LAWRENCE PIZZA

SQL PROJECT ON PIZZA SALES

LAWRENCE PIZZA

IN THIS PROJECT, I HAVE SOLVED VARIOUS ANALYTICAL QUESTIONS RELATED TO PIZZA SALES USING SQL. THE REPOSITORY CONTAINS SQL QUERIES THAT DEMONSTRATE DATA ANALYSIS TECHNIQUES FOR EXTRACTING INSIGHTS FROM A PIZZA SALES DATASET, COVERING:

SALES TRENDS OVER TIME

BEST-SELLING PIZZA TYPES AND SIZES

REVENUE BREAKDOWNS BY TIME, LOCATION, AND PRODUCT CATEGORY

CUSTOMER ORDERING PATTERNS AND PEAK SALES PERIODS

THIS PROJECT HIGHLIGHTS SQL SKILLS SUCH AS AGGREGATIONS, JOINS, SUBQUERIES.

TOTAL NUMBER OF ORDERS PLACED.

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

	total_orders
▶	21350

TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

	total_revenue
▶	827450

HIGHEST-PRICED PIZZA.

```
SELECT
    pizza_types.pizza_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;
```

	pizza_name	price
▶	The Greek Pizza	36

MOST COMMON PIZZA SIZE ORDERED.

```
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
LIMIT 1;
```

	size	order_count
▶	L	18526

LAWRENCE PIZZA

TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
SELECT
    pizza_types.pizza_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.pizza_name
ORDER BY Quantity DESC
LIMIT 5;
```

pizza_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

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.

```
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
23	28
10	8
9	1

CATEGORY-WISE DISTRIBUTION OF PIZZAS.

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

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

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

```sql
SELECT
    ROUND(AVG(Quantity), 0) as 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;
```

	avg_pizza_ordered_per_day
▶	138

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;
```

	avg_pizza_ordered_per_day
▶	138

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;
```

	order_date	cum_revenue
▶	2015-01-01	2746
	2015-01-02	5512
	2015-01-03	8203
	2015-01-04	9983
	2015-01-05	12075
	2015-01-06	14532
	2015-01-07	16761
	2015-01-08	19628

LAWRENCE PIZZA

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_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON pizzas.pizza_id = order_details.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;
```

	category	revenue
▶	Classic	26.96
	Supreme	25.51
	Chicken	24.01
	Veggie	23.53

LAWRENCE PIZZA

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

```sql
select pizza_name, revenue, rnk
from
(select category, pizza_name, revenue,
rank() over(partition by category order by revenue desc) as rnk
from
(select pizza_types.category, pizza_types.pizza_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.pizza_name) as a) as b
where rnk <=3;
```

	pizza_name	revenue	rnk
▶	The Thai Chicken Pizza	44027	1
	The Barbecue Chicken Pizza	43376	2
	The California Chicken Pizza	42002	3
	The Classic Deluxe Pizza	38417	1
	The Hawaiian Pizza	33122	2