

[MENU](#)

[HOME](#)

[PIZZAHUT](#)



PIZZAHUT



[NEXT >](#)



BY ANOUSHKAA DEEPAK GAVANKAR

IN THIS PROJECT I HAVE UTILIZED SQL QUERIES TO SOLVE QUESTIONS THAT WERE RELATED TO PIZZA SALES.

NEXT >

The screenshot shows a MySQL Workbench interface. The toolbar at the top includes icons for file operations, database selection, and various tools. A dropdown menu labeled "Limit to 1000 rows" is visible. The main area contains a numbered list of SQL statements:

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

The screenshot shows the "Result Grid" section of MySQL Workbench. It displays the output of the executed SQL query. The grid has two columns: one for the column name and one for the data. The data row contains a right-pointing arrow icon followed by the value "21350".

	total_orders
▶	21350

NEXT >



```
1  -- Calculate the total revenue generated from pizza sales.  
2  
3 • select  
4  round(sum(order_details.quantity * pizzas.price),2) as total_revenue  
5  from order_details join pizzas  
6  on order_details.pizza_id = pizzas.pizza_id
```

Result Grid |

	total_revenue
▶	817860.05

NEXT >

1 -- Identify the highest-priced pizza.

2

3 • `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`

Result Grid | Filter Rows

	name	price
▶	The Greek Pizza	35.95

NEXT >

```
1  -- Identify the most common pizza size ordered.  
2  
3 • select pizzas.size , count(order_details.order_details_id) as order_count  
4  from pizzas join order_details  
5  on pizzas.pizza_id = order_details.pizza_id  
6  group by pizzas.size order by order_count desc limit 1
```

Result Grid

	size	order_count
▶	L	18526

NEXT >

```
1 -- List the top 5 most ordered pizza types along with their quantities.  
2  
3 • select pizza_types.name,sum(order_details.quantity) as Quantity  
4   from pizza_types join pizzas  
5     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
6   join order_details  
7     on order_details.pizza_id = pizzas.pizza_id  
8   group by pizza_types.name order by Quantity desc limit 5
```

| Result Grid | Filter Rows:

	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

NEXT >

```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.  
2  
3 • select pizza_types.category , sum(order_details.quantity) as Quantity  
4   from pizza_types join pizzas  
5     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
6   join order_details  
7     on order_details.pizza_id = pizzas.pizza_id  
8   group by pizza_types.category order by Quantity desc
```

Result Grid

	category	Quantity
▶	Classic	14888
▶	Supreme	11987
▶	Veggie	11649
▶	Chicken	11050

NEXT >

```
1    -- Determine the distribution of orders by hour of the day.  
2  
3 • select hour(order_time) as hour, count(orders.order_id) as order_count from orders  
4 group by hour
```

Result Grid | Filter

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336

NEXT >

```
1 --- Join relevant tables to find the category-wise distribution of pizzas.  
2  
3 • select category, count(name) as distribution  
4 from pizza_types  
5 group by category
```

Result Grid | Filter

	category	distribution
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

NEXT >

```
1   -- Group the orders by date and calculate the average number of pizzas ordered per day.  
2  
3 • select round(avg(total_pizzas),0) as avg_no_of_pizzas  
4   from  
5   (select orders.order_date , sum(order_details.quantity) as total_pizzas  
6   from orders join order_details  
7   on orders.order_id = order_details.order_id  
8   group by orders.order_date) as total pizza orders resp dates
```

Result Grid

	avg_no_of_pizzas
▶	138

NEXT >

```
1  -- Determine the top 3 most ordered pizza types based on revenue.  
2  
3 • select pizza_types.name ,  
4   sum(pizzas.price*order_details.quantity) as revenue  
5   from pizza_types join pizzas  
6   on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7   join order_details  
8   on pizzas.pizza_id = order_details.pizza_id  
9   group by pizza_types.name order by revenue desc limit 3
```

Result Grid | Filter Rows:

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

NEXT >

```
1      -- Calculate the percentage contribution of each pizza type to total revenue.  
2  
3 • Ⓛ select pizza_types.category ,round(sum(pizzas.price*order_details.quantity)  
4   /  
5   (select round(sum(pizzas.price * order_details.quantity),2)  
6   as total_sales  
7   from pizzas join order_details  
8   on pizzas.pizza_id = order_details.pizza_id)*100,2) as revenue  
9   from pizza_types join pizzas  
10  on pizza_types.pizza_type_id = pizzas.pizza_type_id  
11  join order_details  
12  on pizzas.pizza_id = order_details.pizza_id  
13  group by pizza_types.category  
14  order by revenue desc
```

Result Grid |

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

```
1  -- Analyze the cumulative revenue generated over time.  
2  
3 • select order_date, sum(revenue) over (order by order_date) as cum_revenue  
4  from  
5  (select orders.order_date, sum(order_details.quantity * pizzas.price) as revenue  
6  from orders join order_details  
7  on orders.order_id = order_details.order_id  
8  join pizzas  
9  on pizzas.pizza_id = order_details.pizza_id  
10 group by orders.order_date) as sales
```

Result Grid | Filter Rows:

	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

```

1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2
3 • select category, name, revenue
4   from
5   (select category, name, revenue,
6    rank() over (partition by category order by revenue desc) as rn
7   from
8   (select pizza_types.category , pizza_types.name,
9    sum(order_details.quantity * pizzas.price) as revenue
10  from pizza_types join pizzas
11  on pizza_types.pizza_type_id = pizzas.pizza_type_id
12  join order_details
13  on order_details.pizza_id = pizzas.pizza_id
14  group by pizza_types.category , pizza_types.name) as a) as b
15  where rn<=3

```

Result Grid | Filter Rows:

	Category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5

THANK YOU!

