

↓

SQL project

Pizzas

Sales report



Monirul Islam

2024

-- Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

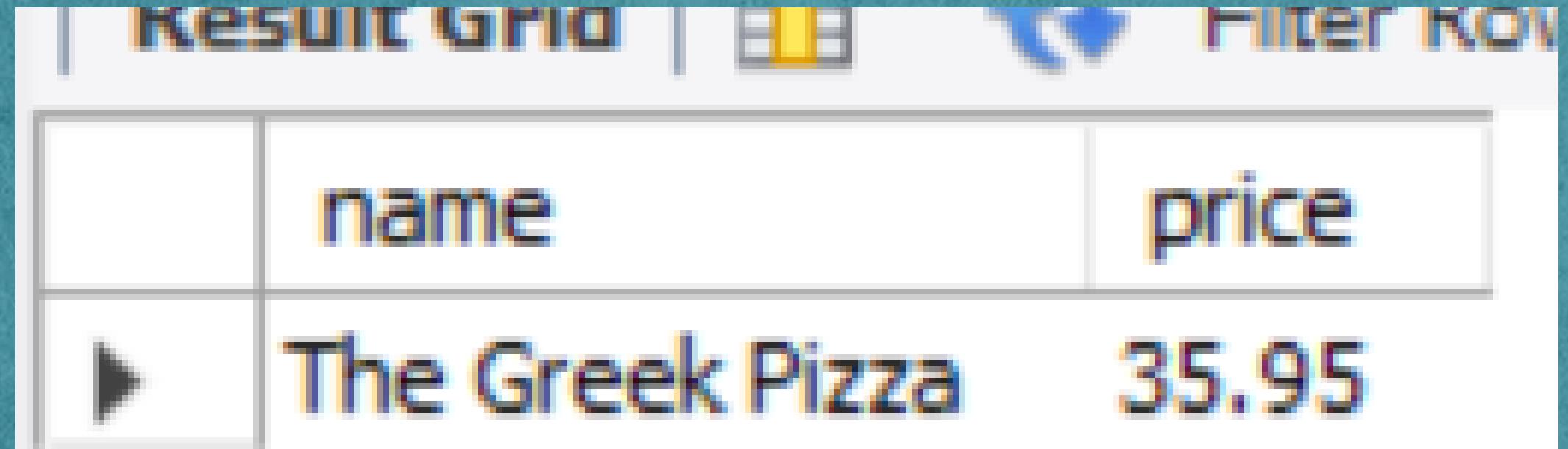
Result Grid	
	total_orders
▶	21350

-- Calculate the total revenue generated from pizza sales.

```
SELECT  
round(SUM(order_details.quantity  
* pizzas.price),2) AS Total_Sales  
FROM  
order_details  
JOIN  
pizzas ON  
order_details.pizza_id =  
pizzas.pizza_id;
```

Result Grid	
	Total_Sales
▶	817860.05

```
##the highest-priced pizza
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;
```



A screenshot of a database query results grid. The grid has a header row with two columns: 'name' and 'price'. Below the header, there is one data row containing the values 'The Greek Pizza' and '35.95'. The 'name' column has a small arrow icon pointing to it.

name	price
The Greek Pizza	35.95

-- Identify the most common pizza size ordered

```
SELECT
    quantity, COUNT(order_details_id)
FROM
    order_details
GROUP BY quantity;
```

```
SELECT
    pizzas.size, COUNT(order_details.order_details_id)
FROM
    pizzas
    JOIN
        order_details ON pizzas.pizza_id =
order_details.pizza_id
GROUP BY pizzas.size
ORDER BY pizzas.size;
```

size	COUNT(order_details.order_details_id)
L	18526
M	15385
S	14137
XL	544
XXL	28

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

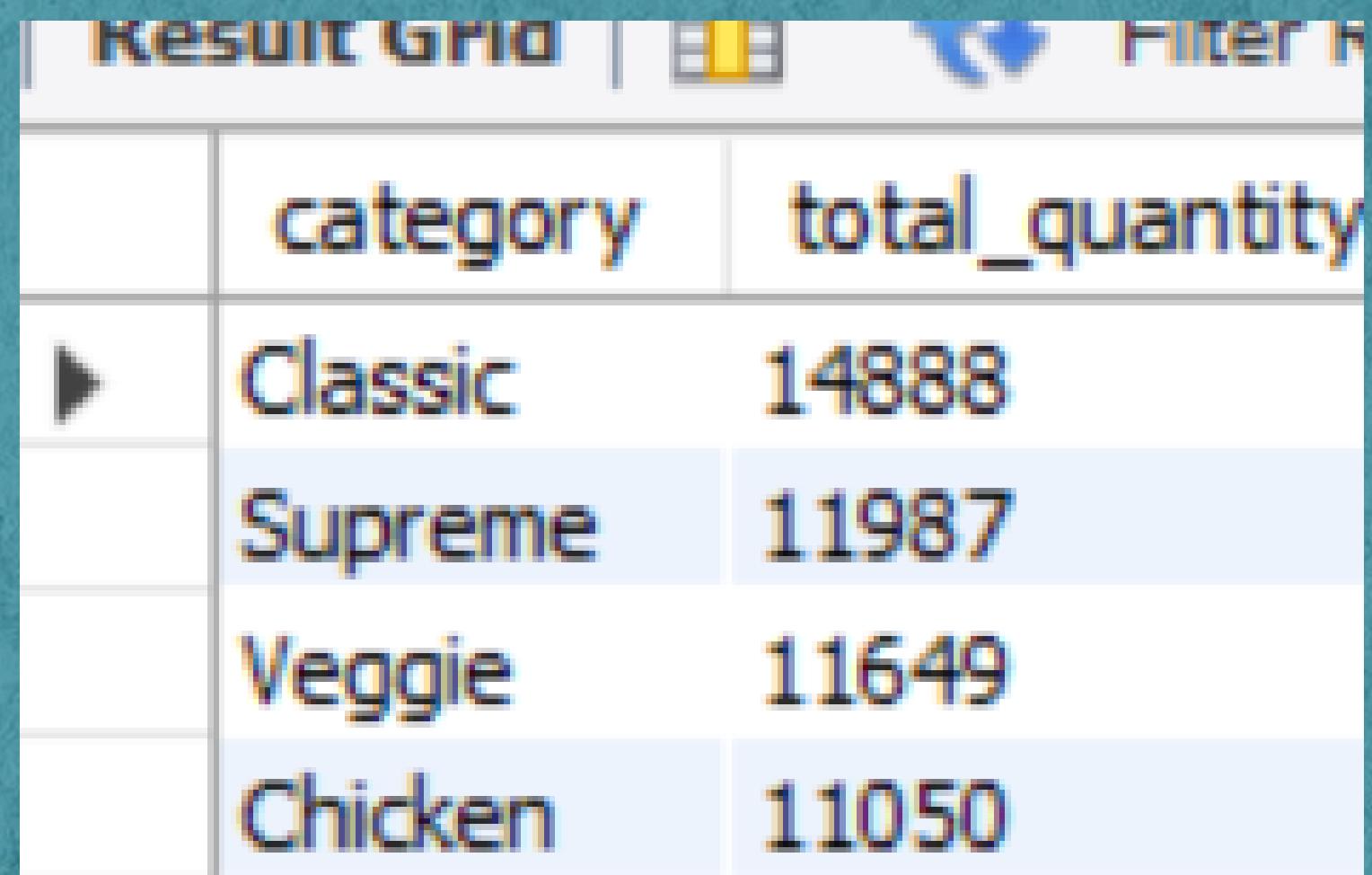
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS
    quantity
FROM
    pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id =
    pizzas.pizza_type_id
    JOIN
        order_details ON order_details.pizza_id =
    pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

The screenshot shows a table with two columns: 'name' and 'quantity'. The table lists five rows of data, each representing a pizza type and its total quantity. The rows are ordered by quantity in descending order. The first row is highlighted in light blue, indicating it is the current selection. The table has a header row with column names and a footer row with summary statistics. The background of the table cells is white, and the text color is black. The overall layout is clean and professional, typical of a database management system interface.

Total Rows: 5	
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.

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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 total_quantity DESC;
```



	category	total_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);
```

Form Editor | Navigate: ⟲ ⟳ 1 / 15 ⟲ ⟳ |

HOUR(order_time):	11
COUNT(order_id):	1231

-- 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(pizza_order), 0) as Avg_pizza_ord
FROM
    (SELECT
        orders.order_date,
        SUM(order_details.quantity) AS
    pizza_order
    FROM
        orders
    JOIN order_details ON orders.order_id =
    order_details.order_id
    GROUP BY order_date) AS order_quantity;
```

Result Grid	
	Avg_pizza_ord
	138

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

SELECT

 pizza_types.name,
 SUM(order_details.quantity * pizzas.price) AS revenue

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 revenue DESC

LIMIT 3;

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

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

```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS Total_Sales
    )
    FROM
        order_details
        JOIN
            pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
    2) AS revenue
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 revenue DESC;
```

Result Grid

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

-- Analyze the cumulative revenue generated over time..

```
select order_date,  
sum(revenue)over(order by order_date as
```

cum_revenue

from

(select orders.order_date,

```
sum(order_details.quantity * pizzas.price) as revenue
```

```
from order_details join orders
```

on order_details.order_id = orders.order_id

join pizzas

on pizzas.pizza_id = order_details.pizza_id

```
group by orders.order_date) as sales;
```

FORM Editor | Navigate: ▲ ▾ ⌂ ⌃ ⌄

Order_date:	2015-01-01
Cum_revenue:	2713.850000000004

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

```
select name, revenue
from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as Rn
from
(select pizza_types.category,pizza_types.name,
sum(order_details.quantity* pizzas.price) as revenue
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;
```

