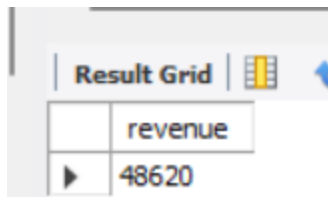


1.Retrieve the total number of orders placed.

Code:

```
SELECT count(ORDER_ID) as revenue FROM order_details;
```

Output:



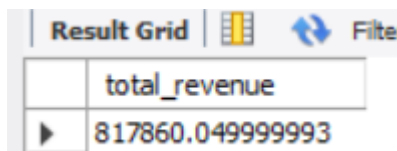
Result Grid	
	revenue
▶	48620

2.Calculate the total revenue generated from pizza sales.

Code:

```
select sum(order_details.quantity*pizzas.price) as total_revenue  
from pizzas join order_details  
on pizzas.pizza_id=order_details.pizza_id
```

Output:



Result Grid	
	total_revenue
▶	817860.0499999993

3.Identify the highest-priced pizza.

Code:

```
select pizzas.price,pizza_types.name  
from pizzas join pizza_types  
on pizzas.pizza_type_id=pizza_types.pizza_type_id  
order by pizzas.price desc limit 1;
```

Output:

Result Grid			Filter Rows:
	price	name	
▶	35.95	The Greek Pizza	

4. Identify the most common pizza size ordered.

Code:

```
select pizzas.size, count(order_details.quantity) as order_cunt
from pizzas join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizzas.size
order by order_cunt desc limit 1;
```

Output:

Result Grid			Filter Ro
	size	order_cunt	
▶	L	18526	

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

Code:

```
select sum(order_details.quantity) as most_ordered, pizza_types.name
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.name
order by most_ordered desc limit 5;
```

Output:

Result Grid			Filter Rows:
	most_ordered	name	
▶	2453	The Classic Deluxe Pizza	
	2432	The Barbecue Chicken Pizza	
	2422	The Hawaiian Pizza	
	2418	The Pepperoni Pizza	
	2371	The Thai Chicken Pizza	

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

Code:

```
select pizza_types.category, sum(order_details.quantity) as total
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.category order by total;
```

Output:

Result Grid			Filter
	category	total	
▶	Chicken	11050	
	Veggie	11649	
	Supreme	11987	
	Classic	14888	

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

Code:

```
select hour(time), count(order_id) from orders group by hour(time) ;
```

Output:

Result Grid			Filter Rows:
	hour(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	

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

Code:

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

Output:

Result Grid			Filter Rows:
	category	count(name)	
▶	Chicken	6	
	Classic	8	
	Supreme	9	
	Veggie	9	

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

Code:

```

select avg(agv) from
> (select orders.date,sum(order_details.quantity) as agv
  from orders join order_details
  on orders.order_id=order_details.order_id
  group by orders.date ) as order_quantity;

```

Output:

Result Grid	
	avg(agv)
▶	138.4749

10.Determine the top 3 most ordered pizza types based on revenue.

Code:

```

select sum(order_details.quantity * pizzas.price) as revenue,pizza_types.name
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.name
order by revenue desc limit 3;

```

Output:

Result Grid		
	revenue	name
▶	43434.25	The Thai Chicken Pizza
	42768	The Barbecue Chicken Pizza
	41409.5	The California Chicken Pizza

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

Code:

```

> select pizza_types.category,(sum(order_details.quantity*pizzas.price)/
> (select(round(sum(order_details.quantity*pizzas.price),2) ) as per
~ from order_details join pizzas on pizzas.pizza_id=order_details.pizza_id))*100 as rev
  from pizzas join pizza_types
  on pizza_types.pizza_type_id=pizzas.pizza_type_id
  join order_details
  on pizzas.pizza_id=order_details.pizza_id
  group by pizza_types.category order by rev desc;

```

Output:

Result Grid			Filter Rows:
	category	rev	
▶	Classic	26.90596025566967	
	Supreme	25.45631126009862	
	Chicken	23.955137556847287	
	Veggie	23.682590927384577	

12. Analyze the cumulative revenue generated over time.

Code:

```

select date,sum(revenue) over(order by date) as cum_revenue
from
(select orders.date,SUM(pizzas.price*order_details.quantity) as revenue
from pizzas join order_details
on pizzas.pizza_id=order_details.pizza_id
join orders
on orders.order_id=order_details.order_id
group by orders.date) as sales;

```

Output:

Result Grid			Filter Rows:
	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	
	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	34343.500000000001	
	2015-01-16	36937.650000000001	
	2015-01-17	39001.750000000001	
	2015-01-18	40978.600000000006	
	2015-01-19	43365.750000000001	
	2015-01-20	45763.650000000001	
	2015-01-21	47804.200000000001	
	2015-01-22	50300.900000000001	
	2015-01-23	52724.600000000006	