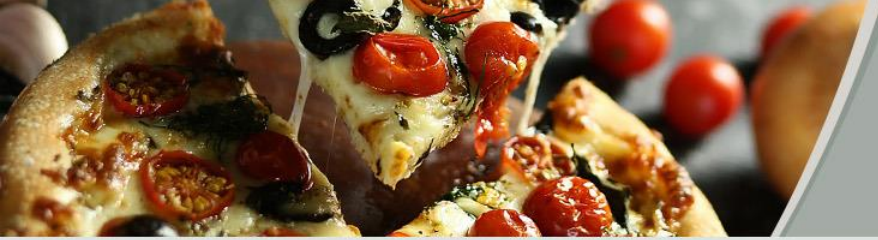
SQL PROJECT

PIZZA SALES ANALYSIS USING SQL

Presented by: Arvind Prasad

Hello !

I'm Arvind Prasad. In this project, I analyzed pizza sales data using SQL. I uncovered insights into customer preferences, sales trends, and popular toppings. This project demonstrates how SQL can transform raw data into valuable information. Join me as I reveal the key findings from my analysis of pizza sales!



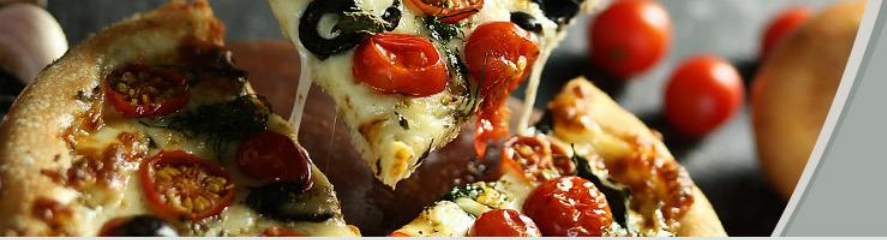
OUR VARIANTS

Chicken

Supreme

Classic

Veggie



Raw DataSets

pizza_id	pizza_type	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75
ckn_alfredo	ckn_alfredo	S	12.75
ckn_alfredo	ckn_alfredo	M	16.75
ckn_alfredo	ckn_alfredo	L	20.75
ckn_pesto_s	ckn_pesto	S	12.75
ckn_pesto_m	ckn_pesto	M	16.75
ckn_pesto_l	ckn_pesto	L	20.75
southw_ckn_s	southw_ckn	S	12.75
southw_ckn_m	southw_ckn	M	16.75
southw_ckn_l	southw_ckn	L	20.75
thai_ckn_s	thai_ckn	S	12.75

Table: pizzas

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue C	Chicken	Barbecued Chicken, Red Peppers, Green Peppers, Tomatoes, Red Onions, Barbecue Sauce
cali_ckn	The California C	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno Peppers, Fontina Cheese, Gouda Cheese
ckn_alfredo	The Chicken Al	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms, Asiago Cheese, Alfredo Sauce
ckn_pesto	The Chicken Pe	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garlic, Pesto Sauce
southw_ckn	The Southwest	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, Jalapeno Peppers, Corn, Cilantro, Chipotle Sauce
thai_ckn	The Thai Chick	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, Thai Sweet Chilli Sauce
big_meat	The Big Meat P	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sausage
classic_dlx	The Classic Del	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppers, Bacon
hawaiian	The Hawaiian P	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
ital_cpcllo	The Italian Cap	Classic	Capocollo, Red Peppers, Tomatoes, Goat Cheese, Garlic, Oregano
napolitana	The Napolitana	Classic	Tomatoes, Anchovies, Green Olives, Red Onions, Garlic
pep_msh_pep	The Pepperoni	Classic	Pepperoni, Mushrooms, Green Peppers
pepperoni	The Pepperoni	Classic	Mozzarella Cheese, Pepperoni
the_greek	The Greek Pizz	Classic	Kalamata Olives, Feta Cheese, Tomatoes, Garlic, Beef Chuck Roast, Red Onions
brie_carre	The Brie Carre	Supreme	Brie Carre Cheese, Prosciutto, Caramelized Onions, Pears, Thyme, Garlic
calabrese	The Calabrese	Supreme	'Nduja Salami, Pancetta, Tomatoes, Red Onions, Friggitello Peppers, Garlic

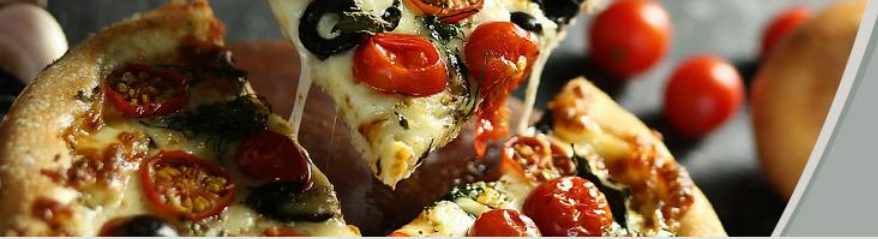
Table: Pizza_types

order_id	date	time
1	#########	11:38:36
2	#########	11:57:40
3	#########	12:12:28
4	#########	12:16:31
5	#########	12:21:30
6	#########	12:29:36
7	#########	12:50:37
8	#########	12:51:37
9	#########	12:52:01
10	#########	13:00:15
11	#########	13:02:59
12	#########	13:04:41
13	#########	13:11:55
14	#########	13:14:19

Table: orders

Table: orders_details

order_det	order_id	pizza_id	quantity
1	1	hawaiian_	1
2	2	classic_dlx	1
3	2	five_chees	1
4	2	ital_supr_	1
5	2	mexicana_	1
6	2	thai_ckn_l	1
7	3	ital_supr_	1
8	3	prsc_argla	1
9	4	ital_supr_	1
10	5	ital_supr_	1
11	6	bbq_ckn_	1



Questions

Basic:

Retrieve the total number of orders placed.

Calculate the total revenue generated from pizza sales.

Identify the highest-priced pizza.

Identify the most common pizza size ordered.

List the top 5 most ordered pizza types along with their quantities.

Intermediate:

Join the necessary tables to find the total quantity of each pizza category ordered.

Determine the distribution of orders by hour of the day.

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

Group the orders by date and calculate the average number of pizzas ordered per day.

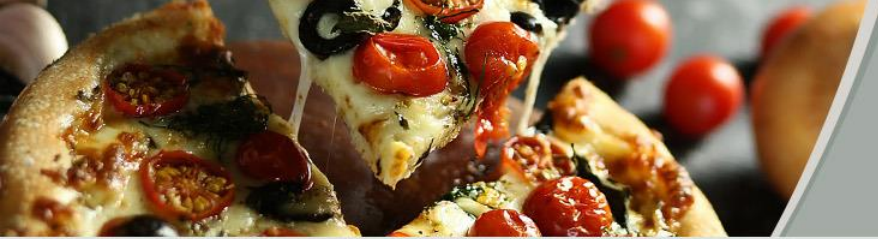
Determine the top 3 most ordered pizza types based on revenue.

Advanced:

Calculate the percentage contribution of each pizza type to total revenue.

Analyze the cumulative revenue generated over time.

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



Basic Questions:

1) Retrieve the total number of orders placed.

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

Result Grid

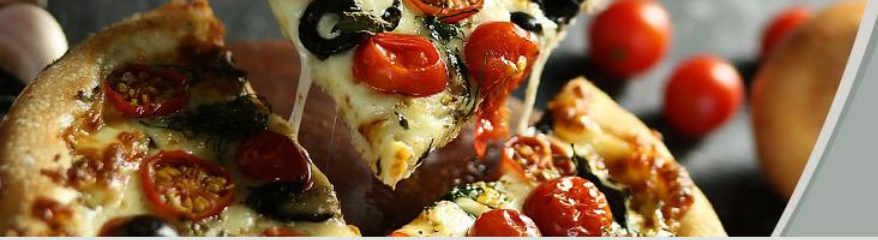
	total_orders
▶	21350

2) Calculate the total revenue generated from pizza sales.

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

Result Grid

	total_price
▶	817860.05



Basic Questions :

3) 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;
```

Result Grid | Filter Rows

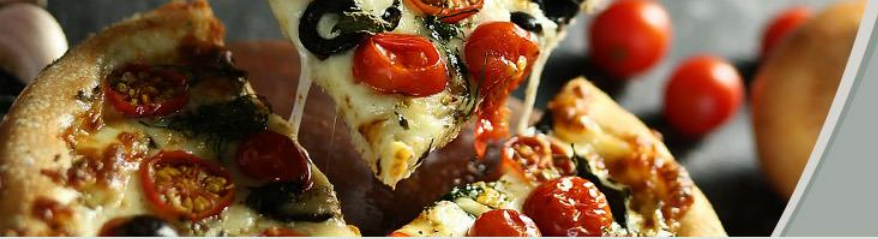
name	price
The Greek Pizza	35.95

4) Identify the most common pizza size ordered.

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

Result Grid | Filter Rc

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



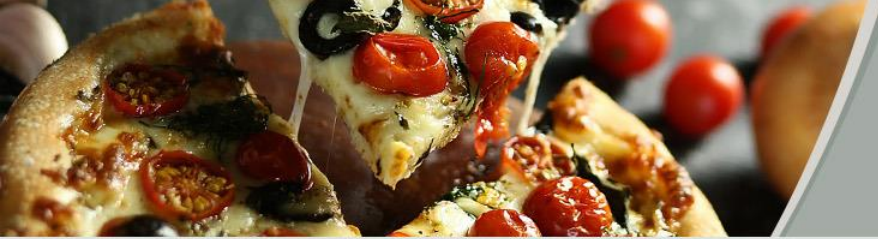
Basic Questions :

5) List the top 5 most ordered pizza types along with their quantities.

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

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



Intermediate Questions:

1) Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Result Grid | Filter Ro

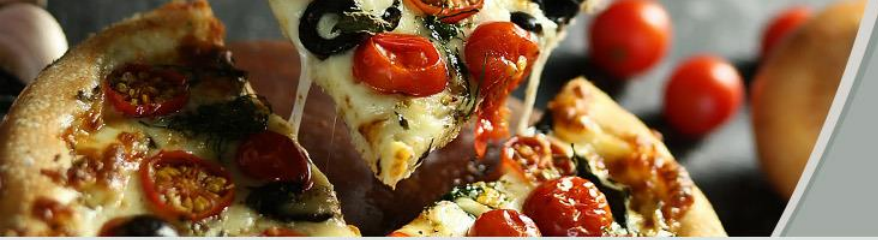
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

2) Determine the distribution of orders by hour of the day.

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

Result Grid | Filter Rows:

HOUR(order_time)	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



Intermediate Questions:

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

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

Result Grid | Filter Rows:

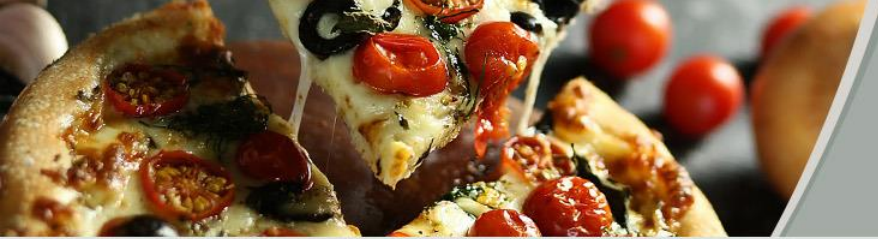
category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

4) Group the orders by date and calculate the average number of pizzas ordered per day.

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizzas_order_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows:

avg_pizzas_order_per_day
138



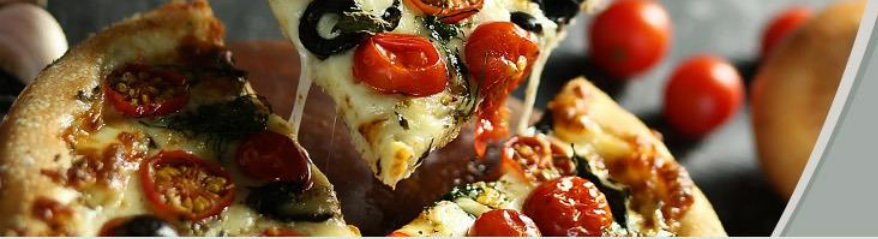
Intermediate Questions:

5) Determine the top 3 most ordered pizza types based on revenue.

```sql
SELECT
    pizza_types.name,
    ROUND(SUM(pizzas.price * orders_details.quantity),
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.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



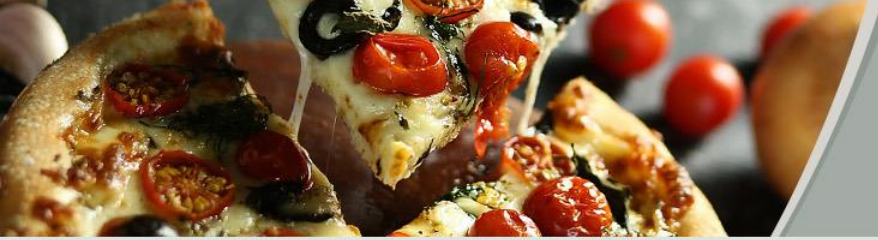
Advance Questions :

1) Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT
    pizza_types.category,
    round((SUM(pizzas.price * orders_details.quantity) / (SELECT
            SUM(pizzas.price * orders_details.quantity)
        FROM
            pizzas
                JOIN
            orders_details ON pizzas.pizza_id = orders_details.pizza_id)) * 100,2) AS rev_percentage
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY rev_percentage DESC;
```

Result Grid | Filter Rows:

category	rev_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



Advance Questions :

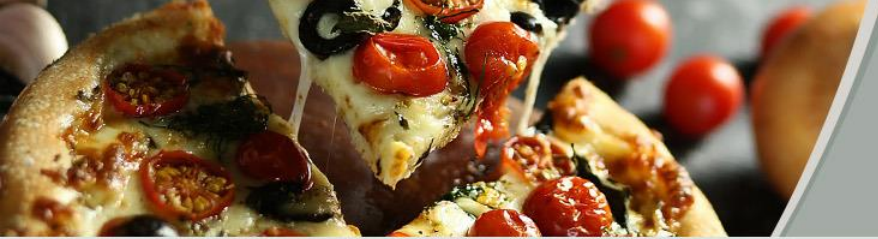
2) Analyze the cumulative revenue generated over time.

```sql
select order_date, revenue, round(sum(revenue)
over(order by order_date),2) as cum_revenue from
(SELECT
    orders.order_date,
    ROUND(SUM(pizzas.price * orders_details.quantity),
            2) AS revenue
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
        JOIN
    orders ON orders.order_id = orders_details.order_id
GROUP BY orders.order_date) as sales;
```

Result Grid | Filter Rows:

order_date	revenue	cum_revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-0	2015-01-03	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2	16560.7
2015-01-08	2838.35	19399.05
2015-01-09	2127.35	21526.4
2015-01-10	2463.95	23990.35
2015-01-11	1872.3	25862.65
2015-01-12	1919.05	27781.7
2015-01-13	2049.6	29831.3
2015-01-14	2527.4	32358.7
2015-01-15	1984.8	34343.5
2015-01-16	2594.15	36937.65

.....



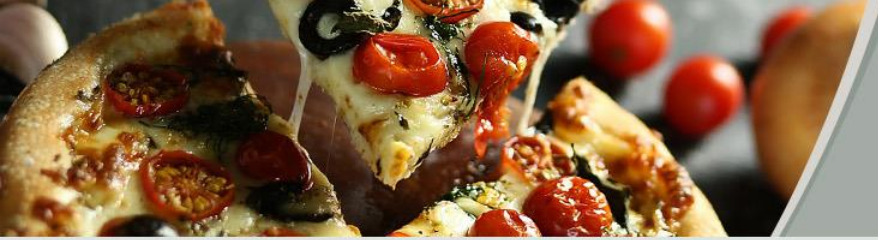
Advance Questions :

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

```sql
select category, 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(pizzas.price * orders_details.quantity) AS revenue
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category , pizza_types.name) as a) as b
where rn <=3;
```

Result Grid | Filter Rows: | Export:

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



CONCLUSION

Through the analysis of multiple raw datasets on pizza sales using SQL, several key insights were uncovered. We identified the most popular pizza toppings, peak sales periods, and customer preferences. These findings highlight important trends that can be leveraged to enhance marketing strategies, optimize inventory management, and improve customer satisfaction.

The pizza business can enhance its operations, better meet customer needs, and ultimately drive growth. The use of SQL for data analysis has proven to be a powerful tool in uncovering valuable insights and informing strategic decisions.

THANK YOU !!