PIZZAZ

HOUSE

@Vaishnavikanase

PROJECT OVERVIEW

The Pizza Sales Analysis project leverages SQL to explore sales patterns, customer preferences, and business performance of a pizza house. Basic analysis includes calculating total orders, revenue, identifying the highest-priced pizza, the most common pizza size, and the top 5 most ordered pizza types. Intermediate analysis involves joining tables to find total quantities of each pizza category, distribution of orders by hour, category-wise pizza distribution, daily average orders, and top 3 pizza types of revenue. Advanced analysis calculates each pizza types's revenue contribution, cumulative revenue over time, and top 3 pizza types by revenue within each category. These insights guide strategic decisions to optimize operations and boost profitability.

Analysis Questions

1.Retrive 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. find the category-wise distribution of pizzas.

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

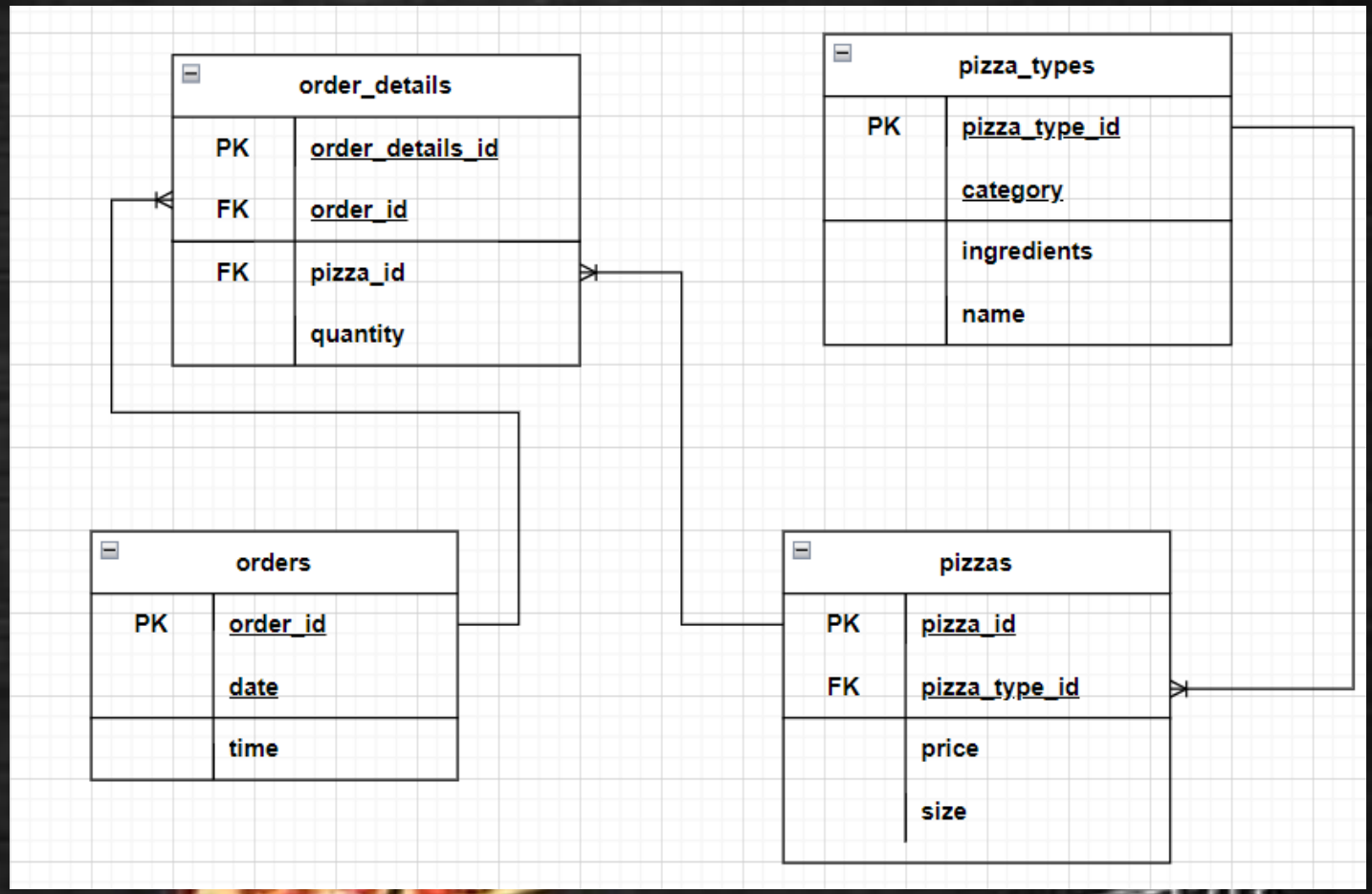
10. Determine the top 3 most ordered pizza types based on revenue.

11. Analyze the cumulative revenue generated over time.

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

SCHEMA

Entity Relationship Diagram



1.Retrive the total number of order placed

```
43
44
45  select count(order_id) as toatal_number_order from orders;
46
47
```

Data Output | Messages | Notifications

	toatal_number_order bigint
1	21350

2.Calculate the total revenue generated by pizza sale.

```
44
45  select round((sum(order_details.quantity * pizzaz.price))::numeric,2)
46      as total_sale
47      from order_details
48      join pizzaz
49      on order_details.pizza_id= pizzaz.pizza_id  ;
50
```

Data Output | Messages | Notifications

SQL

	total_sale numeric
1	2002.85

3. Identify the highest-priced pizza

```
45  select pizza_types.name,pizzaz.price
46        from pizzaz
47       join pizza_types
48       on pizzaz.pizza_type_id = pizzaz.pizza_type_id
49       order by pizzaz.price
50       desc limit 1;
51
52
```

Data Output | Messages | Notifications

SQL

	name character varying (70)	price double precision
1	The California Chicken Pizza	35.95

4.Identify the most common sized pizza ordered.

```
44
45  select pizzaz.size , sum(order_details.quantity)as quantity
46      from pizzaz join order_details
47      on pizzaz.pizza_id = order_details.pizza_id
48      group by pizzaz.size
49      order by quantity
50      desc limit 1;
51
52
```

Data Output | Messages | Notifications

SQL

	size character varying (2)	quantity bigint
1	L	54

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

```
45  select pizza_types.name , sum(order_details.quantity) as quantity
46      from order_details join pizzaz
47      on pizzaz.pizza_id = order_details.pizza_id join  pizza_types
48      on pizza_types.pizza_type_id =  pizzaz.pizza_type_id
49      group by pizza_types.name
50      order by quantity desc limit 5;
51
```

Data Output | Messages | Notifications

SQL

	name character varying (70)	quantity bigint
1	The Italian Supreme Pizza	12
2	The Mexicana Pizza	8
3	The Green Garden Pizza	7
4	The Thai Chicken Pizza	7
5	The Barbecue Chicken Pizza	7

6.Determine the distribution of order by hour per day.

```
8  select extract(hour from time) as hour , count(order_id)
9       as order_count
10           from orders
11               group by hour;
```

Data Output | Messages | Notifications

SQL

	hour numeric	order_count bigint
1	11	1231
2	23	28
3	18	2399
4	19	2009
5	15	1468
6	9	1
7	21	1198
8	17	2336
9	20	1642
10	13	2455
11	10	8
12	16	1920
13	22	663

7.Find the total quantity of each pizza category ordered.

```
15  select category ,sum(quantity) as quantity from  pizza_types join pizzaz
16      on pizza_types.pizza_type_id = pizzaz.pizza_type_id
17          join order_details on pizzaz.pizza_id = order_details.pizza_id
18                group by category
19                   order by  quantity desc ;
20
21
```

Data Output | Messages | Notifications

SQL

	category character varying (20)	quantity bigint
1	Veggie	36
2	Chicken	26
3	Classic	24
4	Supreme	23

8 . find the category-wise distribution of pizzas.

```
14
15  select category, count(name) as count from pizza_types group by category ;
```

Data Output | Messages | Notifications

SQL

	category character varying (20)	count bigint
1	Chicken	5
2	Supreme	1
3	Chicken	1
4	Veggie	9

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

```
14
15  select round((avg(q))::numeric,0) as average_order_per_day from (
16      select orders.date , sum(order_details.quantity) as q from order_details join orders
17      on order_details.order_id = orders.order_id group by orders.date )
18
```

Data Output | Messages | Notifications

SQL

	average_order_per_day numeric
1	120

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

```
15  select pizza_types.name , sum(order_details.quantity * pizzaz.price) as revenue
16  from order_details join pizzaz on pizzaz.pizza_id = order_details.pizza_id
17  join  pizza_types on pizzaz.pizza_type_id = pizza_types.pizza_type_id
18  group by pizza_types.name  order by revenue desc limit 3;
19
```

Data Output | Messages | Notifications

SQL

	name character varying (70)	revenue double precision
1	The Italian Supreme Pizza	211
2	The Mexicana Pizza	145.25
3	The Thai Chicken Pizza	137.25

11. Analyze the cumulative revenue generated over time.

```
12
13  select date, sum(revenue) over (order by date ) as cumulative_revenue from (select orders.date,
14  sum(order_details.quantity * pizzaz.price) as revenue from orders join order_details
15  on order_details.order_id = orders.order_id join pizzaz on
16  order_details.pizza_id=pizzaz.pizza_id group by orders.date)  as sales
17
```

Data Output | Messages | Notifications

SQL

	date date	cumulative_revenue double precision
1	2015-01-01	2002.8500000000001

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

```
13
14
15  select pizzaz.size, sum(order_details.quantity) as total_sold from pizzaz join
16  order_details on pizzaz.pizza_id = order_details.pizza_id
17  group by pizzaz.size order by total_sold  desc;
```

Data Output | Messages | Notifications

SQL

	size character varying (2)	total_sold bigint
1	L	54
2	S	34
3	M	32

THANK YOU!