Home About Contact

SQL PROJECT ON PIZZA SALES

- WHERE EVERY SLICE TELLS A STORY

ABOUT

Hello my name is Kushagra Shukla. In this project I have utilised SQL queries to solve questions that were related to pizza sales.
I used multiple concepts of MySQL like JOINS , SUBQUERIES , RANK etc. in this project

Retrieve the total number of orders placed.

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

Result Grid

	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

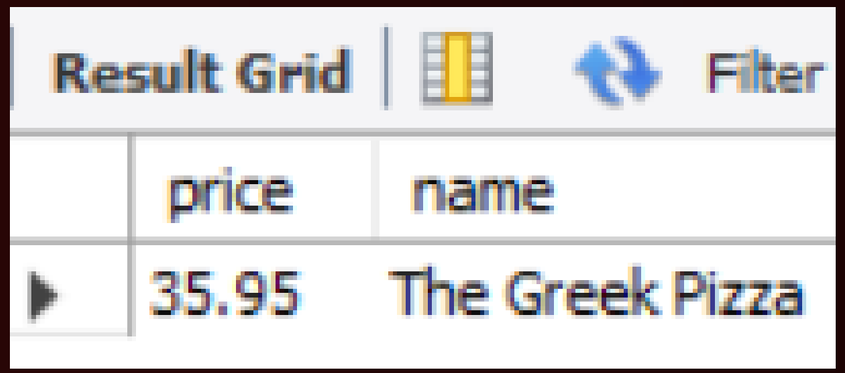
```sql
SELECT
    ROUND(SUM(od.quantity * p.price),2)
          AS total_revenue
FROM
    order_details od
        INNER JOIN
    pizzas p ON p.pizza_id = od.pizza_id;
```

Result Grid

	total_revenue
▶	817860.05

Identify the highest-priced pizza.

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



Determine the distribution of orders by hour of the day.

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

hours	order_count
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468

Identify the most common pizza size ordered.

```sql
SELECT
    p.size, COUNT(od.quantity) AS order_count
FROM
    pizzas p
        INNER JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY p.size
ORDER BY order_count DESC
LIMIT 1;
```

Result Grid | Filter

size	order_count
L	18526

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

```sql
SELECT
    p.pizza_type_id, SUM(od.quantity) AS total_quantity
FROM
    pizzas p
        INNER JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pizza_type_id
ORDER BY total_quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

pizza_type_id	total_quantity
classic_dlx	2453
bbq_ckn	2432
hawaiian	2422
pepperoni	2418
thai_ckn	2371

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

```sql
SELECT
    pt.category, COUNT(od.quantity) AS total_quantity
FROM
    pizza_types pt
        INNER JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        INNER JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.category;
```

Result Grid | Filter Ro

category	total_quantity
Classic	14579
Veggie	11449
Supreme	11777
Chicken	10815

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

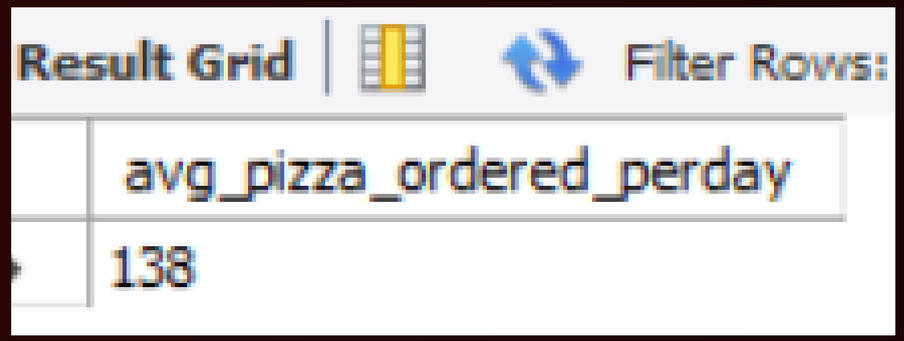
```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Ro

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.

```sql
SELECT
    round(AVG(quantity),0) as avg_pizza_ordered_perday
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS quantity
    FROM
        orders o
    INNER JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quantity;
```



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

```sql
SELECT
    pt.pizza_type_id,
    SUM(od.quantity * p.price) AS total_revenue
FROM
    pizzas p
        INNER JOIN
    order_details od ON od.pizza_id = p.pizza_id
        INNER JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.pizza_type_id
ORDER BY total_revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

pizza_type_id	total_revenue
thai_ckn	43434.25
bbq_ckn	42768
cali_ckn	41409.5

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

```sql
SELECT
    pt.category,
    ROUND(SUM(od.quantity * p.price) / (SELECT
                    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue
                FROM
                    order_details od
                        INNER JOIN
                    pizzas p ON p.pizza_id = od.pizza_id) * 100,
            2) AS revenue
FROM
    pizzas p
        INNER JOIN
    order_details od ON od.pizza_id = p.pizza_id
        INNER JOIN
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category
ORDER BY revenue DESC;
```

Result Grid

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

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

Result Grid | Filter Rows:

order_date	cumm_rev
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
2015-01-10	23990.350000000002
2015-01-11	25862.65

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

```sql
select name , category ,  revenue from
(select category , name , revenue , rank() over(partition by category order by revenue desc) as rn from
(select pt.category , pt.name , sum(od.quantity*p.price) as revenue from
pizza_types pt inner join pizzas p on pt.pizza_type_id = p.pizza_type_id
inner join order_details od on od.pizza_id = p.pizza_id group by pt.category , pt.name) as a) as b
where rn <=3;
```

Result Grid | Filter Rows: | Export:

name	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.70000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5

Home About Contact

THANK YOU

FOR ATTENTION