null?	Type

FNO	NOT NULL	NUMBER(38)
FFROM		VARCHAR2(20)
FTO		VARCHAR2(20)
DISTANCE		NUMBER(8)
DEPARTS		VARCHAR2(10)
ARRIVES		VARCHAR2(10)
PRICE		FLOAT(126)

```
SQL> desc aircraft;
```

Name	Null?	Type

AID	NOT NULL	NUMBER(38)
ANAME		VARCHAR2(20)
CRANGE		NUMBER(38)

```
SQL> desc certified;
```

Name	Null?	Type

EID	NOT NULL	NUMBER(38)

AID NOT NULL NUMBER(38)

SQL> desc employees;

Name	Null?	Type
------	-------	------

EID	NOT NULL	NUMBER(38)
-----	----------	------------

ENAME		VARCHAR2(20)
-------	--	--------------

SALARY		NUMBER(38)
--------	--	------------

b)

INSERT INTO EMPLOYEES VALUES(4214,'AJITH',85000);

INSERT INTO EMPLOYEES VALUES(4215,'JUJARE',35000);

INSERT INTO EMPLOYEES VALUES(4216,'KUMAR',95000);

INSERT INTO AIRCRAFT VALUES(5604,'BOEING111',2600);

INSERT INTO AIRCRAFT VALUES(5605,'BOEING112',2150);

INSERT INTO AIRCRAFT VALUES(5606,'BOEING DAUNTLESS',2500);

INSERT INTO CERTIFIED VALUES(4214,5604);

INSERT INTO CERTIFIED VALUES(4215,5605);

INSERT INTO CERTIFIED VALUES(4216,5606);

INSERT INTO AIRCRAFT VALUES(5608,'BOEING123',2300);

INSERT INTO CERTIFIED VALUES(4214,5605);

INSERT INTO CERTIFIED VALUES(4214,5608);

INSERT INTO CERTIFIED VALUES(4214,5606);

INSERT INTO AIRCRAFT VALUES(9801,'BOEING456',3000);

INSERT INTO CERTIFIED VALUES(4215,9801);

INSERT INTO CERTIFIED VALUES(4214,9801);

```
INSERT INTO FLIGHT VALUES
(8741,'BANGALORE','FRANKFURT',1500,'08:30:00','10:30:00',55000);
```

```
INSERT INTO FLIGHT VALUES(8125,'BANGALORE','NEW
DELHI',5000,'08:30:00','11:30:00',15000);
```

```
INSERT INTO FLIGHT VALUES(8126,'BANGALORE','NEW
DELHI',5000,'09:30:00','12:30:00',16000);
```

```
INSERT INTO FLIGHT VALUES(8127,'BANGALORE','NEW
DELHI',5000,'10:30:00','13:30:00',17000);
```

```
INSERT INTO FLIGHT VALUES(8128,'BANGALORE','NEW
DELHI',2900,'10:30:00','13:30:00',17000);
```

```
SQL> select * from flight;
```

FNO	FFROM	FTO	DISTANCE	DEPARTS	ARRIVES	PRICE
8741	BANGALORE	FRANKFURT	1500	08:30:00	10:30:00	55000
8125	BANGALORE	NEW DELHI	5000	08:30:00	11:30:00	15000
8126	BANGALORE	NEW DELHI	5000	09:30:00	12:30:00	16000
8127	BANGALORE	NEW DELHI	5000	10:30:00	13:30:00	17000
8128	BANGALORE	NEW DELHI	2900	10:30:00	13:30:00	17000

```
SQL> SELECT * FROM AIRCRAFT;
```

AID	ANAME	CRANGE
-----	-------	--------

5604 BOEING111	2600
5605 BOEING112	2150
5606 BOEING DAUNTLESS	2500
5608 BOEING123	2300
9801 BOEING456	3000

SQL> SELECT * FROM CERTIFIED;

EID	AID
4214	5604
4215	5605
4216	5606
4214	5605
4214	5608
4214	5606
4215	9801
4214	9801

8 rows selected.

SQL> SELECT * FROM EMPLOYEES;

EID	ENAME	SALARY
4214	AJITH	85000
4215	JUJARE	35000
4216	KUMAR	95000

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

```
SELECT ANAME
FROM AIRCRAFT A, EMPLOYEES E, CERTIFIED C
WHERE C.EID=E.EID AND C.AID=A.AID AND E.SALARY>80000;
```

D) For each pilot who is certified for more than three aircraft, find the eid and the maximum cruising range of the aircraft that he (or she) is certified for.

```
SELECT C.EID, MAX(A.CRANGE)
FROM EMPLOYEES E,CERTIFIED C,AIRCRAFT A
WHERE E.EID=C.EID AND C.AID=A.AID
GROUP BY C.EID

HAVING COUNT(*)>3;
```

E) Find the names of pilots whose salary is less than the price of the cheapest route from Los Angeles to Honolulu.

```
SELECT DISTINCT E.ENAME
FROM EMPLOYEES E,CERTIFIED C
WHERE E.EID=C.EID AND E.SALARY<
      (SELECT MIN(PRICE) FROM FLIGHT F WHERE
      F.FFROM='BANGALORE' AND F.FTO='FRANKFURT');
```

F) Find the second highest salary of an employee.

```
SELECT MAX(SALARY)
FROM EMPLOYEES
WHERE SALARY NOT IN ( SELECT MAX(SALARY) FROM EMPLOYEES)
```

OR

```
SELECT MAX(SALARY)
FROM EMPLOYEES
WHERE SALARY <> ( SELECT MAX(SALARY) FROM EMPLOYEES)
```

FOR nth highest salary

```
SELECT *
FROM EMPLOYEES EMP1
WHERE (N-1) = ( SELECT COUNT(DISTINCT(EMP2.SALARY))
FROM EMPLOYEES EMP2
WHERE EMP2.SALARY > EMP1.SALARY)
```

G) Create a stored procedure that remove employees based on employee id.

```
SQL>CREATE OR REPLACE PROCEDURE SP_EMPLOYEE_DELETE_BYEID
(
    p_EID_ID IN EMPLOYEES.EID%TYPE
)
AS
BEGIN
DELETE
    FROM EMPLOYEES
    WHERE EID = p_EID_ID ;
COMMIT ;
EXCEPTION
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR (-20001, p_EID_ID || ':$:' || SQLERRM, TRUE) ;
END SP_EMPLOYEE_DELETE_BYEID ;
/
```

to execute the above procedure

```
SQL> BEGIN
SP_EMPLOYEE_DELETE_BYEID(4214);
END
/
```

PL SQL procedure executed successfully.

if exists a child record in CERTIFIED table,make sure that 'on delete cascade' constraint is applied first. If not alter the table by adding on delete cascade to the certified relation.

```
SQL> ALTER TABLE CERTIFIED MODIFY FOREIGN KEY (EID) REFERENCES
EMPLOYEES(EID) ON DELETE CASCADE;
```