

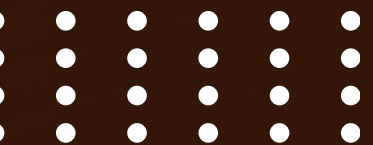
Where Every Slice is a Taste of Perfection

SQL PROJECT ON PIZZA SALES



By Suryansh
Thakur

Start Your Slide





HELLO!

I am Suryansh Thakur, and I am pleased to present this comprehensive SQL report on pizza sales. The following report provides a thorough analysis of pizza sales, leveraging SQL queries to derive key insights and trends. By examining the sales data from various perspectives, this report aims to present a clear picture of the performance of different pizza types, the revenue generated over specific time periods, and the overall sales growth.

Questions



Basic:

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

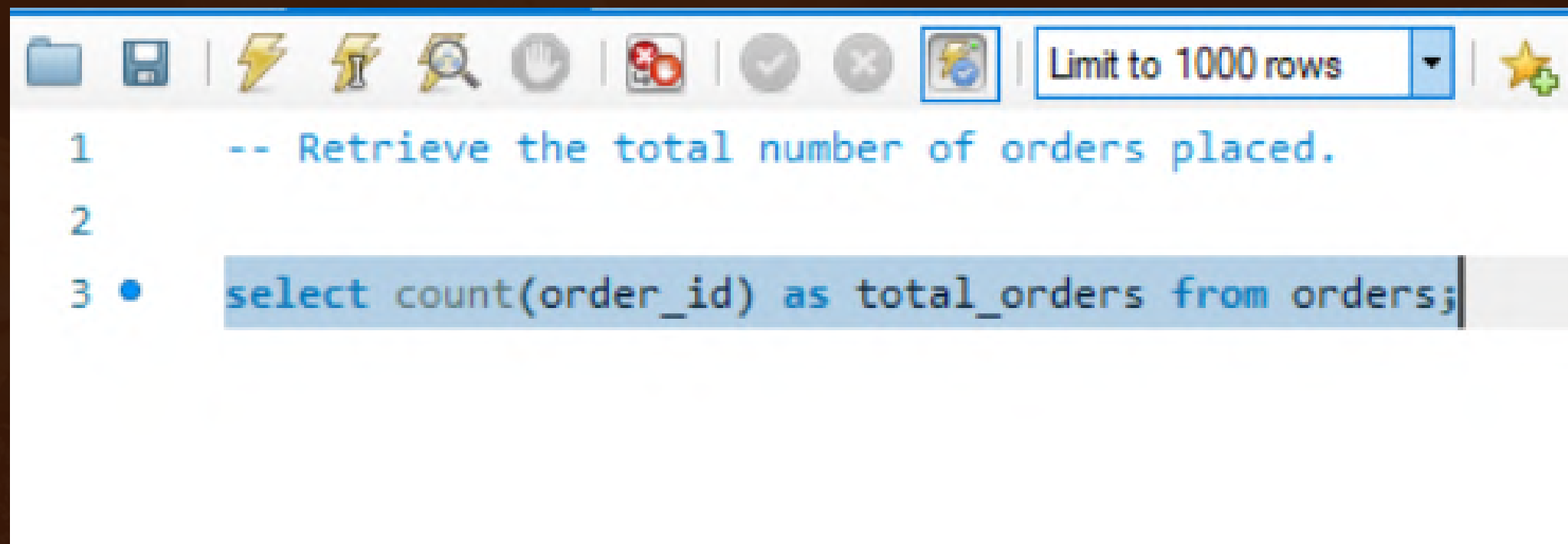
Intermediate:

6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.

Advanced:

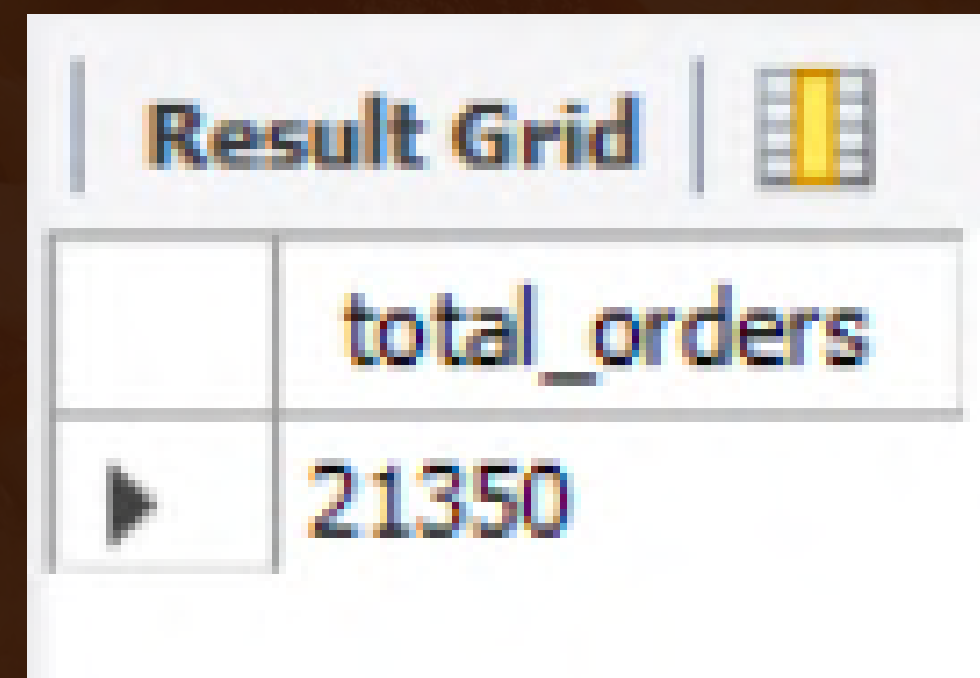
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED. :::::



A screenshot of a SQL query editor window. The toolbar at the top includes icons for file operations, execution, and a dropdown menu set to "Limit to 1000 rows". The query text is as follows:

```
1  -- Retrieve the total number of orders placed.  
2  
3  • select count(order_id) as total_orders from orders;
```



A screenshot of a "Result Grid" window. It displays a single row of results for the query. The column header is "total_orders" and the value is "21350".

	total_orders
▶	21350



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT  
    ROUND(SUM(order_details.quantity * pizzas.price),  
          2) AS total_sales  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid	
	total_sales
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

```
1  -- Identify the highest-priced pizza.
2
3  •  SELECT
4      pizza_types.name, pizzas.price
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9  ORDER BY pizzas.price DESC
10 LIMIT 1;
```

Result Grid			Filter R
	name	price	
▶	The Greek Pizza	35.95	

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



```
1  -- Identify the most common pizza size ordered.
2
3  •  SELECT
4      pizzas.size,
5      COUNT(order_details.order_details_id) AS order_count
6  FROM
7      pizzas
8      JOIN
9      order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC;
```

Result Grid			Filter
	size	order_count	
▶	L	18526	
	M	15385	
	S	14137	
	XL	544	
	XXL	28	



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

```
1  -- Identify the most common pizza size ordered.
2
3  • SELECT
4      pizzas.size,
5      COUNT(order_details.order_details_id) AS order_count
6  FROM
7      pizzas
8      JOIN
9      order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC;
```

Result Grid			Filter
	size	order_count	
▶	L	18526	
	M	15385	
	S	14137	
	XL	544	
	XXL	28	

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

```
1  -- Join the necessary tables to find the
2  -- total quantity of each pizza category ordered.
3
4  • SELECT
5      pizza_types.category,
6      SUM(order_details.quantity) AS quantity
7  FROM
8      pizza_types
9      JOIN
10     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11     JOIN
12     order_details ON order_details.pizza_id = pizzas.pizza_id
13 GROUP BY pizza_types.category
14 ORDER BY quantity DESC;
15
```

Result Grid			Filter
	category	quantity	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	

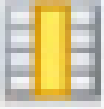

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
1  -- Determine the distribution of orders by hour of the day.
2
3  ●  SELECT
4      HOUR(order_time) AS hour, COUNT(order_id) AS order_count
5  FROM
6      orders
7  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	
	19	2009	
	20	1642	
	21	1198	

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

```
1  -- Join relevant tables to find the
2  -- category-wise distribution of pizzas.
3
4  ●  SELECT
5      category, COUNT(name)
6
7  FROM
8      pizza_types
9  GROUP BY category
```

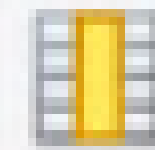
Result Grid				
	category	count(n		
▶	Chicken	Resets a		
	Classic	8		
	Supreme	9		
	Veggie	9		

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

```
-- Group the orders by date and calculate the average  
-- number of pizzas ordered per day.
```

```
SELECT  
    ROUND(AVG(quantity), 0) as avg_pizzas_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date, SUM(order_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid



Filter Rows

	avg_pizzas_ordered_per_day
--	----------------------------

	138
--	-----

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

```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2
3  •  select pizza_types.name,
4     sum(order_details.quantity * pizzas.price) as revenue
5  from pizza_types join pizzas
6     on pizzas.pizza_type_id = pizza_types.pizza_type_id
7  join order_details
8     on order_details.pizza_id = pizzas.pizza_id
9  group by pizza_types.name order by revenue desc limit 3;
```

Result Grid



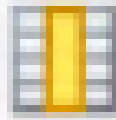

Filter Rows:

	name	revenue
▶	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.



```
1  -- Calculate the percentage contribution of each
2  -- pizza type to total revenue.
3
4  • SELECT
5      pizza_types.category,
6      ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
7          ROUND(SUM(order_details.quantity * pizzas.price),
8              2) AS total_sales
9          FROM
10             order_details
11             JOIN
12                 pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
13          2) AS revenue
14  FROM
15      pizza_types
16      JOIN
17      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
18      JOIN
19      order_details ON order_details.pizza_id = pizzas.pizza_id
20  GROUP BY pizza_types.category
```

Result Grid   Filter Rows		
	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.





```
1  -- Analyze the cumulative revenue generated over time.
2
3
4  •  SELECT order_date,
5         SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
6
7  FROM (
8         SELECT orders.order_date,
9                SUM(order_details.quantity * pizzas.price) AS revenue
10        FROM order_details
11        JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
12        JOIN orders ON orders.order_id = order_details.order_id
13        GROUP BY orders.order_date
14    ) AS sales;
```

Result Grid			Filter Rows:
	order_date	cum_revenue	
▶	2015-01-01	2713.8500000000000004	
	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	

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

```
1  -- Determine the top 3 most ordered pizza types
2  -- based on revenue for each pizza category.
3
4  • select name , revenue from
5  (select category, name , revenue,
6   rank() over(partition by category order by revenue desc) as rn
7   from
8   (select pizza_types.category, pizza_types.name,
9    sum((order_details.quantity) * pizzas.price) as revenue
10   from pizza_types join pizzas
11    on pizza_types.pizza_type_id = pizzas.pizza_type_id
12   join order_details
13    on order_details.pizza_id = pizzas.pizza_id
14   group by pizza_types.category, pizza_types.name) as a) as b
15  where rn <=3 ;
16
```

Result Grid   Filter Rows: <input type="text"/>		
	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	22476.75

Result 4 x

**THANK YOU
FOR ATTENTION**

