

**Dept of Computer Engg. Sub : DBMS Practical & Oral Exam – April – 2022**

Example No: 4a Employee Table (Data)

EMPNO (Primary key) Number(4)	ENAME Varchar2(20)	JOB Varchar2(20)	MGR Number(4)	HIREDATE Date	SAL Number(9,2)	COMM Number(9,2)	DEPT NO Number(2)
7369	Smith	Clerk	7902	17-DEC-80	800	-	20
7499	Allen	Salesman	7698	20-FEB-81	1600	-	30
7521	Ward	Salesman	7698	22-FEB-81	1250	300	30
7566	Jones	Manager	7839	02-APR-81	2975	500	20
7654	Martin	Salesman	7698	28-SEP-81	1250	-	30
7698	Blake	Manager	7839	01-MAY-81	2850	1400	30
7782	Clark	Manager	7839	09-JUN-81	2450	-	10
7783	Scott	Analyst	7566	09-DEC-82	3000	-	20
7839	King	President	-	17-NOV-81	5000	-	10
7844	Turner	Salesman	7698	08-SEP-81	1500	0	30
7876	Adams	Clerk	7788	12-JAN-83	1100		20
7900	James	Clerk	7698	03-DEC-81	950		30
7902	Ford	Analyst	7566	04-DEC-81	3000		20
7934	Miller	Clerk	7782	23-JAN-82	1300		10

Department-TABLE (Data)

DEPT NO Primary key Number(2)	DNAME Varchar2(20)	LOC Varchar2(10)
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

**Write SQL statement with Output for the following:**

- 1. Create Employee and department table with above mentioned schema.**

```
CREATE TABLE Department (  
    DEPTNO INT(2) PRIMARY KEY,  
    DNAME VARCHAR(20),  
    LOC VARCHAR(10)  
);
```

```
CREATE TABLE Employee (  
    EMPNO INT(4) PRIMARY KEY,  
    ENAME VARCHAR(20),  
    JOB VARCHAR(20),  
    MGR INT(4),  
    HIREDATE DATE,  
    SAL DECIMAL(9,2),  
    COMM DECIMAL(9,2),  
    DEPTNO INT(2),  
    FOREIGN KEY (DEPTNO) REFERENCES Department(DEPTNO)  
);
```

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

```
INSERT INTO Department VALUES
(10, 'Accounting', 'New York'),
(20, 'Research', 'Dallas'),
(30, 'Sales', 'Chicago'),
(40, 'Operations', 'Boston');
```

```
INSERT INTO Employee VALUES
(7369, 'Smith', 'Clerk', 7902, '1980-12-17', 800, NULL, 20),
(7499, 'Allen', 'Salesman', 7698, '1981-02-20', 1600, NULL, 30),
(7521, 'Ward', 'Salesman', 7698, '1981-02-22', 1250, 300, 30),
(7566, 'Jones', 'Manager', 7839, '1981-04-02', 2975, 500, 20),
(7654, 'Martin', 'Salesman', 7698, '1981-09-28', 1250, NULL, 30),
(7698, 'Blake', 'Manager', 7839, '1981-05-01', 2850, 1400, 30),
(7782, 'Clark', 'Manager', 7839, '1981-06-09', 2450, NULL, 10),
(7783, 'Scott', 'Analyst', 7566, '1982-12-09', 3000, NULL, 20),
(7839, 'King', 'President', NULL, '1981-11-17', 5000, NULL, 10),
(7844, 'Turner', 'Salesman', 7698, '1981-09-08', 1500, NULL, 30),
(7876, 'Adams', 'Clerk', 7788, '1983-01-12', 1100, NULL, 20),
(7900, 'James', 'Clerk', 7698, '1981-12-03', 950, NULL, 30),
(7902, 'Ford', 'Analyst', 7566, '1981-12-04', 3000, NULL, 20),
(7934, 'Miller', 'Clerk', 7782, '1982-01-23', 1300, NULL, 10);
```

**3. Add new attribute company in the employee table with char (4).**

```
ALTER TABLE Employee ADD company CHAR(4);
```

**4. List the employee name, who are not eligible for commission.**

```
SELECT ENAME FROM Employee WHERE COMM IS NULL;
```

**5. Display names of employees in lower and upper case.**

```
SELECT LOWER(ENAME) AS LowerName, UPPER(ENAME) AS UpperName FROM
Employee;
```

**6. List the name, salary, and PF amount of all employees. (Pf is 10% of salary)**

```
SELECT ENAME, SAL, (SAL * 0.1) AS PF FROM Employee;
```

**7. List the Different designations available in the employee table.**

```
SELECT DISTINCT JOB FROM Employee;
```

**8. List the Employee details not belonging to the department 10, 30, 40.**

```
SELECT * FROM Employee
WHERE DEPTNO NOT IN (10, 30, 40);
```

**9. List the Employee Name and Salary, whose salary is in the range 1000 and 2000.**

```
SELECT ENAME, SAL FROM Employee
WHERE SAL BETWEEN 1000 AND 2000;
```

**10. List the total salary, highest salary and average salary of employees job wise for the department no 20 and display only those rows having average salary greater than 1500 and arrange the result in descending order of total salary.**

```
SELECT JOB,
SUM(SAL) AS TotalSalary,
MAX(SAL) AS HighestSalary,
```

```

    AVG(SAL) AS AverageSalary
FROM Employee
WHERE DEPTNO = 20
GROUP BY JOB
HAVING AVG(SAL) > 1500
ORDER BY TotalSalary DESC;\

```

- 11. Display the list of employees working in each department. Display the department information even if no employees belong to that department.**

```

SELECT d.DEPTNO, d.DNAME, e.ENAME
FROM Department d
LEFT JOIN Employee e ON d.DEPTNO = e.DEPTNO;

```

- 12. Display the name of employee having salary same as that of employee FORD.**

```

SELECT ENAME FROM Employee
WHERE SAL = (SELECT SAL FROM Employee WHERE ENAME = 'Ford');

```

- 13. List the names of employee along with Manager Name, who joined company before their manager.**

```

SELECT
    e.ENAME AS Employee_Name,
    m.ENAME AS Manager_Name
FROM
    Employee e
JOIN
    Employee m ON e.MGR = m.EMPNO
WHERE
    e.HIREDATE < m.HIREDATE;

```

- 14. Write PL SQL program to find factorial of 5.**

```

CREATE PROCEDURE factorial_5()
BEGIN
    DECLARE i INT DEFAULT 1;
    DECLARE fact INT DEFAULT 1;

    WHILE i <= 5 DO
        SET fact = fact * i;
        SET i = i + 1;
    END WHILE;

    SELECT fact AS factorial_of_5;
END;

```

- 15. Increase the salary of employees working as a Manager by 5 %.**

```

UPDATE Employee
SET SAL = SAL + SAL * 0.05
WHERE JOB = 'Manager';

```