```
create table orders (
    order_id int not null,
    order_date date not null,
    order_time time not null,
    primary key(order_id) );

create table order_details (
    order_details_id int not null,
    order_id int not null,
    pizza_id text not null,
    quantity int not null,
    primary key(order_details_id) );

-- Retrieve the total number of orders placed.

SELECT
```



COUNT(order_id) AS total_orders

FROM

orders;

-- Calculate the total revenue generated from pizza sales.

SELECT

ROUND(SUM(order_details.quantity * pizzas.price),

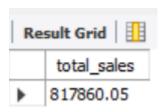
2) AS total_sales

FROM

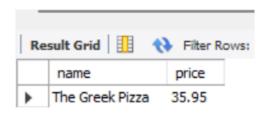
order_details

JOIN

pizzas ON pizzas.pizza_id = order_details.pizza_id



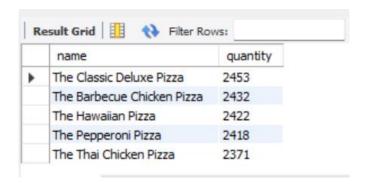
-- Identify the highest-priced pizza.



```
1
       -- Identify the most common pizza size ordered.
 2
 3 •
       SELECT
           pizzas.size,
 4
 5
           COUNT(order_details.order_details_id) AS order_count
       FROM
 7
           pizzas
 8
               JOIN
           order_details ON pizzas.pizza_id = order_details.pizza_id
 9
10
       GROUP BY pizzas.size
       ORDER BY order count DESC;
11
```

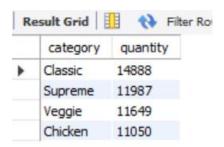
Re	esult Gri	d 🔢 🚷 Filter Rows
	size	order_count
•	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

```
-- List the top 5 most ordered pizza types along with their quantities.
1
2
3 •
       SELECT
           pizza_types.name, SUM(order_details.quantity) AS quantity
4
       FROM
5
6
           pizza_types
           pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9
           order_details ON order_details.pizza_id = pizzas.pizza_id
10
       GROUP BY pizza_types.name
11
       ORDER BY quantity DESC
12
       LIMIT 5;
13
```



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

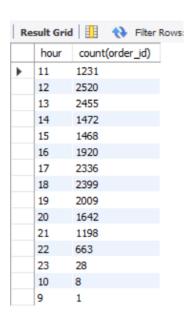
```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity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



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

```
SELECT
```

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



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

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```



-- Group the orders by date and calculate the average number of pizzas ordered per day

SELECT ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day FROM

GROUP BY orders.order_date) AS order_quantity;

(SELECT

orders.order_date, SUM(order_details.quantity) AS quantity
FROM
 orders
JOIN order_details ON orders.order_id = order_details.order_id



```
-- 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 pizzas.pizza_type_id = pizza_types.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	
Þ	The Thai Chicken Pizza	43434.25	
	The Barbecue Chicken Pizza	42768	
	The California Chicken Pizza	41409.5	