PIZZA SALES

Report on SQL

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.

QUESTIONS

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.

QUESTIONS

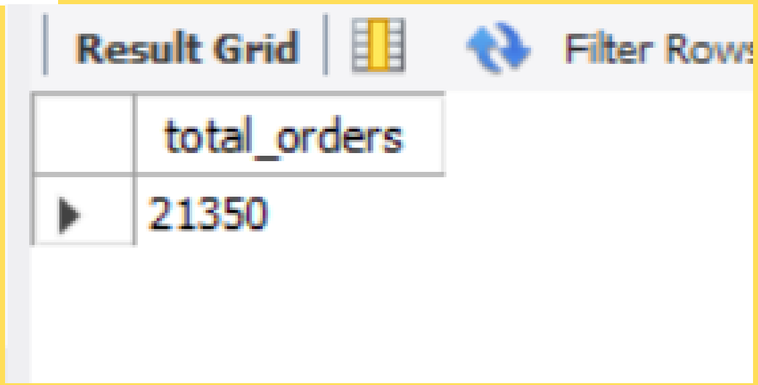
Advanced:

- 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.

SQL Queries

Retrieve the total number of orders placed.

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



SQL Queries

Calculate the total revenue generated from pizza sales.

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

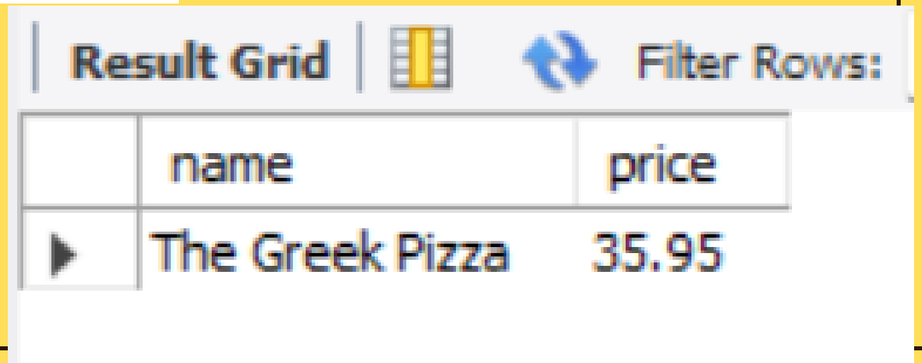
Result Grid

	total_cost
▶	817860.05

SQL Queries

Identify the highest-priced pizza.

```
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;
```



SQL Queries

Identify the most common pizza size ordered.

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

Result Grid | Filte

size	total_count
L	18526
M	15385
S	14137
XL	544
XXL	28

SQL Queries

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

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS total
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 total DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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

SQL Queries

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

```
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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

SQL Queries

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS hours, COUNT(order_id) AS total
FROM
    orders
GROUP BY hours;
```

Result Grid

hours	total
11	1231
12	2520
13	2455
14	1472
15	1468

SQL Queries

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

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

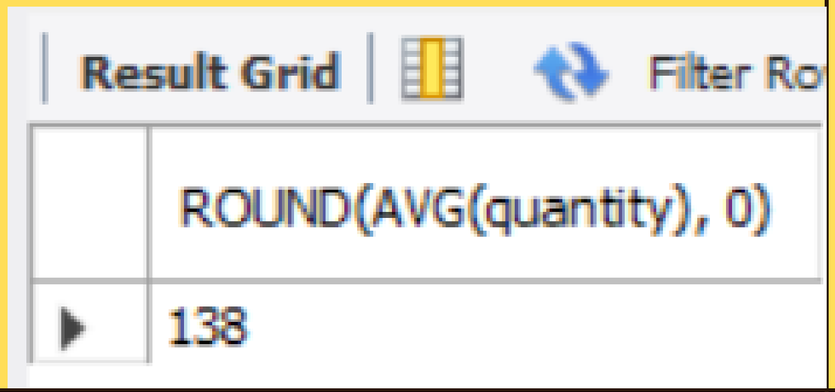
Result Grid | Filter Rows:

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

SQL Queries

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

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



SQL Queries

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
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 revenue DESC
```

Result Grid | Filter Rows:

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

SQL Queries

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

```sql
SELECT
    pizza_types.category,
    round((SUM(order_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM((order_details.quantity * pizzas.price)),
            2) AS total_cost
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id)) * 100 ,2) as revenue
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 revenue DESC
LIMIT 3;
```

Result Grid | Filter Row

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96

SQL Queries

Analyze the cumulative revenue generated over time.

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

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.850000000004
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

SQL Queries

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

```
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(order_details.quantity *
 pizzas.price) as revenue 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, pizza_types.name) as a) as b where rn <=3;
```

Result Grid | Filter Rows:

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

THANK YOU !