SALES PIZZA

PIZZA SQL & DATA ANALISIS PROJECT

LARANA PIZZA

WELCOME TO PIZZA SQL PROJECT

In this Project we have used easy question , Intermidiate question as well as difficult question with the help of SQL command in this project it's used for analysis the dataset or data analysis project so that we can find the ralationship between the table and join the two or three tables in database

-- CALCULATE THE TOTAL REVENUE GENRATED FROM PIZZA SALES.

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) as total_sales
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id;
```

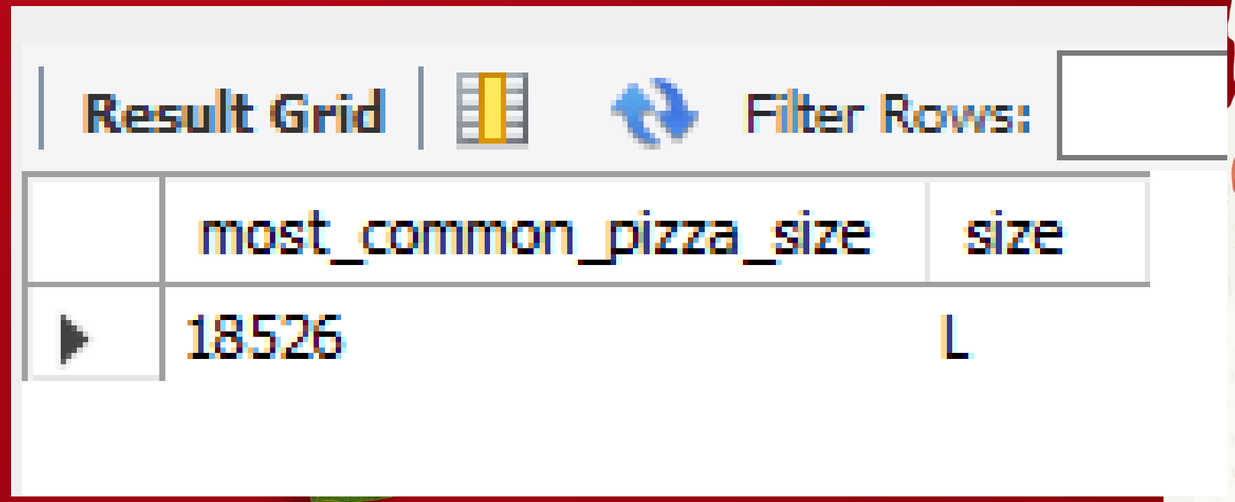
EASY QUESTIONS

Result Grid

	total_sales
▶	817860.05

IDENTIFY THE MOST COMMON PIZZA SIZED ORDER.

```sql
SELECT
    COUNT(order_details.quantity) AS most_common_pizza_size,
    pizzas.size
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY pizzas.size ASC
LIMIT 1;
```



most_common_pizza_size	size
18526	L

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

```sql
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 Ro

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

MEDIUM QUESTIONS

DETERMINE THE DISTRIBUTION OF THE ORDERS BY HOURS OF THE DAY.

```sql
SELECT
    HOUR(order_time) AS hours, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid | Filt

hours	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663

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

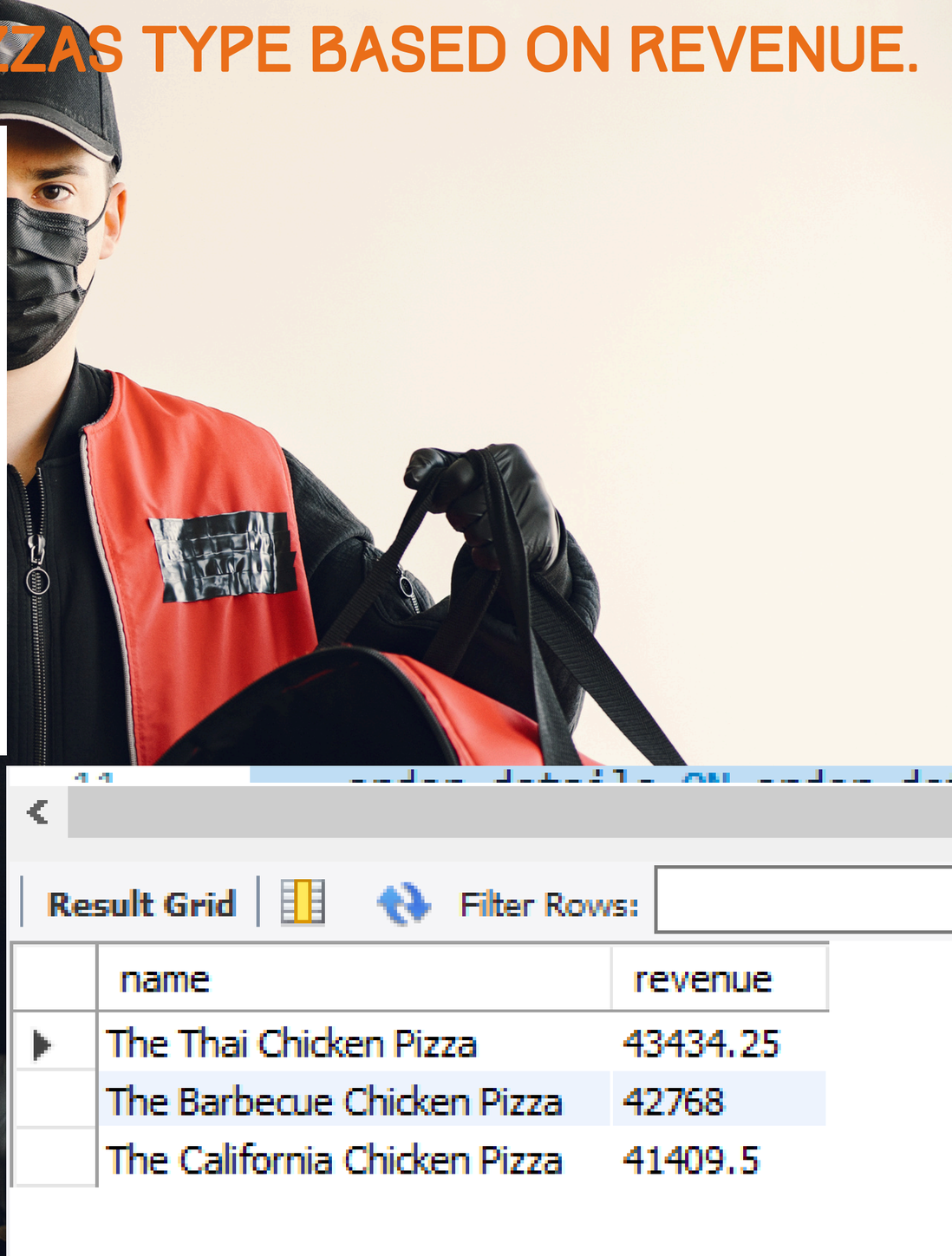
```sql
SELECT
    ROUND(AVG(average_count), 2)
FROM
    (SELECT
        order_date, SUM(quantity) AS average_count
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY order_date) AS order_quant;
```

Result Grid | Filter Rows:

	ROUND(AVG(average_count), 2)
▶	138.47

DETERMINE THE TOP 3 MOST ORDERED PIZZAS TYPE BASED ON REVENUE.

```sql
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 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

HARD QUESTIONS

ANALYZE THE CUMULATIVE REVENUE GENRATED OVER TIME.

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

order_date	cumulator_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 ORDER PIZZA TYPE 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 pizza_rev) 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

LARANA PIZZA

THANK YOU!