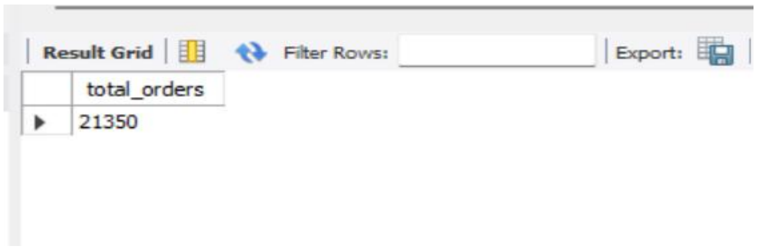
PIZZA SALES ANALYSIS SQL QUERY WITH OUTPUT

BASIC:---

Q. 1.Retrieve the total number of orders placed.

Query:
```
select count(order_id)  as total_orders from pizzahut.orders;
```

Result:

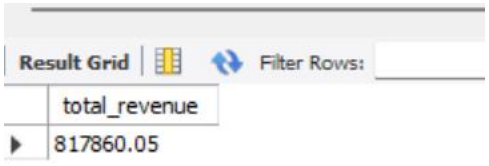


```
total_orders
21350
```

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

Query:
```
SELECT
   ROUND(SUM(price * quantity), 2) AS total_revenue
FROM
   pizzahut.order_details
      JOIN
   pizzahut.pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

RESULT:

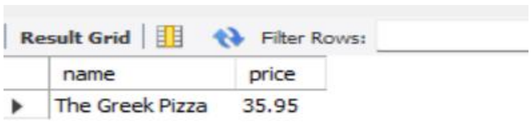


```
total_revenue
817860.05
```

Q.3.Identify the highest-priced pizza.

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

RESULT:

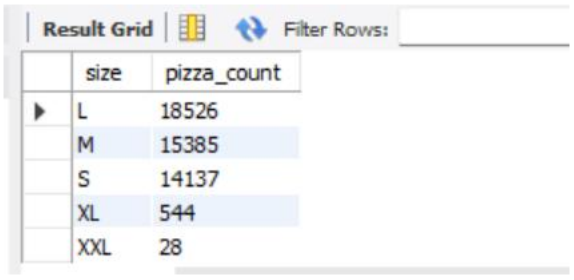

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Q.4.Identify the most common pizza size ordered.

QUERY:
```
SELECT
    size, COUNT(order_details_id) AS pizza_count
FROM
    pizzahut.pizzas
        JOIN
    pizzahut.order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
ORDER BY pizza_count DESC;
```

RESULT:

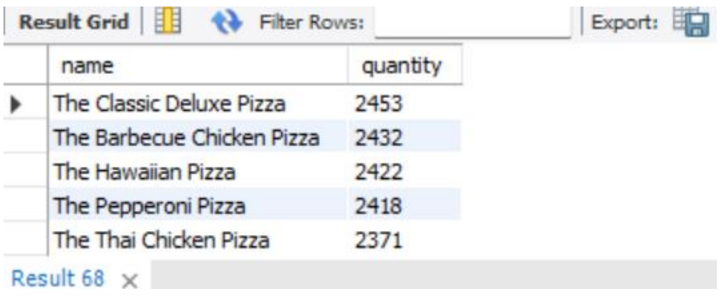

Result Grid | Filter Rows:

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

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

QUERY:
```
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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

RESULT:


Result Grid | Filter Rows: | Export:

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

Result 68

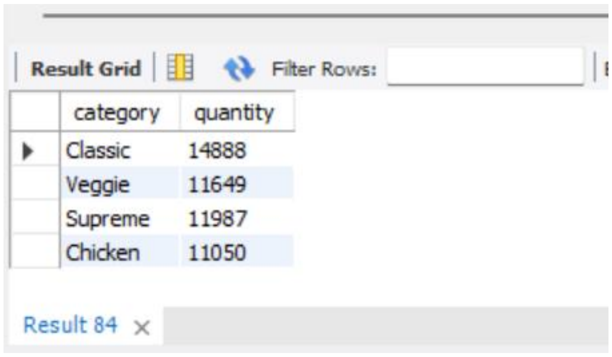
MODERATE:

Q.6.Join the necessary tables to find the total quantity of each pizza category ordered.

QUERY:

```
SELECT
   pizza_types.category,
   SUM(order_details.quantity) AS quantity
FROM
   pizzahut.pizza_types
      JOIN
   pizzahut.pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
      JOIN
   pizzahut.order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category;
```

RESULT:



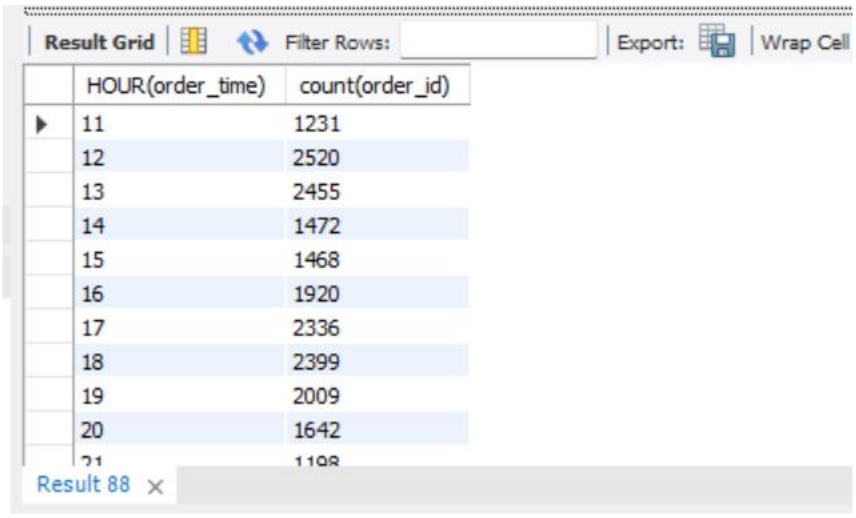
category	quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Q.7.Determine the distribution of orders by hour of the day.

QUERY:

```
SELECT
   HOUR(order_time), COUNT(order_id)
FROM
   orders
GROUP BY HOUR(order_time);
```

RESULT:



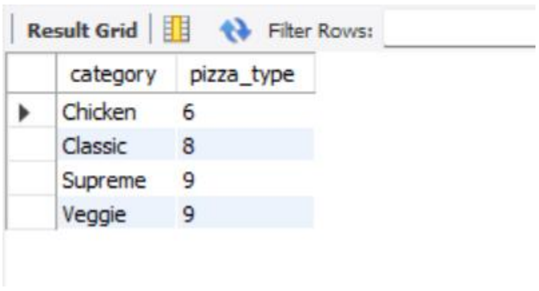
HOUR(order_time)	count(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198

Q.8.Join relevant tables to find the category-wise distribution of pizzas.

QUERY:

```
SELECT
   category, COUNT(pizza_type_id) AS pizza_type
FROM
   pizza_types
GROUP BY category;
```

RESULT:



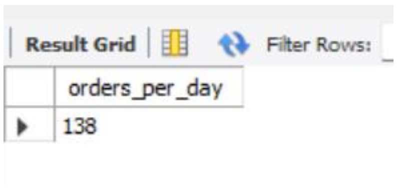
category	pizza_type
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q.9.Group the orders by date and calculate the average number of pizzas ordered per day.

QUERY:

```
SELECT
   ROUND(AVG(QUANTITY), 0) AS orders_per_day
FROM
   (SELECT
      order_date, SUM(quantity) AS QUANTITY
   FROM
      pizzahut.orders
   JOIN pizzahut.order_details ON orders.order_id = order_details.order_id
   GROUP BY order_date) AS ORDER_QUANTITY;
```

RESULT:



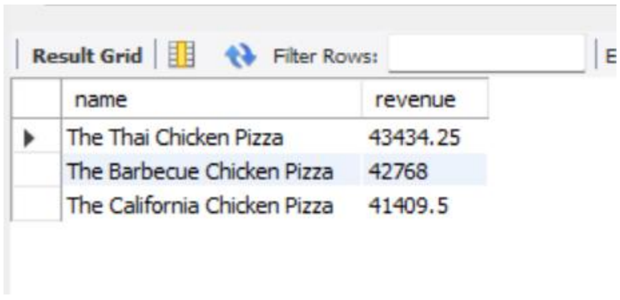
```
Result Grid | Filter Rows:
  orders_per_day
▶ 138
```

Q.10.Determine the top 3 most ordered pizza types based on revenue.

QUERY:

```
SELECT
   pizza_types.name,
   SUM(pizzas.price * order_details.quantity) AS revenue
FROM
   pizza_types
      JOIN
   pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
      JOIN
   order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

RESULT:



```
Result Grid | Filter Rows:
  name                          revenue
▶ The Thai Chicken Pizza        43434.25
  The Barbecue Chicken Pizza    42768
  The California Chicken Pizza  41409.5
```

ADVANCED:--

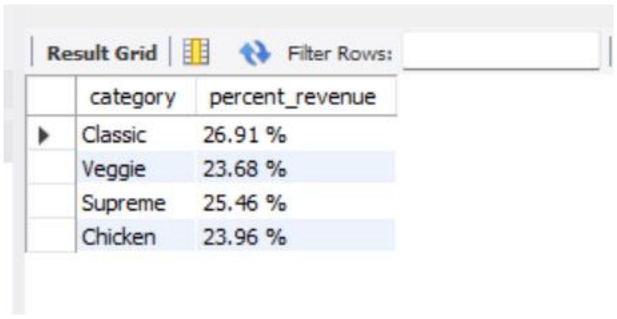
Q.11.Calculate the percentage contribution of each pizza type to total revenue.

QUERY:

```
SELECT
   pizza_types.category,
   CONCAT(ROUND(SUM((order_details.quantity * pizzas.price) / (SELECT
                  ROUND(SUM(order_details.quantity * pizzas.price),
                        2)
               FROM
                  order_details
                     JOIN
                  pizzas ON order_details.pizza_id = pizzas.pizza_id)) * 100,
            2),
         ' %') AS percent_revenue
FROM
   order_details
      JOIN
   pizzas ON order_details.pizza_id = pizzas.pizza_id
      JOIN
   pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

```
GROUP BY pizza_types.category;
```

RESULT:



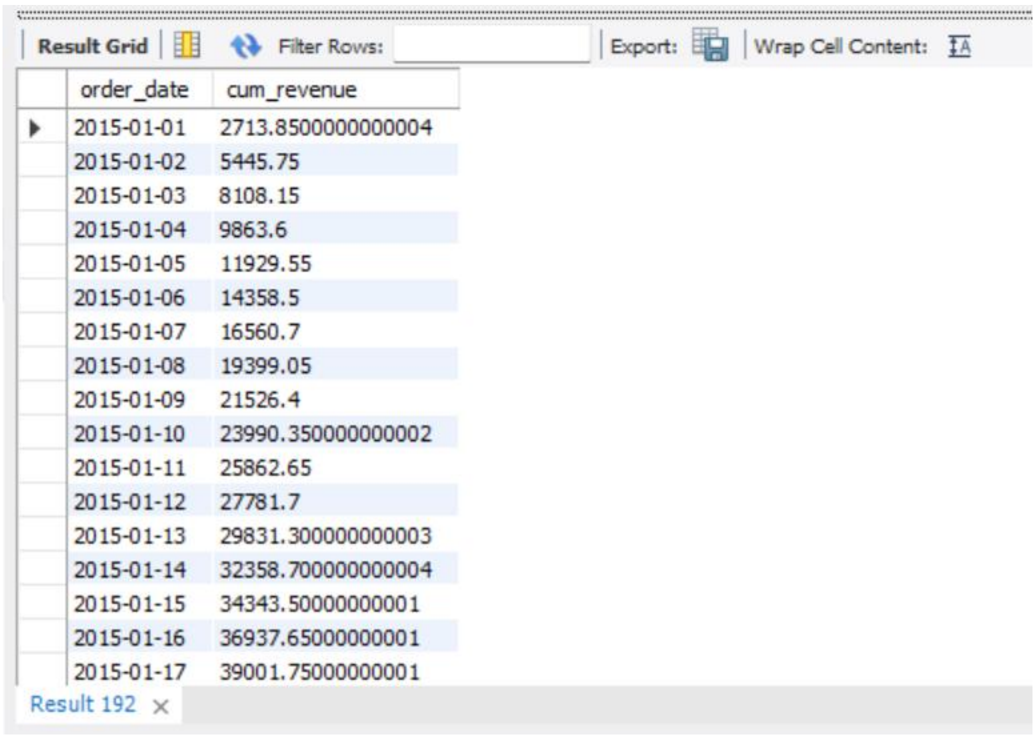
category	percent_revenue
Classic	26.91 %
Veggie	23.68 %
Supreme	25.46 %
Chicken	23.96 %

Q.12.Analyze the cumulative revenue generated over time.

QUERY:

```
SELECT
   order_date,
   SUM(revenue) OVER(ORDER BY order_date) AS cum_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:



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
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001