WEEK 3: READING MATERIAL 4

Self-Join

To join a table to itself means that each row of the table is combined with itself and with every other row of the table. The self-join can be viewed as a join of two copies of the same table. The table is not actually copied, but SQL performs the command as though it were.

The syntax of the command for joining a table to itself is almost the same as that for joining two different tables. To distinguish the column names from one another, aliases for the actual the table name are used, since both the tables have the same name. Table name aliases are defined in the FROM clause of the query. To define the alias, one space is left after the table name and the alias.

Example:

EMP TABLE

EMPNO	**ENAME**	**MGR**
7839	KING	
7566	JONES	7839
7876	ADAMS	7788
7934	MILLER	7782
	………..	………….

Consider the emp table shown above. Primary key of the emp table is empno. Details of each employee’s manager is just another row in the EMP table whose EMPNO is stored in MGR column of some other row. So every employee except manager has a Manager. Therefore MGR is a foreign key that references empno. To list out the names of the manager with the employee record one will have to join EMP itself.

```
SQL>SELECT WORKER. Ename “Ename”, MANAGER.ename “Manager”
        FROM emp WORKER, emp MANAGER
        WHERE WORKER.mgr=MANAGER.empno;
```

Where WORKER and MANAGER are two aliases for the EMP table and acts as a virtual tables.

WORKER

EMPNO	ENAME	MGR
7369	SMITH	7902
7499	ALLEN	7698
7521	WARD	7698
7566	JONES	7839
7654	MARTIN	7698
7698	BLAKE	7839
7782	CLARK	7839
7788	SCOTT	7566
7839	KING	
7844	TURNER	7698
7876	ADAMS	7788
7900	JAMES	7698
7902	FORD	7566
7934	MILLER	7782

MANAGER

EMPNO	ENAME	MGR
7369	SMITH	7902
7499	ALLEN	7698
7521	WARD	7698
7566	JONES	7839
7654	MARTIN	7698
7698	BLAKE	7839
7782	CLARK	7839
7788	SCOTT	7566
7839	KING	
7844	TURNER	7698
7876	ADAMS	7788
7900	JAMES	7698
7902	FORD	7566
7934	MILLER	7782

OUTPUT:

Ename	Manager
SMITH	FORD
ALLEN	BLAKE
WARD	BLAKE
JONES	KING
MARTIN	BLAKE
BLAKE	KING
CLARK	KING
SCOTT	JONES
TURNER	BLAKE
ADAMS	SCOTT

JAMES	BLAKE
FORD	JONES
MILLER	CLARK

13 rows selected.

- List all employees who joined the company before their manager.

```
SQL>Select e.ename, e.hiredate, m.ename manager, m.hiredate FROM emp e, emp m
    WHERE e.mgr= m.empno
     and e.hiredate<m.hiredate;
```

OUTPUT:

ENAME	HIREDATE	MANAGER	HIREDATE
SMITH	17-DEC-80	FORD	03-DEC-81
ALLEN	20-FEB-81	BLAKE	01-MAY-81
WARD	22-FEB-81	BLAKE	01-MAY-81
JONES	02-APR-81	KING	17-NOV-81
BLAKE	01-MAY-81	KING	17-NOV-81
CLARK	09-JUN-81	KING	17-NOV-81

6 rows selected.

Note: If we wish to include those employees name who has no corresponding manager also in above list. Then it becomes the case of self join and outer join. In this scenario, we wish to list all the employees whether it has manager or not. So, worker table, i.e., left table appears full and (+) will appear on manager side so it becomes the case of right outer join. And corresponding query has been shown below.

```
SQL>SELECT WORKER. Ename “Ename”, MANAGER.ename “Manager”
    FROM emp WORKER, emp MANAGER
    WHERE WORKER.mgr=MANAGER.empno(+);
```

OUTPUT:

Ename	**Manager**
SMITH	FORD
ALLEN	BLAKE
WARD	BLAKE
JONES	KING
MARTIN	BLAKE
BLAKE	KING
CLARK	KING
SCOTT	JONES
TURNER	BLAKE
ADAMS	SCOTT
JAMES	BLAKE
FORD	JONES
MILLER	CLARK
KING	

14 rows selected.

It shows that KING appears in the list and he has no corresponding manager or in simple words he is at top of hierarchy.

- List all employees who joined the company before their manager.

```
SQL>Select e.ename, e.hiredate, m.ename manager, m.hiredate FROM emp e, emp m
      WHERE e.mgr= m.empno
       and e.hiredate<m.hiredate;
```