SQL PROJECT ON PIZZA SALES

Start Your Slide

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
SELECT
    COUNT(Order_id) AS Total_orders
FROM
    Orders;
```

Result Grid | Filter

	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 pizzas.pizza_id = order_details.pizza_id
```

Result Grid

	Total_sales
▶	265702.7

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 pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Ro

	name	price
▶	The Greek Pizza	35.95

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 Barbecue Chicken Pizza	14478.25
The Thai Chicken Pizza	13953.75
The California Chicken Pizza	13148.75

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(time) AS HOUR, COUNT(ORDER_ID) AS ORDER_COUNT
FROM
    ORDERS
GROUP BY HOUR(time);
```

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    quantity, COUNT(order_details_id)
FROM
    orders_details
GROUP BY quantity;
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	6039
M	4973
S	4591
XL	185
XXL	12

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

```sql
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;
```

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

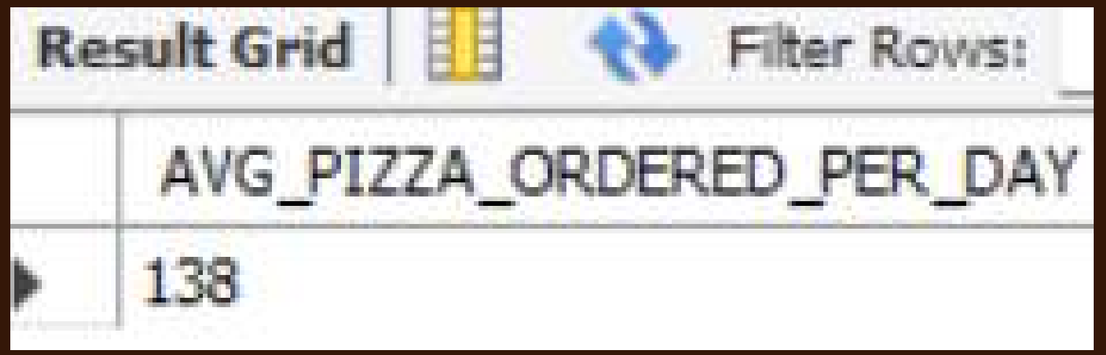
```sql
SELECT DATE,
SUM(REVENUE) OVER(ORDER BY DATE) AS CUM_REVENUE
FROM
(SELECT ORDERS.DATE,
SUM(order_details.QUANTITY * PIZZAS.PRICE) AS REVENUE
FROM order_details JOIN PIZZAS
ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
JOIN ORDERS
ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
GROUP BY ORDERS.DATE) AS SALES;
```

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

```sql
SELECT
    PIZZA_TYPES.CATEGORY,
    (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 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;
```

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(QUANTITY), 0) AS AVG_PIZZA_ORDERED_PER_DAY
FROM
    (SELECT
        ORDERS.DATE, SUM(ORDER_DETAILS.quantity) AS QUANTITY
    FROM
        ORDERS
    JOIN ORDER_DETAILS ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
    GROUP BY ORDERS.DATE) AS ORDER_QUANTITY;
```



THANK YOU FOR ATTENTION

SEE YOU LATER