



Pizza Track



SQL Project





Hello, I'm Subhash

Data Analyst

- Utilized SQL queries to analyze pizza sales data and answer business-related questions.
 - Examined sales performance, customer preferences, and order trends for key insights.
 - Optimized queries to drive data-driven decisions for business growth and revenue analysis.
- 
- 

Calculate the total revenue generated from pizza sales.

```
1  -- Calculate the total revenue generated from pizza sales.
2  • SELECT
3      ROUND(SUM((order_details.quantity * pizzas.price)),
4              2) AS total_sales
5  FROM
6      order_details
7      JOIN
8      pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid		
	total_sales	
▶	817860.05	

Identify the highest-priced pizza.

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

Result Grid



total_sales

817860.05

Identify the most common pizza size ordered.

```
1  -- Identify the most common pizza size ordered.
2  ● SELECT
3      p.size, COUNT(od.order_details_id) as order_count
4  FROM
5      pizzas p
6      JOIN
7      order_details od ON p.pizza_id = od.pizza_id
8  GROUP BY p.size
9  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

```
-- List the top 5 most ordered pizza types along with their quantities
```

```
• SELECT
    pt.name, SUM(od.quantity) AS quantity
FROM
    pizza_types pt
    JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid			Filter Rows:
	name	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.

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

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 hours, COUNT(order_id) AS order_ib
5  FROM
6      orders
7  GROUP BY hours
```

Result Grid				
	hours	order_ib		
▶	11	1231		
	12	2520		
	13	2455		
	14	1472		
	15	1468		
	16	1920		
	17	2336		

Retrieve the total number of orders placed.

```
1  -- Join relevant tables to find the category-wise distribution of pizzas.
2
3 • SELECT category, COUNT(name) from pizza_types
4   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.

...

```
1  -- Group the orders by date and
2  -- calculate the average number of pizzas ordered per day.
3
4  • SELECT
5      ROUND(AVG(quantity), 0) as avq_pizzas_ordered_per_day
6  FROM
7      (SELECT
8          o.order_date, COUNT(od.quantity) AS quantity
9      FROM
10         orders o
11        JOIN order_details od ON o.order_id = od.order_id
12       GROUP BY order_date) AS order_quantity;
```

Result Grid



Filter Rows:

	avq_pizzas_ordered_per_day
--	----------------------------

▶	136
---	-----

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
4      pt.name, SUM(p.price * od.quantity) AS revenue
5  FROM
6      pizza_types pt
7      JOIN
8      pizzas p ON pt.pizza_type_id = p.pizza_type_id
9      JOIN
10     order_details od ON od.pizza_id = p.pizza_id
11 GROUP BY pt.name
12 ORDER BY revenue DESC
13 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 pizza type to total revenue.
2
3  SELECT
4      pt.category,
5      round(SUM(p.price * od.quantity) / (SELECT
6          ROUND(SUM(order_details.quantity * pizzas.price),
7              2) AS total_sales
8          FROM
9              order_details
10             JOIN
11                 pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,2) AS revenue
12  FROM
13      pizza_types pt
14      JOIN
15      pizzas p ON pt.pizza_type_id = p.pizza_type_id
16      JOIN
17      order_details od ON od.pizza_id = p.pizza_id
18  GROUP BY pt.category;
```

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

Analyze the cumulative revenue generated over time.

```
1  -- Analyze the cumulative revenue generated over time.
2
3  • SELECT order_date,
4     sum(revenue) over (order by order_date) as cum_revenue
5  from
6  (SELECT o.order_date,
7     sum(od.quantity * p.price) as revenue
8  from order_details od join pizzas p
9   on od.pizza_id = p.pizza_id
10 join
11 orders o on o.order_id = od.order_id
12 GROUP BY o.order_date) 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	

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  • select name, revenue
4  from
5  (SELECT category, name, revenue,
6   rank() over (PARTITION BY category ORDER BY revenue DESC) as rn
7   from
8   (SELECT pt.category, pt.name,
9    sum(od.quantity * p.price) as revenue
10   from pizza_types pt join pizzas p
11    on pt.pizza_type_id = p.pizza_type_id
12   join order_details od on od.pizza_id = p.pizza_id
13   GROUP BY pt.category, pt.name) as A) as B
14 where rn <= 3;
```

Result Grid			Filter Rows:
	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	