

Structure Query Language (SQL)

Example of MID()

Eid	Name	Age	Salary
401	anu	22	9000
402	shane	29	8000
403	rohan	34	6000
404	scott	44	10000
405	Tiger	35	8000

Consider following **Emp** table

SQL query will be,

```
select MID(name,2,2) from emp;
```

Result will come out to be,

MID(name,2,2)
Nu
ha
oh
co
ig

4) ROUND()

ROUND function is used to round a numeric field to number of nearest integer. It is used on Decimal point values. Syntax of Round function is,

```
SELECT ROUND(column_name, decimals) from table-name
```

Example of ROUND()

Consider following Emp table

Eid	Name	Age	Salary
401	anu	22	9000.67
402	shane	29	8000.98
403	rohan	34	6000.45
404	scott	44	10000
405	Tiger	35	8000.01

SQL query is,

```
SELECT ROUND(salary) from emp;
```

Result will be,

ROUND(salary)
9001
8001
6000
10000
8000

6.15 Join in SQL

SQL Join is used to fetch data from two or more tables, which is joined to appear as single set of data. SQL Join is used for combining column from two or more tables by using values common to both tables. **Join** Keyword is used in SQL queries for joining two or more tables. Minimum required condition for joining table, is **(n-1)** where **n**, is number of tables. A table can also join to itself known as, **Self Join**.

Types of Join

The following are the types of JOIN that we can use in SQL.

- Inner
- Outer
- Left
- Right

6.15.1 Cross JOIN or Cartesian Product

This type of JOIN returns the Cartesian product of rows of 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 is,

```
SELECT column-name-list
```

```
from table-name1
```

```
CROSS JOIN
```

```
table-name2;
```

Example of Cross JOIN

The **class** table,

ID	NAME
1	abhi
2	adam
4	alex

The **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI

Cross JOIN query will be,

```
SELECT *  
from class,  
cross JOIN class_info;
```

The result table will look like,

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	1	DELHI
4	alex	1	DELHI
1	abhi	2	MUMBAI
2	adam	2	MUMBAI
4	alex	2	MUMBAI

1	abhi	3	CHENNAI
2	adam	3	CHENNAI
4	alex	3	CHENNAI

6.15.2 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 query.

Inner Join Syntax is,

```
SELECT column-name-list
```

```
from table-name1
```

```
INNER JOIN
```

```
table-name2
```

```
WHERE table-name1.column-name = table-name2.column-name;
```

Example of Inner JOIN

The **class** table,

ID	NAME
1	abhi
2	adam
3	alex
4	anu

The **class_info** table,

ID	Address
----	---------

1	DELHI
2	MUMBAI
3	CHENNAI

Inner JOIN query will be,

```
SELECT * from class, class_info where class.id = class_info.id;
```

The result table will look like,

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI

6.15.3 Natural JOIN

Natural Join is a type of Inner join which is based on column having same name and same data type present in both the tables to be joined.

Natural Join Syntax is,

```
SELECT *  
from table-name1  
NATURAL JOIN  
table-name2;
```

Example of Natural JOIN

The **class** table,

ID	NAME
1	abhi
2	adam
3	alex
4	anu

The **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI

Natural join query will be,

```
SELECT * from class NATURAL JOIN class_info;
```

The result table will look like,

ID	NAME	Address
1	abhi	DELHI
2	adam	MUMBAI
3	alex	CHENNAI

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

6.15.5 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

6.15.5.1 Left Outer Join

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

Left Outer Join syntax is,

```
SELECT column-name-list  
from table-name1  
LEFT OUTER JOIN  
table-name2  
on table-name1.column-name = table-name2.column-name;
```

Left outer Join Syntax for **Oracle** is,

```
select column-name-list  
from table-name1,  
table-name2  
on table-name1.column-name = table-name2.column-name(+);
```