



SQL WITH Queries:

Common Table Expressions







```
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





```
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
                      Collation | Nullable |
                                                             Default
             Type
                                   not null | nextval('orders_id_seq'::regclass)
 id
            integer |
 region
                                   not null |
            text
 product
            text
                                   not null |
 quantity
                                   not null |
            integer
            numeric
                                   not null |
 amount
```





"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
                       Collation | Nullable |
                                                             Default
             Type
                                   not null | nextval('orders_id_seq'::regclass)
 id
            integer
                                   not null |
 region
            text
 product
                                   not null |
            text
 quantity
            integer
                                   not null |
            numeric
                                   not null |
 amount
```





```
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)
                                                        postgres=# \d orders
SELECT region,
                                                                           Table "public.orders"
                                                                      Collation | Nullable
                                                                                               Default
                                                                Type
        product,
                                                                             | not null | nextval('orders_id_seg'::regclass)
                                                                integer
        SUM(quantity) AS product_units,
                                                         region
                                                                text
                                                                              | not null |
                                                                              not null |
                                                         product
                                                                text
        SUM(amount) AS product_sales
                                                         quantity
                                                                integer
                                                                              not null |
                                                                numeric
                                                                              not null
                                                         amount
FROM orders
WHERE region IN (SELECT region FROM top_regions)
GROUP BY region, product;
```





"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"
                      Collation | Nullable |
  Column
             Type
                                                            Default
 id
            integer
                                  not null | nextval('orders_id_seq'::regclass)
 region
                                  not null
            text
 product
            text
                                  not null
 quantity
            integer
                                  not null
                                  not null |
 amount
            numeric
```





"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(quanti y) AS product_units,

SUM(amount) S product_sales

FROM orders

WHERE region IN (SELECT region FROM top_regions)

GROUP BY region, product;
```

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





"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 region product quantity amount	integer text text integer numeric	 	not null not null not null not null not null not null	nextval('orders_id_seq'::regclass)				





"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 region product quantity amount	integer text text integer numeric	 	not null not null not null not null not null	nextval('orders_id_seq'::regclass) 				





```
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;
```