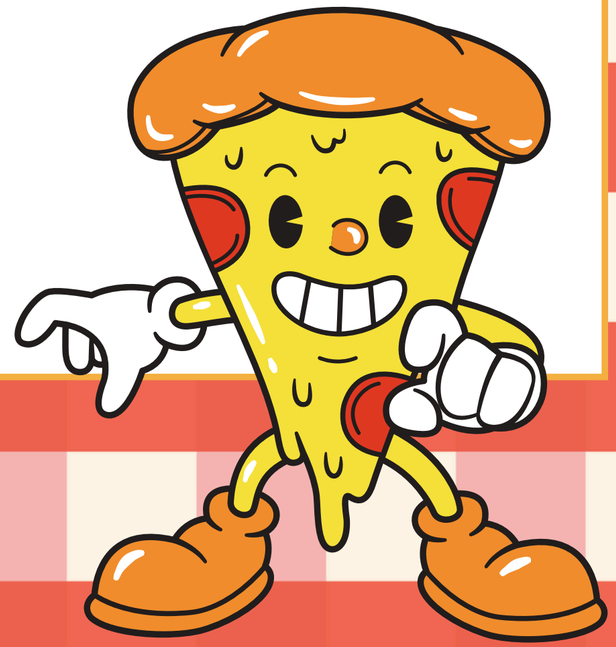
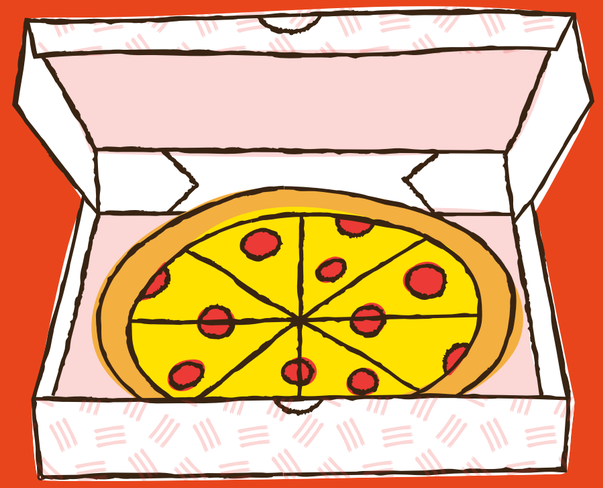
PROJECT: PIZZA SALES ANALYSIS

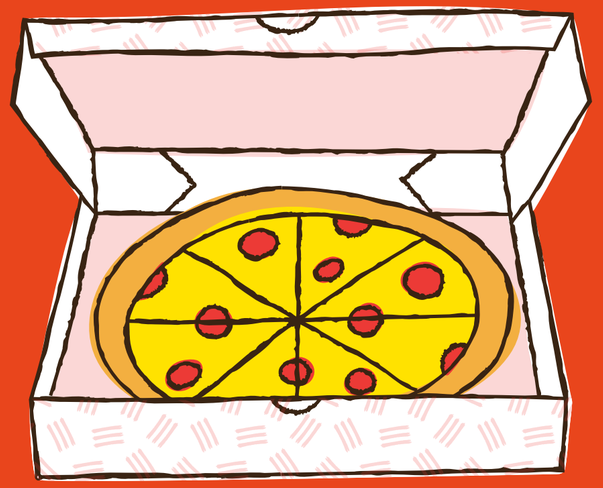


HELLO!

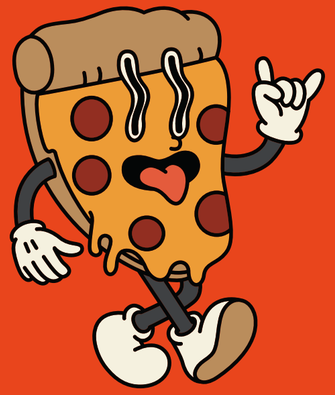
My name is Rohan. In this project I have utilized SQL queries to solve questions related to Pizza Sales



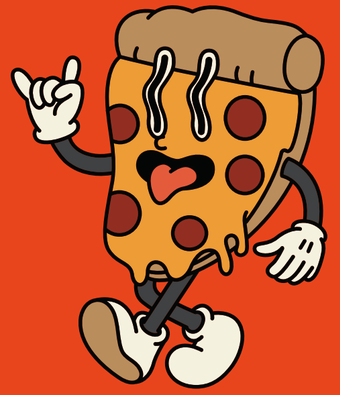
TABLES



1. PIZZAS
2. PIZZA_TYPES
3. ORDERS
4. ORDER_DETAILS



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 revenue generated from pizza sales.

```sql
SELECT
    SUM(order_details.quantity * pizzas.price) AS total_revenue
FROM
    order_details
        INNER JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid

	total_revenue
▶	817860.049999993

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 Rows:

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 quantity
FROM
    pizzas
        INNER JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY quantity DESC
LIMIT 1;
```

Result Grid

	size	quantity
▶	L	18526

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

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) as total
FROM
    pizza_types
        INNER 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 total desc limit 5;
```

Result Grid | Filter Rows:

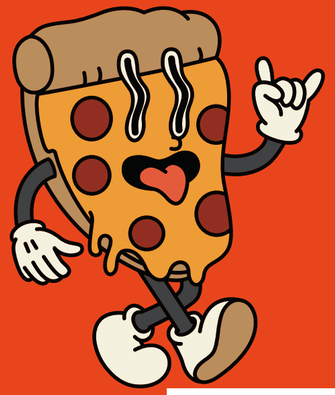
name	total
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

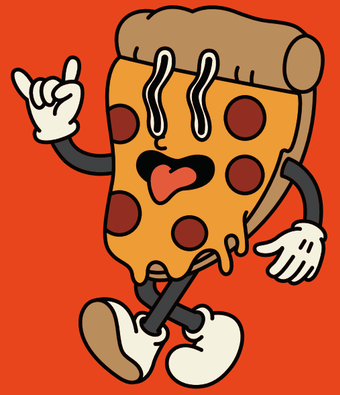
```sql
SELECT
    pizza_types.category, SUM(order_details.quantity)
FROM
    pizza_types
        INNER JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        INNER JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category order by SUM(order_details.quantity) desc;
```

Result Grid | Filter Rows:

category	SUM(order_details.quantity)
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



Determine the distribution of orders by hour of the day.



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

Result Grid | Filter Rows:

HOUR(order_time)	COUNT(order_id)
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

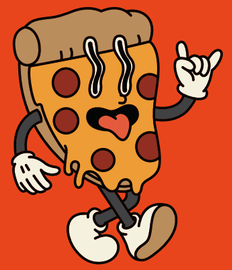
2520

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

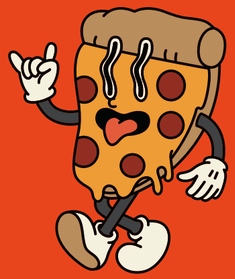
```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Row:

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
     AVG(total)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) as total
    FROM
        orders
    INNER JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS a;
```

Result Grid

	AVG(total)
▶	138.4749

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
        INNER JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        INNER 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

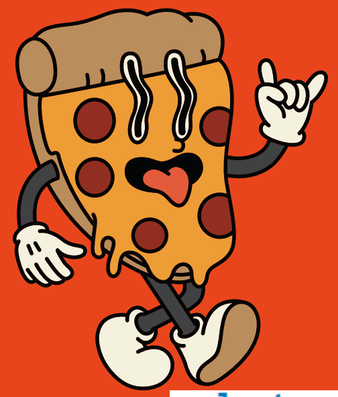
Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT
    pizza_types.category,
    concat(round((SUM(order_details.quantity * pizzas.price)/(select sum(order_details.quantity * pizzas.price) as total_revenue
    from order_details
    inner join pizzas on order_details.pizza_id = pizzas.pizza_id))*100,0),'%') as percentage
    FROM
    pizza_types
        INNER JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        INNER JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id group by pizza_types.category;
```

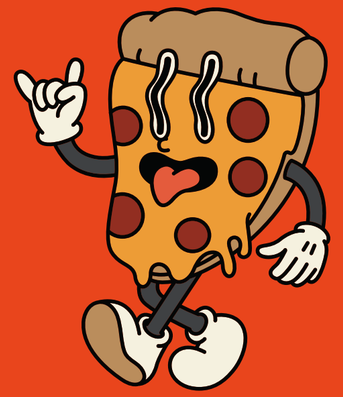
Result Grid | Filter Rows:

category	percentage
Classic	27%
Veggie	24%
Supreme	25%
Chicken	24%

24%



Analyze the cumulative revenue generated over time.



```sql
select order_date, sum(revenue) over(order by order_date)
from
(SELECT
    orders.order_date,
    SUM(order_details.quantity * pizzas.price) as revenue
FROM
    orders
        INNER JOIN
    order_details ON orders.order_id = order_details.order_id
        INNER JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id group by orders.order_date) as
A;
```

Result Grid | Filter Rows: | Expor

order_date	sum(revenue) over(order by order_date)
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
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001

8108.15

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

```sql
select category, name, revenue, rn 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
inner join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
inner 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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell

category	name	revenue	rn
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.70000000065	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

THANK YOU!