

SP_Nov_25

Consider the following schema of a Boat Reservation System relational database:

- Sailors (sid: integer, sname: string, rating: integer, age: integer, phno: integer)
- Boats (bid: integer, bname: string, color: string, type: string)
- Reserves (sid: integer, bid: integer, day: date)

Constraints on the Sailors table:

- sid is the primary key
- sname, rating, age cannot be blank
- rating lies between 1 and 10
- age lies between 20 and 50
- appropriate size should be chosen for the attribute phno

Constraints on the Boats table:

- bid is the primary key
- bname, color, type cannot be empty

Constraints on the Reserves table:

- sid, bid, day together make up the primary key
- sid is the foreign key which depends on sid of Sailors
- bid is the foreign key which depends on bid of Boats

Questions:

PL/SQL

Problem 1:

As a DBA, you have been tasked with segregating the employee details department-wise. This will enable department heads to manage their personnel better. The master table, containing details of all employees from all departments, is provided to you.

Question:

Can this problem be solved using only SQL?

```
DROP TABLE EMPLOYEE;
```

```
CREATE TABLE EMPLOYEE(EID INTEGER PRIMARY KEY, ENAME VARCHAR(100) NOT NULL,  
SALARY INTEGER NOT NULL, DNAME VARCHAR(20) NOT NULL);
```

```
INSERT INTO employee VALUES(1, 'ABC', 4500, 'Marketing');  
INSERT INTO employee VALUES(2, 'ABC', 3500, 'Marketing');  
INSERT INTO employee VALUES(3, 'XYZ', 7500, 'Analytics');  
INSERT INTO employee VALUES(4, 'DEF', 2500, 'Finance');  
INSERT INTO employee VALUES(5, 'PQR', 5000, 'Marketing');  
INSERT INTO employee VALUES(6, 'GHI', 8000, 'Analytics');  
INSERT INTO employee VALUES(7, 'FFF', 6000, 'Finance');  
INSERT INTO employee VALUES(8, 'DWT', 5200, 'Finance');  
INSERT INTO employee VALUES(9, 'FFT', 2000, 'Operations');  
INSERT INTO employee VALUES(10, 'DCT', 3700, 'Analytics');
```

```
SELECT DISTINCT (dname) FROM employee;
```

```
DROP TABLE marketing;
```

```
DROP TABLE finance;
```

```
DROP TABLE analytics;
```

```
DROP TABLE operations;
```

```
CREATE TABLE marketing(eid INTEGER PRIMARY KEY, ename VARCHAR(100) NOT NULL,  
salary INTEGER NOT NULL);
```

```
CREATE TABLE finance(eid INTEGER PRIMARY KEY, ename VARCHAR(100) NOT NULL,  
salary INTEGER NOT NULL);
```

```
CREATE TABLE analytics(eid INTEGER PRIMARY KEY, ename VARCHAR(100) NOT NULL,  
salary INTEGER NOT NULL);
```

```
CREATE TABLE operations(eid INTEGER PRIMARY KEY, ename VARCHAR(100) NOT NULL,  
salary INTEGER NOT NULL);
```

```
DECLARE
```

```
    NOOFROWS INTEGER;
```

```
    DEPTNAME VARCHAR(20);
```

```
    LOOPVAR   INTEGER := 1;
```

```
    EMPNAME   VARCHAR(100);
```

```
    SAL       INTEGER;
```

```
BEGIN
```

```

SELECT
    COUNT(*) INTO NOOFROWS
FROM
    EMPLOYEE;
WHILE LOOPVAR <= NOOFROWS LOOP
    SELECT
        DNAME INTO DEPTNAME
    FROM
        EMPLOYEE
    WHERE
        EID = LOOPVAR;
    SELECT
        ENAME INTO EMPNAME
    FROM
        EMPLOYEE
    WHERE
        EID = LOOPVAR;
    SELECT
        SALARY INTO SAL
    FROM
        EMPLOYEE
    WHERE
        EID = LOOPVAR;
    IF DEPTNAME = 'Marketing' THEN
        INSERT INTO MARKETING VALUES(
            LOOPVAR,
            EMPNAME,
            SAL
        );
    ELSIF DEPTNAME = 'Finance' THEN
        INSERT INTO FINANCE VALUES(
            LOOPVAR,
            EMPNAME,
            SAL
        );
    ELSIF DEPTNAME = 'Analytics' THEN
        INSERT INTO ANALYTICS VALUES(
            LOOPVAR,
            EMPNAME,
            SAL
        );
    ELSIF DEPTNAME = 'Operations' THEN
        INSERT INTO OPERATIONS VALUES(
            LOOPVAR,
            EMPNAME,
            SAL
        );
    END IF;
    LOOPVAR := LOOPVAR + 1;

```

```
END LOOP;  
END;  
/
```

To check:

```
SELECT * FROM marketing;  
SELECT * FROM finance;  
SELECT * FROM analytics;  
SELECT * FROM operations;
```

Problem 2:

As a DBA, you have been tasked with segregating the employee details salary-wise. This will enable the organisation to manage the payroll system better, Employees earning less than 5000 should be placed in one table, while the remaining in the second table. The master table, containing details of all employees from all departments, is provided to you.