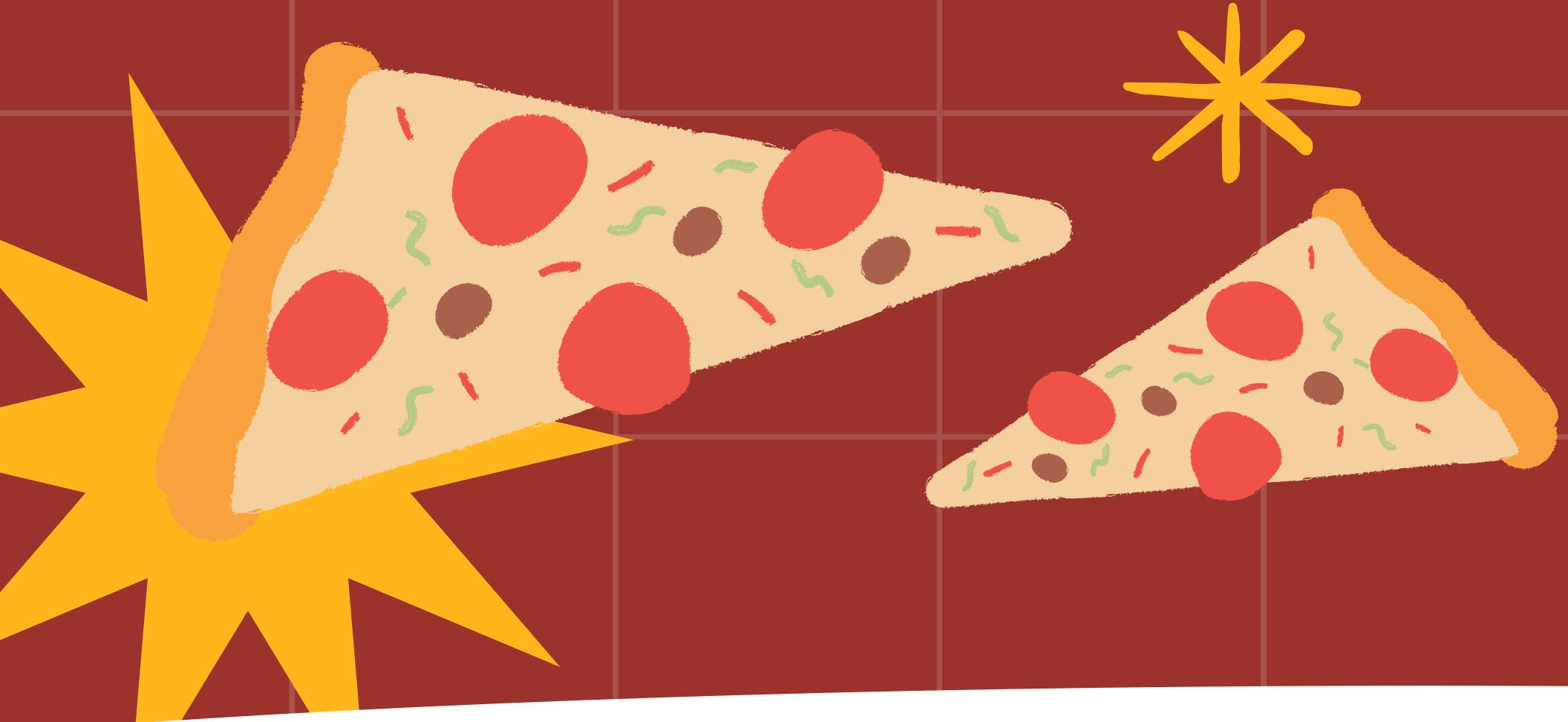
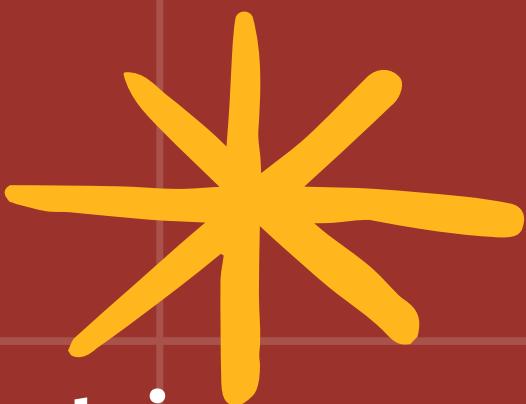


# SQL PROJECT ON PIZZA SALES

By Dhiraj Sakate



# OBJECTIVE -



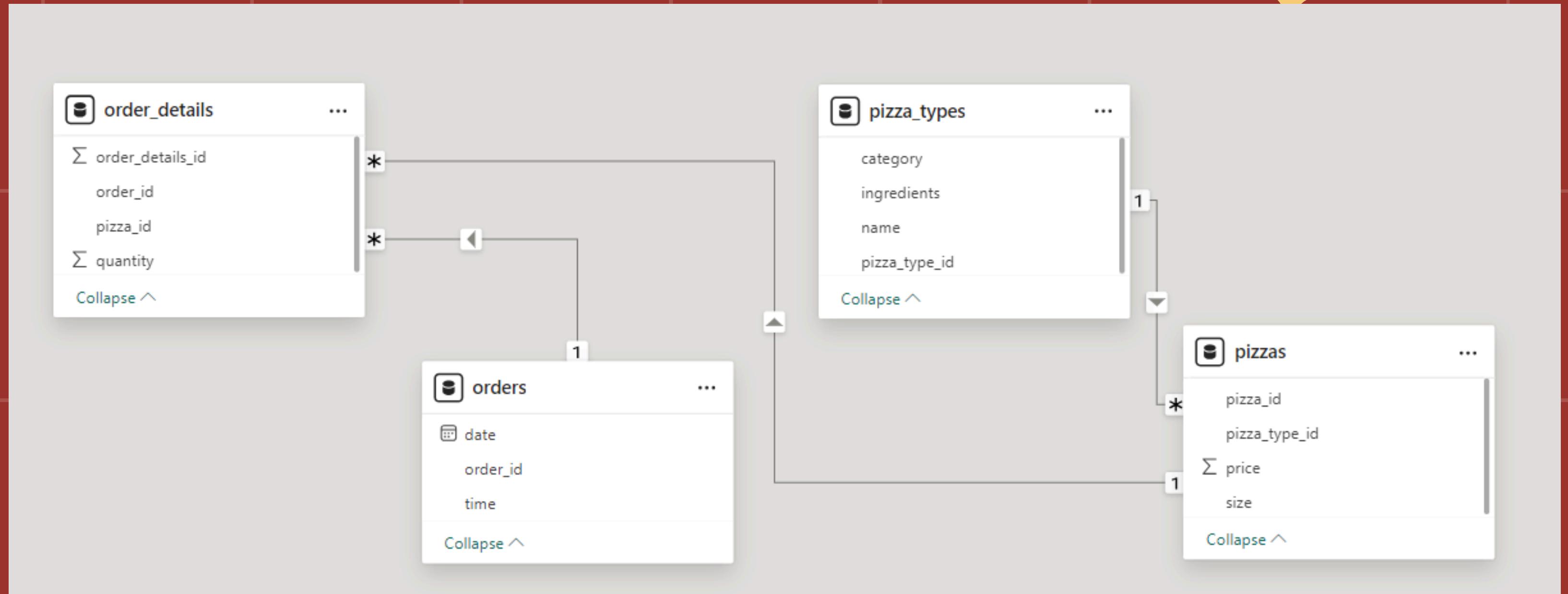
The objective of this SQL pizza sales project is to analyze key sales metrics, including total orders, revenue, and pizza preferences, to gain insights into customer behavior, product performance, and revenue trends.

# SCOPE -

This project involves retrieving and analyzing pizza sales data to identify top-selling pizza types, most common sizes, price-based rankings, time-based order patterns, and category-wise distribution, along with calculating cumulative revenue and the percentage contribution of each pizza type to total revenue.



# DATABASE SCHEMA -



# CREATING DATABASE -

```
1 •   CREATE DATABASE pizzahut;  
2  
3 • ○ create table orders (  
4     order_id int not null,  
5     order_date date not null,  
6     order_time time not null,  
7     primary key(order_id) );  
8  
9  
10 • ○ create table order_details (  
11     order_details_id int not null,  
12     order_id int not null,  
13     pizza_id text not null,  
14     quantity int not null,  
15     primary key(order_details_id) );
```



# Retrieve the total number of orders placed.

```
3 • select count(order_id) as total_orders from orders;
```

	total_orders
▶	21350

# Calculate the total revenue generated from pizza sales.

```
3 • select  
4     round(sum(order_details.quantity * pizzas.price),2) as total_sales  
5     from order_details join pizzas  
6     on pizzas.pizza_id= order_details.pizza_id;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:		
<table><thead><tr><th>total_sales</th></tr></thead><tbody><tr><td>817860.05</td></tr></tbody></table>				total_sales	817860.05	
total_sales						
817860.05						

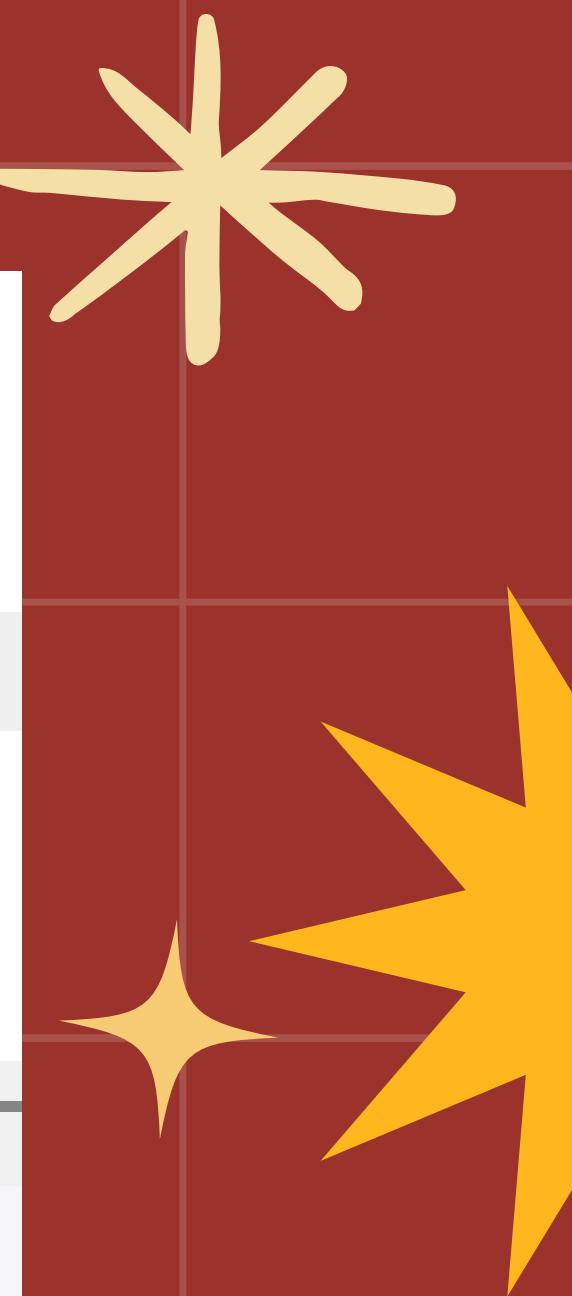
# Identify the highest-priced pizza.

```
3 *   select pizza_types.name, pizzas.price  
4     from pizza_types join pizzas  
5       on pizza_types.pizza_type_id = pizzas.pizza_type_id  
6   order by pizzas.price desc limit 1;
```



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

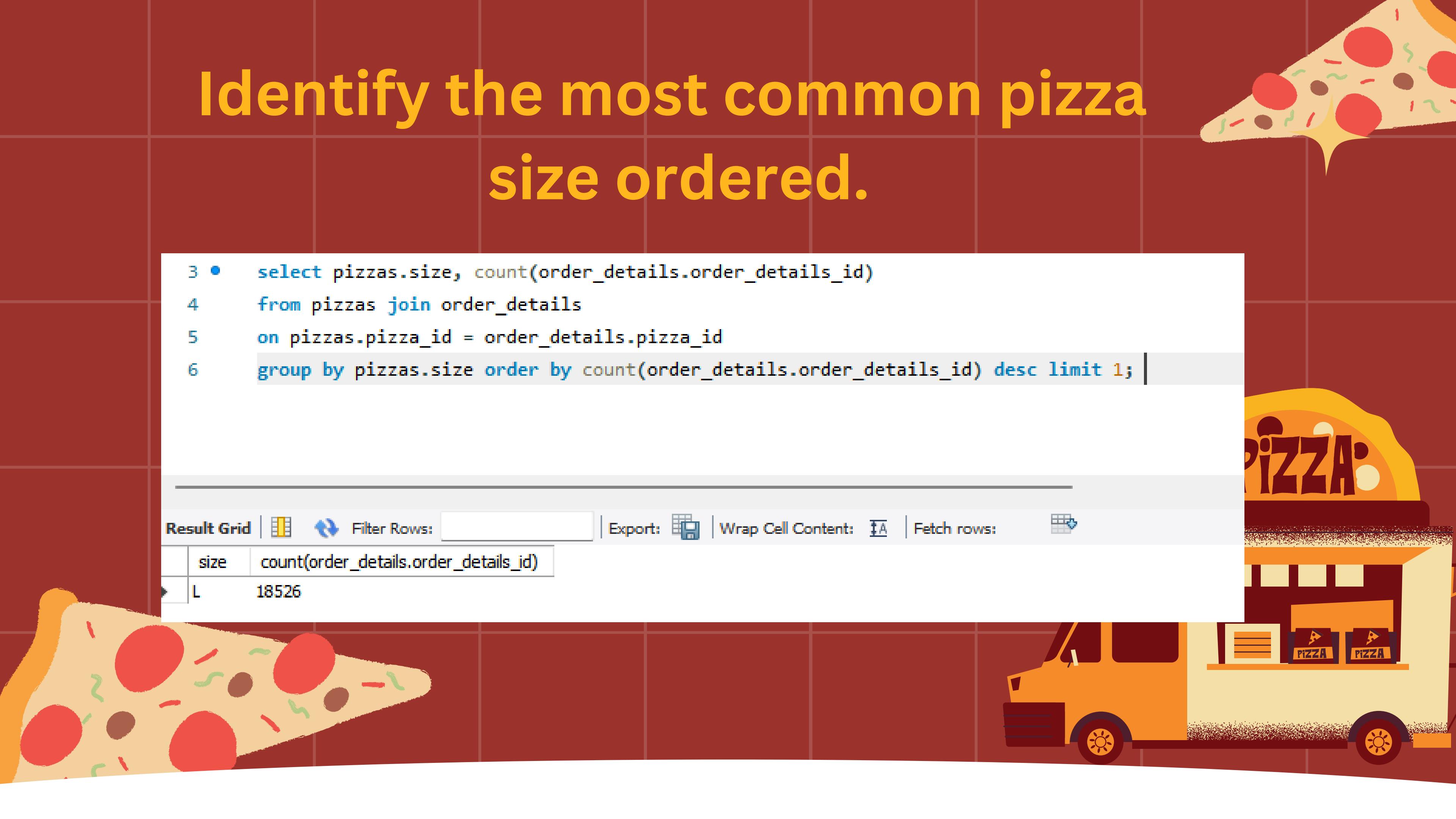
	name	price
▶	The Greek Pizza	35.95



# Identify the most common pizza size ordered.

```
3 •   select pizzas.size, count(order_details.order_details_id)
4     from pizzas join order_details
5       on pizzas.pizza_id = order_details.pizza_id
6   group by pizzas.size order by count(order_details.order_details_id) desc limit 1;
```

size	count(order_details.order_details_id)
L	18526



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

```
3 • select pizza_types.name,  
4     sum(order_details.quantity) as quantity  
5     from pizza_types join pizzas  
6     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7     join order_details  
8     on order_details.pizza_id = pizzas.pizza_id  
9     group by pizza_types.name order by quantity desc limit 5;
```

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

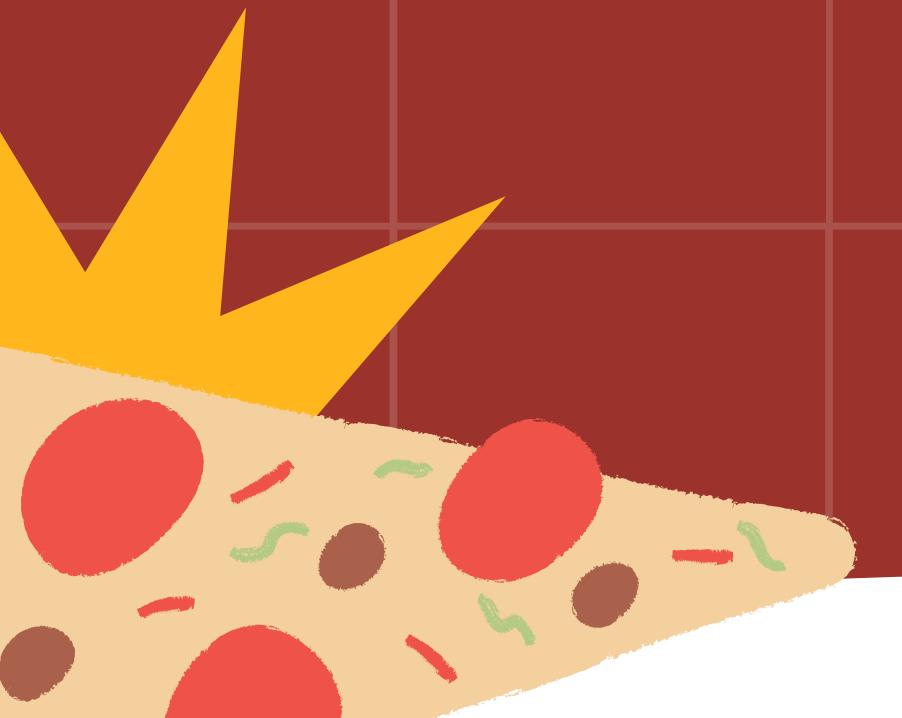
	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

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

```
3 •   select pizza_types.category,  
4       sum(order_details.quantity) as quantity  
5       from pizza_types join pizzas  
6           on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7       join order_details  
8           on order_details.pizza_id = pizzas.pizza_id  
9       group by pizza_types.category  order by quantity desc;  
10
```

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

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



# Determine the distribution of orders by hour of the day.

```
3 • select hour(order_time) as hour, count(order_id) as order_count from orders  
4   group by hour(order_time);
```

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

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	...	-----

Result 1

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

```
3 • select category, count(name) from pizza_types  
4 group by category;
```

Result Grid | Filter Rows:  Export: Wrap C

	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.

```
3 •   SELECT
4       round(AVG(quantity), 0) as avg_pizza_ordered_per_day
5   FROM
6   (SELECT
7       orders.order_date, SUM(order_details.quantity) AS quantity
8   FROM
9       orders
10  JOIN order_details ON orders.order_id = order_details.order_id
11  GROUP BY orders.order_date) AS order_quantity;
```

	avg_pizza_ordered_per_day
▶	138

# \* Determine the top 3 most ordered pizza types based on revenue.

```
3 •   select pizza_types.name,  
4     round (sum(order_details.quantity * pizzas.price)) as revenue  
5   from pizza_types join pizzas  
6     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7   join order_details  
8     on order_details.pizza_id = pizzas.pizza_id  
9   group by pizza_types.name order by revenue desc limit 3 ;  
10
```

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

	name	revenue
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

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

```
3 •   select pizza_types.category,  
4     round(sum(order_details.quantity * pizzas.price) /  
5     (select round(sum(order_details.quantity * pizzas.price),2) as total_sales  
6     from order_details join pizzas  
7     on pizzas.pizza_id= order_details.pizza_id) *100,2) as revenue  
8     from pizza_types join pizzas  
9     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
10    join order_details  
11    on order_details.pizza_id = pizzas.pizza_id  
12    group by pizza_types.category order by revenue desc;
```

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

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# Analyze the cumulative revenue generated over time.

```
3 •   select order_date,  
4       sum(revenue) over(order by order_date) as cum_revenue  
5   from  
6   (select orders.order_date,  
7        sum(order_details.quantity * pizzas.price) as revenue  
8        from order_details join pizzas  
9          on order_details.pizza_id = pizzas.pizza_id  
10       join orders  
11         on orders.order_id = order_details.order_id  
12     group by orders.order_date) as sales;
```

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

	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

Result 1 ×

# Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
3 •   select name,revenue
4   from
5   (select category,name,revenue,
6   rank() over(partition by category order by revenue desc) as ranks
7   from
8   (select pizza_types.category , pizza_types.name,
9   sum(order_details.quantity * pizzas.price) as revenue
10  from pizza_types join pizzas
11  on pizza_types.pizza_type_id = pizzas.pizza_type_id
12  join order_details
13  on order_details.pizza_id = pizzas.pizza_id
14  group by pizza_types.category , pizza_types.name) as a) as b
15  where ranks <= 3 ;
```

Result Grid	
	name
▶	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
	33265.000000000005

Result 1 ×

A festive illustration set against a red background with a white grid. In the center, the words "THANK YOU" are written in large, bold, white capital letters. To the left, a person with dark curly hair and a green shirt holds a slice of pizza. To the right, another person with glasses and a green shirt holds a piece of pie. Above them, a reindeer with a yellow and orange patterned collar and a bell hangs from its neck looks down. The reindeer's collar features a dollar sign (\$) symbol. The scene is decorated with yellow stars and a large yellow starburst at the bottom.

THANK YOU