SQL WITH Queries:

Common Table Expressions

WITH


```
WITH temporary_table_name AS (
    nested_query
)
base_query;
```

Used to form temporary result sets, called common table expressions (CTEs)

Breaks down complex queries into simpler parts

Base query can leverage data in temporary table

Does not create new table, table cleared from memory after query is completed

WITH example


```
CREATE TABLE orders (
    id SERIAL PRIMARY KEY,
    region TEXT NOT NULL,
    product TEXT NOT NULL,
    quantity INT NOT NULL,
    amount NUMERIC NOT NULL
);
```

```
postgres=# \d orders
                                Table "public.orders"
  Column   |  Type   | Collation | Nullable |                Default
-----------+---------+-----------+----------+---------------------------------------
 id        | integer |           | not null | nextval('orders_id_seq'::regclass)
 region    | text    |           | not null |
 product   | text    |           | not null |
 quantity  | integer |           | not null |
 amount    | numeric |           | not null |
```

WITH example

"For the top sales regions (top 10% in total sales), find the total units sold and the total sales for each product

```
postgres=# \d orders
                                  Table "public.orders"
  Column   |  Type   | Collation | Nullable |                Default
-----------+---------+-----------+----------+---------------------------------------
 id        | integer |           | not null | nextval('orders_id_seq'::regclass)
 region    | text    |           | not null |
 product   | text    |           | not null |
 quantity  | integer |           | not null |
 amount    | numeric |           | not null |
```

WITH example


```sql
WITH regional_sales AS (
        SELECT region, SUM(amount) AS total_sales
        FROM orders
        GROUP BY region
     ), top_regions AS (
        SELECT region
        FROM regional_sales
        WHERE total_sales > (SELECT SUM(total_sales)/10 FROM regional_sales)
     )
SELECT region,
       product,
       SUM(quantity) AS product_units,
       SUM(amount) AS product_sales
FROM orders
WHERE region IN (SELECT region FROM top_regions)
GROUP BY region, product;
```

```
postgres=# \d orders
                                  Table "public.orders"
  Column  |  Type   | Collation | Nullable |              Default
----------+---------+-----------+----------+-----------------------------------
 id       | integer |           | not null | nextval('orders_id_seq'::regclass)
 region   | text    |           | not null |
 product  | text    |           | not null |
 quantity | integer |           | not null |
 amount   | numeric |           | not null |
```

WITH example

"For the top sales regions (top 10% in total sales), find the total units sold and the total sales for each product"

```sql
SELECT region,
       product,
       SUM(quantity) AS product_units,
       SUM(amount) AS product_sales
FROM orders
WHERE region IN (SELECT region FROM top_regions)
GROUP BY region, product;
```

```
postgres=# \d orders
                                    Table "public.orders"
  Column  |  Type   | Collation | Nullable |              Default
----------+---------+-----------+----------+-----------------------------------
 id       | integer |           | not null | nextval('orders_id_seq'::regclass)
 region   | text    |           | not null |
 product  | text    |           | not null |
 quantity | integer |           | not null |
 amount   | numeric |           | not null |
```

WITH example

"For the top sales regions (top 10% in total sales), find the total units sold and the total sales for each product"

```
SELECT region,
       product,
       SUM(quantity) AS product_units,
       SUM(amount) AS product_sales
FROM orders
WHERE region IN (SELECT region FROM top_regions)
GROUP BY region, product;
```

?

```
postgres=# \d orders
                                  Table "public.orders"
  Column  |  Type   | Collation | Nullable |              Default
----------+---------+-----------+----------+-----------------------------------
 id       | integer |           | not null | nextval('orders_id_seq'::regclass)
 region   | text    |           | not null |
 product  | text    |           | not null |
 quantity | integer |           | not null |
 amount   | numeric |           | not null |
```

WITH example

"For the top sales regions (top 10% in total sales), find the total units sold and the total sales for each product"

```
postgres=# \d orders
                                  Table "public.orders"
  Column  |  Type   | Collation | Nullable |                Default
----------+---------+-----------+----------+---------------------------------------
 id       | integer |           | not null | nextval('orders_id_seq'::regclass)
 region   | text    |           | not null |
 product  | text    |           | not null |
 quantity | integer |           | not null |
 amount   | numeric |           | not null |
```

```sql
WITH top_regions AS (
        SELECT region
        FROM regional_sales
        WHERE total_sales > (SELECT SUM(total_sales)/10 FROM regional_sales)
    )
SELECT region,
       product,
       SUM(quantity) AS product_units,
       SUM(amount) AS product_sales
FROM orders
WHERE region IN (SELECT region FROM top_regions)
GROUP BY region, product;
```

WITH example

"For the top sales regions (top 10% in total sales), find the total units sold and the total sales for each product"

```
postgres=# \d orders
                                  Table "public.orders"
  Column  |  Type   | Collation | Nullable |                Default
----------+---------+-----------+----------+---------------------------------------
 id       | integer |           | not null | nextval('orders_id_seq'::regclass)
 region   | text    |           | not null |
 product  | text    |           | not null |
 quantity | integer |           | not null |
 amount   | numeric |           | not null |
```

```sql
WITH top_regions AS (
        SELECT region
        FROM regional_sales
        WHERE total_sales > (SELECT SUM(total_sales)/10 FROM regional_sales)
    )
SELECT region,
       product,
       SUM(quantity) AS product_units,
       SUM(amount) AS product_sales
FROM orders
WHERE region IN (SELECT region FROM top_regions)
GROUP BY region, product;
```

?

WITH example


```
WITH regional_sales AS (
        SELECT region, SUM(amount) AS total_sales
        FROM orders
        GROUP BY region
     ), top_regions AS (
        SELECT region
        FROM regional_sales
        WHERE total_sales > (SELECT SUM(total_sales)/10 FROM regional_sales)
     )
SELECT region,
       product,
       SUM(quantity) AS product_units,
       SUM(amount) AS product_sales
FROM orders
WHERE region IN (SELECT region FROM top_regions)
GROUP BY region, product;
```