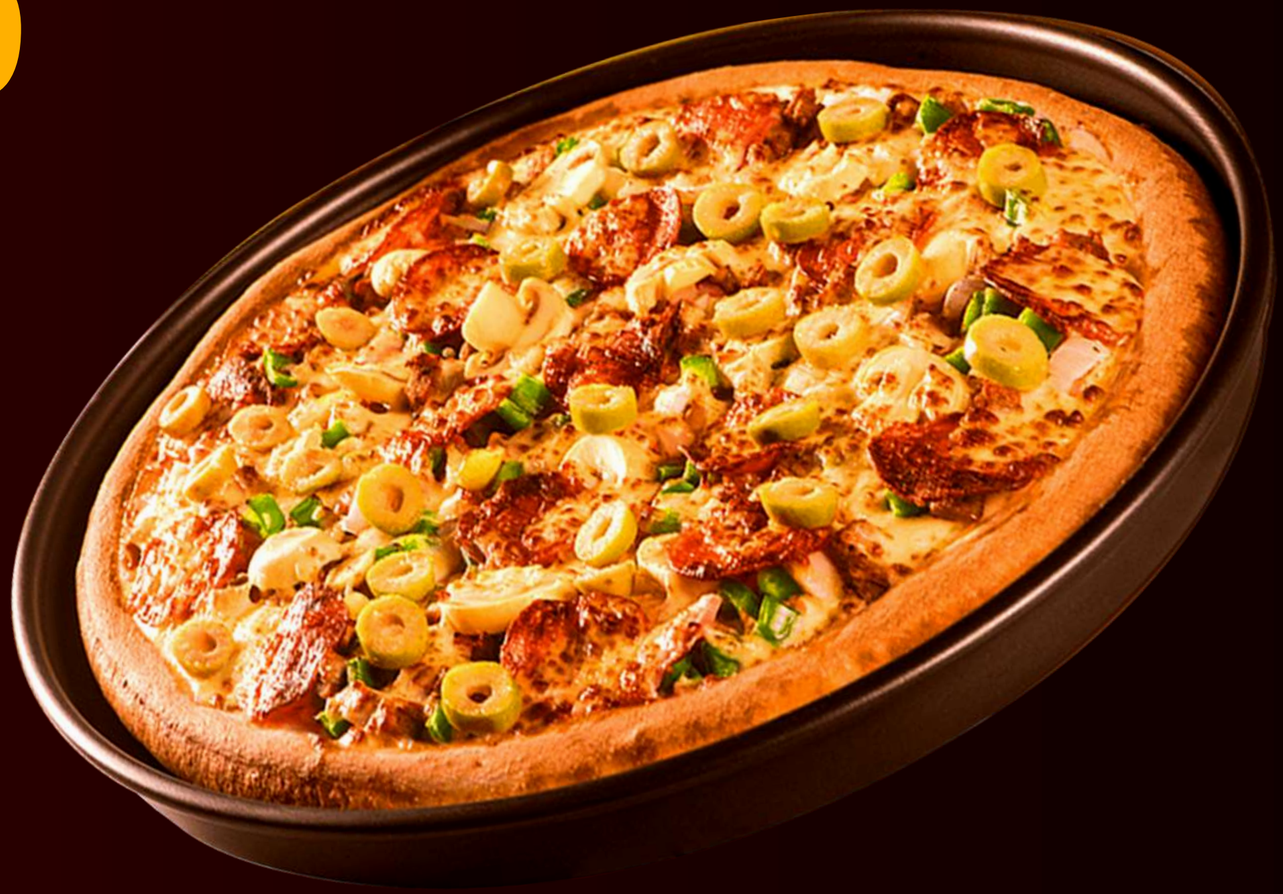
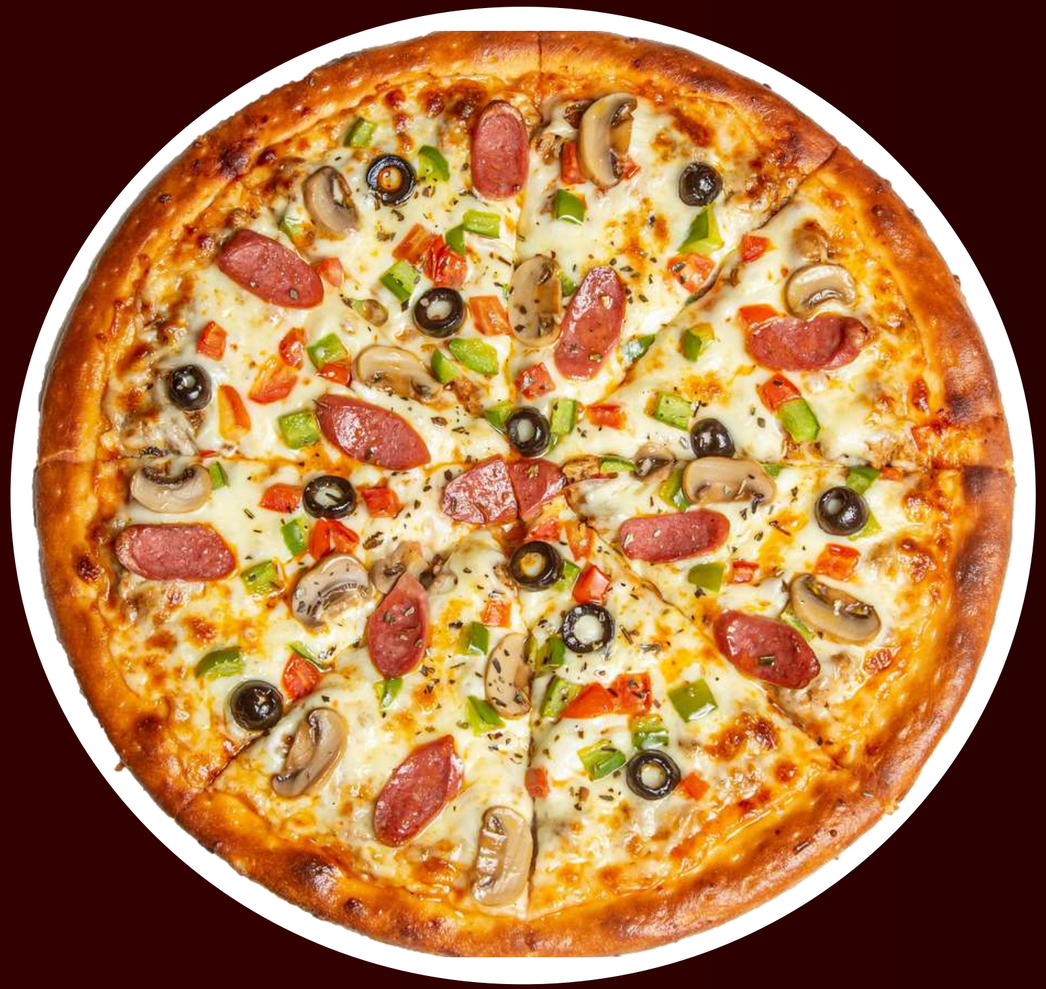
Home About Contact

PIZZA SALES PROJECT USING SQL



Home About Contact



HEELO

MY NAME IS RINKI GOSWAMI .

IN THIS PROJECT I HAVE UTILIZED SQL QUERIES TO SOLVE QUESTIONS THAT WERE RELATED TO P IZZA SALES

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 REVENU GENERATED FROM PIZZA SALES.

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

Result Grid

total_sales
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA

```sql
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 Grid | Filter Ro

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED..

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

Result Grid | Filter

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

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


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

Result Grid | Filter Rows:

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

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


```sql
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;
```

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.


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

Result Grid

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468

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

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

Home About Contact

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

Result Grid | Filter Rows:

avg_pizza_ordered_per_day
138

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 pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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

ALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYP TO TOTAL REVENUE.


```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_sales
        FROM
            order_details
                JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100 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;
```

Result Grid | Filter Rows:

category	revenue
Veggie	23.68259092738457
Chicken	23.955137556847287
Supreme	25.45631126009862
Classic	26.9059602556696

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 reven
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;
```

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


```sql
select name,revenue from
(select category,name,revenue,rank()over (partition by category order by rever
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)b where rn<=3;
select
```

CONCLUSION

This project was developed entirely using SQL, focusing on designing and managing a relational database system. Through this project, I created multiple tables, inserted meaningful data, and wrote queries to retrieve, update, and manipulate the data efficiently. The project helped me understand the core concepts of SQL such as table creation, data types, constraints, joins, subqueries, and aggregate functions. Overall, it was a valuable hands-on experience in working with databases using pure SQL.

THANK YOU