# DATABASE VIEWS

```
SELECT * FROM customers;

 cust_id | cust_name |   address
---------+-----------+--------------
       1 | Alice     | 123 main st
       2 | Bob       | 45 west road
       3 | Charlie   | 6 Second Ave
       4 | Diana     | 78 River st
(4 rows)

SELECT * FROM stores;

 store_id |   city
----------+----------
       11 | Paris
       12 | London
       13 | New York
(3 rows)

SELECT * FROM orders;

 order_no | cust_id | amount |   o_date   | store_id
----------+---------+--------+------------+----------
      101 |       1 |  50.00 | 2020-11-30 |       13
      102 |       1 | 150.00 | 2021-06-08 |       12
      103 |       2 |  80.00 | 2020-12-25 |       13
      105 |       3 |  20.00 | 2020-12-24 |       12
      106 |       4 | 120.00 | 2021-02-21 |       13
      107 |       4 |  60.00 | 2020-12-14 |       12
      104 |       3 | 200.00 | 2021-05-03 |       11
(7 rows)
```

```sql
-- Find customers whose total amount is greater than the average total amount across customers

WITH customer_totals AS (
    SELECT cust_id, sum(amount) AS total_amount
    FROM orders
    GROUP BY cust_id
),
avg_total_amount AS (
    SELECT avg(total_amount) AS avg_total
    FROM customer_totals
)
SELECT cust_id, total_amount
FROM customer_totals, avg_total_amount
WHERE total_amount > avg_total;
```

```
 cust_id | total_amount
---------+--------------
       3 |       220.00
       4 |       180.00
       1 |       200.00
(3 rows)
```

```
 cust_id | total_amount
---------+--------------
       3 |       220.00
       4 |       180.00
       2 |        80.00
       1 |       200.00
(4 rows)
```

```
       avg_total
-----------------------
 170.0000000000000000
(1 row)
```

```
SELECT city, sum(amount_plus_tax) AS city_totals
FROM (
  SELECT city, s.store_id, order_no, o_date, amount * 1.15 AS amount_plus_tax
  FROM stores AS s JOIN orders AS o
  ON s.store_id = o.store_id
) AS order_totals_with_tax
WHERE o_date <= '2020-12-31'
GROUP BY city;

   city   | city_totals
----------+-------------
 New York |    149.5000
 London   |     92.0000
(2 rows)


SELECT store_id, sum(amount_plus_tax) AS store_totals
FROM (
  SELECT city, s.store_id, order_no, o_date, amount * 1.15 AS amount_plus_tax
  FROM stores AS s JOIN orders AS o
  ON s.store_id = o.store_id
) AS order_totals_with_tax
WHERE o_date >= '2021-01-01'
GROUP BY store_id;

 store_id | store_totals
----------+-------------
       13 |    138.0000
       11 |    230.0000
       12 |    172.5000
(3 rows)
```

```
SELECT city, sum(amount_plus_tax) AS city_totals
FROM (
  SELECT city, s.store_id, order_no, o_date, amount * 1.15 AS amount_plus_tax
  FROM stores AS s JOIN orders AS o
  ON s.store_id = o.store_id
) AS order_totals_with_tax
WHERE o_date <= '2020-12-31'
GROUP BY city;

   city   | city_totals
----------+-------------
 New York |    149.5000
 London   |     92.0000
(2 rows)


SELECT store_id, sum(amount_plus_tax) AS store_totals
FROM (
  SELECT city, s.store_id, order_no, o_date, amount * 1.15 AS amount_plus_tax
  FROM stores AS s JOIN orders AS o
  ON s.store_id = o.store_id
) AS order_totals_with_tax
WHERE o_date >= '2021-01-01'
GROUP BY store_id;

 store_id | store_totals
----------+-------------
       13 |    138.0000
       11 |    230.0000
       12 |    172.5000
(3 rows)
```

```
\d
          List of relations
 Schema |   Name    | Type  |  Owner
--------+-----------+-------+----------
 public | customers | table | postgres
 public | orders    | table | postgres
 public | stores    | table | postgres
(3 rows)

CREATE VIEW order_totals_with_tax AS
SELECT city, s.store_id, order_no, o_date, amount * 1.15 AS amount_plus_tax
FROM stores AS s JOIN orders AS o
ON s.store_id = o.store_id;

CREATE VIEW

\d
                List of relations
 Schema |         Name          | Type  |  Owner
--------+-----------------------+-------+----------
 public | customers             | table | postgres
 public | order_totals_with_tax | view  | postgres
 public | orders                | table | postgres
 public | stores                | table | postgres
(4 rows)
```

```
SELECT * FROM pg_views WHERE viewname = 'order_totals_with_tax';

 schemaname |       viewname        | viewowner |                      definition
------------+-----------------------+-----------+-----------------------------------------------------
 public     | order_totals_with_tax | postgres  |  SELECT s.city,                                    +
            |                       |           |     s.store_id,                                    +
            |                       |           |     o.order_no,                                    +
            |                       |           |     o.o_date,                                      +
            |                       |           |     (o.amount * 1.15) AS amount_plus_tax           +
            |                       |           |    FROM (stores s                                  +
            |                       |           |      JOIN orders o ON ((s.store_id = o.store_id)));
(1 row)

SELECT * FROM order_totals_with_tax;

   city   | store_id | order_no |   o_date   | amount_plus_tax
----------+----------+----------+------------+-----------------
 New York |       13 |      101 | 2020-11-30 |         57.5000
 London   |       12 |      102 | 2021-06-08 |        172.5000
 New York |       13 |      103 | 2020-12-25 |         92.0000
 London   |       12 |      105 | 2020-12-24 |         23.0000
 New York |       13 |      106 | 2021-02-21 |        138.0000
 London   |       12 |      107 | 2020-12-14 |         69.0000
 Paris    |       11 |      104 | 2021-05-03 |        230.0000
(7 rows)

EXPLAIN SELECT * FROM order_totals_with_tax;
                            QUERY PLAN
------------------------------------------------------------------
 Hash Join  (cost=36.10..66.68 rows=1360 width=82)
   Hash Cond: (o.store_id = s.store_id)
   ->  Seq Scan on orders o  (cost=0.00..23.60 rows=1360 width=28)
   ->  Hash  (cost=21.60..21.60 rows=1160 width=42)
         ->  Seq Scan on stores s  (cost=0.00..21.60 rows=1160 width=42)
(5 rows)
```

```sql
SELECT store_id, sum(amount_plus_tax) AS store_totals
FROM (
  SELECT city, s.store_id, order_no, o_date, amount * 1.15 AS amount_plus_tax
  FROM stores AS s JOIN orders AS o
  ON s.store_id = o.store_id
) AS order_totals_with_tax
WHERE o_date >= '2021-01-01'
GROUP BY store_id;
```

```
SELECT store_id, sum(amount_plus_tax) AS store_totals
FROM order_totals_with_tax
WHERE o_date >= '2021-01-01'
GROUP BY store_id;

 store_id | store_totals
----------+--------------
       13 |    138.0000
       11 |    230.0000
       12 |    172.5000
(3 rows)

EXPLAIN SELECT store_id, sum(amount_plus_tax) AS store_totals
FROM order_totals_with_tax
WHERE o_date >= '2021-01-01'
GROUP BY store_id;

                                QUERY PLAN
--------------------------------------------------------------------------
 HashAggregate  (cost=67.69..73.35 rows=453 width=36)
   Group Key: s.store_id
   ->  Hash Join  (cost=36.10..64.29 rows=453 width=20)
         Hash Cond: (o.store_id = s.store_id)
         ->  Seq Scan on orders o  (cost=0.00..27.00 rows=453 width=20)
               Filter: (o_date >= '2021-01-01'::date)
         ->  Hash  (cost=21.60..21.60 rows=1160 width=4)
               ->  Seq Scan on stores s  (cost=0.00..21.60 rows=1160 width=4)
(8 rows)
```

```
SELECT * FROM customers;

 cust_id | cust_name |   address
---------+-----------+--------------
       1 | Alice     | 123 main st
       2 | Bob       | 45 west road
       3 | Charlie   | 6 Second Ave
       4 | Diana     | 78 River st
(4 rows)

CREATE VIEW customer_amounts AS
SELECT c.cust_id, cust_name, SUM(amount) AS total_amount
FROM customers AS c JOIN orders AS o
ON c.cust_id = o.cust_id
GROUP BY c.cust_id, cust_name;

CREATE VIEW

SELECT * FROM customers_amounts;

 cust_id | cust_name | total_amount
---------+-----------+--------------
       4 | Diana     |       180.00
       2 | Bob       |        80.00
       1 | Alice     |       200.00
       3 | Charlie   |       220.00
(4 rows)
```

```
\d orders;

                  Table "public.orders"
  Column  |     Type      | Collation | Nullable | Default
----------+---------------+-----------+----------+---------
 order_no | integer       |           | not null |
 cust_id  | integer       |           |          |
 amount   | numeric(10,2) |           |          |
 o_date   | date          |           |          |
 store_id | integer       |           | not null |
Indexes:
    "orders_pkey" PRIMARY KEY, btree (order_no)
Foreign-key constraints:
    "orders_cust_id_fkey" FOREIGN KEY (cust_id) REFERENCES customers(cust_id)
    "orders_store_id_fkey" FOREIGN KEY (store_id) REFERENCES stores(store_id)

SELECT * FROM orders;

 order_no | cust_id | amount |   o_date   | store_id
----------+---------+--------+------------+----------
      101 |       1 |  50.00 | 2020-11-30 |       13
      102 |       1 | 150.00 | 2021-06-08 |       12
      103 |       2 |  80.00 | 2020-12-25 |       13
      105 |       3 |  20.00 | 2020-12-24 |       12
      106 |       4 | 120.00 | 2021-02-21 |       13
      107 |       4 |  60.00 | 2020-12-14 |       12
      104 |       3 | 200.00 | 2021-05-03 |       11
(7 rows)

CREATE VIEW expensive_orders AS SELECT order_no, amount FROM orders WHERE amount > 100;
CREATE VIEW
```

```
SELECT * FROM expensive_orders;

 order_no | amount
----------+--------
      102 | 150.00
      106 | 120.00
      104 | 200.00
(3 rows)

UPDATE expensive_orders SET amount = 300 WHERE order_no = 104;
UPDATE 1

SELECT * FROM expensive_orders;

 order_no | amount
----------+--------
      102 | 150.00
      106 | 120.00
      104 | 300.00
(3 rows)

SELECT * FROM orders;

 order_no | cust_id | amount |   o_date   | store_id
----------+---------+--------+------------+----------
      101 |       1 |  50.00 | 2020-11-30 |       13
      102 |       1 | 150.00 | 2021-06-08 |       12
      103 |       2 |  80.00 | 2020-12-25 |       13
      105 |       3 |  20.00 | 2020-12-24 |       12
      106 |       4 | 120.00 | 2021-02-21 |       13
      107 |       4 |  60.00 | 2020-12-14 |       12
      104 |       3 | 300.00 | 2021-05-03 |       11
(7 rows)
```