Structure Query Language (SQL)

Example of Left Outer Join

The **class** table,

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

The **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

Left Outer Join query will be,

```
SELECT * FROM class LEFT OUTER JOIN class_info ON (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
4	anu	null	null
5	ashish	null	null

6.15.5.2 Right Outer Join

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

Right Outer Join Syntax is,

```
select column-name-list
from table-name1
RIGHT OUTER JOIN
table-name2
on table-name1.column-name = table-name2.column-name;
```

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

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

Example of Right Outer Join

The **class** table,

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

The **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

Right Outer Join query will be,

```
SELECT * FROM class RIGHT OUTER JOIN class_info on (class.id=class_inf
o.id);
```

The result table will look like,

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI
null	null	7	NOIDA
null	null	8	PANIPAT

6.15.3 Full Outer Join

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

Full Outer Join Syntax 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 is,

The **class** table,

ID	NAME
1	abhi
2	adam
3	alex

4	anu
5	ashish

The **class_info** table,

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA
8	PANIPAT

Full Outer Join query will be like,

```
SELECT * FROM class F
ULL OUTER JOIN class_i
nfo on (class.id=class_i
nfo.id);
```

The result table will look like,

ID	NAME	ID	Address
1	abhi	1	DELHI
2	adam	2	MUMBAI
3	alex	3	CHENNAI
4	anu	null	null
5	ashish	null	null
Null	null	7	NOIDA
Null	null	8	PANIPAT

6.16 SQL Alias

Alias is used to give an alias name to a table or a column. This is quite useful in case of large or complex queries. Alias is mainly used for giving a short alias name for a column or a table with complex names.

Syntax of Alias for table names,

```
SELECT column-name
from table-name
as alias-name
```

Following is an Example using Alias,

```
SELECT * from Employee_detail as ed;
```

Alias syntax for columns will be like,

```
SELECT
column-name as alias-name
fromtable-name
```

Example using alias for columns,

```
SELECT customer_id as cid from Emp;
```

Example of Alias in SQL Query

Consider the following two tables,

The **class** table,The **class_info** table,

ID	Name
1	abhi
2	adam
3	alex
4	anu

ID	Address
1	DELHI
2	MUMBAI
3	CHENNAI
7	NOIDA

5	ashish

8	PANIPAT

Below is the Query to fetch data from both the tables using SQL Alias,

```
SELECT C.id, C.Name, Ci.Address from Class as C, Class_info as Ci where C
.id=Ci.id;
```

Result table look like,

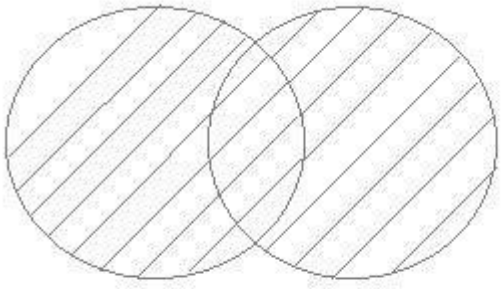
ID	Name	Address
1	abhi	DELHI
2	adam	MUMBAI
3	alex	CHENNAI

6.17 Set Operation in SQL

SQL supports few Set operations to be performed on table data. These are used to get meaningful results from data, under different special conditions.

6.17.1 Union

UNION is used to combine the results of two or more Select statements. However it will eliminate duplicate rows from its result set. In case of union, number of columns and data type must be same in both the tables.



Example of UNION

The **First** table,

ID	Name
1	Abhi
2	Adam

The **Second** table,

ID	Name
2	adam
3	Chester

Union SQL query will be,

```
select * from First
```

```
UNION
select * from second
```

The result table will look like,

ID	NAME
1	abhi
2	adam
3	Chester