Pizza Sales Analysis Project

Objective: The Pizza Sales Analysis Project aims to uncover valuable insights from pizza sales data to inform business decisions. The project involves querying a comprehensive database of orders, pizzas, and order details to analyze sales patterns, revenue, and customer preferences.

Key Analyses:

- **Total Orders:** Determined the total number of orders placed.
- **Revenue Generation:** Calculated the total revenue from pizza sales.
- **Pricing Insights:** Identified the highest-priced pizza.
- **Order Patterns:** Analyzed the most common pizza size ordered and the top five most ordered pizza types.
- **Category Analysis:** Explored the total quantity of each pizza category ordered and the distribution of pizzas by category.
- **Temporal Trends:** Examined the distribution of orders by hour and the average number of pizzas ordered per day.
- **Revenue Insights:** Assessed the top three most ordered pizza types based on revenue and their percentage contribution to total revenue.
- **Cumulative Revenue:** Analyzed cumulative revenue generated over time.
- **Category-wise Revenue Leaders:** Determined the top three most ordered pizza types based on revenue for each category.

Database creation and table manipulation:

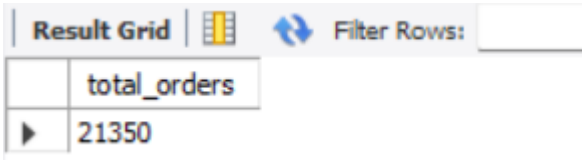
```
1  create database pizzahut;
2
3  create table orders (
4  order_id int not null,
5  order_date date not null,
6  order_time time not null,
7  primary key(order_id));
8
9  create table orders_details
10 (
11 order_details_id int not null,
12 order_id int not null,
13 pizza_id text not null,
14 quantity int not null,
15 primary key(order_details_id));
```

1 Retrieve the total number of orders placed.

Query:

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

Results:



Result Grid | Filter Rows:

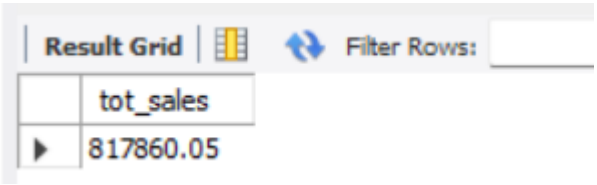
	total_orders
▶	21350

2 Calculate the total revenue generated from pizza sales.

Query:

```
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS tot_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Result:



Result Grid | Filter Rows:

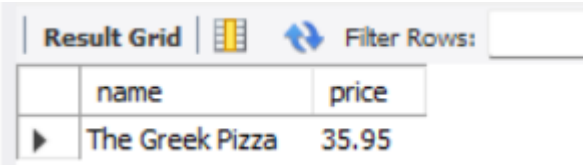
	tot_sales
▶	817860.05

3 Identify the highest-priced pizza.

Query:

```
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
LIMIT 1;
```

Result:



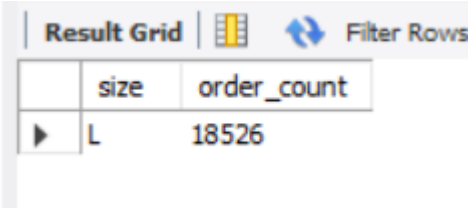
	name	price
▶	The Greek Pizza	35.95

4 Identify the most common pizza size ordered.

Query:

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

Result:



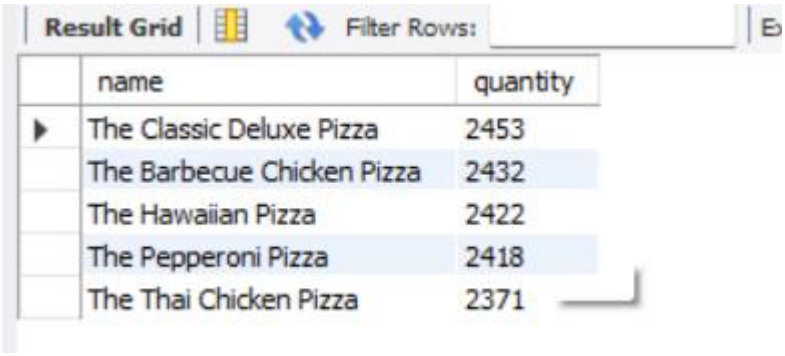
	size	order_count
▶	L	18526

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

Query:

```
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Result:



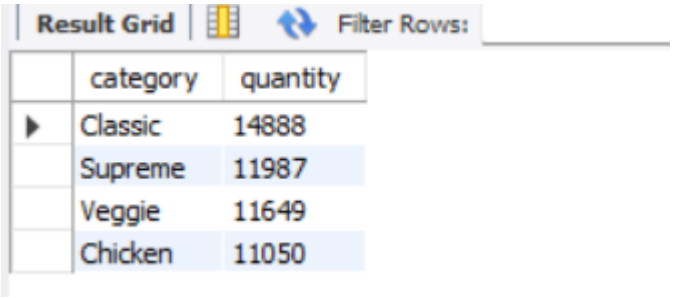
	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

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

Query:

```
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result:



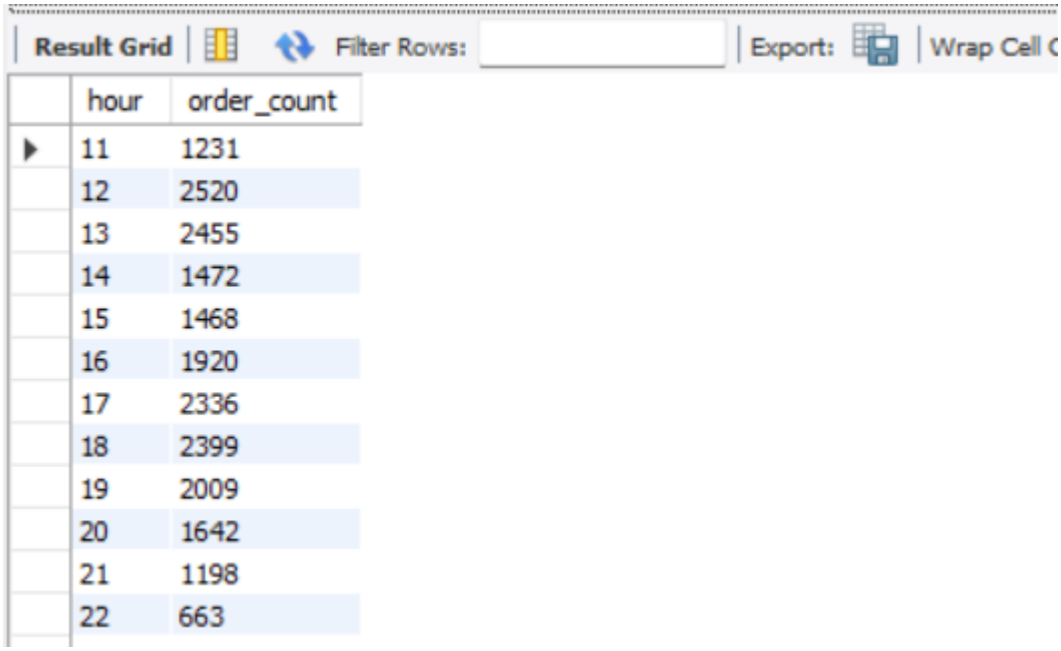
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

Query:

```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result:



Result Grid | Filter Rows: | Export: | Wrap Cell

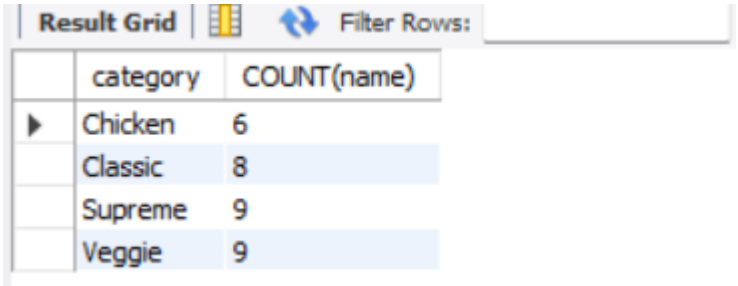
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663

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

Query:

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result:



Result Grid | Filter Rows:

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

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

Query:

```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizzaorder_perday
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result:

Result Grid | Filter Rows:

avg_pizzaorder_perday
138

10 Top three most ordered pizza types based on revenue

Query:

```
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result:

Result Grid | Filter Rows: | Expo

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

11 Percentage contribution of each pizza type to total revenue

Query:

```
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(orders_details.quantity * pizzas.price),
                                2) AS tot_sales
                FROM
                    orders_details
                        JOIN
                    pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
            2) AS revenue_percentage
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue_percentage DESC;
```

Result:

Result Grid | Filter Rows: | Exp

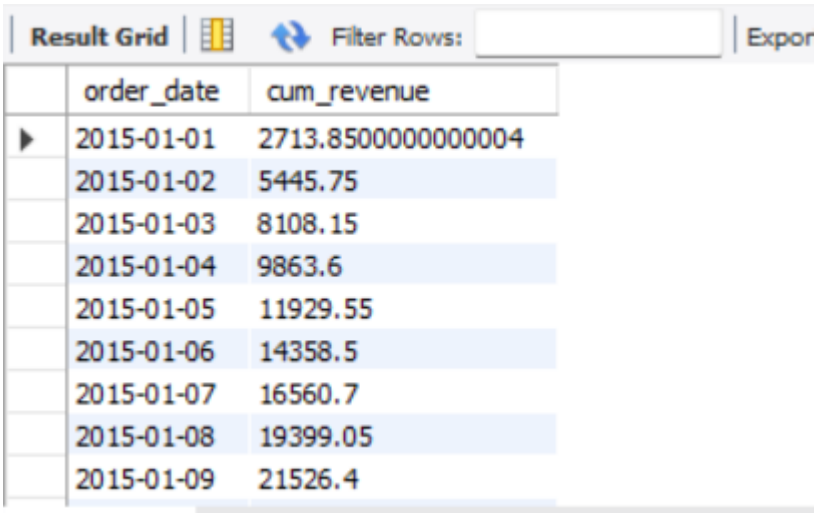
category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

12 Analyse the cumulative revenue generated over time

Query:

```
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id=pizzas.pizza_id
join orders on orders.order_id =orders_details.order_id
group by orders.order_date) as sales;
```

Result:



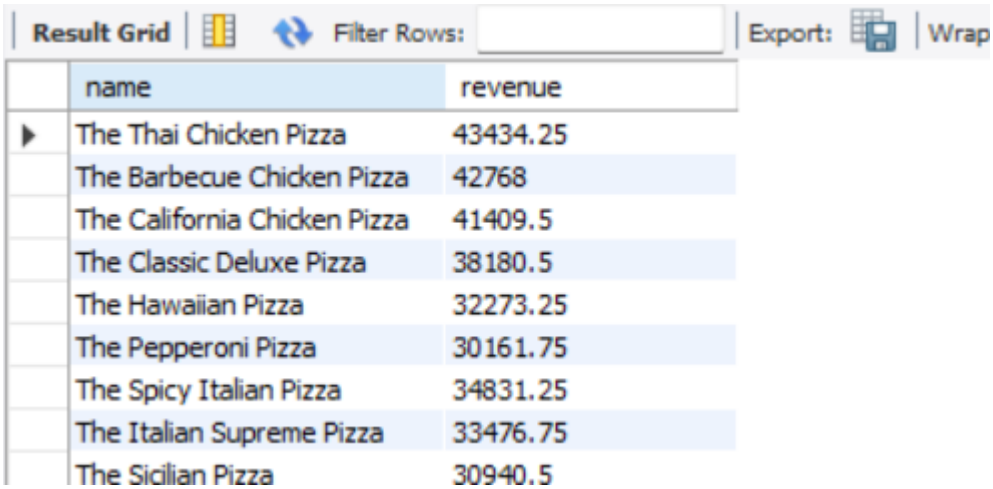
order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4

13 Determine top three most ordered pizza types based on revenue for each pizza category

Query:

```
select name, revenue from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum((orders_details.quantity)*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id=pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
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

Resources:

Dataset: [PizzaSales]