PIZZA SALES PROJECT SQL

Presented by: Sourabh Sharma

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

Query 1 | -- Retrieve the total number of or...

Limit to 1000 rows

```sql
create database pizzamo;
-- 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.

Query 1 -- Retrieve the total number of or... -- Calculate the total revenue ge...

Limit to 1000 rows

```sql
-- 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.

```sql
-- Identify 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;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

Identify the most common pizza...

Limit to 1000 rows

```sql
-- Identify the most common pizza size ordered.
SELECT
    quantity, COUNT(order_details_id)
FROM
    orders_details
GROUP BY quantity;
```

Result Grid | Filter Rows:

quantity	COUNT(order_details_id)
1	47693
2	903
3	21
4	3

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

Limit to 1000 rows

```sql
-- List the top 5 most ordered pizza
-- types along with their quantities.
-- So we have to join three tables because of
-- dissimilarities bewteen 2 table
SELECT
    pizza_types.name, 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.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

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
-- 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;
```

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
-- Determine the distribution of orders by hour of the day.
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid

hour	order_cou
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.

Join relevant tables to find the c...

Limit to 1000 rows

```sql
-- 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:

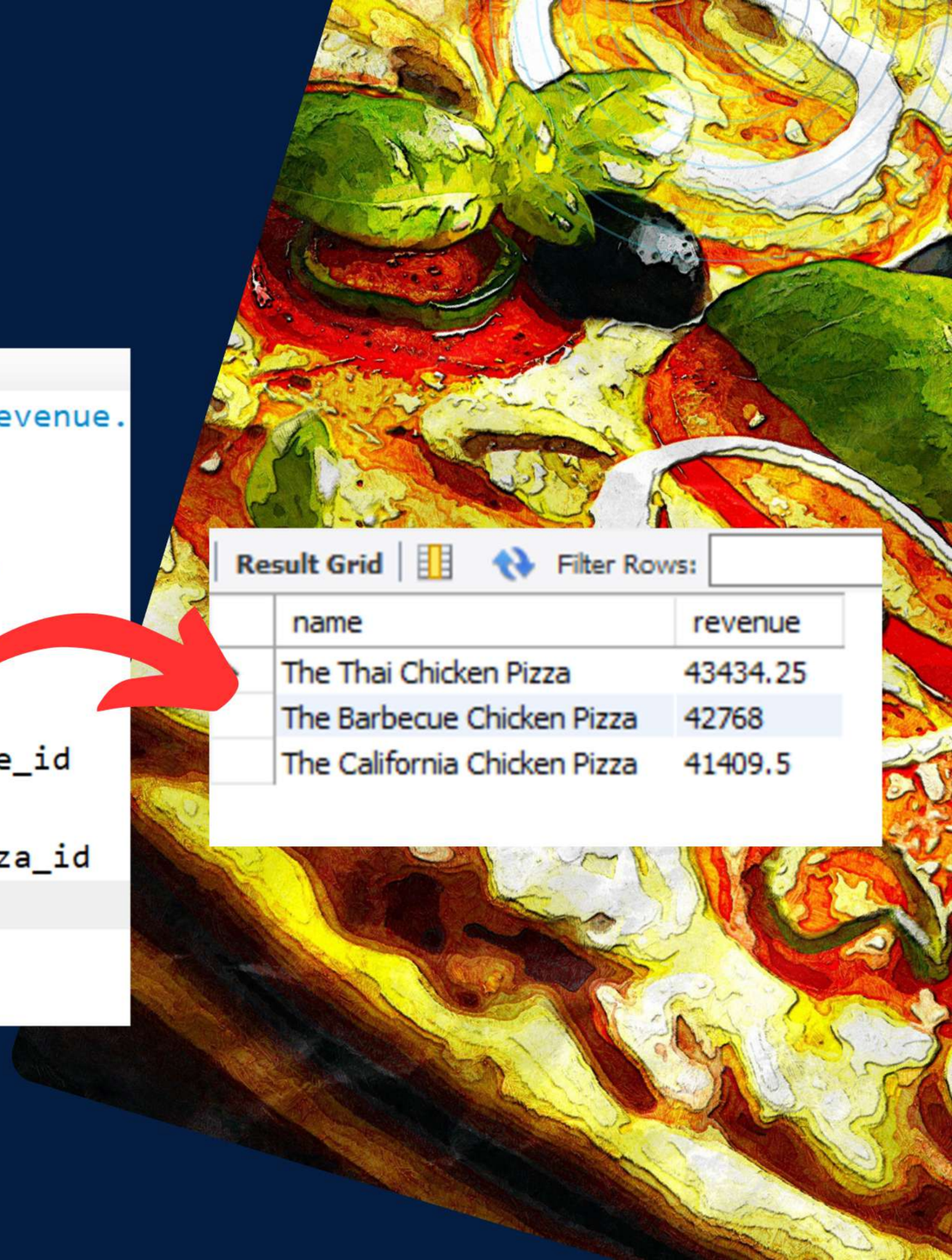
category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    pizza_types.name,
    SUM(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
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



CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pizza_types.category,
    ROUND((SUM(orders_details.quantity * pizzas.price) / (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)) * 100,
            2) 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
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Result Grid | Filter

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
select order_date,
sum(revenue)over (order by order_date) as cum_revenue
from
(SELECT
    orders.order_date,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    orders_details
JOIN
    pizzas
ON
    orders_details.pizza_id = pizzas.pizza_id
JOIN
    orders
ON
    orders.order_id = orders_details.order_id
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
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
-- 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(orders_details.quantity * pizzas.price) AS revenue
        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,
            pizza_types.name
    ) AS a) as b
    where rn <= 3;
```

Result Grid | Filter Rows:

name	revenue
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
The Five Cheese Pizza	26066.5

THANK YOU SO MUCH
BEST REGARDS
~SOURABH SHARMA