

SQL PROJECT ON PIZZA SALES



A completely new pizza

@ Koushik_Talluri



Agenda

1

Project description

2

Schema

3

Basic questions

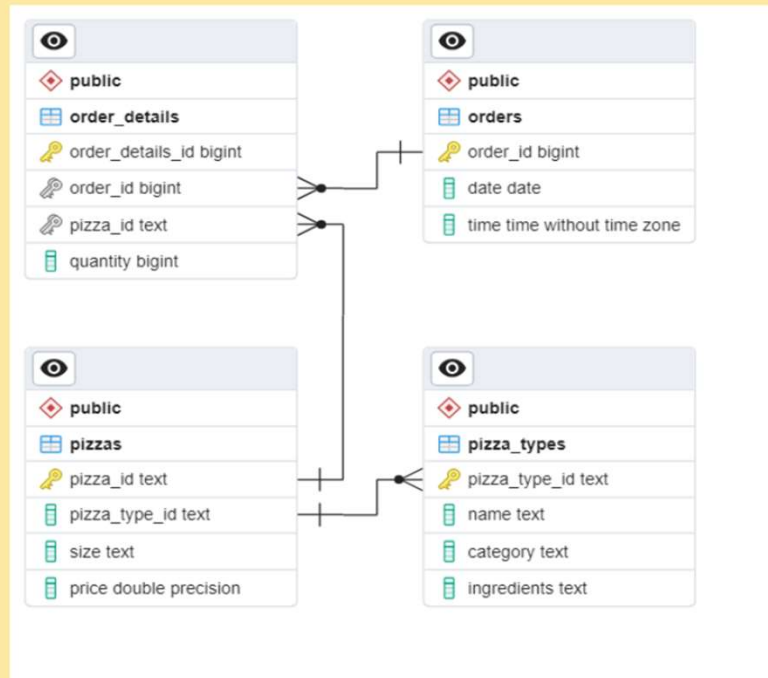
4

Advanced questions



1

Schema













2

Basic Questions

#1. Retrieve the total number of orders placed.

```
23  
24 SELECT count(order_id) AS Total_orders  
25 FROM orders
```











Data Output		Messages	Notifications
        			
	total_orders bigint 		
1	21350		



2

#2. Calculate the total revenue generated from pizza sales.

```
29 SELECT round(sum(od.quantity*p.price)) AS Total_revenue
30 FROM order_details AS od
31 JOIN pizzas AS p ON p.pizza_id= od.pizza_id
```

Data Output		Messages	Notifications
        			
	total_revenue double precision 		
1	817860		



2

#3. Identify the highest-priced pizza.

```
SELECT pt.name, p.price
FROM pizza_types AS pt
JOIN pizzas AS p ON p.pizza_type_id= pt.pizza_type_id
ORDER BY p.price DESC
LIMIT 1
```

Data Output

Messages

Notifications











	<div>name</div> <div>text</div> <div></div>	<div>price</div> <div>double precision</div> <div></div>
1	The Greek Pizza	35.95



2

#4. Identify the most common pizza size ordered.

```
43 SELECT p.size,  
44         count(od.order_details_id) AS Order_count  
45 FROM pizzas AS p  
46 JOIN order_details AS od ON p.pizza_id = od.pizza_id  
47 GROUP BY(p.size)  
48 ORDER BY Order_count DESC  
49 LIMIT 1|
```

Data Output		Messages	Notifications
        			
	size text		order_count bigint
1	L		18526



2

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

```
SELECT pt.name,  
       count(od.order_details_id) 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(pt.name)  
ORDER BY count(od.order_details_id) DESC  
LIMIT 5
```

	name text	quantity bigint
1	The Classic Deluxe Pizza	2416
2	The Barbecue Chicken Pizza	2372
3	The Hawaiian Pizza	2370
4	The Pepperoni Pizza	2369
5	The Thai Chicken Pizza	2315



2

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

```
SELECT pt.category,  
       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(pt.category)  
ORDER BY count(od.order_details_id) DESC
```

	category text	quantity numeric
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050



2

#7. Determine the distribution of orders by hour of the day.

```
74 SELECT extract(HOUR
75             FROM o.time) AS HOUR,
76        count(o.order_id) AS order_count
77 FROM orders AS o
78 GROUP BY (HOUR)
79 ORDER BY order_count DESC |
```

	hour numeric 🔒	order_count bigint 🔒
1	12	2520
2	13	2455
3	18	2399
4	17	2336
5	19	2009



2

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

```
83 SELECT category,  
84      count(name)  
85 FROM pizza_types  
86 GROUP BY(category)
```

	category text	count bigint
1	Supreme	9
2	Chicken	6
3	Classic	8
4	Veggie	9



2

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

```
90 SELECT round(avg(quantity), 2) AS average_quantity
91 FROM
92     (SELECT date(o.date) AS date,
93          sum(od.quantity) AS quantity
94     FROM orders AS o
95     JOIN order_details AS od ON od.order_id = o.order_id
96     GROUP BY(date))AS qnt
```

Data Output		Messages
	average_quantity numeric	
1	138.47	



2

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

```
100 SELECT pt.name,  
101         sum(od.quantity*p.price) AS revenue  
102 FROM order_details AS od  
103 JOIN pizzas AS p ON od.pizza_id=p.pizza_id  
104 JOIN pizza_types AS pt ON pt.pizza_type_id= p.pizza_type_id  
105 GROUP BY(pt.name)  
106 ORDER BY revenue DESC  
107 LIMIT 3
```

	name text	revenue double precision
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5



3



Advanced Questions



3

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

```
111 WITH revenue_per_category AS
112     (SELECT pt.category,
113         round(cast(sum(od.quantity*p.price)AS numeric), 2) AS category_revenue
114     FROM order_details AS od
115     JOIN pizzas AS p ON od.pizza_id=p.pizza_id
116     JOIN pizza_types AS pt ON pt.pizza_type_id= p.pizza_type_id
117     GROUP BY(pt.category)
118     ORDER BY category_revenue DESC),
119     total_revenue AS
120     (SELECT SUM(category_revenue) AS total_revenue
121     FROM revenue_per_category)
122 SELECT rpc.category,
123     rpc.category_revenue,
124     tr.total_revenue,
125     round(cast((rpc.category_revenue / tr.total_revenue) * 100 AS numeric), 2) AS percentage
126 FROM revenue_per_category rpc,
127     total_revenue tr
128 ORDER BY percentage DESC;
```

category text	category_revenue numeric	total_revenue numeric	percentage numeric
Classic	220053.10	817860.05	26.91
Supreme	208197.00	817860.05	25.46
Chicken	195919.50	817860.05	23.96
Veggie	193690.45	817860.05	23.68



3

#2. Analyze the cumulative revenue generated over time.

```
132 SELECT date, sum(revenue) OVER (ORDER BY date) AS cumulative_revenue
133 FROM
134     (SELECT o.date AS date,
135          round(cast(sum(od.quantity*p.price) AS numeric), 2) AS revenue
136     FROM order_details AS od
137     JOIN pizzas AS p ON od.pizza_id=p.pizza_id
138     JOIN orders AS o ON o.order_id=od.order_id
139     GROUP BY (date)) sales
```

	date date	cumulative_revenue numeric
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.60
5	2015-01-05	11929.55



3

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

```
143 SELECT category,  
144         name,  
145         revenue,  
146         rank  
147 FROM  
148     (SELECT category,  
149         name,  
150         revenue,  
151         rank()over(PARTITION BY category  
152                     ORDER BY revenue DESC) AS rank  
153 FROM  
154     (SELECT pt.category,  
155         pt.name,  
156         sum(od.quantity*p.price) AS revenue  
157 FROM pizza_types AS pt  
158 JOIN pizzas AS p ON pt.pizza_type_id=p.pizza_type_id  
159 JOIN order_details AS od ON od.pizza_id=p.pizza_id  
160 GROUP BY (pt.category,  
161         pt.name)) AS a) AS b  
162 WHERE rank<4
```



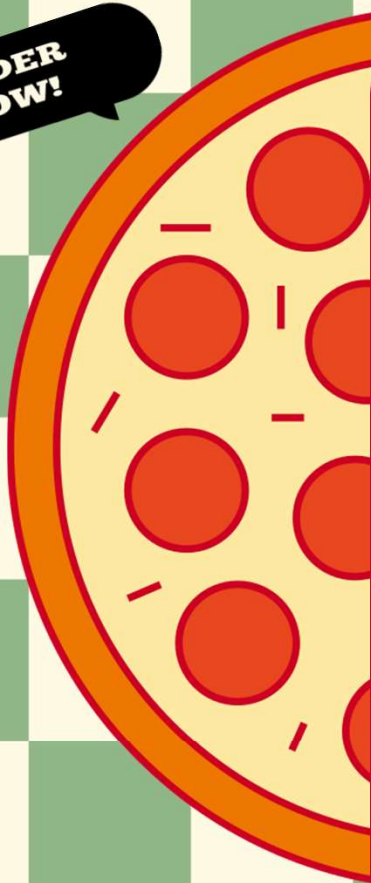
3

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

Data Output Messages Notifications					
	category text	name text	revenue double precision	rank bigint	
1	Chicken	The Thai Chicken Pizza	43434.25	1	
2	Chicken	The Barbecue Chicken Pizza	42768	2	
3	Chicken	The California Chicken Pizza	41409.5	3	
4	Classic	The Classic Deluxe Pizza	38180.5	1	
5	Classic	The Hawaiian Pizza	32273.25	2	
6	Classic	The Pepperoni Pizza	30161.75	3	
7	Supreme	The Spicy Italian Pizza	34831.25	1	
8	Supreme	The Italian Supreme Pizza	33476.75	2	
9	Supreme	The Sicilian Pizza	30940.5	3	
10	Veggie	The Four Cheese Pizza	32265.70000000065	1	
11	Veggie	The Mexicana Pizza	26780.75	2	
12	Veggie	The Five Cheese Pizza	26066.5	3	

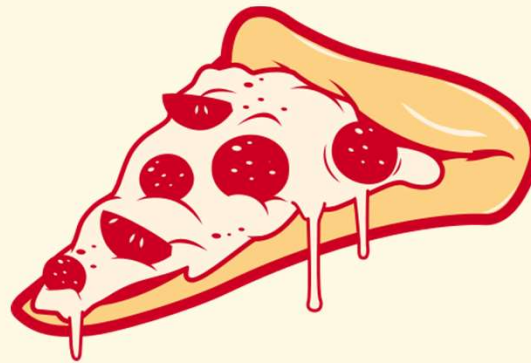


**ORDER
NOW!**



THANKYOU

Like & Comment



Only at # tkoushik77@gmail.com
