SQL PROJECT ON PIZZA SALES

HELLO

MY NAME IS ROHIT

THIS PROJECT FOCUSES ON ANALYZING PIZZA SALES DATA USING SQL. THE OBJECTIVE IS TO EXTRACT MEANINGFUL INSIGHTS FROM RAW TRANSACTIONAL DATA, SUCH AS IDENTIFYING TOP-SELLING PIZZAS, PEAK SALES HOURS, AND REVENUE TRENDS. BY LEVERAGING SQL QUERIES, WE AIM TO OPTIMIZE BUSINESS DECISIONS RELATED TO INVENTORY, MARKETING, AND CUSTOMER PREFERENCES. THE ANALYSIS HIGHLIGHTS THE POWER OF DATA-DRIVEN STRATEGIES IN ENHANCING OPERATIONAL EFFICIENCY AND BOOSTING OVERALL SALES PERFORMANCE IN THE FOOD INDUSTRY.

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
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
```

total_sales
817860.05

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;
```

	name	price
▶	The Greek Pizza	35.95

RETRIEVE THE TOTAL NUMBER

```sql
select count(order_id) as total_orders from orders;
```

total_orders
21350

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.ALTER

```
select pizzas.size, count(order_details.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;
```

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_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by quantity desc limit 5;
```

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 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;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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(order_time);
```

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

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NO. OF PIZZAS ORDERED PER DAY.

```
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;
```

round(avg(quantity),0)
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;
```

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

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE 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 DESC;
```

category	revenue
Classic	26.9059602556967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

THANK YOU!