**In this project, we’ll look into a fictional pizza delivery database to uncover valuable insights and trends that could drive better business decisions. Using SQL queries, we'll**

- Analyze order trends by calculating the total number of orders and revenue.
- Spot customer preferences by identifying the most popular pizza size and the top 5 pizza types ordered.
- Highlight key performers like the highest-priced pizza and top-revenue generators.
- Explore ordering patterns by breaking down order distribution by hours and days.
- Slice the data further to understand pizza demand by category and revenue contribution.
- Through table joins, groupings, and aggregations, this project showcases how SQL can turn raw data into actionable insights for any business.

# 1) Retrieve the total number of orders placed.

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

# 2) Calculate the total revenue generated from pizza sales.

```sql
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 |

# 3) Identify the highest-priced pizza.

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows

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

# 4) Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

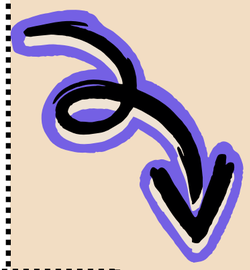
sult Grid | Filter Ro

| size | order_count |
|---|---|
| L | 18526 |
| M | 15385 |
| S | 14137 |
| XL | 544 |
| XXL | 28 |

15385

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

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
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.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 |

# 6) Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
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 quantity DESC;
```

Result Grid | Filter Rows:

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

# 7) Determine the distribution of orders by hour of the day.

```
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 | Filte

| hour | order_count |
|---|---|
| 11 | 1231 |
| 12 | 2520 |
| 13 | 2455 |
| 14 | 1472 |
| 15 | 1468 |
| 16 | 1920 |
| 17 | 2336 |
| 18 | 2399 |
| 19 | 2009 |
| 20 | 1642 |
| 21 | 1198 |
| 22 | 663 |

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

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

Result Grid | Filter Rows:

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

# 9) Determine the top 3 most ordered pizza types based on revenue.

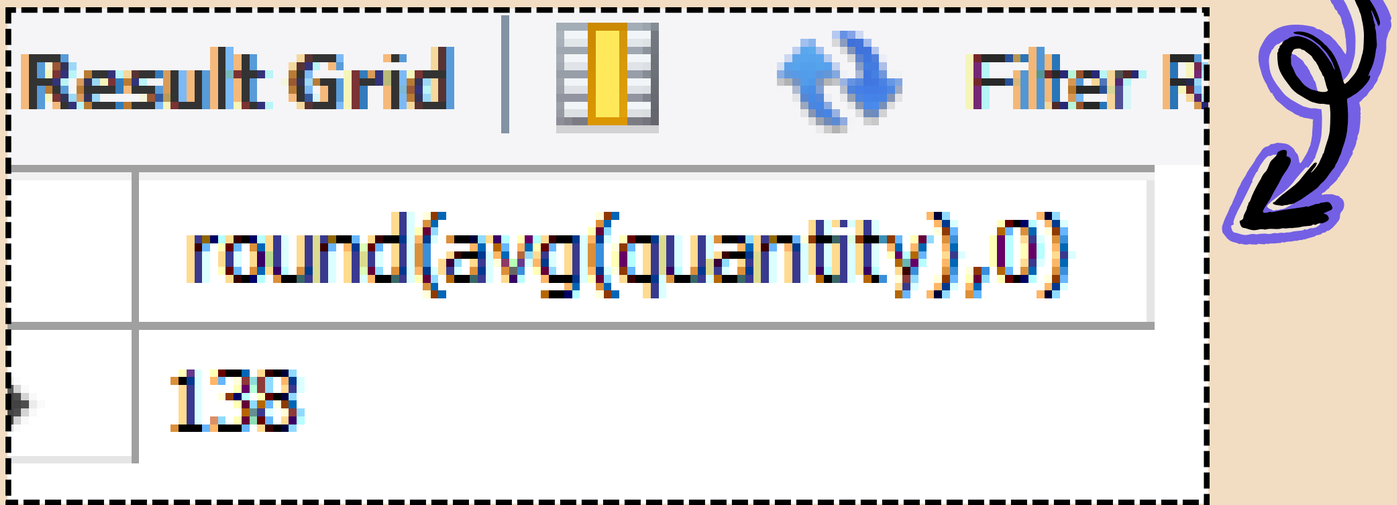
```
3 •   select pizza_types.name,
4     sum(order_details.quantity * pizzas.price) as revenue
5     from pizza_types join pizzas
6     on pizzas.pizza_type_id = pizza_types.pizza_type_id
7     join
8     order_details
9     on order_details.pizza_id = pizzas.pizza_id
10    group by pizza_types.name 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 |

# 10) Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizzas_ordered_perday
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;
```



# THANKS