Sure! In PostgreSQL, when you use the ORDER BY clause, NULL values can be ordered in various ways. By default, NULL values are sorted at the end for ascending orders and at the beginning for descending orders. However, you can control this behavior using NULLS FIRST or NULLS LAST.

Here's how you can include queries that demonstrate ordering with NULL values:

**Sample Table: products (reused)**

1. **Creating the Table**

sql

Copy code

CREATE TABLE products (

product\_id SERIAL PRIMARY KEY,

product\_name VARCHAR(100) NOT NULL,

price NUMERIC(10, 2),

quantity INT,

description TEXT

);

1. **Inserting Data (Including NULL Values)**

sql

Copy code

INSERT INTO products (product\_name, price, quantity, description) VALUES

('Laptop', 999.99, 50, 'High performance laptop'),

('Smartphone', NULL, 100, 'Latest model smartphone'), -- price is NULL

('Tablet', 299.99, NULL, NULL), -- quantity and description are NULL

('Smartwatch', 199.99, 75, NULL); -- description is NULL

**Queries for Ordering by NULL Values**

1. **Ordering by Price (Ascending)**

By default, NULL values will be sorted last:

sql

Copy code

SELECT \* FROM products

ORDER BY price; -- NULLs will be at the end

1. **Ordering by Price (Descending)**

By default, NULL values will be sorted first:

sql

Copy code

SELECT \* FROM products

ORDER BY price DESC; -- NULLs will be at the beginning

1. **Ordering by Price (Ascending) with NULLS FIRST**

This query explicitly puts NULL values at the beginning of the sorted result:

sql

Copy code

SELECT \* FROM products

ORDER BY price NULLS FIRST; -- NULLs will be at the beginning

1. **Ordering by Price (Descending) with NULLS LAST**

This query explicitly puts NULL values at the end of the sorted result:

sql

Copy code

SELECT \* FROM products

ORDER BY price DESC NULLS LAST; -- NULLs will be at the end

1. **Ordering by Multiple Columns**

You can also order by multiple columns, handling NULL values in one of them. For example, first by price, and then by quantity:

sql

Copy code

SELECT \* FROM products

ORDER BY price NULLS LAST, quantity NULLS FIRST; -- Sort by price (NULLs last), then by quantity (NULLs first)

**Summary of Ordering with NULLs**

* **ORDER BY price**: By default, NULL values are sorted last in ascending order.
* **ORDER BY price DESC**: By default, NULL values are sorted first in descending order.
* **ORDER BY price NULLS FIRST**: Explicitly places NULL values at the beginning of the sorted result.
* **ORDER BY price DESC NULLS LAST**: Explicitly places NULL values at the end of the sorted result.
* **ORDER BY multiple columns**: Allows for complex sorting behaviors, handling NULL values individually.

These queries illustrate how to manage and order NULL values in PostgreSQL effectively!