

## Nested Queries and Join

### The data contained in the EMP and DEPT tables

The data in the EMP table contains the following 14 rows:

EMPNO	ENAME	JOB	HIREDATE	MGR	SAL	COMM	DEPTNO
7369	SMITH	CLERK	17-DEC-80	7902	800		20
7499	ALLEN	SALESMAN	20-FEB-81	7698	1600	300	30
7521	WARD	SALESMAN	22-FEB-81	7698	1250	500	30
7566	JONES	MANAGER	02-APR-81	7839	2975		20
7654	MARTIN	SALESMAN	28-SEP-81	7698	1250	1400	30
7698	BLAKE	MANAGER	01-MAY-81	7839	2850		30
7782	CLARK	MANAGER	09-JUN-81	7839	2450		10
7788	SCOTT	ANALYST	19-APR-87	7566	3000		20
7839	KING	PRESIDENT	17-NOV-81		5000		10
7844	TURNER	SALESMAN	08-SEP-81	7698	1500	0	30
7876	ADAMS	CLERK	23-MAY-87	7788	1100		20
7900	JAMES	CLERK	03-DEC-81	7698	950		30
7902	FORD	ANALYST	03-DEC-81	7566	3000		20
7934	MILLER	CLERK	23-JAN-82	7782	1300		10

```
create table dept_0096(  
  deptno number(2),  
  dname varchar2(30),  
  loc varchar2(20)  
);
```

```
INSERT INTO DEPT_0096 (DEPTNO, DNAME, LOC) VALUES (10, 'ACCOUNTING', 'NEW YORK');  
INSERT INTO DEPT_0096 (DEPTNO, DNAME, LOC) VALUES (20, 'RESEARCH', 'DALLAS');  
INSERT INTO DEPT_0096 (DEPTNO, DNAME, LOC) VALUES (30, 'SALES', 'CHICAGO');  
INSERT INTO DEPT_0096 (DEPTNO, DNAME, LOC) VALUES (40, 'OPERATIONS', 'BOSTON');
```

```
alter table dept_0096  
add primary key(deptno);
```

```
create table emp_0096(  
  empno number(4),  
  ename varchar2(30),  
  job varchar2(20),  
  hiredate date,
```

## Nested Queries and Join

```
mgr number(4),  
sal number(7,2),  
comm number(7,2),  
deptno number(2) );
```

```
alter table emp_0096
```

```
add primary key(empno);
```

```
alter table emp_0096
```

```
add foreign key(deptno) references dept_0096(deptno);
```

```
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7369, 'SMITH', 'CLERK', TO_DATE('17-DEC-1980', 'DD-MON-YYYY'), 7902, 800, NULL, 20);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7499, 'ALLEN', 'SALESMAN', TO_DATE('20-FEB-1981', 'DD-MON-YYYY'), 7698, 1600, 300, 30);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7521, 'WARD', 'SALESMAN', TO_DATE('22-FEB-1981', 'DD-MON-YYYY'), 7698, 1250, 500, 30);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7566, 'JONES', 'MANAGER', TO_DATE('02-APR-1981', 'DD-MON-YYYY'), 7839, 2975, NULL, 20);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7654, 'MARTIN', 'SALESMAN', TO_DATE('28-SEP-1981', 'DD-MON-YYYY'), 7698, 1250, 1400, 30);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7698, 'BLAKE', 'MANAGER', TO_DATE('01-MAY-1981', 'DD-MON-YYYY'), 7839, 2850, NULL, 30);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7782, 'CLARK', 'MANAGER', TO_DATE('09-JUN-1981', 'DD-MON-YYYY'), 7839, 2450, NULL, 10);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7788, 'SCOTT', 'ANALYST', TO_DATE('19-APR-1987', 'DD-MON-YYYY'), 7566, 3000, NULL, 20);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7839, 'KING', 'PRESIDENT', TO_DATE('17-NOV-1981', 'DD-MON-YYYY'), NULL, 5000, NULL, 10);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7844, 'TURNER', 'SALESMAN', TO_DATE('08-SEP-1981', 'DD-MON-YYYY'), 7698, 1500, 0, 30);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7876, 'ADAMS', 'CLERK', TO_DATE('23-MAY-1987', 'DD-MON-YYYY'), 7788, 1100, NULL, 20);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7900, 'JAMES', 'CLERK', TO_DATE('03-DEC-1981', 'DD-MON-YYYY'), 7698, 950, NULL, 30);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7902, 'FORD', 'ANALYST', TO_DATE('03-DEC-1981', 'DD-MON-YYYY'), 7566, 3000, NULL, 20);  
INSERT INTO EMP_0096 (EMPNO, ENAME, JOB, HIREDATE, MGR, SAL, COMM, DEPTNO) VALUES (7934, 'MILLER', 'CLERK', TO_DATE('23-JAN-1982', 'DD-MON-YYYY'), 7782, 1300, NULL, 10);
```

Figure 3.1

The DEPT table contains the following four rows:

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Problem 4.1: Select all employees from 'maintenance' and 'development' dept.

```
SELECT * FROM EMP WHERE DEPTNO IN (SELECT DEPTNO FROM DEPT WHERE  
DNAME='MAINTAINANCE' OR DNAME='DEVELOPMENT')
```

Problem 4.2: Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with 'M'.

```
SELECT ENAME, SAL
```

## Nested Queries and Join

```
FROM EMP
WHERE SAL > (SELECT MIN(SAL) FROM EMP)
AND JOB LIKE 'M%';
```

Problem 4.3: Issue a query to find all the employees who work in the same job as jones.

```
SELECT *
FROM EMP
WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME = 'JONES');
```

Problem 4.4: Issue a query to display information about employees who earn more than any employee in dept 30.

```
SELECT *
FROM EMP
WHERE SAL > ANY (
    SELECT SAL
    FROM EMP
    WHERE DEPTNO = 30
);
```

Problem 4.5: Display the employees who have the same job as jones and whose salary >= fords

```
SELECT *
FROM EMP
WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME = 'JONES')
AND SAL >= (SELECT SAL FROM EMP WHERE ENAME = 'FORD');
```

**Problem 4.6: Write a query to display the name and job of all employees in Management dept.**

```
SELECT E.ENAME, E.JOB
FROM EMP E
JOIN DEPT D ON E.DEPTNO = D.DEPTNO
WHERE D.DNAME = 'MANAGEMENT';
```

Problem 4.7: Issue a query to list all the employees who salary is > the average salary of their own dept.

```
SELECT *
FROM EMP E
WHERE SAL > (SELECT AVG(SAL)
    FROM EMP
    WHERE DEPTNO = E.DEPTNO);
```

**Problem 4.8: Write a query that would display the empname, job, location and the name of their dept.**

```
SELECT E.ENAME, E.JOB, D.LOC, D.DNAME
FROM EMP E
JOIN DEPT D ON E.DEPTNO = D.DEPTNO;
```

## Nested Queries and Join

Problem 4.9: Write a query to list the employees having the same job as employees located in 'mainblock'. (use multiple subquery)

```
SELECT *
FROM EMP
WHERE JOB IN (
    SELECT JOB
    FROM EMP E1
    WHERE DEPTNO IN (
        SELECT DEPTNO
        FROM DEPT
        WHERE LOC = 'MAINBLOCK'
    )
);
```

Problem 4.10: Write a query to list the employees in dept 10 with the same job as anyone in the development dept.

```
SELECT *
FROM EMP
WHERE DEPTNO = 10
AND JOB IN (
    SELECT JOB
    FROM EMP
    WHERE DEPTNO = (SELECT DEPTNO FROM DEPT WHERE DNAME = 'DEVELOPMENT')
);
```

Problem 4.11: Write a query to list the employees with the same job and salary as 'ford'.

```
SELECT *
FROM EMP
WHERE JOB = (SELECT JOB FROM EMP WHERE ENAME = 'FORD')
AND SAL = (SELECT SAL FROM EMP WHERE ENAME = 'FORD');
```

Problem 4.12: Write a query to list all depts. with at least 2 salesman.

```
SELECT DEPTNO
FROM EMP
WHERE JOB = 'SALESMAN'
GROUP BY DEPTNO
HAVING COUNT(*) >= 2;
```

Problem 4.13: Write a query to list the employees in dept 20 with the same job as anyone in dept 30.

```
SELECT *
FROM EMP
WHERE DEPTNO = 20
AND JOB IN (SELECT JOB FROM EMP WHERE DEPTNO = 30);
```

Problem 4.14: List out the employee names who get the salary greater than the maximum salaries of dept with dept no 20,30

```
SELECT *
FROM EMP
WHERE SAL > (SELECT MAX(SAL)
    FROM EMP
    WHERE DEPTNO IN (20, 30));
```

## Nested Queries and Join

Problem 4.15: Display the maximum salaries of the departments whose maximum salary is greater than 9000.

```
SELECT DEPTNO, MAX(SAL)
FROM EMP
GROUP BY DEPTNO
HAVING MAX(SAL) > 9000;
```

Problem 4.16: Display the maximum salaries of the departments whose minimum salary is greater than 1000 and lesser than 5000.

```
SELECT DEPTNO, MAX(SAL)
FROM EMP
GROUP BY DEPTNO
HAVING MIN(SAL) > 1000 AND MIN(SAL) < 5000;
```

**For exercises 4.17 and 4.18 Create one table named Accredited with columns deptno (foreign key of department table), Rank varchar(20)**

```
CREATE TABLE ACCREDIT_0096 (
    DEPTNO NUMBER(2),
    RANK VARCHAR2(20),
    CONSTRAINT FOREIGN KEY (DEPTNO) REFERENCES DEPT_0096 (DEPTNO)
);
```

```
INSERT INTO ACCREDIT_0096 (DEPTNO, RANK) VALUES (10, 'Gold');
INSERT INTO ACCREDIT_0096 (DEPTNO, RANK) VALUES (20, 'Silver');
```

EQUI-JOIN

~~~~~

**Problem 4.17: Display the department names that are accredited by the quality council.**

```
SELECT D.DNAME
FROM DEPT D
JOIN ACCREDIT A ON D.DEPTNO = A.DEPTNO;
```

NON-EQUIJOIN

~~~~~

Problem 4.18: Display the employees of departments which are not accredited by the quality council

```
SELECT E.*
FROM EMP E
WHERE E.DEPTNO NOT IN (SELECT DEPTNO FROM ACCREDIT);
```

LEFTOUT-JOIN

~~~~~

Problem 4.19:        Display all        the employees and the departments        implementing a left

## Nested Queries and Join

outer join.

```
SELECT E.*, D.*
FROM EMP E
LEFT OUTER JOIN DEPT D ON E.DEPTNO = D.DEPTNO;
```

RIGHT OUTER JOIN

~~~~~

Problem 4.20: Display the employee name and department name in which they are working implementing a right outer join.

```
SELECT E.ENAME, D.DNAME
FROM EMP E
RIGHT OUTER JOIN DEPT D ON E.DEPTNO = D.DEPTNO;
```

FULL OUTER JOIN

~~~~~

Problem 4.21: Display the employee name and department name in which they are working implementing a full outer join.

```
SELECT E.ENAME, D.DNAME
FROM EMP E
FULL OUTER JOIN DEPT D ON E.DEPTNO = D.DEPTNO;
```

SELF JOIN

~~~~~

Problem 4.22: Write a query to display their employee names and their managers name.

```
SELECT E.ENAME AS EMPLOYEE, M.ENAME AS MANAGER
FROM EMP E
LEFT JOIN EMP M ON E.MGR = M.EMPNO;
```

Problem 4.23: Write a query to display their employee names and their managers salary for every employee

```
SELECT E.ENAME AS EMPLOYEE, M.SAL AS MANAGER_SALARY
FROM EMP E
LEFT JOIN EMP M ON E.MGR = M.EMPNO;
```

Problem 4.24: Write a query to output the name, job, empno, deptname and location for each dept, even if there are no employees.

```
SELECT E.ENAME, E.JOB, E.EMPNO, D.DNAME, D.LOC
FROM DEPT D
LEFT JOIN EMP E ON D.DEPTNO = E.DEPTNO;
```

Problem 4.25: Find the name of the manager for each employee. Include the following in the output: empno, empname, job and his manager's name.

```
SELECT E.EMPNO, E.ENAME AS EMPLOYEE, E.JOB, M.ENAME AS MANAGER
FROM EMP E
LEFT JOIN EMP M ON E.MGR = M.EMPNO;
```

Problem 4.26: Display the details of those who draw the same salary.

```
SELECT *
FROM EMP
WHERE SAL IN (SELECT SAL
```

```
FROM EMP  
GROUP BY SAL  
HAVING COUNT(*) > 1);
```