



PIZZA

SALES REPORT



HELLO

MY NAME IS ASHISH KUMAR, AND IN THIS PROJECT, I HAVE EMPLOYED SQL QUERIES TO EFFICIENTLY ADDRESS AND SOLVE COMPLEX QUESTIONS RELATED TO PIZZA SALES.



QUESTIONS

Basic:

Retrieve the total number of orders placed.

Calculate the total revenue generated from pizza sales.

Identify the highest-priced pizza.

Identify the most common pizza size ordered.

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

Intermediate:

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

Determine the distribution of orders by hour of the day.

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

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

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

Advanced:

Calculate the percentage contribution of each pizza type to total revenue.

Analyze the cumulative revenue generated over time.

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



Retrieve the total number of orders placed.

```
3 •  SELECT
4      COUNT(order_id) AS total_orders
5  FROM
6      order_type;
```

Result ~

Result Grid	
	total_orders
	21350

Calculate the total revenue generated from pizza sales.

```
3 •  SELECT
4   ⊖    ROUND(SUM(order_details.quantity * pizzas.price),
5              2) AS total_sales
6   FROM
7     order_details
8   JOIN
9     pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result ~

total_sales
817860.05

Identify the highest-priced pizza.

```
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 ~

Result Grid   Filter Rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

```
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 ~

Result Grid   Filter Rows:

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.

```
3 •  SELECT
4      pizza_types.name, SUM(order_details.quantity) AS quantity
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizza_types.pizza_type_id = pizzas.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 quantity DESC
13    LIMIT 5;
```

Result ~

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.

```
3 •  SELECT
4      pizza_types.category,
5          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.category
13     ORDER BY quantity DESC;
```

Result ~

Result Grid

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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      order_type
7  GROUP BY HOUR(order_time);
```

Result ~

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
23	28
10	8
9	1

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

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

Result ~

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.

```
3 •  SELECT
4      ROUND(AVG(quantity), 0) as average_pizza_ordered_per_day
5  FROM
6  (SELECT
7      order_type.order_date,
8      SUM(order_details.quantity) AS quantity
9  FROM
10  order_type
11  JOIN order_details ON order_type.order_id = order_details.order_id
12  GROUP BY order_type.order_date) AS order_quantity;
```

Result ~

average_pizza_ordered_per_day
138

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

```
3 •  SELECT
4      pizza_types.name,
5      SUM(order_details.quantity * pizzas.price) AS revenue
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizzas.pizza_type_id = pizza_types.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 revenue DESC
14  LIMIT 3;
```

Result ~

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.

```
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) AS total_sales
8
9          FROM
10         order_details
11        JOIN
12          pizzas ON pizzas.pizza_id = order_details.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        GROUP BY pizza_types.category
22        ORDER BY revenue DESC;
```

Result ~

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```
3 •  SELECT
4      order_date,
5      SUM(revenue) OVER(ORDER BY order_date) AS cum_revenue
6  FROM (
7      SELECT
8          ot.order_date,
9          SUM(od.quantity * p.price) AS revenue
10     FROM
11         order_details od
12     JOIN
13         pizzas p ON od.pizza_id = p.pizza_id
14     JOIN
15         order_type ot ON ot.order_id = od.order_id
16     GROUP BY
17         ot.order_date
18 ) AS sales;
```

Result ~

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

Determine the top 3 most ordered pizza types 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 pizza_types.category, pizza_types.name,
9           sum((order_details.quantity) * pizzas.price) AS revenue
10    FROM pizza_types JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11      JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
12      GROUP BY pizza_types.category, pizza_types.name
13  ) AS a
14 ) AS b
15 WHERE rn <= 3;
```

Result ~

	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	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5