

DATA ANALYTICS WITH SQL





PIZZAHUT SALES

Portfolio Project By
Qurban Khan



PIZZAHUT SALES DATA

Tables

Name	
	orders
	orders_details
	pizza_types
	pizzas

Columns

Table	Column	Type
orders	◇ order_id	int
orders	◇ order_date	date
orders	◇ order_time	time
orders_details	◇ order_details_id	int
orders_details	◇ order_id	int
orders_details	◇ pizza_id	text
orders_details	◇ quantity	int
pizza_types	◇ pizza_type_id	text
pizza_types	◇ name	text
pizza_types	◇ category	text
pizza_types	◇ ingredients	text
pizzas	◇ pizza_id	text
pizzas	◇ pizza_type_id	text
pizzas	◇ size	text
pizzas	◇ price	double

QUESTIONS

Basic:

- Q1 - Retrieve the total number of orders placed.
- Q2 - Calculate the total revenue generated from pizza sales.
- Q3 - Identify the highest-priced pizza.
- Q4 - Identify the most common pizza size ordered.
- Q5 - List the top 5 most ordered pizza types along with their quantities.

Intermediate:

- Q1 - Join the necessary tables to find the total quantity of each pizza category ordered.
- Q2 - Determine the distribution of orders by hour of the day.
- Q3 - Join relevant tables to find the category-wise distribution of pizzas.
- Q4 - Group the orders by date and calculate the average number of pizzas ordered per day.
- Q5 - Determine the top 3 most ordered pizza types based on revenue.

Advanced:

- Q1 - Calculate the percentage contribution of each pizza type to total revenue.
- Q2 - Analyze the cumulative revenue generated over time.
- Q3 - Determine the top 3 most ordered pizza types based on revenue for each pizza category.

BASIC

Q1 - Retrieve the total number of orders placed.

Query.

```
1 • SELECT
2 COUNT(order_id) AS total_orders
3 FROM
4 orders;
```

Result

Result Grid	
	total_orders
▶	21350




BASIC

Q2 - Calculate the total revenue generated from pizza sales.

Query:

```
1 • SELECT
2     ROUND(SUM(od.quantity * p.price), 2) AS total_revenue
3 FROM
4     orders_details od
5     JOIN
6     pizzas p ON p.pizza_id = od.pizza_id;
```

Result:

Result Grid			
	total_revenue		
	817860.05		




BASIC

Q3 - Identify the highest-priced pizza.

Query:

```
1 • SELECT
2     pt.name, p.*
3 FROM
4     pizzas p
5     JOIN
6     pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
7 ORDER BY price DESC
8 LIMIT 1;
```

Result:

Result Grid   Filter Rows: <input type="text"/> Export:  <input type="text"/>					
	name	pizza_id	pizza_type_id	size	price
▶	The Greek Pizza	the_greek_xxl	the_greek	XXL	35.95

BASIC

Q4 - Identify the most common pizza size ordered.

Query:

```
1 • SELECT
2     p.size, SUM(od.quantity) AS order_quantity
3 FROM
4     orders_details od
5     JOIN
6     pizzas p ON p.pizza_id = od.pizza_id
7 GROUP BY p.size
8 ORDER BY order_quantity DESC
9 LIMIT 1;
```

Result:

Result Grid			Filter F
	size	order_quantity	
▶	L	18956	



BASIC

Q5 - List the top 5 most ordered pizza types along with their quantities.

Query:

```
1 • SELECT
2     pt.name, SUM(od.quantity) AS total_quantity
3 FROM
4     pizza_types pt
5     JOIN
6     pizzas p ON p.pizza_type_id = pt.pizza_type_id
7     JOIN
8     orders_details od ON od.pizza_id = p.pizza_id
9 GROUP BY pt.name
10 ORDER BY total_quantity DESC
11 LIMIT 5;
```

Result:

Result Grid   Filter Rows:		
	name	total_quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



INTERMEDIATE

Q1 - Join the necessary tables to find the total quantity of each pizza category ordered.

Query:

```
1 • SELECT
2     pt.category, SUM(od.quantity) AS total_quantity
3 FROM
4     pizza_types pt
5     JOIN
6     pizzas p ON p.pizza_type_id = pt.pizza_type_id
7     JOIN
8     orders_details od ON od.pizza_id = p.pizza_id
9 GROUP BY pt.category
10 ORDER BY total_quantity DESC;
```

Result:

Result Grid   Filter Rows		
	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

INTERMEDIATE

Q2 - Determine the distribution of orders by hour of the day.

Query:

```
1 • SELECT
2     HOUR(order_time) AS hour, COUNT(order_id) AS total_orders
3 FROM
4     orders
5 GROUP BY hour
6 ORDER BY hour ASC;
```

Result:

Result Grid			Filter
	hour	total_orders	
▶	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	
	22	663	
	23	28	

INTERMEDIATE

Q3 - Join relevant tables to find the category-wise distribution of pizzas.

Query:

```
1 • SELECT
2     pt.category, SUM(od.quantity) AS qty_sold
3 FROM
4     pizza_types pt
5     JOIN
6     pizzas p ON p.pizza_type_id = pt.pizza_type_id
7     JOIN
8     orders_details od ON od.pizza_id = p.pizza_id
9 GROUP BY pt.category;
```

Result:

Result Grid			Filter
	category	qty_sold	
▶	Classic	14888	
	Veggie	11649	
	Supreme	11987	
	Chicken	11050	

INTERMEDIATE

Q4 - Group the orders by date and calculate the average number of pizzas ordered per day.

Query:

```
1 • SELECT
2     ROUND(AVG(qty), 2) AS avg_per_day
3 FROM
4     (SELECT
5         o.order_date AS date, SUM(od.quantity) AS qty
6     FROM
7         orders o
8     JOIN orders_details od ON od.order_id = o.order_id
9     GROUP BY date
10    ORDER BY date ASC) AS sql;
```

Result:

Result Grid	
	avg_per_day
▶	138.47

INTERMEDIATE

Q5 - Determine the top 3 most ordered pizza types based on revenue.

Query:

```
1 • SELECT
2     pt.name, SUM(p.price * od.quantity) AS revenue
3 FROM
4     pizza_types pt
5     JOIN
6     pizzas p ON p.pizza_type_id = pt.pizza_type_id
7     JOIN
8     orders_details od ON od.pizza_id = p.pizza_id
9 GROUP BY pt.name
10 ORDER BY revenue DESC
11 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	

ADVANCED

Q1 - Calculate the percentage contribution of each pizza type to total revenue.

Query:

```
1 • SELECT
2     pt.name,
3     ROUND((SUM(p.price * od.quantity) / (SELECT
4         SUM(od.quantity * p.price)
5     FROM
6         orders_details od
7     JOIN
8         pizzas p ON p.pizza_id = od.pizza_id) * 100),2) AS per_revenue
9 FROM
10    pizza_types pt
11    JOIN
12    pizzas p ON p.pizza_type_id = pt.pizza_type_id
13    JOIN
14    orders_details od ON od.pizza_id = p.pizza_id
15 GROUP BY pt.name
16 ORDER BY per_revenue DESC;
```

Result:

Result Grid			Filter Rows:
	name	per_revenue	
▶	The Thai Chicken Pizza	5.31	
	The Barbecue Chicken Pizza	5.23	
	The California Chicken Pizza	5.06	
	The Classic Deluxe Pizza	4.67	
	The Spicy Italian Pizza	4.26	
	The Southwest Chicken Pizza	4.24	
	The Italian Supreme Pizza	4.09	
	The Hawaiian Pizza	3.95	
	The Four Cheese Pizza	3.95	
	The Sicilian Pizza	3.78	
	The Pepperoni Pizza	3.69	
	The Greek Pizza	3.48	
	The Mexicana Pizza	3.27	
	The Five Cheese Pizza	3.19	
	The Pepper Salami Pizza	3.12	
	The Italian Capocollo Pizza	3.07	
	The Vegetables + Vegetable...	2.98	
	The Prosciutto and Arugula ...	2.96	
	The Napolitana Pizza	2.95	
	The Spinach and Feta Pizza	2.85	
	The Big Meat Pizza	2.81	
	The Pepperoni, Mushroom, ...	2.3	
	The Chicken Alfredo Pizza	2.07	
	The Chicken Pesto Pizza	2.04	
	The Soppressata Pizza	2.01	
	The Italian Vegetables Pizza	1.96	
	The Calabrese Pizza	1.95	
	The Spinach Pesto Pizza	1.91	
	The Mediterranean Pizza	1.88	
	The Spinach Supreme Pizza	1.87	
	The Green Garden Pizza	1.71	
	The Brie Carre Pizza	1.42	



ADVANCED

Q2 - Analyze the cumulative revenue generated over time.

Query:

```
1 • SELECT order_date, round(sum(revenue) over (order by order_date),2) as cum_revenue
2 FROM
3 (
4 SELECT orders.order_date, SUM(pizzas.price * orders_details.quantity) AS revenue
5 FROM orders
6 JOIN orders_details
7 ON orders_details.order_id = orders.order_id
8 JOIN pizzas
9 ON pizzas.pizza_id = orders_details.pizza_id
10 GROUP BY orders.order_date
11 )
12 as sq3;
```

Result:

Result Grid   Filter Rows: <input type="text"/>		
	order_date	cum_revenue
▶	2015-01-01	2713.85
	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	23990.35



ADVANCED

Q3 - Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Query:

```
1 • SELECT *
2 FROM
3 (Select category, name, revenue, (RANK() over (Partition by category order by revenue)) as Ranked
4 FROM
5 (SELECT pt.category, pt.name, ROUND(SUM(od.quantity * p.price),2) as revenue
6 FROM pizza_types pt
7 JOIN pizzas p
8 ON p.pizza_type_id = pt.pizza_type_id
9 JOIN orders_details od
10 ON od.pizza_id = p.pizza_id
11 GROUP BY pt.category, pt.name
12 ORDER BY pt.category, revenue DESC, pt.name
13 ) a
14 ) b
15 WHERE ranked <= 3;
```

Result:

category	name	revenue	Ranked
Chicken	The Chicken Pesto Pizza	16701.75	1
Chicken	The Chicken Alfredo Pizza	16900.25	2
Chicken	The Southwest Chicken Pizza	34705.75	3
Classic	The Pepperoni, Mushroom, and Peppers Pizza	18834.5	1
Classic	The Big Meat Pizza	22968	2
Classic	The Napolitana Pizza	24087	3
Supreme	The Brie Carre Pizza	11588.5	1
Supreme	The Spinach Supreme Pizza	15277.75	2
Supreme	The Calabrese Pizza	15934.25	3
Veggie	The Green Garden Pizza	13955.75	1
Veggie	The Mediterranean Pizza	15360.5	2
Veggie	The Spinach Pesto Pizza	15596	3

The image features a white background with the text 'THANK YOU' centered. In the four corners, there are abstract, organic shapes in shades of orange and brown. These shapes have wavy, irregular edges and some contain small white dots or dashes, resembling stylized clouds or watercolor splatters.

THANK YOU