



SQL PROJECT ON PIZZA SALES

DANISH ZAIDI



Hello everyone my name is Danish. In this project I have utilized SQL queries to solve all the question related to Pizza sales.

Retrieve the total number of orders placed.

- `SELECT COUNT(order_id) AS total_orders
FROM orders;`

Result Grid	
	<code>total_orders</code>
▶	21350



Calculate the total revenue generated from pizza sales.

- **SELECT**

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

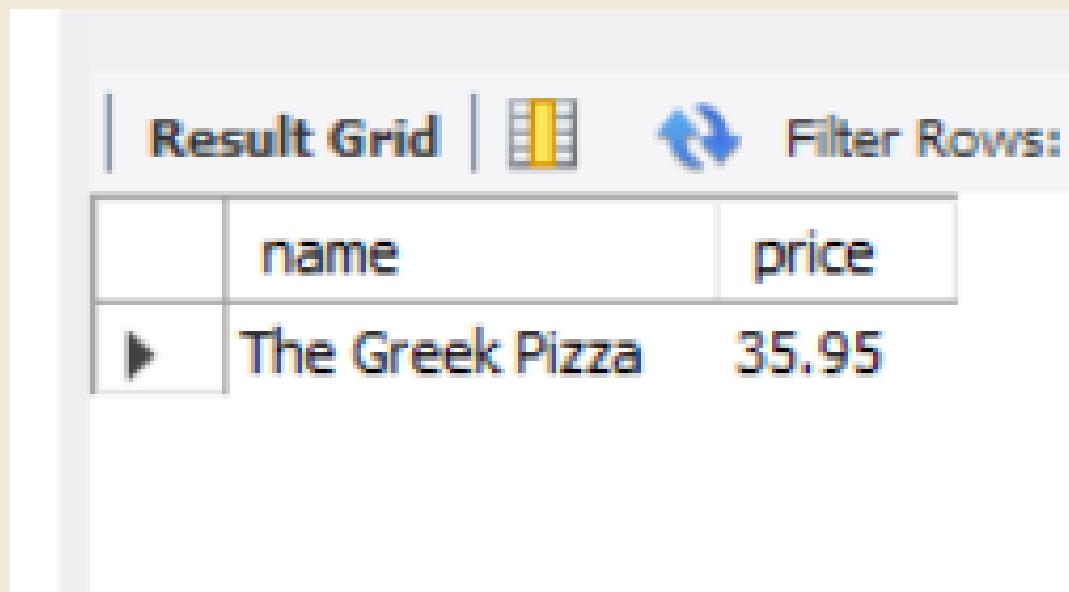
Result Grid	
	<u>total_order</u>
▶	817860.05



Identify the highest-priced pizza.

- **SELECT**

```
    pizza_types.name, pizzas.price  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY pizzas.price DESC
```



The screenshot shows a MySQL Workbench result grid. The grid has two columns: 'name' and 'price'. There is one row of data: 'The Greek Pizza' with a price of '35.95'. The grid includes standard database navigation buttons like 'Result Grid', 'Filter Rows', and a search icon.

	name	price
▶	The Greek Pizza	35.95



Identify the most common pizza size ordered.

```
32 • SELECT
33     pizzas.size,
34     COUNT(order_details.order_details_id) AS total_orders
35 FROM
36     pizzas
37     JOIN
38         order_details ON pizzas.pizza_id = order_details.pizza_id
39 GROUP BY pizzas.size
40 ORDER BY total_orders DESC;
```

Result Grid | Filter Rows:

	size	total_orders
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

```
5 • SELECT
6     pizza_types.name, SUM(order_details.quantity) AS quantity
7 FROM
8     pizzas
9     JOIN
10    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
11    JOIN
12    order_details ON order_details.pizza_id = pizzas.pizza_id
13 GROUP BY pizza_types.name
14 ORDER BY quantity DESC
15 LIMIT 5;
```

	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

```
61 • SELECT
62     pizza_types.category,
63     SUM(order_details.quantity) AS quantity
64 FROM
65     pizzas
66     JOIN
67     pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
68     JOIN
69     order_details ON order_details.pizza_id = pizzas.pizza_id
70 GROUP BY pizza_types.category
71 ORDER BY quantity DESC;
72
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



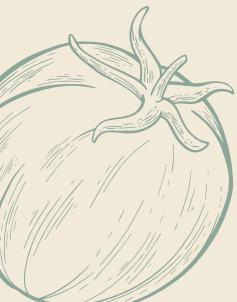
Determine the distribution of orders by hour of the day

```
76 • SELECT
77      HOUR(order_time) AS hours, COUNT(order_id) AS order_count
78  FROM
79    orders
80 GROUP BY HOUR(order_time);
81
```

Result Grid | Filter Rows

	hours	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009

Result 10 ×



Join relevant tables to find the category-wise distribution of pizzas

```
85 • SELECT
```

```
86      category, COUNT(name)
```

```
87 FROM
```

```
88 pizza_types
```

```
89 GROUP BY category;
```

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

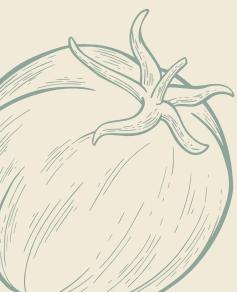


Group the orders by date and calculate the average number of pizzas ordered per day

1 • SELECT

```
5     ROUND(AVG(quantity), 0) AS avg_pizza_order_perday
5 FROM
7   (SELECT
3       orders.order_date, SUM(order_details.quantity) AS quantity
9 FROM
0   orders
L JOIN order_details ON orders.order_id = order_details.order_id
2 GROUP BY orders.order_date) AS order_quantity;
```

	avg_pizza_order_perday
▶	138



Determine the top 3 most ordered pizza types based on revenue

```
07 • SELECT
08     pizza_types.name,
09     SUM(order_details.quantity * pizzas.price) AS revenue
10    FROM
11        pizza_types
12            JOIN
13        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
14            JOIN
15        order_details ON order_details.pizza_id = pizzas.pizza_id
16    GROUP BY pizza_types.name
17    ORDER BY revenue DESC
18    LIMIT 3;
```



	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

```
3 • SELECT |
4     pizza_types.category,
5     ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
6         ROUND(SUM(order_details.quantity * pizzas.price),
7             2)
8     )
9     FROM
10        order_details
11        JOIN
12            pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
13    2) AS revenue
14
15    FROM
16        pizza_types
17        JOIN
18            pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
19        JOIN
20            order_details ON order_details.pizza_id = pizzas.pizza_id
21
22    GROUP BY pizza_types.category
23
24    ORDER BY revenue DESC;
```

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.

```
SELECT order_date, SUM(revenue) OVER(ORDER BY order_date) AS cum_revenue
FROM
(SELECT orders.order_date,
SUM(order_details.quantity * pizzas.price) AS revenue
FROM order_details JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id
JOIN orders
ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) AS sales;
```

Result Grid | Filter Rows:

	order_date	cum_revenue
▶	2015-01-01	2713.850000000004
	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
	2015-01-09	21526.4

Result 20 ×

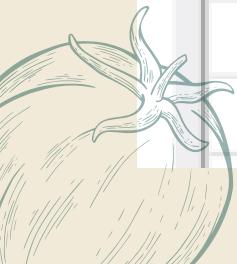


Determine the top 3 most ordered pizza types based on revenue for each pizza category

```
L58 • SELECT name, revenue
L59   FROM
L60   (SELECT category, name, revenue,
L61     RANK() OVER(partition by category ORDER BY revenue DESC) AS rn
L62   FROM
L63   (SELECT pizza_types.category, pizza_types.name,
L64     SUM(order_details.quantity * pizzas.price) AS revenue
L65   FROM pizza_types JOIN pizzas
L66     ON pizza_types.pizza_type_id = pizzas.pizza_type_id
L67   JOIN order_details
L68     ON order_details.pizza_id = pizzas.pizza_id
L69   GROUP BY pizza_types.category, pizza_types.name) AS a) AS b
L70   WHERE rn <= 3;
L71
```

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





THANK
YOU



DANISH ZAIDI