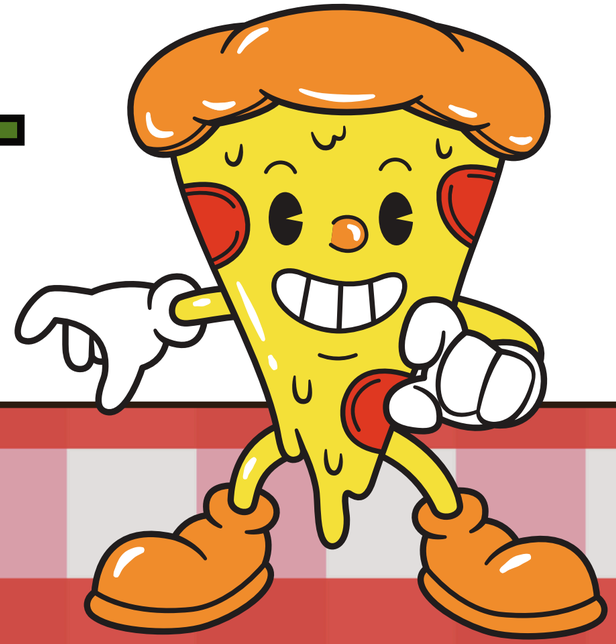
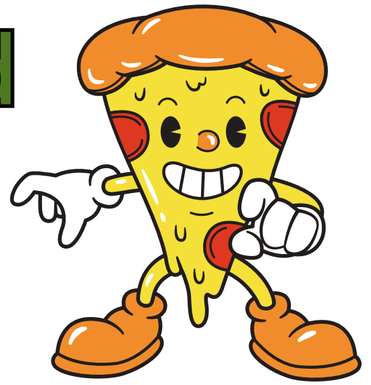
PIZZA- SALES DATA USING SQL



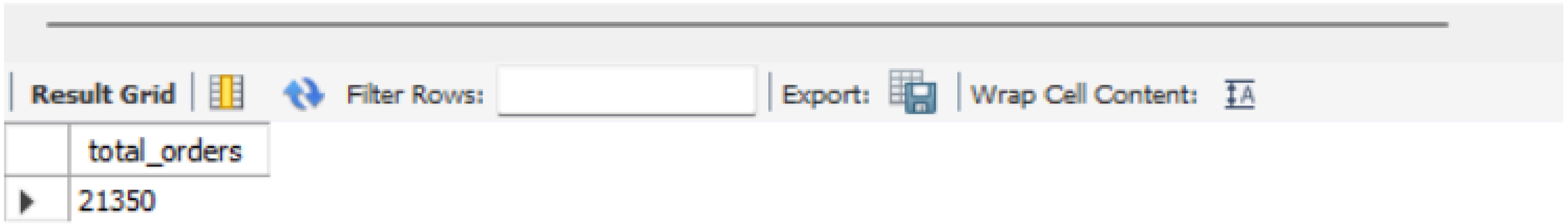
Hello !

My name is shaileja. I have analyzed the pizza sales data using sql queries and solved some amazing questions



Retrieve the total number of orders placed

```
1      # Retrieve the total number of orders placed.
2
3 •    select count(order_id) as total_orders from orders;
4
```



	total_orders
▶	21350

Calculate the total revenue generated from pizza sales

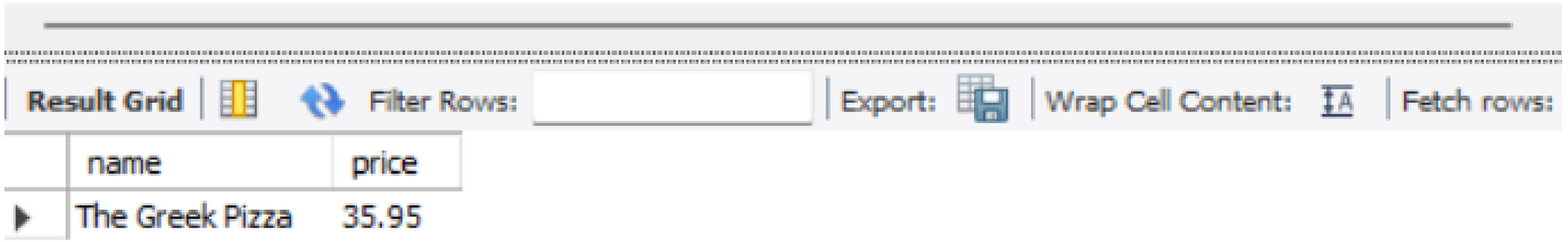
```
1     # Calculate the total revenue generated from pizza sales.
2 •   select
3     round(sum(order_details.quantity * pizzas.price),2) as sales_revenue
4     from order_details join pizzas
5     on pizzas.pizza_id = order_details.pizza_id
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	sales_revenue
▶	817860.05

Identify the highest-priced pizza

```
1     # Identify the highest-priced pizza.
2 •   select pizza_types.name, pizzas.price
3     from pizza_types join pizzas
4     on pizza_types.pizza_type_id = pizzas.pizza_type_id
5     order by pizzas.price desc limit 1
```



name	price
The Greek Pizza	35.95

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: | Export: | Wrap Cell Content:

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

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: | Export: | Wrap Cell Content: | Fetch 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(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
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	quantity
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

Determine the distribution of orders by hour of the day

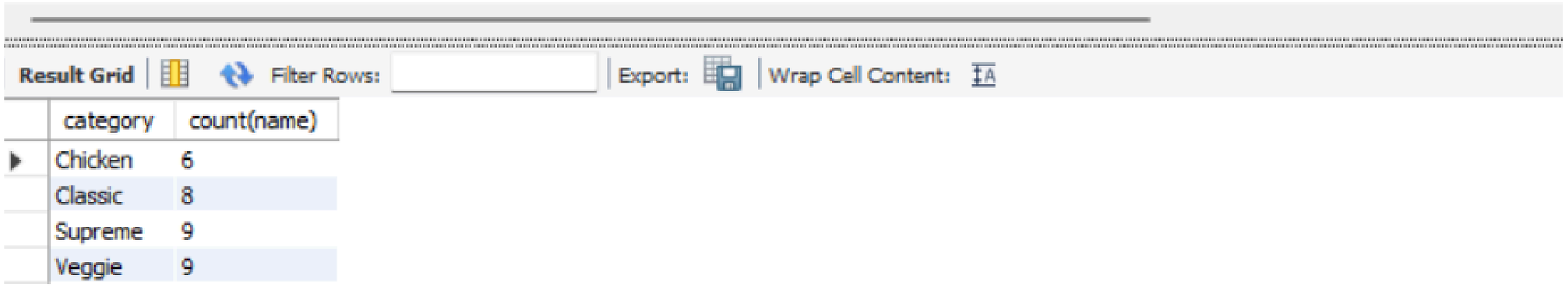
```sql
# Determine the distribution of orders by hour of the day.
select hour(order_time), count(order_id)
from orders
group by hour(order_time)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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
22	663
23	28
10	8
9	1

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

```
1    # Join relevant tables to find the category-wise distribution of pizzas.
2 •  select category, count(name) from pizza_types
3    group by category
```


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

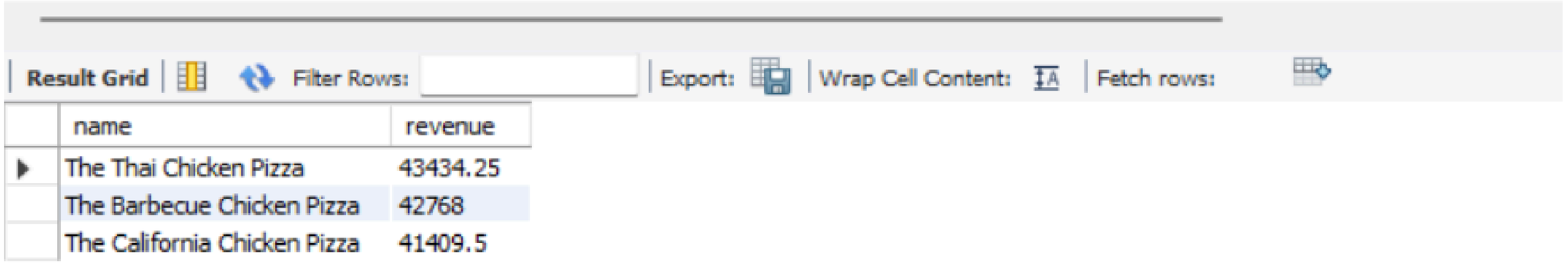
```sql
# Group the orders by date and calculate the average number of pizzas ordered per day.
select avg(quantity) from
(select orders.order_date, sum(order_details.quantity) as quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as order_quantity
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg(quantity)
138.4749

Determine the top 3 most ordered pizza types based on revenue

```sql
# Determine the top 3 most ordered pizza types based on revenue.
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 pizza_types.name order by revenue desc limit 3
```



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(order_details.quantity * pizzas.price)/(select
sum(order_details.quantity * pizzas.price) as sales_revenue
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id) * 100,2)
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
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	revenue
Veggie	23.68
Chicken	23.96
Supreme	25.46
Classic	26.91

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_rev
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 order_details.order_id = orders.order_id
group by orders.order_date
order by revenue) as rev
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_date	cum_rev
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

THANKYOU!