



LARANA PIZZA

WELCOME TO
LARANA PIZZA

THIS IS A PROJECT ON PIZZA
SALES

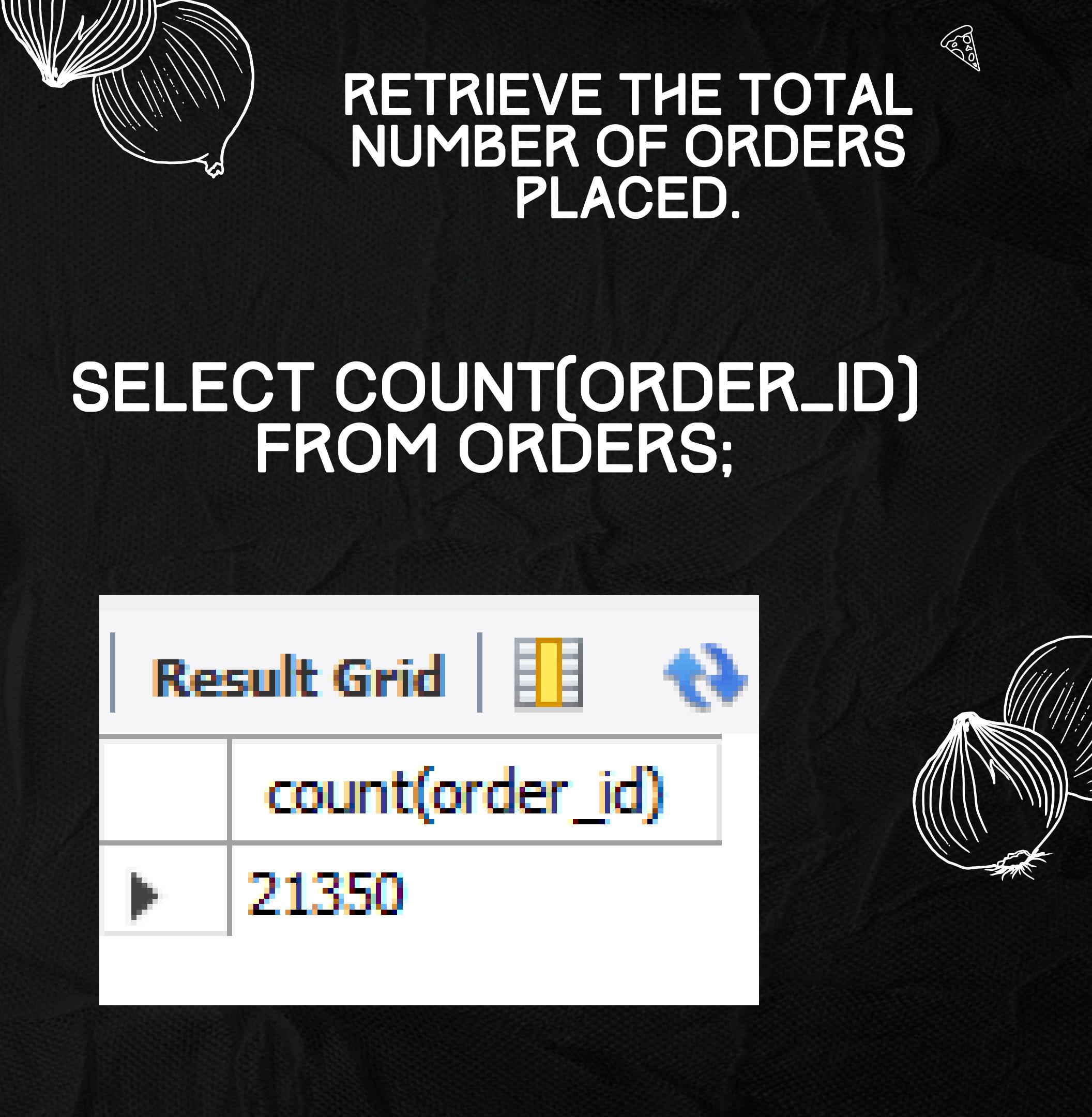




HELLO MY NAME
IS ABHAY AND IN
THIS PROJECT I
HAVE UTILIZED
SQL QUERIES TO
SOLVE
QUESTIONS THAT
WERE RELATED
TO PIZZA SALES

MEET OUR CHEF

- 1 Basic:
 - 2 Retrieve the total number of orders placed.
 - 3 Calculate the total revenue generated from pizza sales.
 - 4 Identify the highest-priced pizza.
 - 5 Identify the most common pizza size ordered.
 - 6 List the top 5 most ordered pizza types along with their quantities.
- 7
- 8
- 9 Intermediate:
 - 10 Join the necessary tables to find the total quantity of each pizza category ordered.
 - 11 Determine the distribution of orders by hour of the day.
 - 12 Join relevant tables to find the category-wise distribution of pizzas.
 - 13 Group the orders by date and calculate the average number of pizzas ordered per day.
 - 14 Determine the top 3 most ordered pizza types based on revenue.
- 15
- 16 Advanced:
 - 17 Calculate the percentage contribution of each pizza type to total revenue.
 - 18 Analyze the cumulative revenue generated over time.



RETRIEVE THE TOTAL
NUMBER OF ORDERS
PLACED.

SELECT COUNT(ORDER_ID)
FROM ORDERS;

Result Grid

	count(order_id)
▶	21350



CALCULATE THE TOTAL
REVENUE GENERATED
FROM PIZZA SALES.

SELECT SUM(QUANTITY * PRICE)FROM
ORDERS_DETAILS AS LEFTT JOIN
PIZZAS AS RIGHTT ON LEFTT.PIZAA_ID =
RIGHTT.PIZZA_ID;

Result Grid

Filter

	sum(quantity * price)
▶	817860.049999993





IDENTIFY THE HIGHEST-PRICED PIZZA.

SELECT NAME,PRICE FROM
PIZZA_TYPES AS LEFTT JOIN PIZZAS AS
RIGHTT ON LEFTT.PIZZA_TYPE_ID =
RIGHTT.PIZZA_TYPE_ID ORDER BY
PRICE DESC;

Result Grid Filter Rows:

	name	price
▶	The Greek Pizza	35.95
	The Greek Pizza	25.5
	The Brie Carre Pizza	23.65
	The Italian Vegetables Pizza	21
	The Barbecue Chicken Pizza	20.75
	The Spinach Supreme Pizza	20.75
	The Italian Supreme Pizza	20.75
	The California Chicken Pizza	20.75





IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

SELECT PIZZAS.SIZE,
COUNT(ORDERS_DETAILS.ORDERS_DETAILS_ID) FROM PIZZAS

JOIN ORDERS_DETAILS ON PIZZAS.PIZZA_ID =
ORDERS_DETAILS.PIZZA_ID GROUP BY PIZZAS.SIZE ORDER
BY COUNT(ORDERS_DETAILS.ORDERS_DETAILS_ID) DESC ;

Result Grid | Filter Rows:

	size	count(orders_details.orders_details_id)
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28





LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

SELECT
PIZZA_TYPES.NAME, SUM(ORDERS_DETAILS.QUANTITY) FROM ORDERS_DETAILS JOIN
PIZZAS ON ORDERS_DETAILS.PIZAA_ID =
PIZZAS.PIZZA_ID

JOIN PIZZA_TYPES ON PIZZA_TYPES.PIZZA_TYPE_ID =
PIZZAS.PIZZA_TYPE_ID GROUP BY PIZZA_TYPES.NAME ORDER BY
SUM(ORDERS_DETAILS.QUANTITY) DESC LIMIT 5;

Result Grid | Filter Rows: Export:

	name	sum(orders_details.quantity)
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371





JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

SELECT
PIZZA_TYPES.CATEGORY,SUM(ORDERS_DETAILS.QUANTITY) FROM ORDERS_DETAILS
JOIN PIZZAS ON
ORDERS_DETAILS.PIZAA_ID =
PIZZAS.PIZZA_ID

JOIN PIZZA_TYPES ON PIZZA_TYPES.PIZZA_TYPE_ID =
PIZZAS.PIZZA_TYPE_ID GROUP BY PIZZA_TYPES.CATEGORY ;

Result Grid | Filter Rows:

	category	sum(orders_details.quantity)
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050





DETERMINE THE
DISTRIBUTION OF ORDERS
BY HOUR OF THE DAY.

SELECT HOUR(ORDER_TIME),
COUNT(ORDER_ID) FROM ORDERS
GROUP BY HOUR(ORDER_TIME);

Result Grid | Filter Rows:

	hour(order_time)	count(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399

Result 11 ×





JOIN RELEVANT TABLES TO
FIND THE CATEGORY-WISE
DISTRIBUTION OF PIZZAS.

SELECT CATEGORY, COUNT(NAME)
FROM PIZZA_TYPES GROUP BY
CATEGORY;

Result Grid | Filter Rows

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9





GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

SELECT AVG(QUANTITY) FROM
(SELECT ORDERS.ORDER_DATE,
SUM(ORDERS_DETAILS.QUANTITY) AS
QUANTITY FROM ORDERS JOIN
ORDERS_DETAILS ON ORDERS.ORDER_ID=
ORDERS_DETAILS.ORDER_ID GROUP BY
ORDERS.ORDER_DATE) AS
ORDER_QUANTITY;

Result Grid

	avg(quantity)
▶	138.4749





DETERMINE THE TOP 3
MOST ORDERED PIZZA
TYPES BASED ON REVENUE.

```
SELECT PIZZA_TYPES.NAME,  
SUM(ORDERS_DETAILS.QUANTITY *  
PIZZAS.PRICE) FROM  
ORDERS_DETAILS JOIN PIZZAS ON  
ORDERS_DETAILS.PIZAA_ID =  
PIZZAS.PIZZA_ID  
JOIN PIZZA_TYPES ON  
PIZZA_TYPES.PIZZA_TYPE_ID =  
PIZZAS.PIZZA_TYPE_ID  
GROUP BY PIZZA_TYPES.NAME ORDER BY  
SUM(ORDERS_DETAILS.QUANTITY *  
PIZZAS.PRICE) DESC LIMIT 3 ;
```

Result Grid		Filter Rows:	Export:	Wrap Cell
	name	sum(orders_details.quantity * pizzas.price)		
▶	The Thai Chicken Pizza	43434.25		
	The Barbecue Chicken Pizza	42768		
	The California Chicken Pizza	41409.5		





CALCULATE THE PERCENTAGE
CONTRIBUTION OF EACH PIZZA
TYPE TO TOTAL REVENUE.

```
SELECT PIZZA_TYPES.CATEGORY,  
ROUND(SUM(ORDER_DETAILS.QUANTITY* PIZZAS.PRICE) /  
(SELECT ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),2)  
AS TOTAL_SALES FROM ORDERS_DETAILS  
JOIN PIZZAS ON PIZZAS.PIZZA_ID =ORDERS_DETAILS.PIZAA_ID)  
*100,2) AS REVENUE  
FROM PIZZA_TYPES JOIN PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID =  
PIZZAS.PIZZA_TYPE_ID  
JOIN ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID =  
PIZZAS.PIZZA_ID  
GROUP BY PIZZA_TYPES.CATEGORY ORDER BY REVENUE DESC;
```

Default Grid Filter Rows

	Category	Revenue
1	Classic	25.91
	Supreme	25.45
	Chicken	21.94
	Veggie	22.68





ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT ORDER_DATE, SUM(REVENUE) OVER(ORDER BY ORDER_DATE) AS CUM_REVENUE
FROM (SELECT ORDERS.ORDER_DATE,
SUM(ORDERS_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
FROM ORDERS_DETAILS JOIN PIZZAS ON
ORDERS_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
JOIN ORDERS ON ORDERS.ORDER_ID = ORDERS_DETAILS.ORDER_ID
GROUP BY ORDERS.ORDER_DATE) AS SALES;
```

Result Grid | Filter Rows:

	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05

Result 16 ×





ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT ORDER_DATE, SUM(REVENUE) OVER(ORDER BY ORDER_DATE) AS CUM_REVENUE
  FROM (SELECT ORDERS.ORDER_DATE,
    SUM(ORDERS_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
      FROM ORDERS_DETAILS JOIN PIZZAS ON
    ORDERS_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
    JOIN ORDERS ON ORDERS.ORDER_ID = ORDERS_DETAILS.ORDER_ID
    GROUP BY ORDERS.ORDER_DATE) AS SALES;
```

Result Grid | Filter Rows:

	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05

Result 16 ×





LARANA PIZZA

THANK YOU!

