LARANA PIZZA

NEXT >

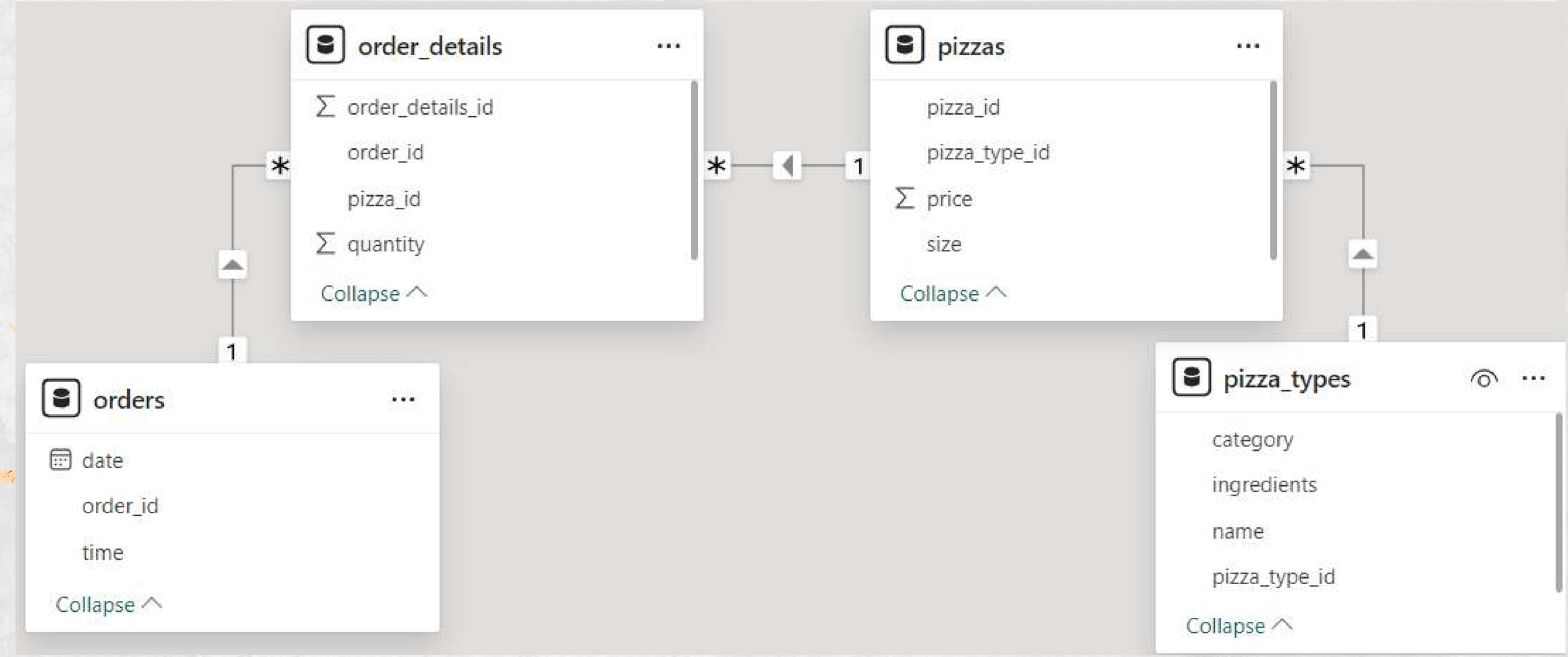
ABOUT PROJECT

Hello My name is Priyanshu Srivastav. The data for this project is provided by WSCube Tech.

In this project, I used multiple SQL queries to extract information from the database.

DATA MODEL

LARANA PIZZA



< BACK

NEXT >

QUERIES QUESTONS

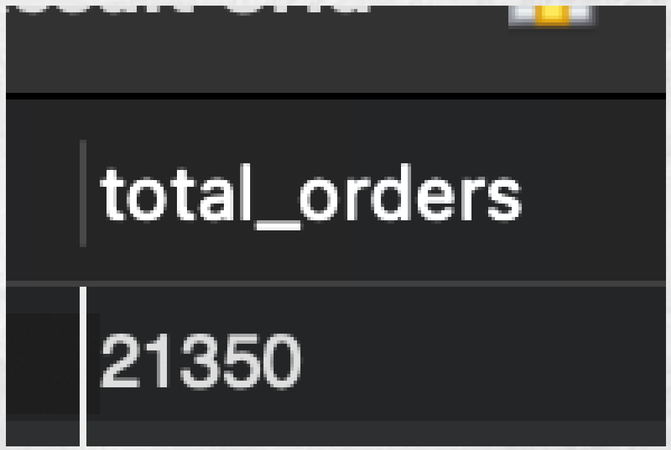
1- Retrieve the total number of orders placed.
2- Calculate the total revenue generated from pizza sales.
3- Identify the highest-priced pizza.
4- Identify the most common pizza size ordered.
5- List the top 5 most ordered pizza types along with their quantities.

6- Join the necessary tables to find the total quantity of each pizza category ordered.
7- Determine the distribution of orders by hour of the day.
8- Join relevant tables to find the category-wise distribution of pizzas.
9- Group the orders by date and calculate the average number of pizzas ordered per day.
10- Determine the top 3 most ordered pizza types based on revenue.

11- Calculate the percentage contribution of each pizza type to total revenue.
12- Analyze the cumulative revenue generated over time.
13- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
#Q1 Retrive total number of orders placed

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



total_orders
21350

< BACK

NEXT >

```sql
#Q2 Calculate the total revenue generated from pizza sales.

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;
```

total_sales
817860.05

< BACK

NEXT >

```sql
# Q3 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

< BACK

NEXT >

```sql
#Q4 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;
```

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

< BACK

NEXT >

```sql
#Q5 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

< BACK

NEXT >

```sql
#Q6 Join the necessary tables to find the total quantity of each pizza category ordered.

SELECT pizza_types.category,
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.category ORDER BY quantity DESC;
```

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

< BACK

NEXT >

```sql
#Q7 Determine the distribution of orders by hour of the day.

SELECT hour(order_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
9	1

< BACK

NEXT >

```sql
# Q8 Join relevant tables to find the category-wise distribution of pizzas.

SELECT
    category, COUNT(name) AS pizza
FROM
    pizza_types
GROUP BY category
ORDER BY pizza DESC;
```

category	pizza
Supreme	9
Veggie	9
Classic	8
Chicken	6

< BACK

NEXT >

```sql
# Q9 Group the orders by date and calculate the average number of pizzas ordered per day.

SELECT
    ROUND(AVG(quantity), 0) as Average_pizza_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 order_quantity;
```

Average_pizza_per_day
138

< BACK

NEXT >

```sql
#Q10 Determine the top 3 most ordered pizza types based on revenue.

SELECT 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.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

< BACK

NEXT >

```sql
#Q11 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_percent
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
```

category	revenue_percent
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

< BACK

NEXT >

```sql
#Q12 Analyze the cumulative revenue generated over time.

SELECT order_date,
SUM(revenue) over(order by order_date) as cum_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 orders.order_date) AS sales;
```

order_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
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

< BACK

NEXT >

```sql
#Q13 Determine the top 3 most ordered pizza types based on revenue

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

< BACK

NEXT >

LARANA PIZZA

THANK YOU!

< BACK