Dominoes Pizza Sale

Introduction

Hello, my name is Atharv. In this project, I am showcasing my SQL skills and practicing my abilities by creating a comprehensive database. I've developed queries, imported data, and utilized ChatGPT for generating questions, all to demonstrate my proficiency in SQL and my ability to analyze and present data effectively.

Project Goals

Create Database

Goal: Set up a SQL database named "dominoes" to store the dataset.

Import Data

Goal: Import a CSV file from Kaggle into the database accurately.

Develop Queries

Goal: Create and execute SQL queries at three difficulty levels (Simple, Medium, Intermediate).

Generate Questions

Goal: Use Chatgpt to create questions based on the SQL queries.

Analyze Results

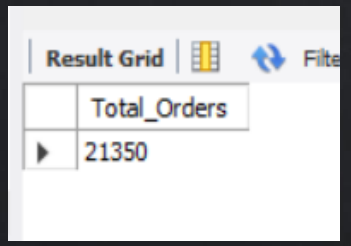
Goal: Analyze query results and provide clear insights.

Prepare Presentation

Goal: Compile the project details into a concise PowerPoint presentation.

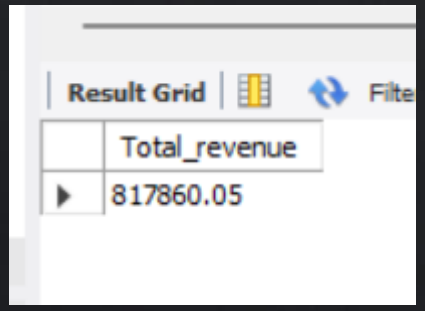
Retrieve the total number of orders placed.

```sql
select count(order_id) as Total_Orders from orders ;
```



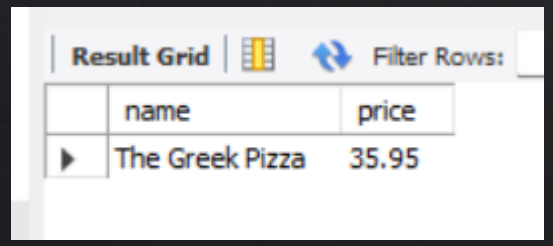
Calculate the total revenue generated from pizza sales.

```
SELECT
    ROUND(SUM(order_details.Quantity * pizzas.price),
            2) AS Total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.Pizza_id = pizzas.pizza_id
```



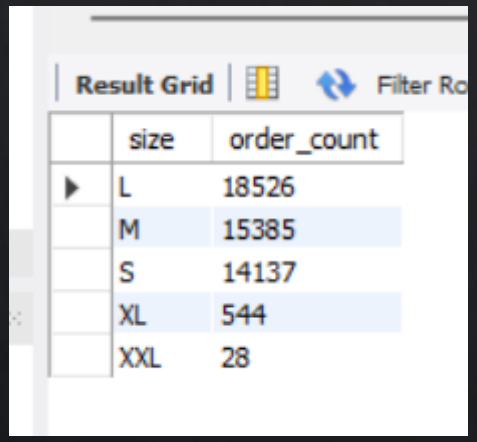
Identify the highest-priced pizza.

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1 ;
```



Identify the most common pizza size ordered.

```
SELECT
    pizzas.size, COUNT(Order_details_id) as order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.Pizza_id
GROUP BY pizzas.size order by order_count desc ;
```



Result Grid | Filter Ro

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

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

```sql
SELECT
    pizza_types.name AS pizza_name,
    SUM(order_details.Quantity) AS Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.Pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Quantity DESC limit  5 ;
```

Result Grid | Filter Rows:

pizza_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

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

```sql
SELECT
    pizza_types.category,
    SUM(order_details.Quantity) AS Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.Pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category order by Quantity desc ;
```

Result Grid | Filter Rows:

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS Hours, COUNT(order_id) AS count
FROM
    orders
GROUP BY Hours
```

Result Grid

Hours	count
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

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

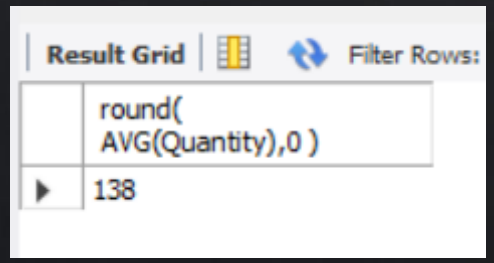
```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

Result Grid | Filter Rows:

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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 AS Date,
            SUM(order_details.quantity) AS Quantity
    FROM
        orders
    JOIN order_details ON orders.Order_id = order_details.Order_id
    GROUP BY Date) as order_quantity;
```



Result Grid | Filter Rows:

	round(AVG(Quantity),0)
▶	138

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.Quantity * pizzas.price) AS reveneu
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.Pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY reveneu desc
LIMIT 3;
```

Result Grid | Filter Rows:

name	reveneu
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Analyze the cumulative revenue generated over time.

```
select order_date, sum(revenue) over (order by Order_date) as cumulative_Revenue
from
(select orders.Order_date,
sum(pizzas.price * order_details.Quantity) as revenue  from
pizzas join order_details on pizzas.pizza_id = order_details.Pizza_id
join
orders on orders.Order_id = order_details.Order_id
group by orders.Order_date) as sales
```

Result Grid | Filter Rows:

order_date	cumulative_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

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

```sql
Select name , reveneu
from
(Select category , name , reveneu , rank() over ( partition by category order by reveneu desc) as Rn
from
( select pizza_types.category ,pizza_types.name ,
sum(order_details.Quantity * pizzas.price)as reveneu from
pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details on order_details.Pizza_id = pizzas.pizza_id
group by  pizza_types.category ,pizza_types.name  ) as a) as b Where rn <= 3 ;
```

Result Grid | Filter Rows:

name	reveneu
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75

THANK YOU