SQL Chapter 7 Join FSN

Joins are used when we need to fetch the data from multiple tables

Types of Joins

- Cartesian (Product) Join
- Inner (Equi) join
- Outer Join
  - Left Outer Join
  - Right Outer Join
  - Full Outer join
- Self join

Cartesian Join

- based on cartesian product theory
- T_A – m rows
  T_B – n rows
- not preferred for realtime scenarios

$T_A \overset{\text{cartesian}}{\text{join}} T_B$: m x n rows

ideally expected: max (m, n) rows

```
select * from emp, dept;
```

Inner Join | Equi joins

- they return the matching records b/w tables
- In real time scenarios, this is most frequently used join

Conventional Style

```
select A.ename, B.dept_name
from emp A, emp B
where A.dep_id = B.dep_id
and A.sal > 2000;
```

Join Condition

(Filter condition) optional

This join condition is mandatory for removing Cartesian o/p.

---

if join 3 table A B & C.

```
select * from A,B,C
where A.x = B.x
and A.x = C.x
```

No. of Tables = No of joins + 1

No of Joins = No. of tables − 1

Note: If there are no common columns, then 2 tables can't be joined

Note: common columns can be in different name.
cust_id & customer_id

ANSII style join

```
select * from
emp A join dept B
on A.deptId = B.dep_id;
```

'where'

Outer Join

↳ It returns both matching & non matching records

Outer join = Inner Join + Non matching records

⇓

It means data present in one table, but absent in another table wrt. common columns

7 ≠ 3

3 match 3

A (10 rows) B (6 rows)

Cartesian	10X6 = 60 rows
Left join	All matching 3 (10) + 7 non match from A left
Right join	All matching 3 (6) + 3 non match right B
Full join	3 matching (13) + 7 non match A + 3 non match B

SELF JOIN

- Joining a table to itself is called Self join

Correlated SubQueries

- Special type of sub queries
- both outer & inner queries are interdependent
- for each & every record of outer query, entire inner query will be executed

- they work on both principles of sub-queries & joins

Display employee who earning the highest salary

```
select * from emp A
where (1)-0 = (select count(distinct (B.sal)) from emp B
               where A.sal < B.sal);
```

(1) → ↑

A.sal < B.sal → ↓ for min salary