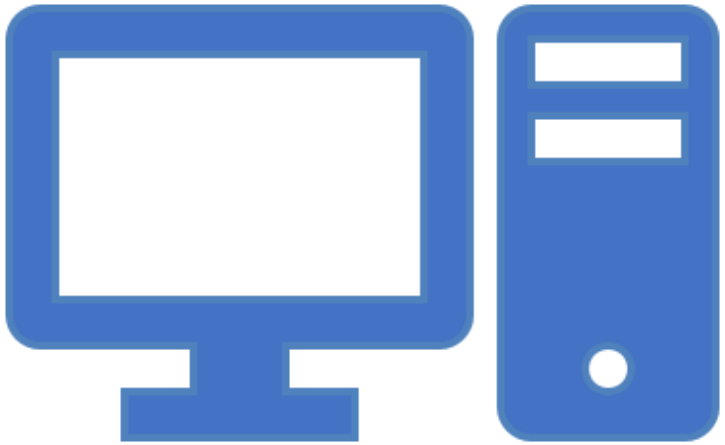
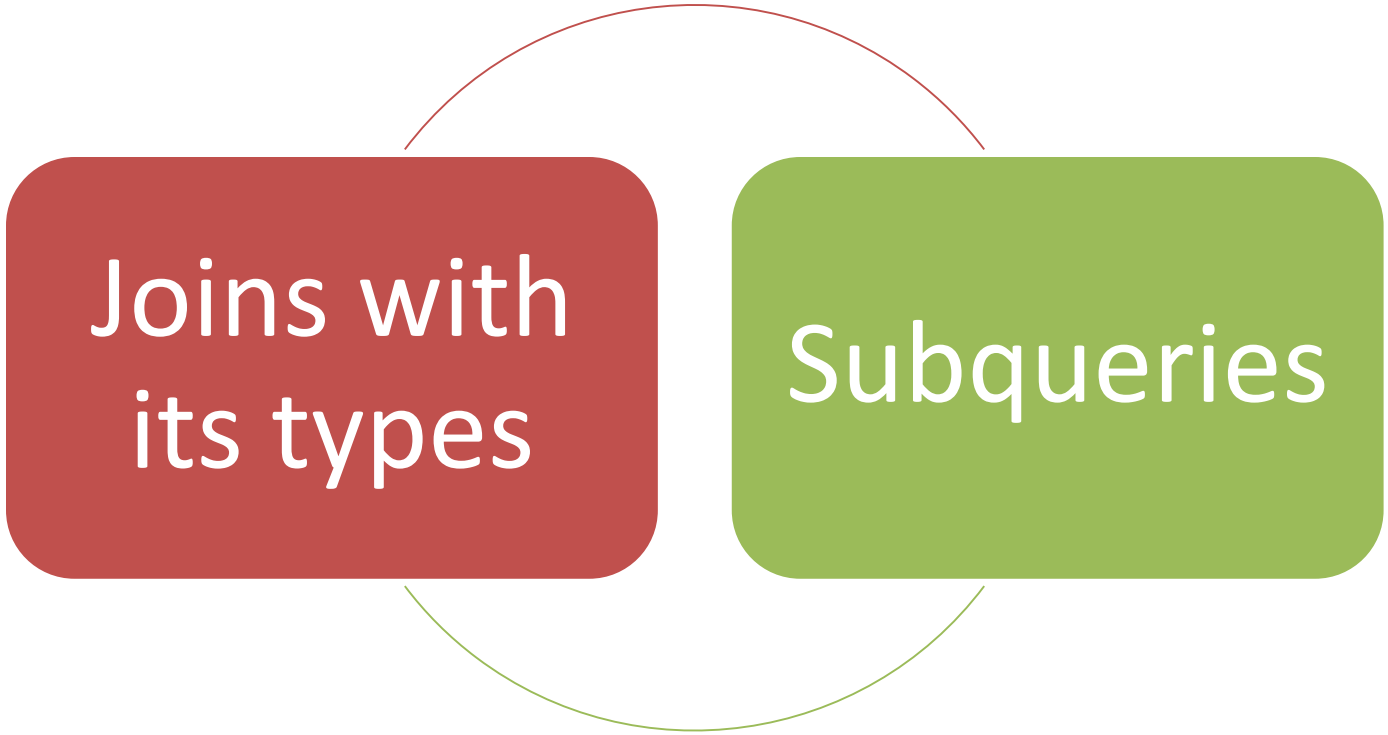


SYIT Sem III
Database Management System
Practical #5

Contents



SQL Join

SQL Join is used to fetch data from two or more tables, which is joined to appear as single set of data.

It is used for combining column from two or more tables by using values common to both tables.

Types of JOIN

Following are the types of JOIN that we can use in SQL:

1.Cross Join/ Cartesian Product

2.Inner/Equi Join

3.Natural Join

3.Outer Join

------Left Outer Join

------Right Outer Join

------Full Outer Join

Cross JOIN or Cartesian Product

- This type of JOIN returns the Cartesian product of rows from the tables in Join.
- It will return a table which consists of records which combines each row from the first table with each row of the second table.
- Cross JOIN Syntax:

-----SELECT column-name-list FROM table-name1 CROSS JOIN table-name2;

Example of Cross JOIN

class table

ID	NAME
1	abhi
2	adam
3	Anu
4	alex

class_info table

ID	ADDRESS
1	delhi
2	mumbai
3	chennai

Cross JOIN query

- SELECT * FROM class CROSS JOIN class_info;

- The resultset table will look like,

ID	NAME	ID	ADDRESS
1	abhi	1	delhi
2	adam	1	delhi
3	Anu	1	delhi
4	alex	1	delhi
1	abhi	2	mumbai
2	adam	2	mumbai
3	Anu	2	mumbai
4	alex	2	mumbai
1	abhi	3	chennai
2	adam	3	chennai
3	Anu	3	chennai
4	alex	3	chennai

As you can see, this join returns the cross product of all the records present in both the tables.

INNER Join or EQUI Join

- This is a simple JOIN in which the result is based on matched data as per the equality condition specified in the SQL query.

- Inner Join Syntax :

-----SELECT column-name-list FROM table-name1 INNER JOIN table-name2 ON table-name1.column-name = table-name2.column-name;

Example of INNER JOIN

Consider a class table

ID	NAME
1	abhi
2	adam
3	Anu
4	alex

class_info table

ID	ADDRESS
1	delhi
2	mumbai
3	chennai

Inner JOIN query

```
--SELECT * from class INNER JOIN
class_info on class.id =
class_info.id;
```

- The resultset table will look like,

ID	NAME	ID	ADDRESS
1	abhi	1	delhi
2	adam	2	mumbai
3	Anu	3	chennai

Natural JOIN

- Natural Join is a type of Inner join which is based on column having same name and same datatype present in both the tables to be joined.
- The syntax for Natural Join :
- SELECT * FROM table-name1 NATURAL JOIN table-name2;

Example of Natural JOIN

Consider a class table

ID	NAME
1	abhi
2	adam
3	Anu
4	alex

class_info table

ID	ADDRESS
1	delhi
2	mumbai
3	chennai

Natural join query :

- SELECT * from class NATURAL JOIN class_info;
- The resultset table will look like,

ID	NAME	ADDRESS
1	abhi	delhi
2	adam	mumbai
3	Anu	chennai

- In the above example, both the tables being joined have ID column(same name and same datatype), hence the records for which value of ID matches in both the tables will be the result of Natural Join of these two tables.

OUTER JOIN

- Outer Join is based on both matched and unmatched data. Outer Joins subdivide further into,
- Left Outer Join
- Right Outer Join
- Full Outer Join

LEFT Outer Join

The left outer join returns a resultset table with the **matched data** from the two tables and then the remaining rows of the **left** table and null from the **right** table's columns.

Syntax for Left Outer Join is,

SELECT column-name-list FROM table-name1 LEFT OUTER JOIN table-name2 ON table-name1.column-name = table-name2.column-name;

To specify a condition, we use the ON keyword with Outer Join.

Example of Left Outer Join

- Here is the class table, and the class_info table,

ID	NAME
1	abhi
2	adam
3	Anu
4	alex
5	ashish

ID	ADDRESS
1	delhi
2	mumbai
3	chennai
7	Noida
8	Panipat

Syntax and Output

Left Outer Join query will be,

SELECT * FROM class LEFT OUTER JOIN class_info ON (class.id = class_info.id);

The resultset table will look like,

ID	NAME	ID	ADDRESS
1	abhi	1	delhi
2	adam	2	mumbai
3	Anu	3	chennai
5	ashish	-	-
4	alex	-	-

RIGHT Outer Join

- The right outer join returns a resultset table with the **matched data** from the two tables being joined, then the remaining rows of the **right** table and null for the remaining **left** table's columns.
- Syntax for Right Outer Join is,

SELECT column-name-list FROM table-name1 RIGHT OUTER JOIN table-name2 ON table-name1.column-name = table-name2.column-name;

Example of Right Outer Join

- Here is the class table, and the class_info table,

ID	NAME
1	abhi
2	adam
3	Anu
4	alex
5	ashish

ID	ADDRESS
1	delhi
2	mumbai
3	chennai
7	Noida
8	Panipat

Syntax and Output

- Right Outer Join query will be,
- SELECT * FROM class RIGHT OUTER JOIN class_info ON (class.id = class_info.id);
- The resultant table will look like,

ID	NAME	ID	ADDRESS
1	abhi	1	delhi
2	adam	2	mumbai
3	Anu	3	chennai
-	-	7	Noida
-	-	8	Panipat

Full Outer Join

- The full outer join returns a result set table with the matched data of two table then remaining rows of both left table and then the right table.

- Syntax of Full Outer Join is,

SELECT column-name-list FROM table-name1 FULL OUTER JOIN table-name2 ON table-name1.column-name = table-name2.column-name;

Example of Full Outer Join

- Here is the class table, and the class_info table,

ID	NAME
1	abhi
2	adam
3	Anu
4	alex
5	ashish

ID	ADDRESS
1	delhi
2	mumbai
3	chennai
7	Noida
8	Panipat

Syntax and Output

- Full Outer Join query will be like,

- SELECT * FROM class FULL OUTER JOIN class_info ON (class.id = class_info.id);
- The resultant table will look like,

ID	NAME	ID	ADDRESS
1	abhi	1	delhi
2	adam	2	mumbai
3	Anu	3	chennai
5	ashish	-	-
4	alex	-	-
-	-	7	Noida
-	-	8	Panipat

SQL Sub Query

A Subquery is a query within another SQL query and embedded within the WHERE clause

A subquery can be placed in a number of SQL clauses like WHERE clause, FROM clause, HAVING clause.

You can use Subquery with SELECT, UPDATE, INSERT, DELETE statements along with the operators like =, <, >, >=, <=, IN, ANY, ALL, BETWEEN, etc.

A subquery is a query within another query. The outer query is known as the main query, and the inner query is known as a subquery.

Subqueries are on the right side of the comparison operator.

A subquery is enclosed in parentheses.

In the Subquery, ORDER BY command cannot be used. But GROUP BY command can be used to perform the same function as ORDER BY command.

Customers Table

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	35	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

1. Subqueries with the Select Statement

- SQL subqueries are most frequently used with the Select statement.
- **Syntax** :

```
SELECT column_name
FROM table_name
WHERE column_name expression operator
( SELECT column_name  from table_name WHERE ... );
```

- The subquery with a SELECT statement will be:

```
SQL> SELECT *
FROM CUSTOMERS
WHERE ID IN (SELECT ID
FROM CUSTOMERS
WHERE SALARY > 4500) ;
```

```
ID | NAME     | AGE | ADDRESS | SALARY
---+----------+-----+---------+----------
 4 | Chaitali |  25 | Mumbai  |  6500.00
 5 | Hardik   |  27 | Bhopal  |  8500.00
 7 | Muffy    |  24 | Indore  | 10000.00
```

2. Subqueries with the INSERT Statement

- SQL subquery can also be used with the Insert statement. In the insert statement, data returned from the subquery is used to insert into another table.
- In the subquery, the selected data can be modified with any of the character, date functions.
- **Syntax:**

```
INSERT INTO table_name (column1, column2, column3....)
SELECT * FROM table_name WHERE VALUE OPERATOR
```

- **Example**

```
INSERT INTO CUSTOMERS_BKP
SELECT * FROM  CUSTOMERS
WHERE ID IN (SELECT ID FROM  CUSTOMERS);
```

3. Subqueries with the UPDATE Statement

- The subquery of SQL can be used in conjunction with the Update statement. When a subquery is used with the Update statement, then either single or multiple columns in a table can be updated.
- **Syntax**

  UPDATE table SET column_name = new_value WHERE VALUE OPERATOR

  (SELECT COLUMN_NAME FROM TABLE_NAME WHERE condition);

- **Example**

  UPDATE CUSTOMERS

  SET SALARY = SALARY * 0.25

  WHERE AGE IN (SELECT AGE FROM

  CUSTOMERS_BKP WHERE AGE >= 27);

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	35	Ahmedabad	125.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	2125.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

4. Subqueries with the DELETE Statement

- The subquery of SQL can be used in conjunction with the Delete statement just like any other statements mentioned above
- **Syntax**

```
DELETE FROM TABLE_NAME
WHERE VALUE OPERATOR
(SELECT COLUMN_NAME FROM TA
```

- **Example**

```
DELETE FROM CUSTOMERS
WHERE AGE IN (SELECT AGE
FROM CUSTOMERS_BKP
WHERE AGE >= 27 );
```

ID	NAME	AGE	ADDRESS	SALARY
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Examples of Subqueries

- Display topic for least sequence value
- Select topic, seq from help where seq=(select min(seq) from help)

- Display info for highest sequence value
- Select info, seq from help where seq=(select max(seq) from help)

- Display topic and sequence from help for less than total number of sequence.
- Select topic, seq from help where seq<(select count(seq) from help)

- Display the topic and sequence from help table for sequences having value greater than minimum value.
- Select topic, seq from help where seq>(select min(seq) from help)

Emp Table

EMPNO	ENAME	JOB	DEPTNO	SAL
101	King	President	10	20000
102	Blake	Manager	10	12000
103	Clark	Manager	10	11000
104	Jones	Manager	20	10000
105	Scott	Analyst	20	5000
106	Ford	Analyst	30	6000
107	Allen	Salesman	40	950
110	Jimmy	President	30	25000
109	James	Salesman	10	1500
108	Turner	Salesman	20	1100
112	Killer	Developer	50	450
113	Jones	Leader	70	450
111	Menon	Developer	50	1200

Examples of Subqueries

- Write query to find the salaries for all the employees who have a higher salary than King.
- Select sal, ename from Emp where sal>(select sal from Emp where ename='King')

- Display details of an employee whose salary is greater than avg salary of employees who work as salesman.
- Select * from Emp where sal>(select avg(sal) from Emp where job='Salesman')

- Display the name of the employees whose salary is greater than the emplyee with empno 102.
- Select ename from Emp where sal>(select sal from Emp where empno=102);

Examples of Subqueries

- Display ename, salary deptno and job of the employees whose job is same as the job of empno 105.
- Select ename, sal, deptno, job from Emp where job= (select job from Emp where empno=105);

- Display ename, sal, deptno of the employess with minimum salary in each department
- Select ename, sal, deptno from Emp where sal IN(select min(sal) from Emp group by deptno);

- Display empno, ename and job of the employees having salary less than president.
- Select empno, ename, job from Emp where sal< ANY(select sal from Emp where job='President')

Examples of Subqueries

- Write query to display ename, job of an employee whose salaries are greater than all salaries of employees who work as Analyst.
- Select empno, ename, job from Emp where sal> ALL(select sal from Emp where job= 'Analyst' and job<>'Analyst');

- Write a query to display the employees having salary greater than the average salary of all departments.
- Select empno, ename, job from Emp where sal> ALL(select avg(sal) from Emp group by deptno);

Thank You