

# Pizza Sales Database project

- By Rajashree Chatterjee



# About the project :

There are total 4 database present in pizza-hut file

1. order\_details
2. orders
3. pizza\_types
4. Pizzas

Tries to solve queries using Mysql through the database

Orders :-

A1	B	C	D	E	F	G
order_id	date	time				
1	1/1/2015	11:38:36				
2	1/1/2015	11:57:40				
3	1/1/2015	12:12:28				
4	1/1/2015	12:16:31				

Order\_details :-

A	B	C	D	E
order_det	order_id	pizza_id	quantity	
1	1	hawaiian	1	
2	2	classic_dlx	1	
2	2	five cheese	1	

Pizza-types :-

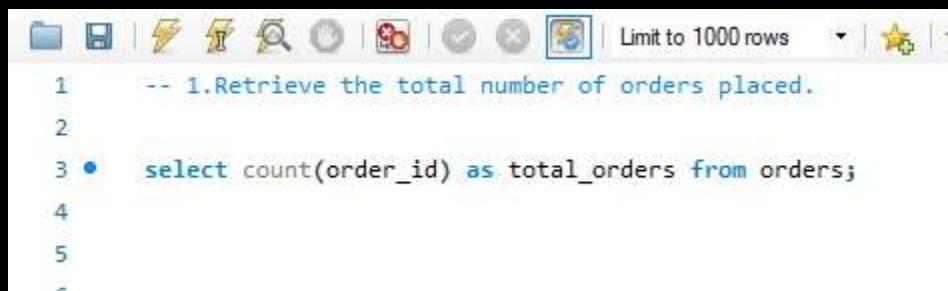
name	category	ingredients
bbq_ckn	The Barbe Chicken	Barbecued Chicken, Red Peppers, Green Peppers, Tomatoes, Red Onions, Barbecue Sauce
cali_ckn	The Califo Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno Peppers, Fontina Cheese, Gouda Cheese
ckn_alfred	The Chick'n Chicken	Chicken, Red Onions, Red Peppers, Mushrooms, Asiago Cheese, Alfredo Sauce
ckn_pesto	The Chick'n Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garlic, Pesto Sauce

Pizzas :-

pizza_id	pizza_type	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75

# 1. Queries:-

-- 1. Retrieve the total number of orders placed.



The screenshot shows a MySQL Workbench interface with a query editor window. The window title bar includes standard icons for file operations, search, and refresh, followed by a dropdown menu set to 'Limit to 1000 rows'. The main area contains the following SQL code:

```
1 -- 1.Retrieve the total number of orders placed.  
2  
3 • select count(order_id) as total_orders from orders;  
4  
5  
6
```

output-



The screenshot shows the 'Result Grid' tab of MySQL Workbench displaying the output of the executed query. The grid has one column labeled 'total\_orders' and one row containing the value '21350'. There are navigation arrows at the bottom left of the grid.

total_orders
21350

## 2. Queries:-

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

```
8      -- 2.Calculate the total revenue generated from pizza sales.  
9  
10 •  SELECT  
11     ROUND(sum(order_details.quantity * pizzas.price),  
12             2) AS total_revenue  
13   FROM  
14     order_details  
15   JOIN  
16     pizzas ON pizzas.pizza_id = order_details.pizza_id;  
17  
18
```

output-

Result Grid	
	total_revenue
▶	817860.05

# 3. Queries:-

-- 3.Identify the highest-priced pizza.

```
21
22      -- 3.Identify the highest-priced pizza.
23
24 •  SELECT
25      pizza_types.name, pizzas.price
26  FROM
27      pizza_types
28      JOIN
29          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
30  ORDER BY pizzas.price DESC
31  LIMIT 1;
32
33
```

output-

Result Grid		Filter Rows:
	name	price
▶	The Greek Pizza	35.95

# 4. Queries:-

-- 4.Identify the most common pizza size ordered.

```
36  
37      -- 4.Identify the most common pizza size ordered.  
38 •  SELECT  
39      pizzas.size,  
40      COUNT(order_details.order_details_id) AS order_count  
41  FROM  
42      pizzas  
43      JOIN  
44      order_details ON pizzas.pizza_id = order_details.pizza_id  
45  GROUP BY pizzas.size  
46  ORDER BY order_count DESC;  
47  
48
```

output-

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

# 5. Queries:-

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

```
50
51      -- 5.List the top 5 most ordered pizza types along with their quantities.
52 •  SELECT
53      pizza_types.name, SUM(order_details.quantity) AS order_pizza
54  FROM
55      pizza_types
56      JOIN
57      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
58      JOIN
59      order_details ON order_details.pizza_id = pizzas.pizza_id
60  GROUP BY pizza_types.name
61  ORDER BY order_pizza
62  LIMIT 5;
63
```

output-

	name	order_pizza
▶	The Brie Carre Pizza	490
	The Mediterranean Pizza	934
	The Calabrese Pizza	937
	The Spinach Supreme Pizza	950
	The Soppressata Pizza	961

# 6. Queries:-

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

```
1      -- 6.Join the necessary tables to find the total quantity of each pizza category ordered.
2 •  use pizzahut;
3 •  SELECT
4      pizza_types.category,
5      SUM(order_details.quantity) AS quantity
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10     JOIN
11     order_details ON order_details.pizza_id = pizzas.pizza_id
12     GROUP BY pizza_types.category
13     ORDER BY quantity DESC;
14
15
```

output-

Result Grid		Filter Row
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

# 7. Queries:-

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

```
18  
19      -- 7.Determine the distribution of orders by hour of the day.  
20 •  SELECT  
21      HOUR(order_time), COUNT(order_id)  
22  FROM  
23    orders  
24 GROUP BY HOUR(order_time);  
25
```

output-

	HOUR(order_time)	COUNT(order_id)
▶	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

# 8. Queries:-

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

```
28  
29      -- 8.Join relevant tables to find the category-wise distribution of pizzas.  
30 •  SELECT  
31      category, COUNT(name)  
32  FROM  
33      pizza_types  
34  GROUP BY category;  
35  |  
36  
37
```

output-

Result Grid		Filter Rows:
	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

# 9. Queries:-

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

```
39      -- 9.Group the orders by date and calculate the average number of pizzas ordered per day.
40 •  SELECT
41      ROUND(AVG(quantity), 0) as avg_order
42  FROM
43  (
44      SELECT
45          orders.order_date, SUM(order_details.quantity) AS quantity
46      FROM
47          orders
48      JOIN order_details ON orders.order_id = order_details.order_id
49      GROUP BY orders.order_date) AS order_quantity;
50
```

output-

Result Grid	
	avg_order
▶	138

# 10. Queries:-

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

```
52
53      -- 10.Determine the top 3 most ordered pizza types based on revenue.
54 •  SELECT
55      pizza_types.name, SUM(quantity * pizzas.price) AS revenue
56  FROM
57      pizza_types
58      JOIN
59      pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
60      JOIN
61      order_details ON order_details.pizza_id = pizzas.pizza_id
62  GROUP BY pizza_types.name
63  ORDER BY revenue DESC
64  LIMIT 3;
```

output-

Result Grid		Filter Rows:	Export
	name	revenue	
▶	The Thai Chicken Pizza	43434.25	
	The Barbecue Chicken Pizza	42768	
	The California Chicken Pizza	41409.5	

**Solved some queries using Mysql . Its helps to gathering knowledge and strive to run quires in optimal way . My first project on Mysql database . Tried to solve as my level best . It's help to understand database and how it works...**

**Thank you ...**