



Hi, my name is Pratik Pawar, and I am a B-Tech student specializing in Artificial Intelligence and Data Science. For my SQL project, I have created a database system to manage and analyze pizza sales. This project demonstrates my ability to design and implement a relational database, query data for insights, and apply my learning in a real-world scenario.



RETRIEVE THE TOTAL NUMBER OF ORDER PLACED



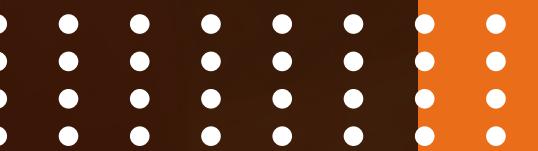
```
3
4 •  SELECT
5      *
6  FROM
7      orders;
8 •  SELECT
9      COUNT(order_id) AS total_order
10  FROM
11     orders;
12
```

ES

Result Grid

| | total_order |
|---|-------------|
| ▶ | 21350 |

calculate the total revenue generated from pizza sales



```
# 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 |

Highest priced of pizza



```
1  # highest priced of 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 Rows

| | name | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |

Identify the most common pizza size ordered



```
1  # identify the most common pizza size ordered #
2 • SELECT
3      pizzas.size,
4      COUNT(order_details.order_details_id) AS order_count
5  FROM
6      pizzas
7      JOIN
8          order_details ON pizzas.pizza_id = order_details.pizza_id
9  GROUP BY pizzas.size
10 ORDER BY order_count DESC; # limit1
```

Result Grid

| | 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;
```

Result Grid | Filter F

| | category | quantity |
|---|----------|----------|
| ▶ | Classic | 14888 |
| | Supreme | 11987 |
| | Veggie | 11649 |
| | Chicken | 11050 |

list the top 5 most ordered pizza
type along with their quantities



```
1  # list the top 5 most ordered pizza
2  # type along with their quantities
3
4 • SELECT
5      pizza_types.name, SUM(order_details.quantity) AS quantity
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10     JOIN
11     order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY quantity DESC limit 5; # limtt 5
14
```

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 |

Determine the distribution of orders by hour of the day



```
3 •  SELECT
4      HOUR(order_time), COUNT(order_id)
5  FROM
6  orders
7 GROUP BY HOUR(order_time);
```

| | HOUR(order_time) | COUNT(order_id) |
|--|------------------|-----------------|
| | 9 | 1 |
| | 10 | 8 |
| | 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



```
2
3
4 •  SELECT
5      category, COUNT(name)
6  FROM
7      pizza_types
8  GROUP BY category;
```

Result Grid | Filter Rows:

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

```
1      # group the orders by date the calculate the ave
2      # number of pizzas ordered per day
3 •  SELECT
4          AVG(quantity)
5      FROM
6      ( SELECT
7          orders.order_date, SUM(order_details.quantity) AS quantity
8      FROM
9          orders
10     JOIN order_details ON orders.order_id = order_details.order_id
11     GROUP BY orders.order_date) AS order_quantity;
```

| 11 | GROUP | | | | |
|------|--|--|---------------|---|----------|
| ures | | | | | |
| | Result Grid   | | | | |
| | <table border="1"><thead><tr><th></th><th>AVG(quantity)</th></tr></thead><tbody><tr><td>▶</td><td>138.4749</td></tr></tbody></table> | | AVG(quantity) | ▶ | 138.4749 |
| | AVG(quantity) | | | | |
| ▶ | 138.4749 | | | | |

determine the top 3 ordered pizza type based on revenue



```
1  # determine the top 3 ordered pizza type based on revenue
2 • SELECT
3      pizza_types.name,
4      SUM(order_details.quantity * pizzas.price) AS revenue
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9      JOIN
10     order_details ON order_details.pizza_id = pizzas.pizza_id
11    GROUP BY pizza_types.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 |

analyze the cumulative revenue generated over time.



```
1  # analyze the cumulative revenue generated over time.
2  • select order_date,
3    sum(revenue)over (order by order_date)as cum_revenue
4    from
5    (select orders.order_date,
6      sum(order_details.quantity*pizzas.price) as revenue
7      from order_details join pizzas
8      on order_details.pizza_id = pizzas.pizza_id
9      join orders
10     on orders.order_id = order_details.order_id
11    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 |
| | 2015-01-09 | 21526.4 |
| | 2015-01-10 | 23990.350000000002 |
| | 2015-01-11 | 25862.65 |
| | 2015-01-12 | 27781.7 |
| | 2015-01-13 | 29831.300000000003 |



THANK YOU