

# SQL Project



## Questions

- Retrieve the total number of orders placed.
- Identify the highest-priced pizza.
- Calculate the total revenue generated from pizza sales.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- 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.
- 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.

# Pizza Table

pizza_id	pizza_type_id	size	price
bbq_dkn_s	bbq_dkn	S	12.75
bbq_dkn_m	bbq_dkn	M	16.75
bbq_dkn_l	bbq_dkn	L	20.75
cali_dkn_s	cali_dkn	S	12.75
cali_dkn_m	cali_dkn	M	16.75

# Pizza\_type Table

pizza_type_id	name	category	ingredients
bbq_dkn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red
cali_dkn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach
dkn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red
southw_dkn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red

# Order\_details Table

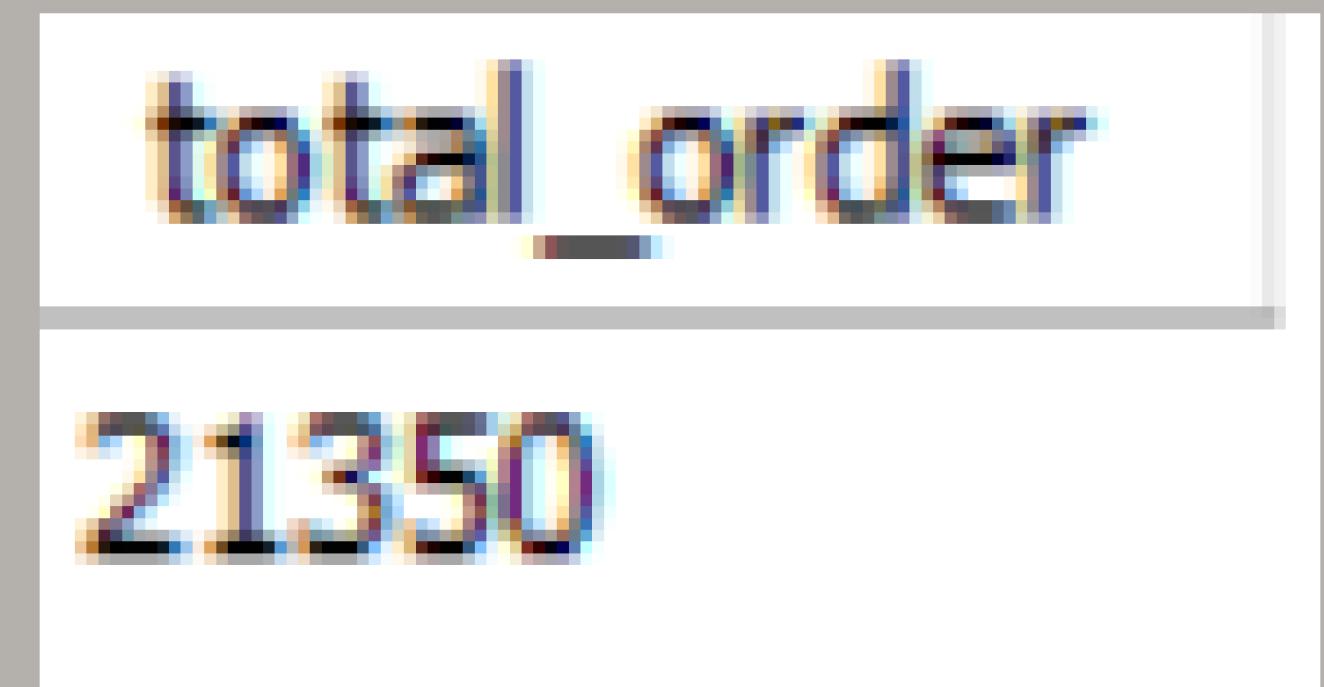
order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	dassic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1

# Orders Table

order_id	date	time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30

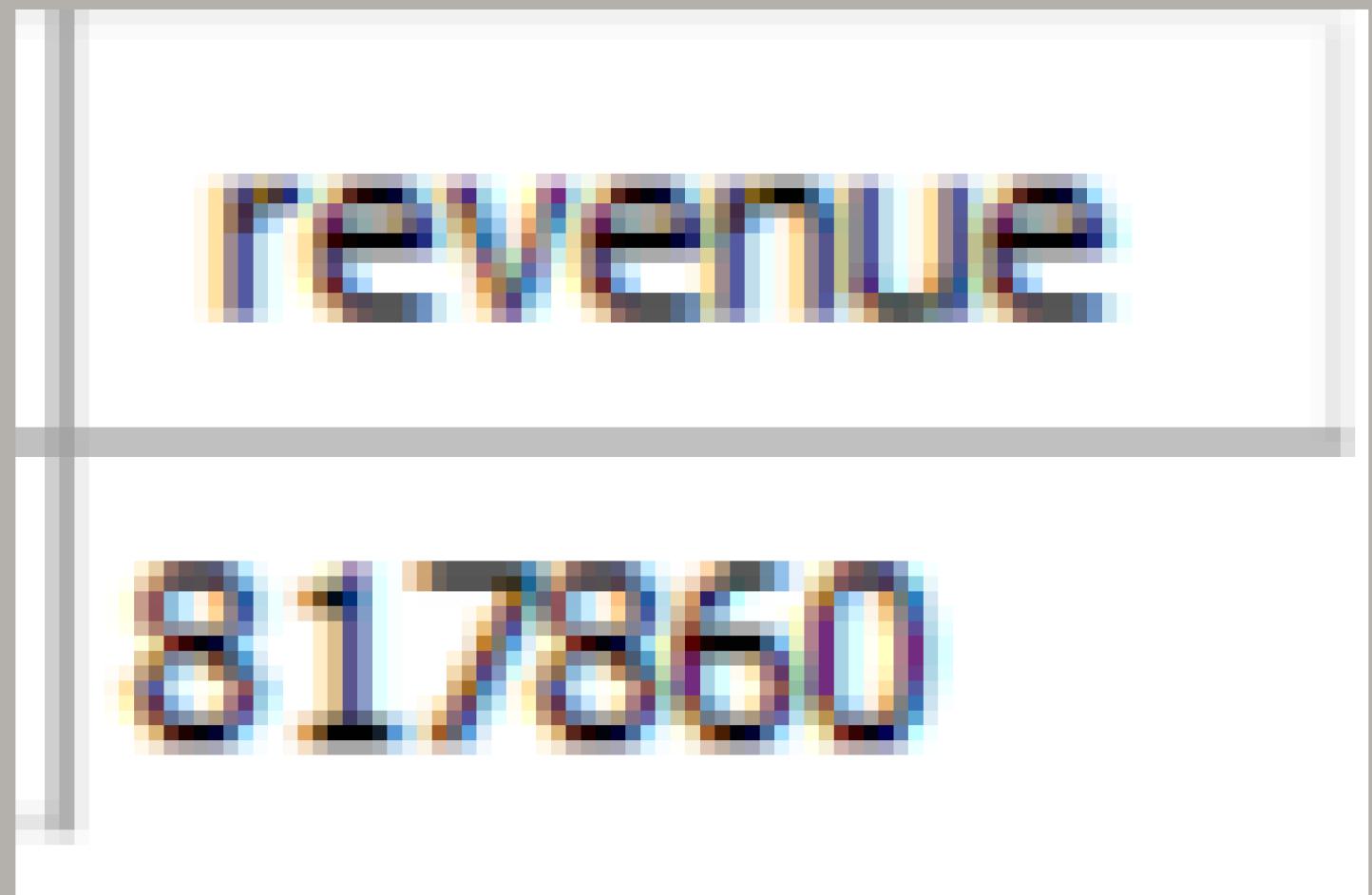
# Retrieve the total number of orders placed.

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



# Calculate the total revenue generated from pizza sales.

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



# 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.

```
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

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

```
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 order_details.pizza_id =
    pizzas.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.

```
 pizza_types.category, SUM(order_details.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(orders.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

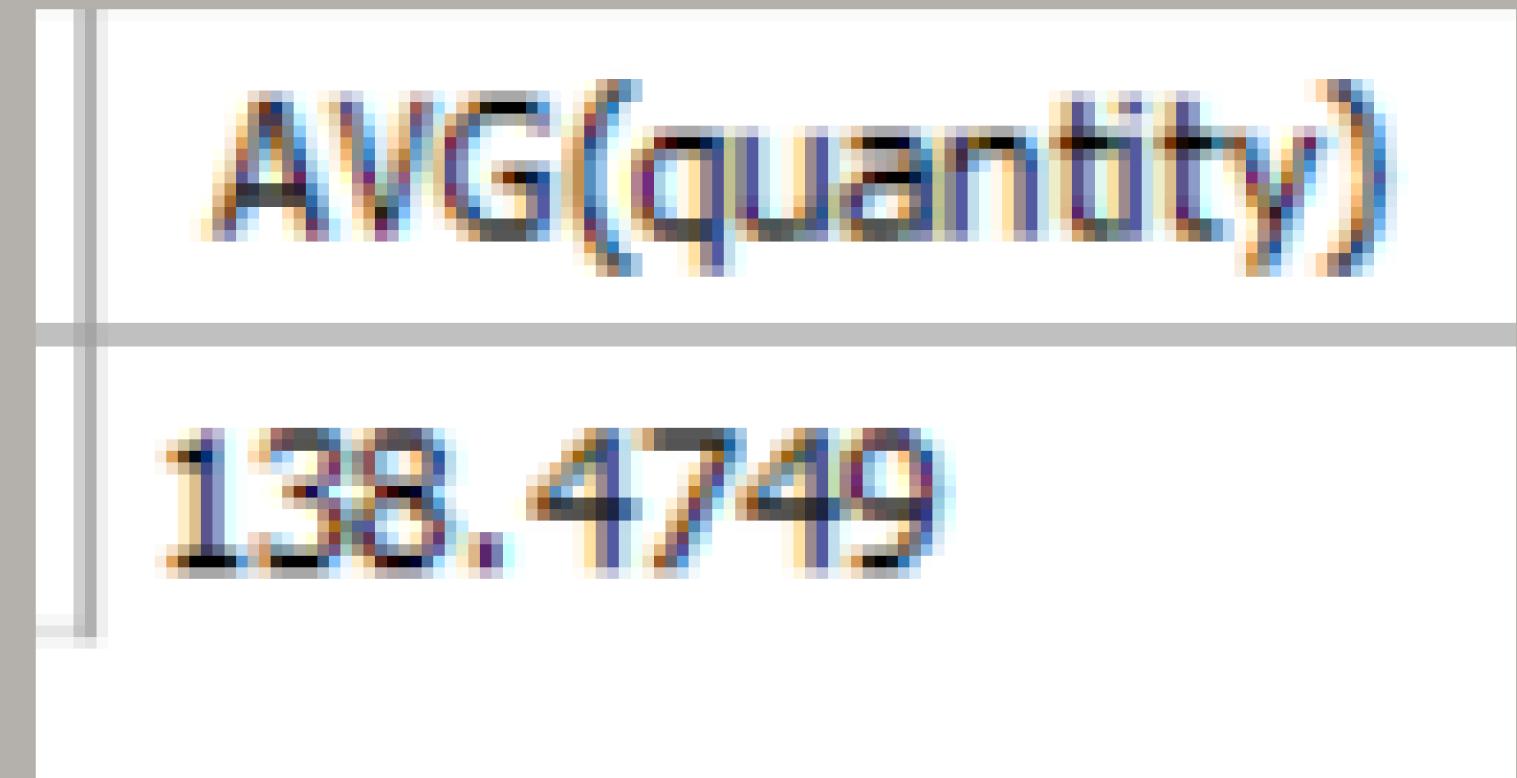
# Join relevant tables to find the category-wise distribution of pizzas.

```
SELECT  
    pizza_types.name, pizza_types.category  
FROM  
    pizza_types  
GROUP BY pizza_types.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.

```
SELECT  
    AVG(quantity)  
FROM  
    (SELECT  
        orders.date, SUM(order_details.quantity) as  
        quantity  
    FROM  
        orders  
    JOIN order_details ON order_details.order_id =  
        orders.order_id  
    GROUP BY orders.date) AS order_quantity;
```



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

```
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) 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;
```

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,
    SUM(pizzas.price * order_details.quantity) /
    (SELECT
        ROUND(SUM(pizzas.price *
    order_details.quantity)) AS revenue
    FROM
        pizzas
        JOIN
        order_details ON pizzas.pizza_id =
    order_details.pizza_id) 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.category
    ORDER BY revenue DESC;
```

category	revenue
Classic	0.2690596190056979
Supreme	0.2545631281637422
Chicken	0.23955139021348398
Veggie	0.23682592375223507

# Analyze the cumulative revenue generated over time.

```
select date,sum(revenue)
over(order by date) as cum_revenue
from (select
orders.date,sum(order_details.quantity*pizzas.price)
as revenue
from order_details join pizzas on
pizzas.pizza_id=order_details.pizza_id join orders on
orders.order_id=order_details.order_id
group by orders.date) as sales ;
```

date	cum_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

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

```
lselect 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 pizzas  
join order_details on  
pizzas.pizza_id=order_details.pizza_id join  
pizza_types on  
pizza_types.pizza_type_id=pizzas.pizza_type_id  
group by pizza_types.category,pizza_types.name  
order by revenue desc limit 3) 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