
Example No 2a : **College :**

SNO (Primary key) Number(4)	SNAME Varchar2(20)	Year Char(15)	DOA (Date of Admission) Date	PCM Marks (PCM) Number(3)	DEPTNO Number(2)
7369	Adams	First	01-jun-08	275	20
7499	Brooks	Second	01-Jun-07	265	10
7521	Curry	Second	01-Jun-07	255	30
7566	Glenn	Third	01-jun-06	215	30
7654	Green	Second	01-Jun-07	220	30
7698	Hayes	Third	01-jun-06	235	30
7782	Johnson	Third	01-jun-06	290	10
7783	Jones	Third	01-jun-06	189	20
7839	Lindsay	Final	01-jun-05	175	10
7844	Smith	Second	01-Jun-07	263	30
7876	Turner	First	01-jun-08	285	20
7900	Williams	First	01-jun-08	275	30
7902	Adams	Third	01-Jun-06	274	20
7934	Brooks	First	01-jun-08	167	10

Department: -

DEPTNO Primary key Number(2)	DNAME Varchar2(20)
10	Computer
20	Electronics
30	Civil
40	Chemical

Write SQL statement with Output for the following:

1. **Create College and Department table with above mentioned schema.**

➢ CREATE TABLE Department(
-> DEPTNO INT(2) PRIMARY KEY,
-> DNAME VARCHAR(20)
->);

CREATE TABLE College(
-> SNO INT(4) PRIMARY KEY,
-> SNAME VARCHAR(20),
-> Year CHAR(15),
-> DOA DATE,
-> PCM INT(3),
-> DEPTNO INT(2),
-> FOREIGN KEY (DEPTNO) REFERENCES Department(DEPTNO)
->);

2. **Insert all tuples in both the tables.**

➢ INSERT INTO Department VALUES (10, 'Computer'), (20, 'Electronics'), (30, 'Civil'), (40, 'Chemical');

```
INSERT INTO College VALUES
  -> (7369, 'Adams', 'First', '2008-06-01', 275, 20),
  -> (7499, 'Brooks', 'Second', '2007-06-01', 265, 10),
  -> (7521, 'Curry', 'Second', '2007-06-01', 255, 30),
  -> (7566, 'Glenn', 'Third', '2006-06-01', 215, 30),
  -> (7654, 'Green', 'Second', '2007-06-01', 220, 30),
  -> (7698, 'Hayes', 'Third', '2006-06-01', 235, 30),
  -> (7782, 'Johnson', 'Third', '2006-06-01', 290, 10),
  -> (7783, 'Jones', 'Third', '2006-06-01', 189, 20),
  -> (7839, 'Lindsay', 'Final', '2005-06-01', 175, 10),
  -> (7844, 'Smith', 'Second', '2007-06-01', 263, 30),
  -> (7876, 'Turner', 'First', '2008-06-01', 285, 20),
  -> (7900, 'Williams', 'First', '2008-06-01', 275, 30),
  -> (7902, 'Adams', 'Third', '2006-06-01', 274, 20),
  -> (7934, 'Brooks', 'First', '2008-06-01', 167, 10);
```

3. Add new attribute Sports Marks (SM) in college table with number (2).

- ALTER TABLE College ADD SM INT(2);

4. Modify the SM attribute value by 15 to students related to department 10.

- UPDATE College SET SM = 15
  -> WHERE DEPTNO = 10;

5. List the Student names, which are eligible for Sports marks.

- SELECT SNAME FROM College
  -> WHERE SM is NOT NULL;

6. Display only names of students (first letter of name in capital only).

- SELECT SNAME FROM College WHERE SNAME = CONCAT(UCASE(LEFT(SNAME, 1)), SUBSTRING(SNAME, 2));

7. List the name, PCM marks, and percentage of all students.

- SELECT Sname, PCM, (PCM / 3) AS Percentage FROM College;

8. List the Different years in college table.

- SELECT DISTINCT Year FROM College;

9. List the player's details not belonging to the teamid 20, 30 40.

- SELECT * FROM College
  -> WHERE DEPTNO NOT IN (20, 30, 40);

10. List the Students Name and PCM marks, whose PCM are in the range of 250 and 300.

- SELECT SNAME, PCM FROM College
  WHERE PCM BETWEEN 250 AND 300;

11. List the total pcm, highest pcm and average pcm of students deptno wise for the department no 30 and display only those rows having average pcm grater than 250 and arrange the result in Descending order of the total pcm.

SELECT DEPTNO, SUM(PCM) AS Total_Pcm, MAX(PCM) AS Highest_PCM, AVG(PCM) AS AVG_PCM FROM College
  -> WHERE DEPTNO = 30
  -> GROUP BY DEPTNO

```
-> HAVING AVG_PCM > 250
-> ORDER BY Total_Pcm DESC;
```

12. Display the list of Students in each department. Display the department information even if no students is belong to that department.

```
SELECT D.DEPTNO, D.Dname, C.Sname
-> FROM Department D
-> LEFT JOIN College C ON D.DEPTNO = C.DEPTNO;
```

13. Display the names of student having pcm marks same as that of student 'Williams'.

```
SELECT SNAME FROM College
-> WHERE PCM = (SELECT PCM FROM College WHERE SName = 'Williams');
```

14. Write PL SQL program to find total no of students in the student table.

```
SELECT COUNT(*) AS TotalStudents FROM College;
CREATE PROCEDURE CountStudents()
BEGIN
   SELECT COUNT(*) AS TotalStudents FROM College;
END;
CALL CountStudents();
```

15. Display the different year present in department no 10 and 20.

```
SELECT DISTINCT Year FROM College
-> WHERE DEPTNO IN (10, 20);
```