netlink
digital solutions

DATABASE

www.netlink.com

PostgreSQL Table Aliases

Table aliases temporarily assign tables new names during the execution of a query.

The basic syntax of table alias is as follows −

```
SELECT column1, column2....
FROM table_name AS alias_name
WHERE [condition];
```

The basic syntax of column alias is as follows −

```
SELECT column_name AS alias_name
FROM table_name
WHERE [condition];
```

Using table aliases for the long table name to make queries more readable:

For example, instead of using the following expression in a query:

```
a_very_long_table_name.column_name
```

you can assign the table `a_very_long_table_name` an alias like this:

```
a_very_long_table_name AS alias
```

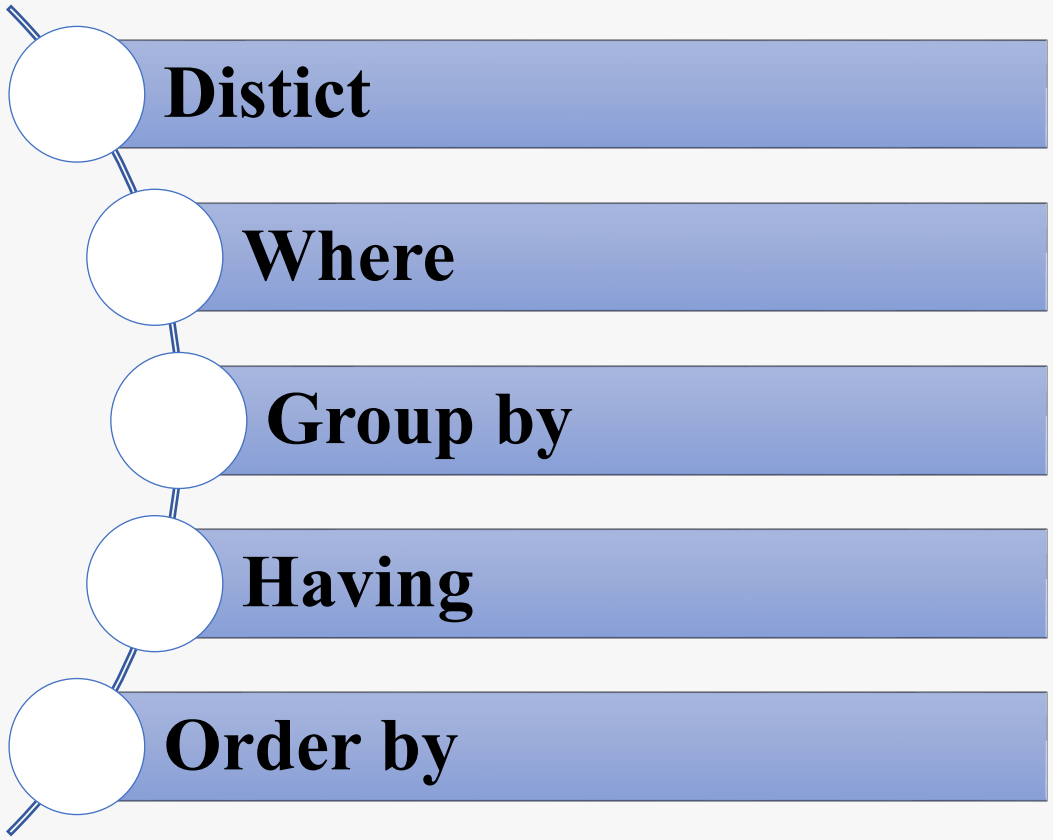
And reference the `column_name` in the table `a_very_long_table_name` using the table alias:

```
alias.column_name
```

Clauses

- Clauses are in-built functions available to us in SQL. **With the help of clauses, we can deal with data easily stored in the table.**
- Clauses help us filter and analyze data quickly. When we have large amounts of data stored in the database, we use Clauses to query and get data required by the user.
- Some of the examples of clauses are –

Types of Clauses



Distinct

Distinct

Distinct clause is used to remove duplicate value from resultant table

In select clause either distinct or astrict (*) should be the first Executable statement

We can pass multiple column name to distinct clause

Syntax :-

```
SELECT  DISTINCT column1 FROM   table_name;
```

Syntax :-

```
SELECT  DISTINCT column1, column2
FROM  table_name;
```

If you specify multiple columns, the `DISTINCT` clause will evaluate the duplicate based on the combination of values of these columns.

Where clause

Where Clause

- It used to filter records from the table
- Where clause execute row by row or record by record
- Where clause execute after from clause
- We can use multiple condition in where clause

Rules and Usage:

1. Rules:

- When using a Where clause, we need to mention at least one condition.

2. Usage:

- Used to query and transact the database.
- Used with an update and delete statements to make sure the right data points are deleted.

Syntax:

```
SELECT * FROM tableName WHERE condition ;
```

Row by row

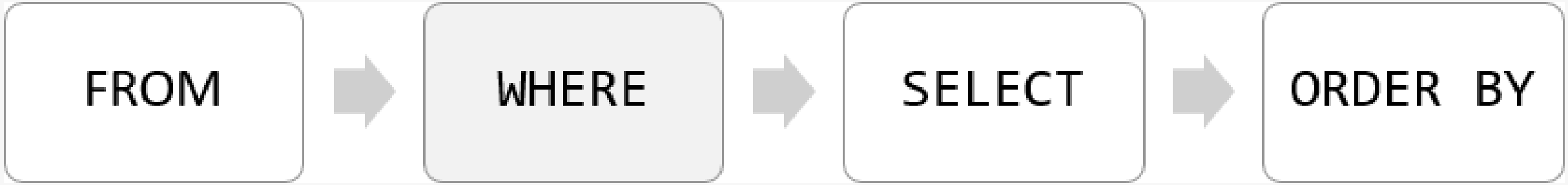
Id	Name	Age	Dept
101	ravi	20	10
102	Mack	21	20
103	King	22	30
104	Jack	23	40

Where clause

The syntax of the PostgreSQL `WHERE` clause is as follows:

```
SELECT select_list FROM table_name WHERE condition
ORDER BY sort_expression
```

PostgreSQL evaluates the `WHERE` clause after the `FROM` clause and before the `SELECT` and `ORDER BY` clause:



The `condition` must evaluate to true, false, or unknown. It can be a boolean expression or a combination of boolean expressions using the AND and OR operators.

Group By

Group by

- Group by is used to create group of records
- Group by clause execute row by row
- With or without using where clause we can use group by clause
- Group by execute after from clause
- Group by clause execute row by row but after execute it will create group
- Any clause which execute after group by clause it execute group by group

Rules and Usage:

1. Rules:

- Columns appearing in the Select clause can only be taken care of in the Group By clause.
- Columns we are passing to the Group By clause should be of comparable type.

2. Usage:

- We use Group by clause to get the groups present in data.

Note:

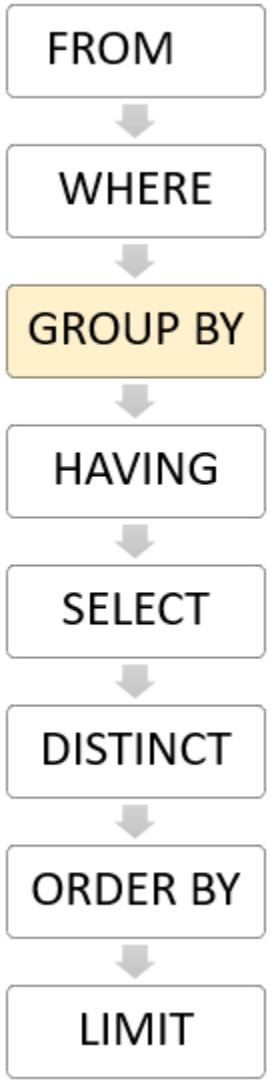
The Group by clause perform its operation in buffer (temp storage) not in database.

Group By

PostgreSQL evaluates the GROUP BY clause after the FROM and WHERE clauses and before the HAVING SELECT, DISTINCT, ORDER BY and LIMIT clauses.

Syntax:

```
SELECT
   column_1,
   column_2,
   ...,
   aggregate_function(column_3)
FROM
   table_name
GROUP BY
   column_1,
   column_2,
   ...;
```



Having Clause

Having Clause

- Having clause is used to filter a group
- Having clause execute after group by group
- Since the HAVING clause is evaluated before the SELECT clause, you cannot use column aliases in the HAVING clause.
- We can have multirow function in having clause

Syntax;

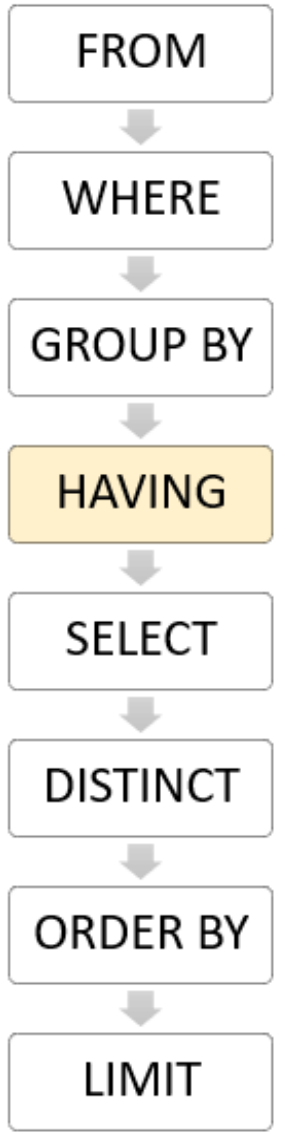
```
SELECT
   column1,
   aggregate_function (column2)
FROM
   table_name
GROUP BY
   column1
HAVING
   condition;
```

Having Clause

PostgreSQL evaluates the HAVING clause after the FROM, WHERE, GROUP BY, and before the SELECT, DISTINCT, ORDER BY and LIMIT clauses.

HAVING vs. WHERE:

- The `WHERE` clause allows you to filter rows based on a specified condition. However, the HAVING clause allows you to filter groups of rows according to a specified condition.
- In other words, the `WHERE` clause is applied to rows while the HAVING clause is applied to groups of rows.



HAVING vs. WHERE:

- In some cases, you need to filter out the individual records. In such cases, you can use WHERE Clause, Whereas in other cases you need to filter the groups with the specific condition. In such cases, you can use HAVING Clause.

- WHERE Clause can be utilized with SELECT, UPDATE, DELETE, and INSERT, whereas HAVING can be utilized only with SELECT statement.

Sno	Where Clause	Having Clause
1	The WHERE clause specifies the criteria which individual records must meet to be selcted by a query. It can be used without the GROUP BY clause	The HAVING clause cannot be used without the GROUP BY clause.
2	The WHERE clause selects rows before grouping.	The HAVING clause selects rows after grouping.
3	The WHERE clause cannot contain aggregate functions	The HAVING clause can contain aggregate functions.
4	WHERE clause is used to impose condition on SELECT statement as well as single row function and is used before GROUP BY clause	HAVING clause is used to impose condition on GROUP Function and is used after GROUP BY clause in the query
5	SELECT Column,AVG(Column_nmae)FROM Table_name WHERE Column > value GROUP BY Column_nmae	SELECT Columnq, AVG(Column_nmae)FROM Table_name WHERE Column > value GROUP BY Column_nmae Having column_name>or<value

Order by

Order by

- It is used to arrange the resultant table in ascending or descending order
- Order by clause execute after select clause
- By default order by clause arrange the resultant table in ascending order
- We can use Alias name in order by clause
- We can pass multiple column name in order
- Order by clause should be the last argument for all the queries

Rules and Usage:

1. Rules:

- A comparable data column should be passed in the query.
- Any column can be used in the order by clause, even those which do not appear in our select statement.
- We can sort data in ascending or descending order (by default sorting is done in ascending order).

Click to add text

2. Usage:

- Order by clause is useful to get data in required sorting orders.

Order of Execution

1 From
2 Where -----row by row
3 Group by-----row by row
4 Having -------group by group
5 Select --------group by group
6 Order by

Order by

```
SELECT
select_list
FROM
table_name
ORDER BY
sort_expression1 [ASC | DESC],
        ...
sort_expressionN [ASC |
DESC];
```

PostgreSQL evaluates the clauses in the SELECT statment in the following order: FROM, SELECT, and ORDER BY:

NOTE :

Due to the order of evaluation, if you have a column alias in the SELECT clause, you can use it in the ORDER BY clause.


netlink
digital solutions
Thank you
www.netlink.com