



# SQL PIZZA SALES PROJECT



Fiza Khan

# Introduction



In this project we intertwined  
SQL and Pizza Sales Data to fetch  
the queries to get insights

Fiza Khan

# Queries

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
  - Identify the highest-priced pizza.
  - Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Calculate the contribution of each pizza type to total revenue.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

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

3

4

5

6

7

8

Result Grid



Filter Rows:



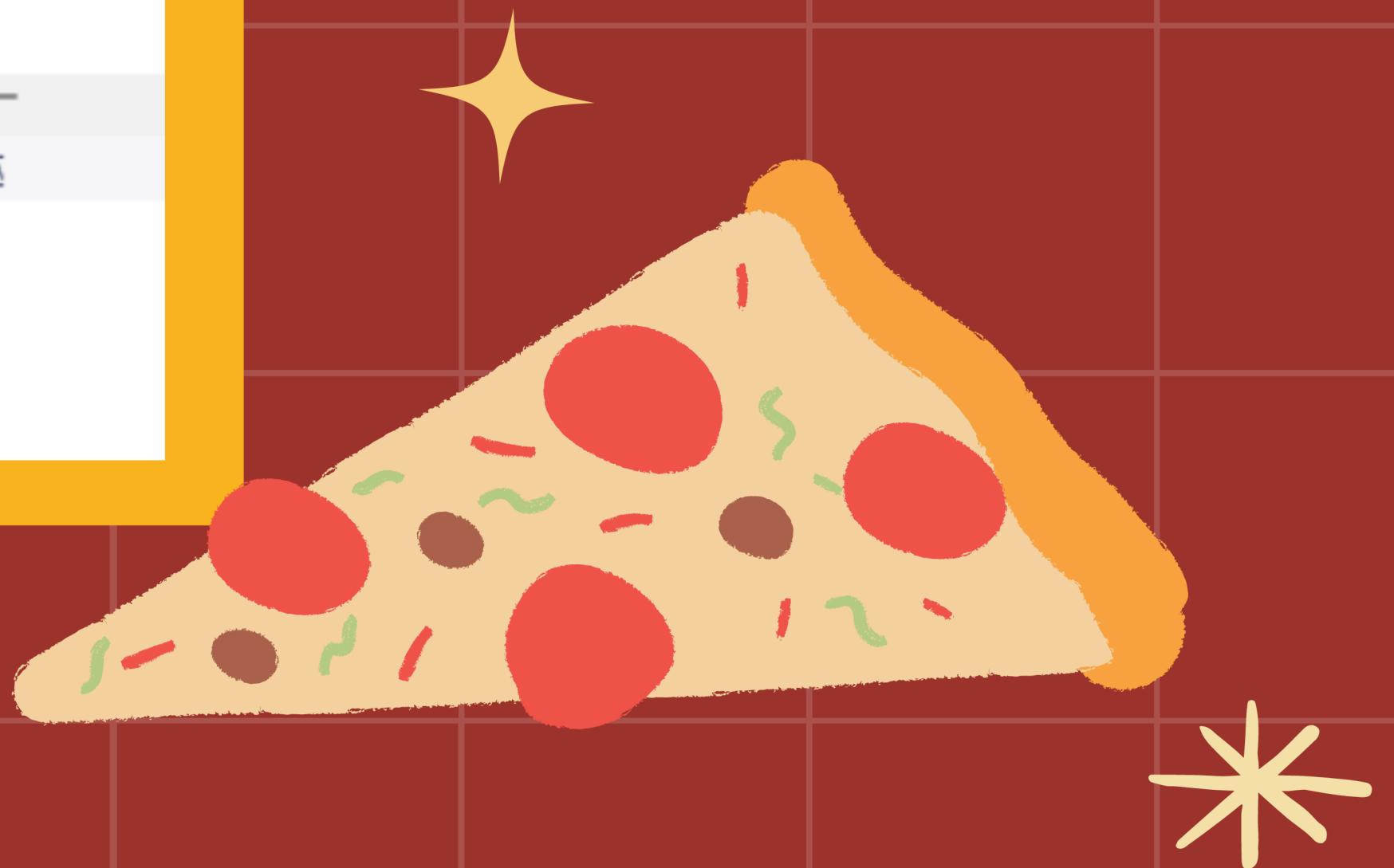
Export:



Wrap Cell Content:



total_orders
436



Fiza Khan



A screenshot of a database management software interface. The top bar includes standard icons like file, save, and search, along with a "Don't Limit" button. The main area contains a SQL query:

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

The result grid shows one row with the column "total\_revenue" containing the value "3146.8".

total_revenue
3146.8

Fiza Khan

```
1  -- Identify the highest-priced pizza.
2 • select pizza_types.name , pizzas.price
3  from pizza_types join pizzas on pizza_types.pizza_type_id=pizzas.pizza_type_id
4  order by pizzas.price desc
5  limit 1;
6
```

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

	name	price
▶	The Greek Pizza	35.95

Fiza Khan

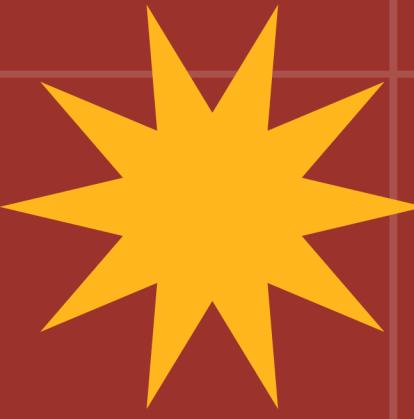


```
1 -- Identify the most common pizza size ordered.  
2 • select pizzas.size , count(order_details.order_details_id) as torder  
3 from pizzas join order_details  
4 on pizzas.pizza_id=order_details.pizza_id  
5 group by pizzas.size  
6 order by pizzas.size;  
7
```

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

	size	torder
▶	L	84
	M	50
	S	49
	XL	1

Fiza Khan





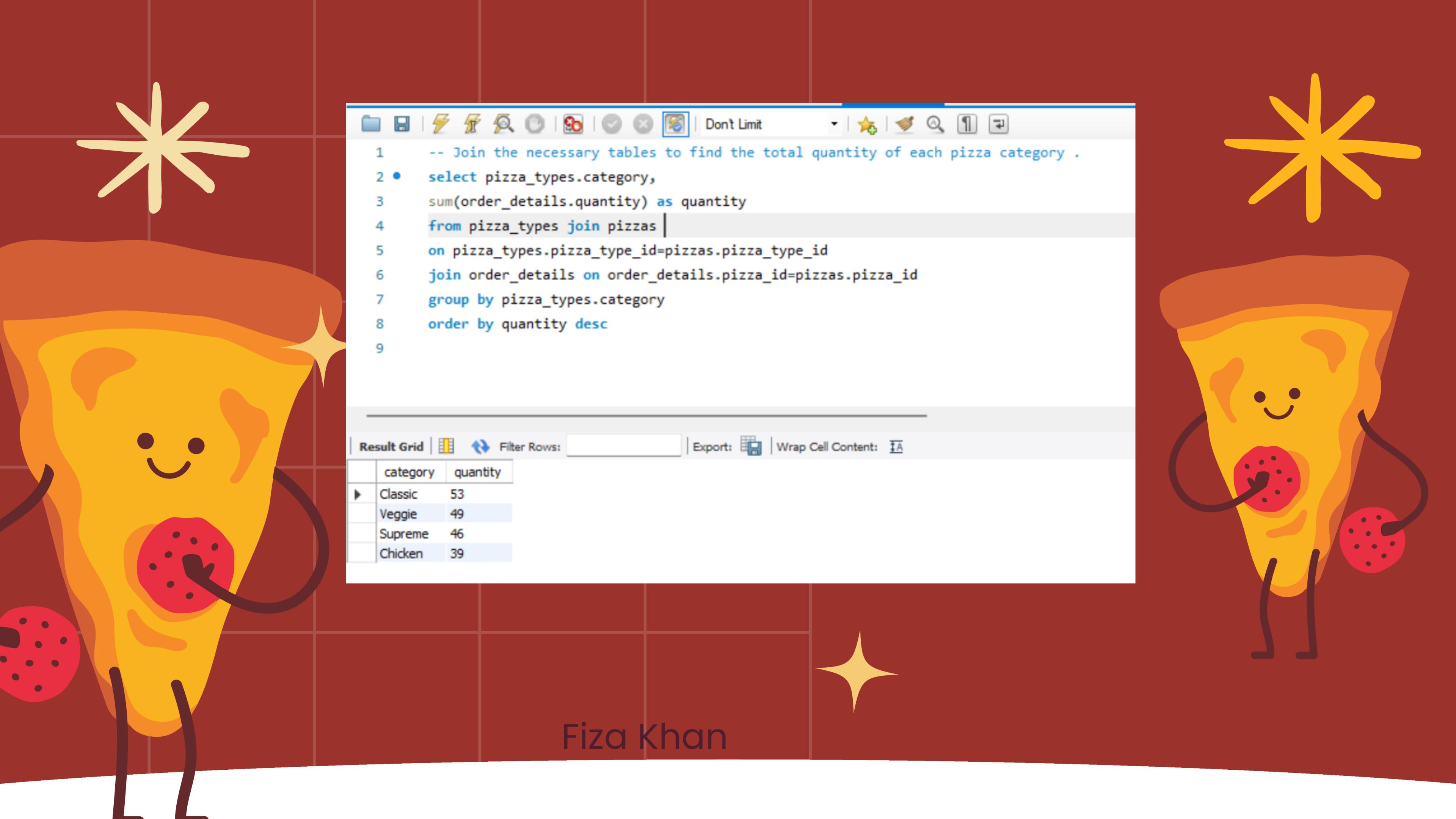
Dont Limit

```
1 -- List the top 5 most ordered pizza types along with their quantities.
2 • select pizza_types.name ,sum(order_details.quantity) as quantity
3 from pizzas join pizza_types on pizzas.pizza_type_id=pizza_types.pizza_type_id
4 join order_details on order_details.pizza_id = pizzas.pizza_id
5 group by pizza_types.name
6 order by quantity desc
7 limit 5 ;
```

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

	name	quantity
▶	The Italian Supreme Pizza	16
	The Barbecue Chicken Pizza	12
	The Classic Deluxe Pizza	11
	The Mexicana Pizza	10
	The Thai Chicken Pizza	10

Fiza Khan



```
1 -- Join the necessary tables to find the total quantity of each pizza category .
2 • select pizza_types.category,
3     sum(order_details.quantity) as quantity
4     from pizza_types join pizzas |
5     on pizza_types.pizza_type_id=pizzas.pizza_type_id
6     join order_details on order_details.pizza_id=pizzas.pizza_id
7     group by pizza_types.category
8     order by quantity desc
9
```

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

	category	quantity
▶	Classic	53
	Veggie	49
	Supreme	46
	Chicken	39

Fiza Khan

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

	category	total
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Fiza Khan



```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.
2 • with cte as (select orders.date , sum(order_details.quantity) as quantity
3   from orders join order_details
4     on orders.order_id = order_details.order_id
5   group by orders.date)
6
7  select round(avg (quantity) ,0)  as avg from cte ;
```

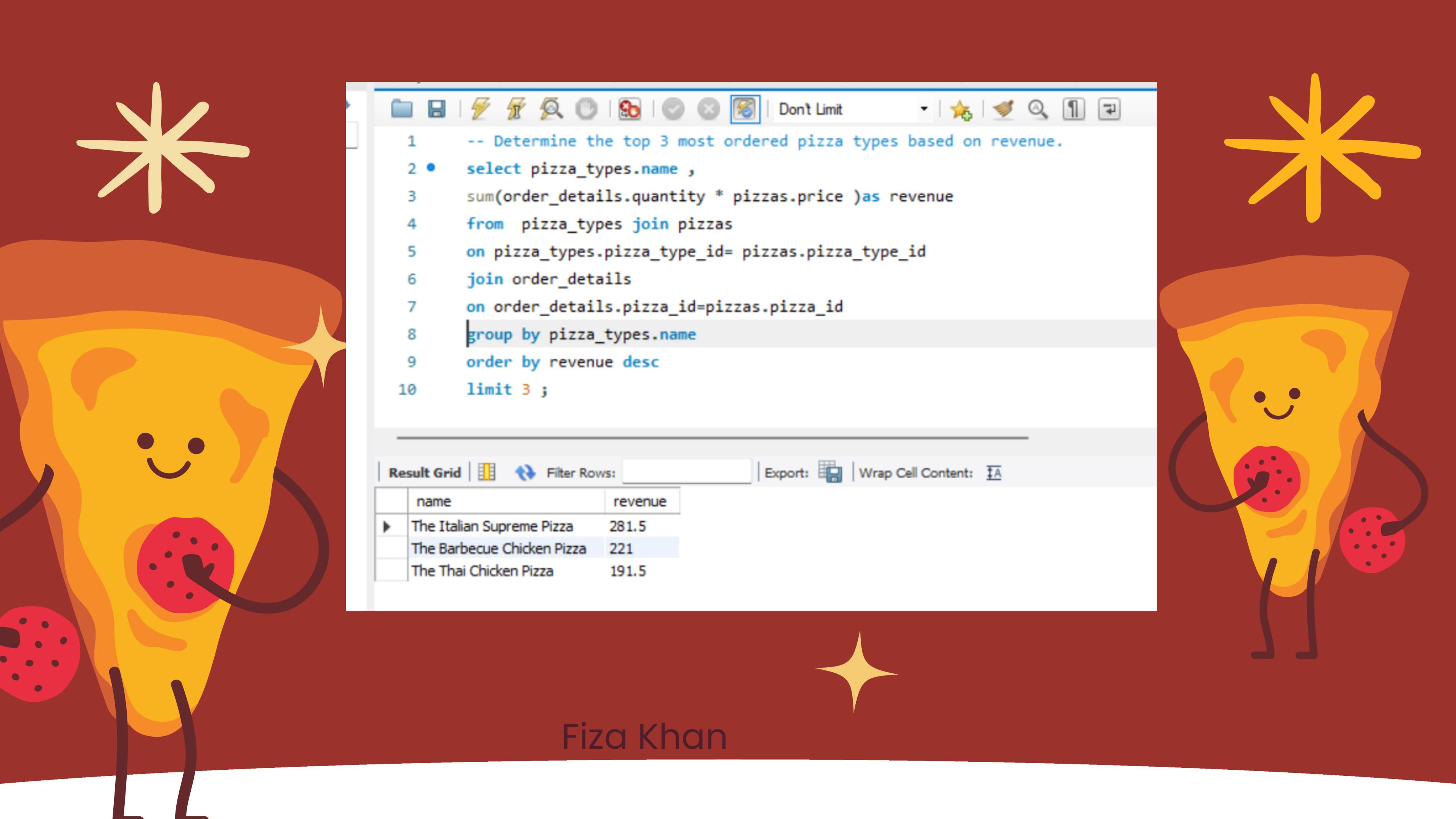
The screenshot shows a database query editor window with a yellow border. The query itself is as follows:

```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.
2 • with cte as (select orders.date , sum(order_details.quantity) as quantity
3   from orders join order_details
4     on orders.order_id = order_details.order_id
5   group by orders.date)
6
7  select round(avg (quantity) ,0)  as avg from cte ;
```

The result grid below the query shows a single row with the column 'avg' containing the value '94'. The interface includes standard SQL navigation buttons (Result Grid, Filter Rows, Export, Wrap Cell Content) at the bottom.

avg
94

Fiza Khan



```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2 • select pizza_types.name ,
3   sum(order_details.quantity * pizzas.price )as revenue
4   from pizza_types join pizzas
5   on pizza_types.pizza_type_id= pizzas.pizza_type_id
6   join order_details
7   on order_details.pizza_id=pizzas.pizza_id
8   group by pizza_types.name
9   order by revenue desc
10  limit 3 ;
```

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

	name	revenue
▶	The Italian Supreme Pizza	281.5
	The Barbecue Chicken Pizza	221
	The Thai Chicken Pizza	191.5

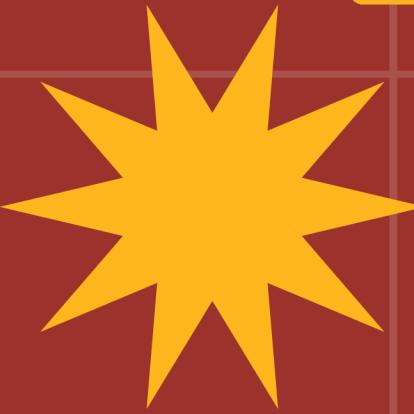
Fiza Khan

```
1 -- Calculate the contribution of each pizza type to total revenue.
2 • select pizza_types.category ,
3     sum(order_details.quantity * pizzas.price )as revenue
4   from pizza_types join pizzas
5     on pizza_types.pizza_type_id= pizzas.pizza_type_id
6   join order_details
7     on order_details.pizza_id=pizzas.pizza_id
8   group by pizza_types.category
9   order by revenue desc ;
```

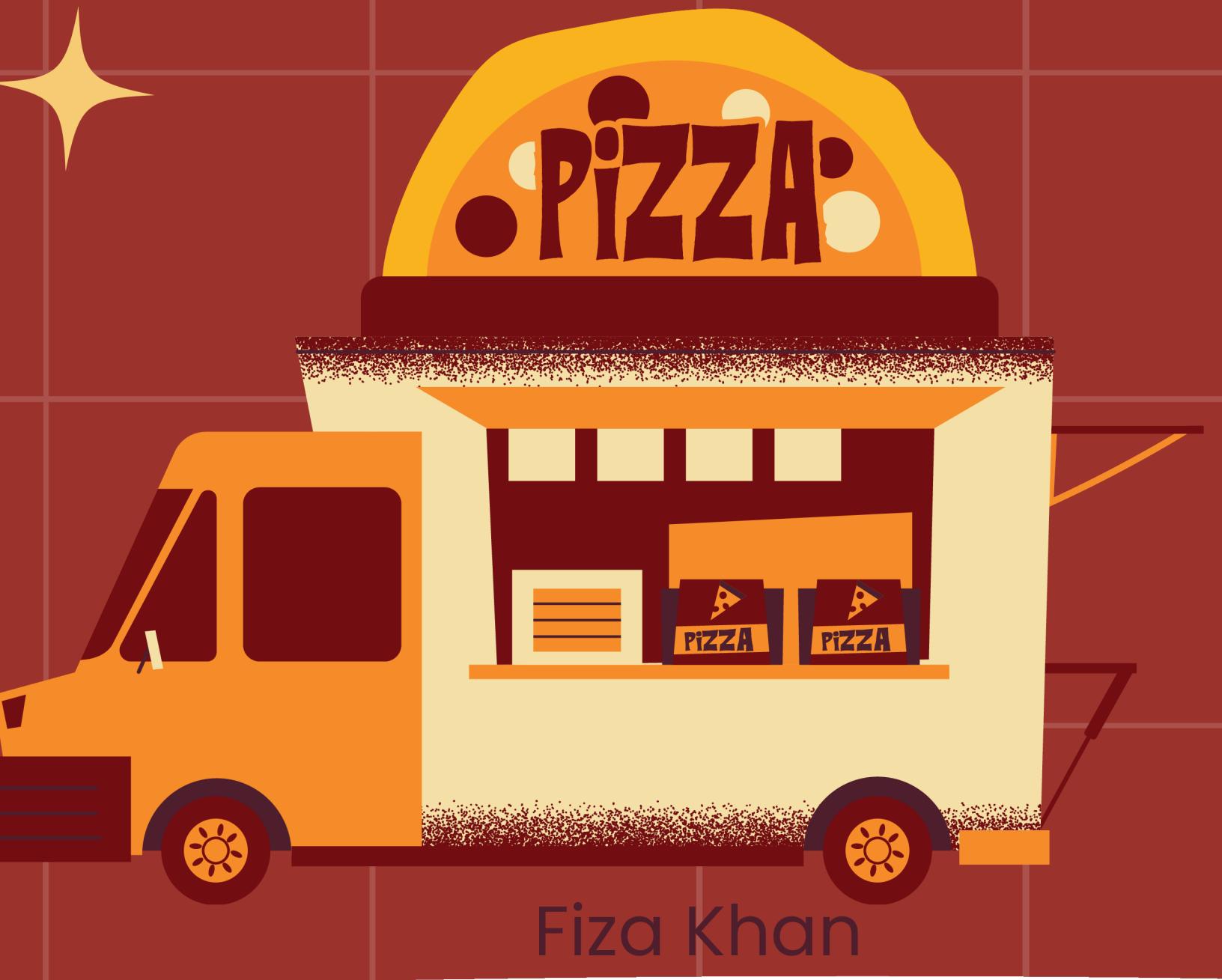
---

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

	category	revenue
▶	Veggie	828.5500000000001
	Supreme	814.75
	Classic	786.25
	Chicken	717.25



# THANK YOU



Fiza Khan

