



sql project on pizza sales

By smile

Introduction

Hello, my name is smile and in this project I utilize SQL queries to solve questions that were related to pizza sales.



SCHEMAS

Filter objects

▼ pizzahut

▼ Tables

orders

orders_details

pizza_types

pizzas

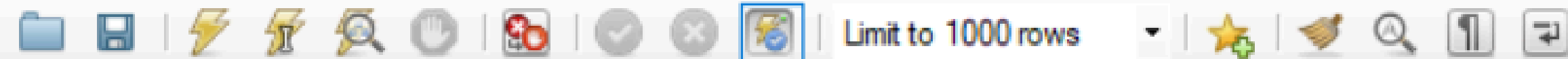
Views

Stored Procedures

Functions

▶ sys

▶ xyz



Limit to 1000 rows

```
1 create database pizzahut;
2 • create table orders (
3     order_id int not null,
4     order_date date not null,
5     order_time time not null,
6     primary key(order_id) );
7
8 • create table orders_details (
9     order_details_id int not null,
10    order_id int not null,
11    pizza_id text not null,
12    quantity int not null,
13    primary key(order_details_id) );
```

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

Navigator

SCHEMAS

Filter objects

pizzahut

Tables

orders

orders_details

pizza_types

pizzas

Views

Stored Procedures

Functions

sys

xyz

Query 1SQL File 2*SQL File 3*SQL File 4*pizzahut



InfoTablesColumnsIndexesTriggersViewsStored ProceduresFunctionsGrantsEvents

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length	Index Length	Data Free	Auto Incre...
orders	InnoDB	10	Dynamic	22037	72	1.5 MiB	0.0 bytes	0.0 bytes	4.0 MiB	0
orders_details	InnoDB	10	Dynamic	48758	54	2.5 MiB	0.0 bytes	0.0 bytes	4.0 MiB	0
pizza_types	InnoDB	10	Dynamic	32	512	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
pizzas	InnoDB	10	Dynamic	96	170	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0

Query 1 SQL File 2* SQL File 3* SQL File 4* pizzahut ×								
Info Tables Columns Indexes Triggers Views Stored Procedures Functions Grants Events								
Table	Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra
orders	order_date	date		NO			select,insert,update,references	
orders	order_id	int		NO			select,insert,update,references	
orders	order_time	time		NO			select,insert,update,references	
orders_details	order_details_id	int		NO			select,insert,update,references	
orders_details	order_id	int		NO			select,insert,update,references	
orders_details	pizza_id	text		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
orders_details	quantity	int		NO			select,insert,update,references	
pizza_types	category	text		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
pizza_types	ingredients	text		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
pizza_types	name	text		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
pizza_types	pizza_type_id	text		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
pizzas	pizza_id	text		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
pizzas	pizza_type_id	text		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
pizzas	price	double		YES			select,insert,update,references	
pizzas	size	text		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	

Count: 15

Refresh

Table	Name	Unique	Index...	Index Comment	Column	Seq in Index	Packed	Collat...	Cardi...	S
 orders	PRIMARY	Yes	BTREE		order_id	1		A	21853	
 orders_details	PRIMARY	Yes	BTREE		order_details_id	1		A	47445	

Count: 2

Refresh



Questions

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

```
select count(order_id) as total_orders from orders;
```

Result Grid	
	total_orders
▶	21350

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

```
SELECT  
ROUND(SUM(orders_details.quantity * pizzas.price),  
2) AS total_sales  
FROM  
orders_details  
JOIN  
pizzas ON pizzas.pizza_id = orders_details.pizza_id;  
.
```

Result Grid	
	total_sales
▶	817860.05

-- Identify the highest-priced pizza.

```
select pizzas_types.name, pizza.price  
  from pizza_types join pizzas  
 on pizza_types.pizza_type_id=pizza.pizza_type_id  
 order by pizzas.price desc;
```

.

Result Grid	
	total_sales
▶	817860.05

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


```
select pizza_types.name,  
       sum( orders_details.quantity)  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
  join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name  
order by pizza_types.name  
;
```

Result Grid			Filter Rows:
	size	order_count	
▶	L	18526	
	M	15385	
	S	14137	
	XL	544	
	XXL	28	

-- Determine the most ordered pizza types based on revenue.

```
select pizza_types.name,  
orders_details.quantity * pizzas.price as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id;
```

Result Grid

 Filter Rows:

	name	revenue
▶	The Hawaiian Pizza	13.25
	The Classic Deluxe Pizza	16
	The Five Cheese Pizza	18.5
	The Italian Supreme Pizza	20.75
	The Mexicana Pizza	16
	The Thai Chicken Pizza	20.75
	The Italian Supreme Pizza	16.5
	The Prosciutto and Arugula Pizza	20.75
	The Italian Supreme Pizza	16.5
	The Italian Supreme Pizza	16.5
	The Barbecue Chicken Pizza	12.75
	The Greek Pizza	12

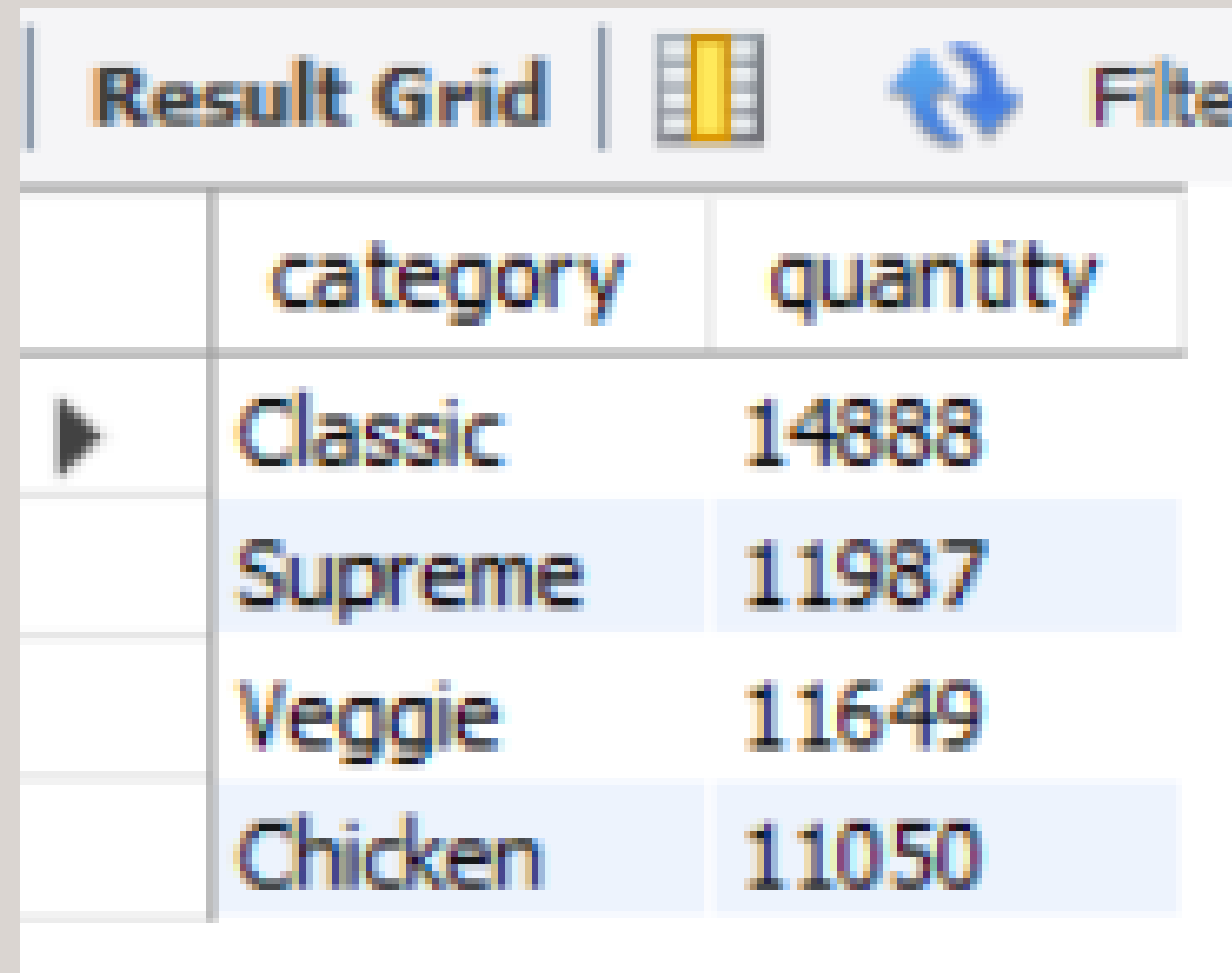
-- **Group the orders by date and calculate the average number of pizzas ordered per day.**

```
select round(avg(quantity),0) from  
(select orders.order_date, sum(orders_details.quantity) as quantity  
  from orders join orders_details  
    on orders.order_id=orders_details.order_id  
 group by orders.order_date) AS order_quantity;
```

Result Grid		Filter Rows:	
	round(avg(quantity),0)		
▶	138		

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

```
select pizza_types.category,  
sum(orders_details.quantity) as quantity  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by quantity desc;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains a table with two columns: 'category' and 'quantity'. The data is sorted in descending order of quantity. The first row is 'Classic' with a quantity of 14888. The second row is 'Supreme' with a quantity of 11987. The third row is 'Veggie' with a quantity of 11649. The fourth row is 'Chicken' with a quantity of 11050. The table has a light blue header and alternating light blue and white rows for the data.

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



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

```
SELECT  
  HOUR(order_time), COUNT(order_id)  
FROM  
  orders  
GROUP BY HOUR(order_time);
```

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

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

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

Result Grid   Filter Rows: <input type="text"/>		
	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

The background is a light gray color. It features several abstract, thin gray lines. On the left side, there are wavy, organic lines. On the right side, there is a prominent spiral pattern that starts from the top right and moves towards the center. Below the spiral, there are more wavy lines.

Thank You

smile Jhorar