PIZZA SALES ANALYSIS

SQL - PROJECT

BY - PRAJAL JOSHI

ABOUT PIZZA SALES (SQL PROJECT)

Hi, I'm Prajal Joshi.
In this project, I used MySQL to analyze pizza sales data and answer key questions like:

- Best-selling pizzas
- Peak sales hours and days
- Impact of size and category on revenue

The goal was to gain useful business insights and improve my SQL and data analysis skills.

SCHEMA

pizza_types
pizza_type_id CHAR(30)
name TEXT
category TEXT
ingredients TEXT
Indexes
orders
order_id INT
order_date DA...
order_time TIME
Indexes
pizzas
pizza_id CHAR(30)
pizza_type_id CHAR(30)
size TEXT
price DOUBLE
Indexes
orders_deta...
order_details_id INT
order_id INT
pizza_id CHAR(30)
quantity INT
Indexes

DATABASE CREATION

```sql
create database pizzahut;

create table orders(
order_id int not null,
order_date date not null,
order_time time not null,
primary key(order_id)
);

create table order_details(
order_details_id int not null,
order_id int not null,
pizza_id text not null,
quantity INT not null,
primary key(order_details_id)
);
```

Retrieve the total number of orders placed.

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

total_orders
21350

Calculate the total revenue generated from pizza sales.

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

revenue
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

Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS total_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_count DESC
LIMIT 1;
```

size	total_count
L	18526

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 pizzas.pizza_id = order_details.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 the total quantity of each pizza category ordered.

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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
```

category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
select
    HOUR(order_time) as hour, COUNT(order_id) as count_order
FROM
    orders
GROUP BY HOUR(order_time);
```

hour	count_order
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

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

```
select category, count(name) from pizza_types group by category ;
```

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_order_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 orderr_quantity_data;
```

avg_pizza_order_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 pizzas.pizza_id = order_details.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.

```
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	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```sql
select order_date, sum(revenue) over(order by order_date) as cumi_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 order_date) as sales;
```

order_date	cumi_revenue
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
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001

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

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

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

THANKYOU