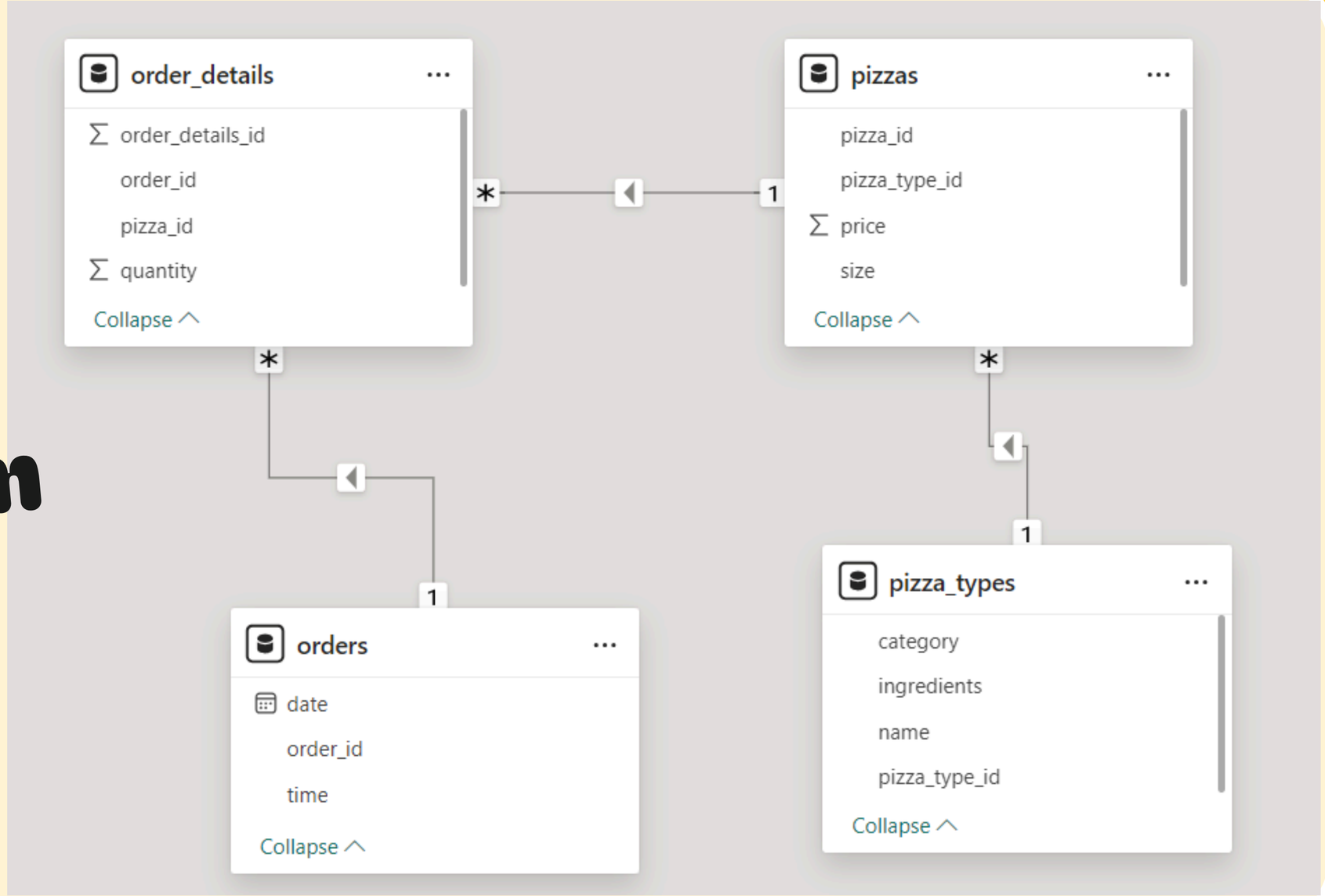
SQL Pizza Sales Project

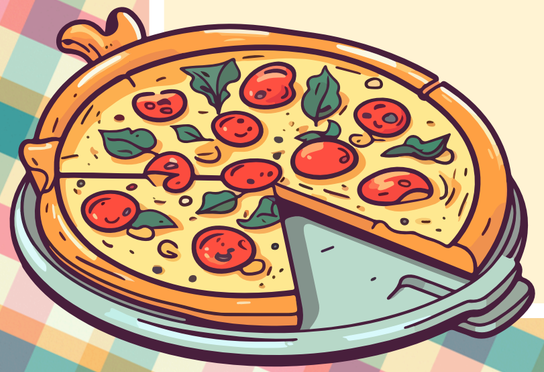
Presented by Deepa Kumari

Hello!

My name is Deepa Kumari and in this project, I have used SQL queries to solve some EDA questions related to pizza sales

ER Diagram


order_details
order_details_id
order_id
pizza_id
quantity
Collapse
pizzas
pizza_id
pizza_type_id
price
size
Collapse
orders
date
order_id
time
Collapse
pizza_types
category
ingredients
name
pizza_type_id
Collapse
*
1

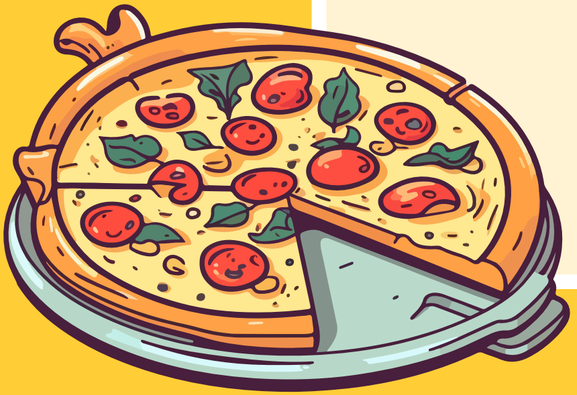


Retrieve the total number of order placed

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

Result Grid

	total_num_orders
▶	21350



calculate total revenue generated by pizza sales

```sql
select round(sum(od.quantity * p.price), 2) as total_revenue
from order_details as od
join pizzas as p on od.pizza_id = p.pizza_id;
```

Result Grid

total_revenue
817860.05

Identify the highest priced pizza

```sql
select pt.name as Name, max(p.price) as highest_price
from pizzas as p
join pizza_types as pt
on p.pizza_type_id = pt.pizza_type_id
group by 1
order by 2 desc
limit 1;
```

Result Grid | Filter Rows:

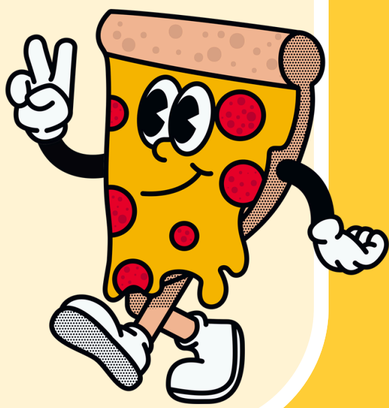
Name	highest_price
The Greek Pizza	35.95

Identify the most common pizza size ordered

```sql
select p.size as most_common_pizza_size, count(od.order_id) as Total_order
from pizzas as p
join order_details as od
on p.pizza_id = od.pizza_id
group by 1
order by 2 desc;
```

Result Grid | Filter Rows:

most_common_pizza_size	Total_order
L	18526
M	15385
S	14137
XL	544
XXL	28



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

```sql
select pt.name, sum(od.quantity) as Quantity
from pizza_types as pt
join pizzas as p
on pt.pizza_type_id = p.pizza_type_id
join order_details as od
on od.pizza_id = p.pizza_id
group by 1
order by 2 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

find the total quantity of each pizza category ordered

```sql
select pt.category, sum(od.quantity) as total_quantity
from order_details as od
join pizzas as p
on od.pizza_id = p.pizza_id
join pizza_types as pt
on pt.pizza_type_id = p.pizza_type_id
group by 1
order by 2 desc;
```

Result Grid | Filter Rows:

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Distribution of orders by hour of the day

```sql
select hour(order_time) as Hour, count(order_id) as orders
from orders
group by 1
order by 1;
```

Result Grid

Hour	orders
9	1
10	8
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

Find category wise distribution of pizzas

```sql
select category as Category, count(pizza_type_id) as count_pizzas
from pizza_types
group by category
```

Result Grid | Filter Rows

Category	count_pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

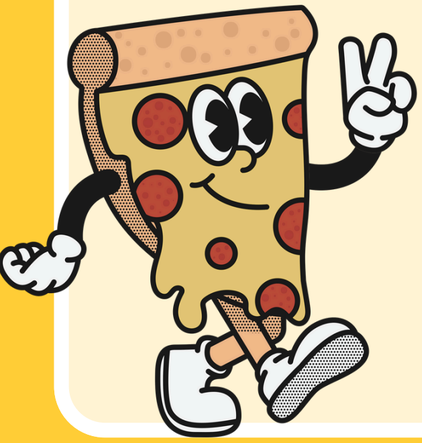
```sql
with order_quantity as (
select o.order_date, sum(od.quantity) as total_quantity
from orders as o
join order_details as od
on o.order_id = od.order_id
group by 1
)
select round(avg(total_quantity), 0) as avg_num_pizza_per_day
from order_quantity;
```

Result Grid | Filter Rows:

	avg_num_pizza_per_day
▶	138

Determine the top 3 most ordered pizza type based on revenue

```sql
select pt.name, count(od.order_id) as total_orders, round(sum(p.price*od.quantity), 2) as total_revenue
from pizza_types as pt
join pizzas as p
on pt.pizza_type_id = p.pizza_type_id
join order_details as od
on p.pizza_id = od.pizza_id
group by 1
order by 3 desc
limit 3;
```



Result Grid | Filter Rows: | Export:

name	total_orders	total_revenue
The Thai Chicken Pizza	2315	43434.25
The Barbecue Chicken Pizza	2372	42768
The California Chicken Pizza	2302	41409.5

calculate the percentage contribution of each pizza type to total revenue

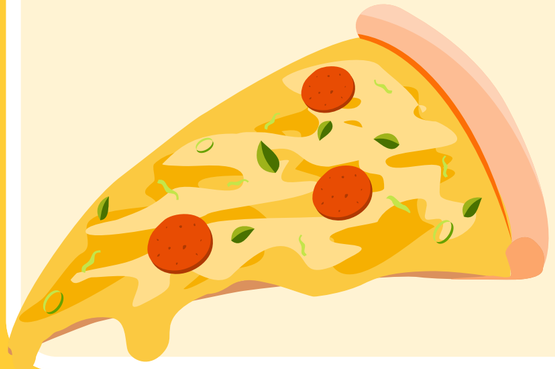
```sql
select pt.category,
round(sum(p.price*od.quantity)/(select round(sum(od.quantity * p.price), 2) as total_revenue
from order_details as od
join pizzas as p on od.pizza_id = p.pizza_id)*100, 2) as percentage_contri_each_pizza
from pizza_types as pt
join pizzas as p
on pt.pizza_type_id = p.pizza_type_id
join order_details as od
on p.pizza_id = od.pizza_id
group by 1;
```

Result Grid | Filter Rows:

category	percentage_contri_each_pizza
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Analyze the cumulative revenue generated over time

```sql
with sales as (
select o.order_date, round(sum(p.price*od.quantity), 2) as revenue
from order_details as od
join pizzas as p
on od.pizza_id = p.pizza_id
join orders as o
on o.order_id = od.order_id
group by 1
)
select order_date, sum(revenue) over(order by order_date) as cum_revenue
from sales
```



Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.85
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

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

```sql
with sales as (
select pt.name, pt.category, round(sum(p.price*od.quantity), 2) as total_revenue
from pizza_types as pt
join pizzas as p
on pt.pizza_type_id = p.pizza_type_id
join order_details as od
on p.pizza_id = od.pizza_id
group by 1, 2
),
rank_category as(
select category, name, total_revenue,
dense_rank() over(partition by category order by total_revenue desc) as top_3_pizza_per_category
from sales
)
select category, name, total_revenue
from rank_category
where top_3_pizza_per_category<=3;
```

Result Grid | Filter Rows: | Export:

category	name	total_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	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Thank You