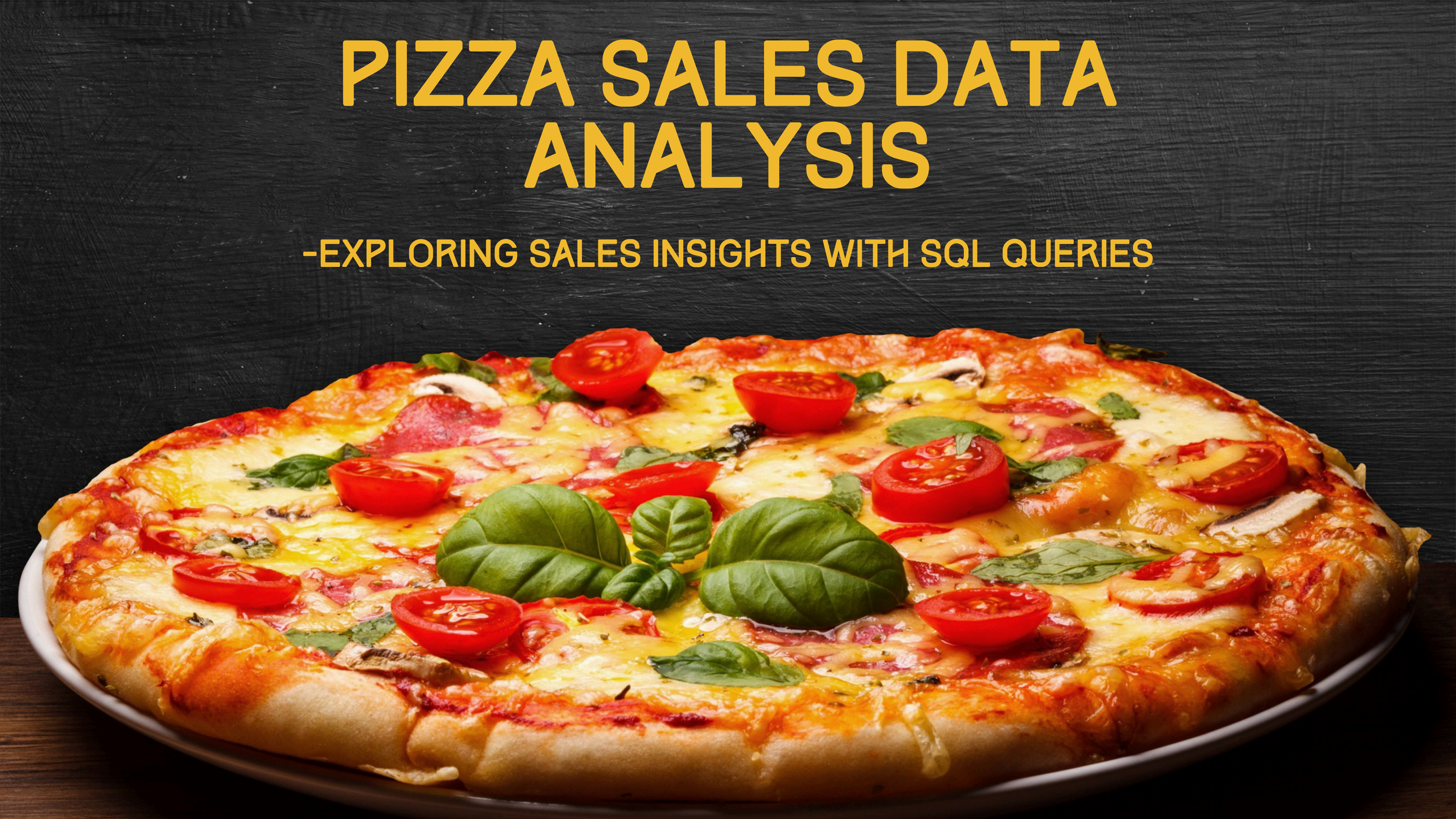
PIZZA SALES DATA ANALYSIS

-EXPLORING SALES INSIGHTS WITH SQL QUERIES



Objective:

To analyze pizza sales data using SQL and extract meaningful insights that can drive business decisions.

Key Areas of Focus:

- Basic insights: Orders, revenue, popular pizzas.
- Intermediate insights: Time-based and category-based distributions.
- Advanced insights: Revenue contributions and cumulative analysis.

DATASET OVERVIEW

The pizza sales dataset comprises four key files: order_details.csv detailing order-specific data like pizza IDs and quantities; orders.csv capturing order-level information including dates and times; pizza_types.csv describing pizza categories and their ingredients; and pizzas.csv providing details on individual pizzas, such as size and price. Together, these files enable comprehensive analysis of order patterns, revenue insights, and category-based distributions, forming the foundation for SQL-based business insights.

DATABASE DESIGN

To ensure seamless data integration, the following tables were created in the database:

```sql
CREATE DATABASE PIZZAHOME;

CREATE TABLE ORDERS(
ORDER_ID INT NOT NULL, ORDER_DATE DATE NOT NULL, ORDER_TIME TIME NOT NULL,
PRIMARY KEY(ORDER_ID));

CREATE TABLE ORDER_DETAILS(
ORDER_DETAILS_ID INT NOT NULL,
ORDER_ID INT NOT NULL, PIZZA_ID TEXT NOT NULL, QUANTITY INT NOT NULL,
PRIMARY KEY(ORDER_DETAILS_ID));
```

Beautify/reformat the SQL

BASIC ANALYSIS

1. Retrieve the total number of orders placed.

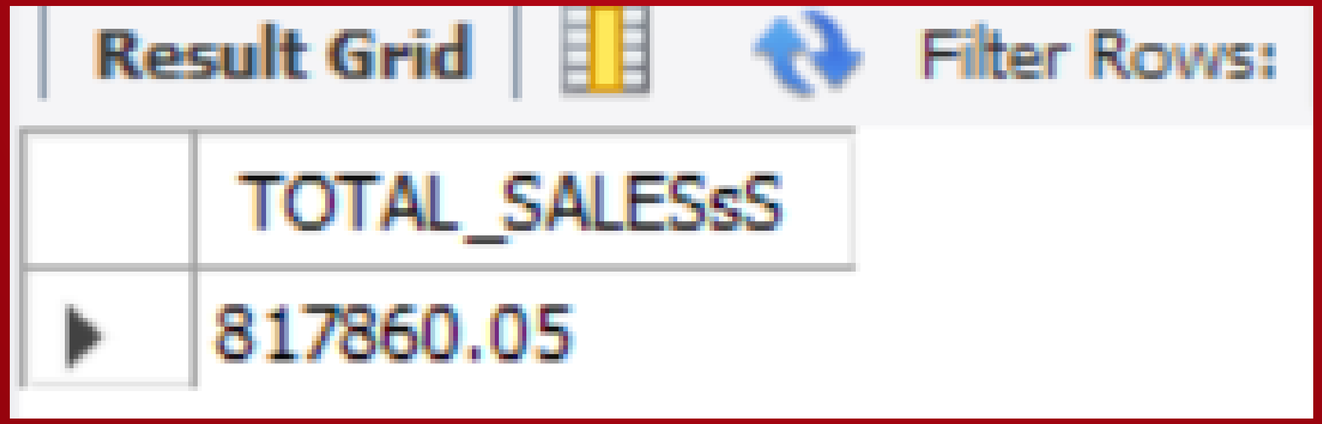
```
1    -- Retrieve the total number of orders placed.
2 •  SELECT COUNT(ORDER_ID) AS TOTAL_ORDERS FROM orders;
```

Result Grid | Filter Rows:

TOTAL_ORDERS
21350

2. Calculate the total revenue generated from pizza sales..

```sql
-- Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),
            2) AS TOTAL_SALESsS
FROM
    ORDER_DETAILS
        JOIN
    PIZZAS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
```



3. 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

4. Identify the most common pizza size ordered.

```sql
-- Identify the most common pizza size ordered.
SELECT
    PIZZAS.SIZE,
    COUNT(ORDER_DETAILS.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 Rows:

SIZE	ORDER_COUNT
L	18526
M	15385
S	14137
XL	544
XXL	28

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

```sql
-- 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;
```

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

INTERMEDIATE ANALYSIS

1. 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(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

2. 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 | Filter Rows:

HOUR	ORDER_COUNT
11	1231
12	2520
13	2455
14	1472
15	1468

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

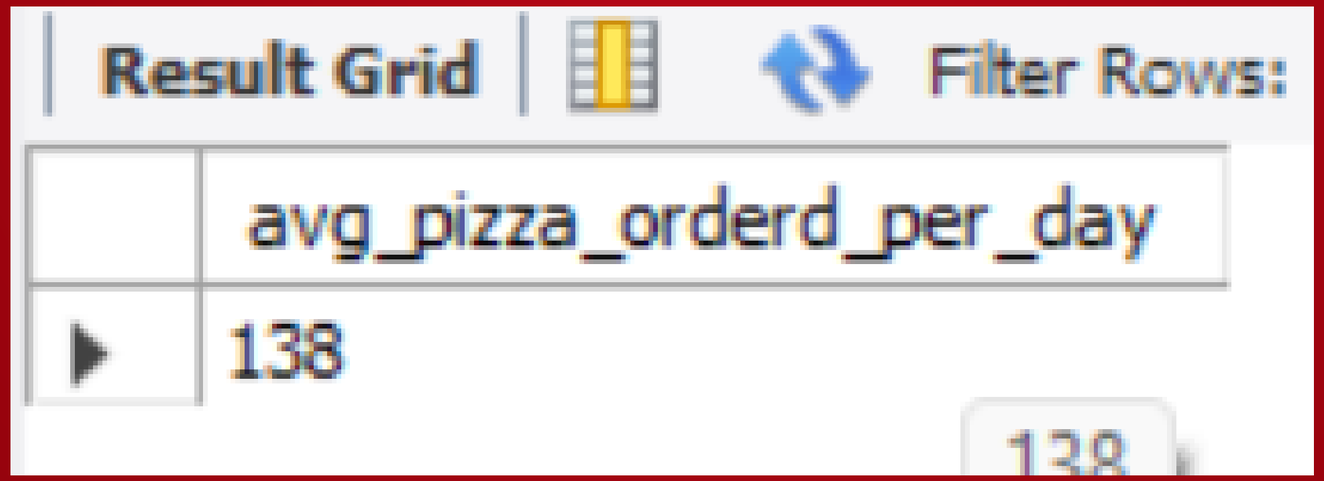
```sql
-- Join relevant tables to find the category-wise distribution of pizzas.
SELECT
    CATEGORY, COUNT(NAME)
FROM
    PIZZA_TYPES
GROUP BY CATEGORY;
```

Result Grid | Filter Rows:

CATEGORY	COUNT(NAME)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.
SELECT
    ROUND(AVG(QUANTITY), 0) as avg_pizza_orderd_per_day
FROM
    (SELECT
        ORDERS.ORDER_DATE, SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
    FROM
        ORDERS
    JOIN ORDER_DETAILS ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
    GROUP BY ORDERS.ORDER_DATE) AS ORDER_QUANTITY;
```



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

```sql
SELECT
    PIZZA_TYPES.NAME,
    SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
FROM
    PIZZA_TYPES
        JOIN
    PIZZAS ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.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;
```

Result Grid | Filter Rows:

NAME	REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

ADVANCED ANALYSIS

1. **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(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) / (SELECT
                    ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),
                                2) AS TOTAL_SALES
                FROM
                    ORDER_DETAILS
                        JOIN
                    PIZZAS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID)) * 100,
            2) AS REVENUE
FROM
    PIZZA_TYPES
        JOIN
    PIZZAS ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.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 | Filter Rows:

CATEGORY	REVENUE
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

2. Analyze the cumulative revenue generated over time.

```sql
-- 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 PIZZAS
ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
JOIN ORDERS
ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
GROUP BY ORDERS.ORDER_DATE) AS SALES;
```

Result Grid | Filter Rows: | Export:

ORDER_DATE	CUM_REVENUE
2015-01-01	2713.850000000004
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

9863.6

3. 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((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;
```

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!

-Riya Saraf