**Исследуем данные ETL**

SELECT \* FROM customers;

SELECT \* FROM restaurants;

SELECT \* FROM orders;

SELECT \* FROM riders;

SELECT \* FROM deliveries;

SELECT COUNT(\*) FROM customers

WHERE

customer\_name IS NULL

OR

reg\_date IS NULL

SELECT COUNT(\*) FROM restaurants

WHERE

restaurant\_name IS NULL

OR

city IS NULL

OR

opening\_hours IS NULL

SELECT \* FROM orders

WHERE

order\_item IS NULL

OR

order\_date IS NULL

OR

order\_time IS NULL

OR

order\_status IS NULL

OR

total\_amount IS NULL

--Проверили на наличие null,не обнаружено, а так бы удалили нижеследующим кодом

--DELETE FROM orders

--WHERE

--order\_item IS NULL

**Отвечаем на вопросы**

--1 Напишите запрос, чтобы найти топ-5 наиболее часто заказываемых клиентом Арджун Мехта блюд за последний 1 год.

--нужен join customers orders

--фильтры-за последний год и Арджун Мехта

SELECT c.customer\_name, o.order\_item as dish, count(\*) as total\_orders

FROM orders as o

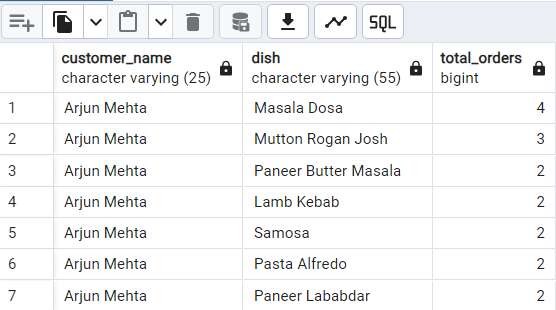
join customers as c

ON c.customer\_id=o.customer\_id

where (o.order\_date>= current\_date-interval '1 Year') and c.customer\_name='Arjun Mehta'

group by c.customer\_name, o.order\_item

order by total\_orders desc



--2.Определите временные интервалы, в течение которых размещается наибольшее количество заказов. Исходя из 2-часовых интервалов.

SELECT (CASE

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 0 AND 1 THEN '00:00 - 02:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 2 AND 3 THEN '02:00 - 04:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 4 AND 5 THEN '04:00 - 06:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 6 AND 7 THEN '06:00 - 08:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 8 AND 9 THEN '08:00 - 10:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 10 AND 11 THEN '10:00 - 12:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 12 AND 13 THEN '12:00 - 14:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 14 AND 15 THEN '14:00 - 16:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 16 AND 17 THEN '16:00 - 18:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 18 AND 19 THEN '18:00 - 20:00'

WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 20 AND 21 THEN '20:00 - 22:00'

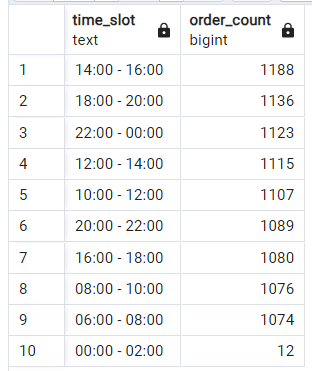
WHEN EXTRACT(HOUR FROM order\_time) BETWEEN 22 AND 23 THEN '22:00 - 00:00'

END) AS time\_slot , COUNT(order\_id) as order\_count

FROM Orders

GROUP BY time\_slot

ORDER BY order\_count DESC;



--3. Найдите среднюю стоимость заказа для каждого клиента, который разместил более 750 заказов.

SELECT c.customer\_name, COUNT(o.order\_id) as total\_orders, AVG(o.total\_amount) as avg\_money

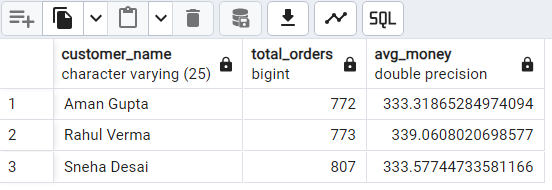
FROM orders o

JOIN customers c

ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_name

HAVING COUNT(order\_id) > 750



--4.Перечислите клиентов, которые потратили в общей сложности более 100 тысяч долларов на заказы еды.

SELECT c.customer\_name, SUM(o.total\_amount) as total\_money

FROM orders o

JOIN customers c

ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_name

HAVING SUM(o.total\_amount) > 100000



--5.Напишите запрос, чтобы найти заказы, которые были размещены, но не доставлены. Укажите название каждого ресторана, город и количество не доставленных заказов

SELECT r.restaurant\_name, r.city, COUNT(o.order\_id) as cnt\_not\_delivered\_orders

FROM orders as o

LEFT JOIN

restaurants r

ON r.restaurant\_id = o.restaurant\_id

LEFT JOIN

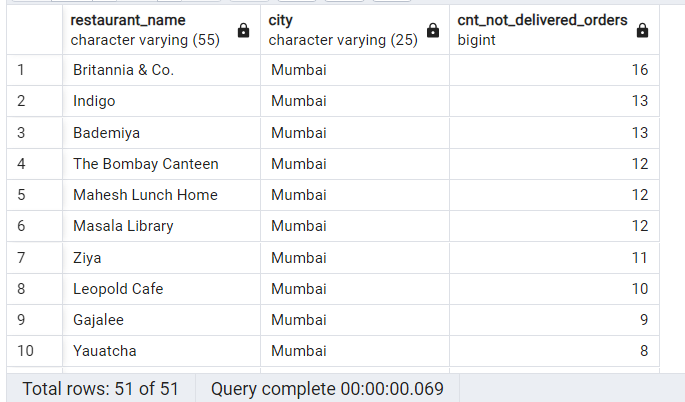
deliveries d

ON d.order\_id = o.order\_id

WHERE d.delivery\_id IS NULL

GROUP BY r.restaurant\_name,r.city

ORDER BY cnt\_not\_delivered\_orders DESC



--6.Проранжируйте рестораны по их общему доходу с клиентов за последний год для каждого города, выведите топовые рестораны для каждого города

WITH tab AS (SELECT r.city, r.restaurant\_name, SUM (o.total\_amount) as revenue,

DENSE\_RANK() OVER(PARTITION BY r.city ORDER BY SUM(o.total\_amount) DESC) as rank1

FROM orders o

JOIN

restaurants r

ON r.restaurant\_id = o.restaurant\_id

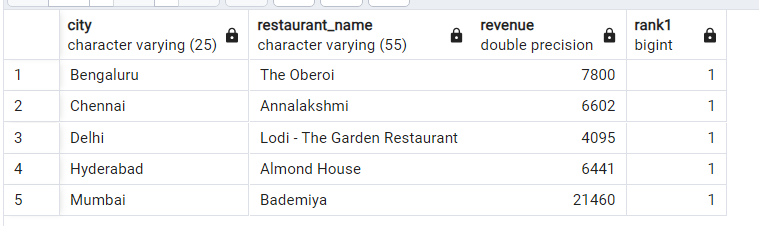
WHERE o.order\_date >= CURRENT\_DATE - INTERVAL '1 year'

GROUP BY r.city, r.restaurant\_name )

SELECT \*

FROM tab

WHERE rank1 = 1



--7. Определите самое популярное блюдо в каждом городе, основываясь на количестве заказов.

WITH cte as (SELECT r.city, o.order\_item as dish, COUNT(order\_id) as total\_orders, RANK() OVER(PARTITION BY r.city ORDER BY COUNT(order\_id) DESC) as rank1

FROM orders o

JOIN restaurants r

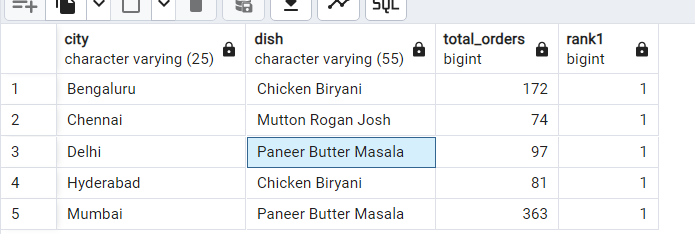
ON r.restaurant\_id = o.restaurant\_id

GROUP BY r.city, o.order\_item )

SELECT\*

from cte

WHERE rank1 = 1

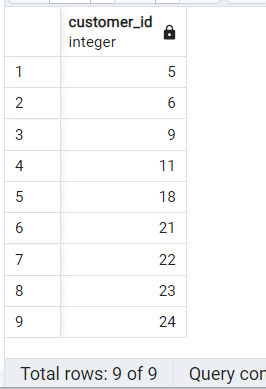


-- 8.Найдите клиентов, которые не размещали заказы в 2024 году, но сделали это в 2023 году.

SELECT DISTINCT customer\_id FROM orders

WHERE EXTRACT(YEAR FROM order\_date) = 2023

AND customer\_id NOT IN (SELECT DISTINCT customer\_id FROM orders WHERE EXTRACT(YEAR FROM order\_date) = 2024)



-- 9.Рассчитайте и сравните частоту отмены заказов в каждом ресторане за текущий и предыдущий годы.

WITH cte\_2023 AS (

SELECT o.restaurant\_id,

COUNT(o.order\_id) AS total\_orders,

COUNT(CASE WHEN d.delivery\_id IS NULL THEN 1 END) AS count\_not\_delivered2023

FROM orders o

LEFT JOIN deliveries d ON o.order\_id = d.order\_id

WHERE EXTRACT(YEAR FROM order\_date) = 2023

GROUP BY o.restaurant\_id

),

cte\_2024 AS (

SELECT o.restaurant\_id,

COUNT(o.order\_id) AS total\_orders,

COUNT(CASE WHEN d.delivery\_id IS NULL THEN 1 END) AS count\_not\_delivered2024

FROM orders o

LEFT JOIN deliveries d ON o.order\_id = d.order\_id

WHERE EXTRACT(YEAR FROM order\_date) = 2024

GROUP BY o.restaurant\_id

),

Rat\_2024 AS (

SELECT restaurant\_id,

total\_orders,

count\_not\_delivered2024,

ROUND((count\_not\_delivered2024::numeric / total\_orders::numeric) \* 100, 2) AS cancel\_rat24

FROM cte\_2024

),

Rat\_2023 AS (

SELECT restaurant\_id,

total\_orders,

count\_not\_delivered2023,

ROUND((count\_not\_delivered2023::numeric / total\_orders::numeric) \* 100, 2) AS cancel\_rat23

FROM cte\_2023

)

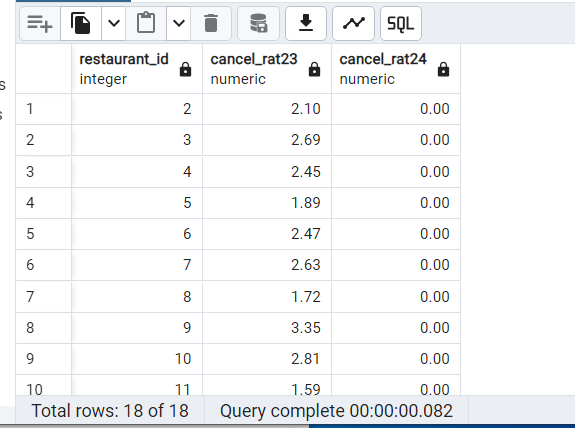
SELECT Rat\_2024.restaurant\_id,

Rat\_2023.cancel\_rat23,

Rat\_2024.cancel\_rat24

FROM Rat\_2023

JOIN Rat\_2024 ON Rat\_2023.restaurant\_id = Rat\_2024.restaurant\_id;



--10.Определите время доставки каждого доставщика.

SELECT d.rider\_id, o.order\_id,o.order\_time,d.delivery\_time,

d.delivery\_time - o.order\_time AS time\_difference,

ROUND(EXTRACT(EPOCH FROM (d.delivery\_time - o.order\_time +

CASE WHEN d.delivery\_time < o.order\_time THEN INTERVAL '1 day' ELSE

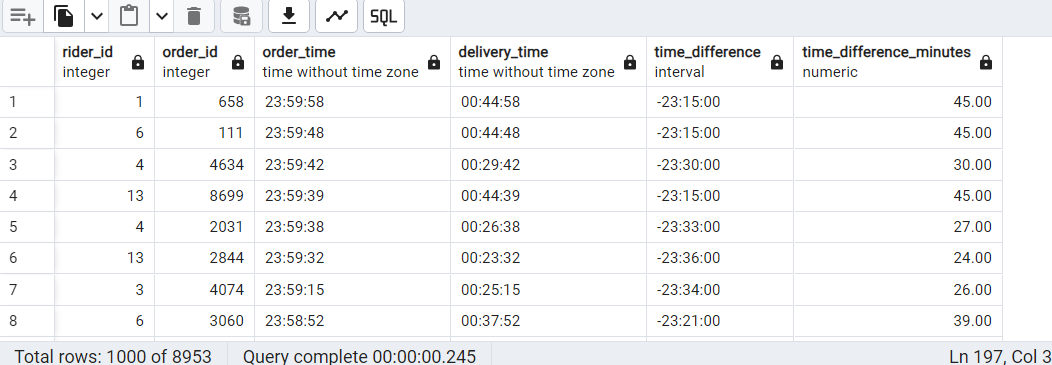
INTERVAL '0 day' END))/60,2) as time\_difference\_minutes

FROM orders o

JOIN deliveries d

ON o.order\_id = d.order\_id

WHERE d.delivery\_status = 'Delivered'



--11. Рассчитайте рост заказов в процентах от предыдущего месяца для каждого ресторана на основе количества доставленных заказов.

WITH tab AS

(SELECT o.restaurant\_id,

EXTRACT(YEAR FROM o.order\_date) as year1,

EXTRACT(MONTH FROM o.order\_date) as month1,

COUNT(o.order\_id) as month\_orders,

LAG(COUNT(o.order\_id), 1) OVER(PARTITION BY o.restaurant\_id ORDER BY EXTRACT(YEAR FROM o.order\_date), EXTRACT(MONTH FROM o.order\_date)) as prev\_month\_orders

FROM orders o

JOIN deliveries d

ON o.order\_id = d.order\_id

WHERE d.delivery\_status = 'Delivered'

GROUP BY o.restaurant\_id, EXTRACT(YEAR FROM o.order\_date), EXTRACT(MONTH FROM o.order\_date)

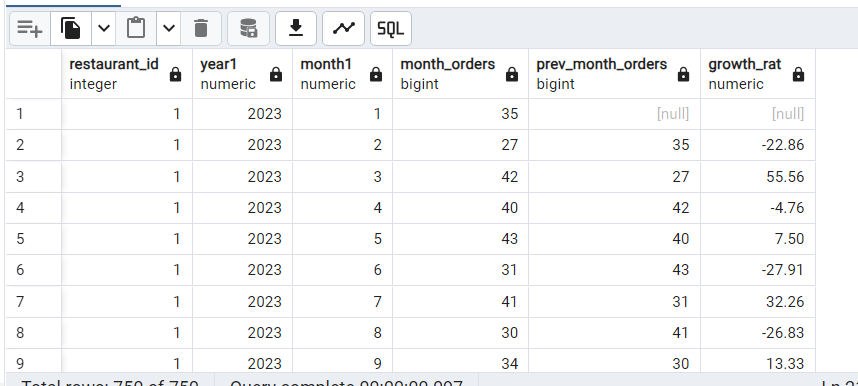
ORDER BY o.restaurant\_id, year1, month1

)

SELECT restaurant\_id, year1, month1, month\_orders, prev\_month\_orders,

ROUND((month\_orders::numeric -prev\_month\_orders::numeric)/prev\_month\_orders::numeric \* 100,2) as growth\_rat

FROM tab;



--12.Сегментация клиентов: Разделите клиентов на "Золотые" или "Серебряные" группы на основе их общих расходов.

-- в сравнении со средней стоимостью заказа (AOV). Если общие расходы клиента превышают AOV,

-- обозначьте их как "Золотые", в противном случае - как "Серебряные". Напишите SQL-запрос, чтобы определить

-- общее количество заказов в каждом сегменте и общий доход.

WITH tab AS

(SELECT customer\_id,

SUM(total\_amount) as total\_spent,

COUNT(order\_id) as total\_orders,

CASE WHEN SUM(total\_amount) > (SELECT AVG(total\_amount) FROM orders) THEN 'Gold' ELSE 'silver' END as category

FROM orders

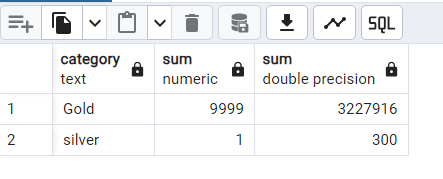
GROUP BY customer\_id

)

SELECT category, SUM(total\_orders), SUM (total\_spent)

FROM tab

GROUP BY category



--13. Рассчитайте общий ежемесячный заработок каждого доставщика, предполагая, что он зарабатывает 8% от суммы заказа.

SELECT

d.rider\_id,

TO\_CHAR(o.order\_date, 'mm-yy') as month1,

SUM(total\_amount) as revenue,

SUM(total\_amount)\* 0.08 as riders\_earning

FROM orders as o

JOIN deliveries as d

ON o.order\_id = d.order\_id

GROUP BY d.rider\_id,TO\_CHAR(o.order\_date, 'mm-yy')

ORDER BY d.rider\_id,month1

