PIZZA SALES ANALYSIS

WELCOME TO

PIZZA SQL ANALYSIS

This project, developed by Devank Verma, focuses on analyzing sales data using MySQL to derive actionable insights that can help improve business performance. The core objective was to answer key data-driven questions related to sales trends, customer behavior, and product performance using structured SQL queries. By writing optimized queries and using functions like GROUP BY, JOIN, ORDER BY, CUMULATIVE SUM, and DATE functions, meaningful patterns were extracted from the raw data. The project demonstrates how relational databases and SQL can be powerful tools in the decision-making process for sales strategy. Through this analysis, I have provided solutions and interpretations that can help businesses make smarter, data-backed decisions to increase their revenue.

LARANA PIZZA

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

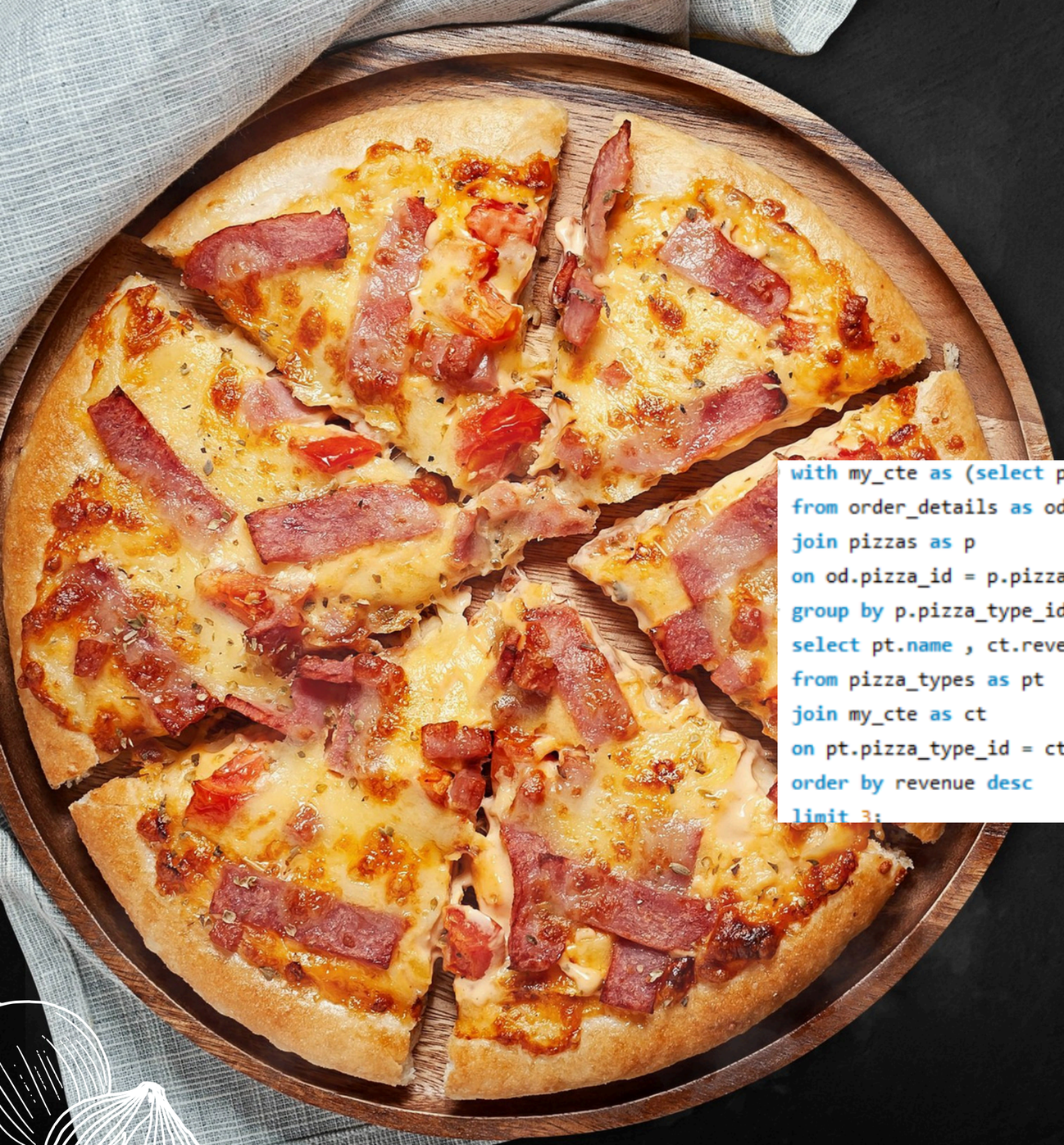
```sql
with my_cte as (select date , sum(od.quantity * p.price) as revenue
from orders as o
join order_details as od
on o.order_id = od.order_id
join pizzas as p
on p.pizza_id = od.pizza_id
group by date)

select date , sum(revenue) over (order by date ) as cum_sum
from my_cte
```

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
with my_cte as (select pt.category , pt.name , sum(p.price * od.quantity) as revenue
from pizza_types as pt
join pizzas as p
on pt.pizza_type_id = p.pizza_type_id
join order_details as od
on od.pizza_id = p.pizza_id
group by  pt.category , pt.name
order by category asc , revenue desc),
tt as (select category , name , revenue , rank() over (partition by category order by revenue desc) as rn
from my_cte)
select category , name , revenue
from tt where rn<4;
```

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
with my_cte as (select o.date , sum(quantity) as total_pizzas
from order_details as od
join orders as o
on o.order_id = od.order_id
group by o.date)
```



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
with my_cte as (select p.pizza_type_id , sum(p.price*od.quantity)  as revenue , count(order_id) as total_orders
from order_details as od
join pizzas as p
on od.pizza_id = p.pizza_id
group by p.pizza_type_id)
select pt.name , ct.revenue , ct.total_orders
from pizza_types as pt
join my_cte as ct
on pt.pizza_type_id = ct.pizza_type_id
order by revenue desc
limit 3;
```

IDENTIFY THE HIGHEST-PRICED PIZZA.

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

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
with my_cte as (select pt.category , sum(od.quantity*p.price) as total_revenue
from order_details as od
join pizzas as p
on od.pizza_id = p.pizza_id
join pizza_types as pt
on pt.pizza_type_id = p.pizza_type_id
group by pt.category)
select category , round((total_revenue/(select sum(total_revenue) from my_cte))*100,2) as conntri_to_revenue
from my_cte
```

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
with my_cte as (select p.pizza_id , pt.pizza_type_id , pt.category
from pizzas as p
join pizza_types as pt
on p.pizza_type_id = pt.pizza_type_id)
select ct.category , sum(od.quantity) as quantity
from order_details as od
join my_cte as ct
on od.pizza_id = ct.pizza_id
group by ct.category;
```

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```sql
SELECT
    category, COUNT(pizza_type_id) AS total_pizzas
FROM
    pizza_types
GROUP BY category;
```

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    p.size, COUNT(od.order_id) AS total_orders
FROM
    order_details AS od
        JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY total_orders DESC;
```

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
with my_cte as (select p.pizza_type_id ,  sum(od.quantity) as total_quantity
from pizzas as p
join order_details as od
on p.pizza_id = od.pizza_id
group by p.pizza_type_id
order by  total_quantity desc
limit 5)
select pt.name ,  my_cte.total_quantity
from my_cte
join pizza_types as pt
on pt.pizza_type_id = my_cte.pizza_type_id
order by total_quantity desc;
```

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
with my_cte as (select pt.category , sum(od.quantity*p.price) as total_revenue
from order_details as od
join pizzas as p
on od.pizza_id = p.pizza_id
join pizza_types as pt
on pt.pizza_type_id = p.pizza_type_id
group by pt.category)
select category , round((total_revenue/(select sum(total_revenue) from my_cte))*100,2) as conntri_to_revenue
from my_cte
```

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(time) AS hour, COUNT(order_id) as total_orders
FROM
    orders
GROUP BY hour;
```

THANK YOU!