

PostgreSQL Querying Guide



Section 1. Querying Data

Select – show you how to query data from a single table.

Column aliases – learn how to assign temporary names to columns or expressions within a query.

Order By – guide you on how to sort the result set returned by a query.

Select Distinct – show you how to remove duplicate rows from the result set.

```
SELECT
  select_list
FROM
  table_name;
```

PostgreSQL evaluates the FROM clause before the SELECT clause in the SELECT statement:

```
SELECT column_name AS alias_name
FROM table_name;
```

```
SELECT
  select_list
FROM
  table_name
ORDER BY
  sort_expression1 [ASC | DESC],
  sort_expression2 [ASC | DESC],
  ...;
```

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

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

Make Document for this



PostgreSQL: Querying Data

1. SELECT Statement

The `SELECT` statement is used to retrieve data from one or more tables.

Syntax:

```
sql
```

```
SELECT select_list
FROM table_name;
```

- ``select_list``: Specifies the columns or expressions to be retrieved.
- ``table_name``: Specifies the table to query from.

Execution Order of SELECT Statement:

PostgreSQL evaluates the clauses in the following order:

1. **FROM** clause (to determine the data source).
2. **SELECT** clause (to choose the columns).
3. **ORDER BY** clause (to sort the result set).

2. Column Aliases

Column aliases are used to assign a temporary name to a column or an expression in the result set. This can make the output more readable.

Syntax:

```
sql
```

```
SELECT column_name AS alias_name
FROM table_name;
```

Example:

Suppose you have a table ``employees`` with columns ``first_name`` and ``last_name``.

```
sql
```

```
SELECT first_name AS "First Name", last_name AS "Last Name"
FROM employees;
```

Output:

First Name	Last Name
John	Doe
Jane	Smith

3. ORDER BY Clause

The ``ORDER BY`` clause is used to sort the result set based on one or more columns or expressions. You can specify ascending (``ASC``, default) or descending (``DESC``) order.

Syntax:

sql

```
SELECT select_list
FROM table_name
ORDER BY sort_expression1 [ASC | DESC], sort_expression2 [ASC | DESC], ...;
```

Example:

Sort employees by `last_name` in ascending order and `first_name` in descending order.

sql

```
SELECT first_name, last_name
FROM employees
ORDER BY last_name ASC, first_name DESC;
```

Output:

First Name	Last Name
Jane	Doe
John	Doe
Alan	Smith

4. SELECT DISTINCT

The `SELECT DISTINCT` statement is used to remove duplicate rows from the result set.

Syntax:

sql

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

Example:

Retrieve distinct department names from the `employees` table.

sql

```
SELECT DISTINCT department
FROM employees;
```

Output:

Department
HR
IT
Sales

Summary of Clauses Evaluation Order in PostgreSQL

1. **FROM:** Specifies the table(s) to query.
2. **SELECT:** Defines the columns or expressions to return.
3. **ORDER BY:** Sorts the result set based on specified criteria.

Document Example Query

```
sql
```

```
-- Query to retrieve distinct department names, sort by name in ascending order:
SELECT DISTINCT department AS "Department Name"
FROM employees
ORDER BY "Department Name" ASC;
```

Output:

Department Name
HR
IT
Sales