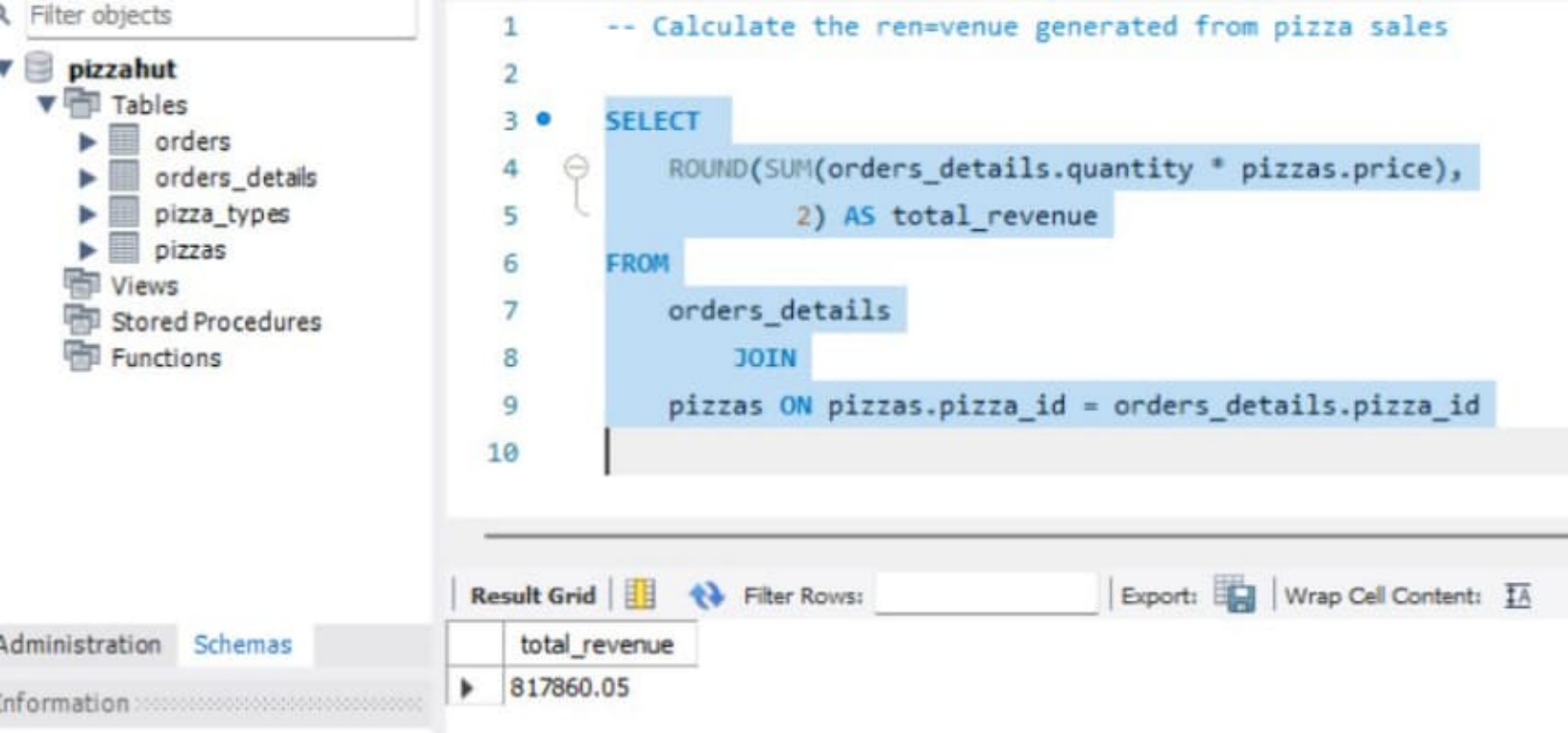
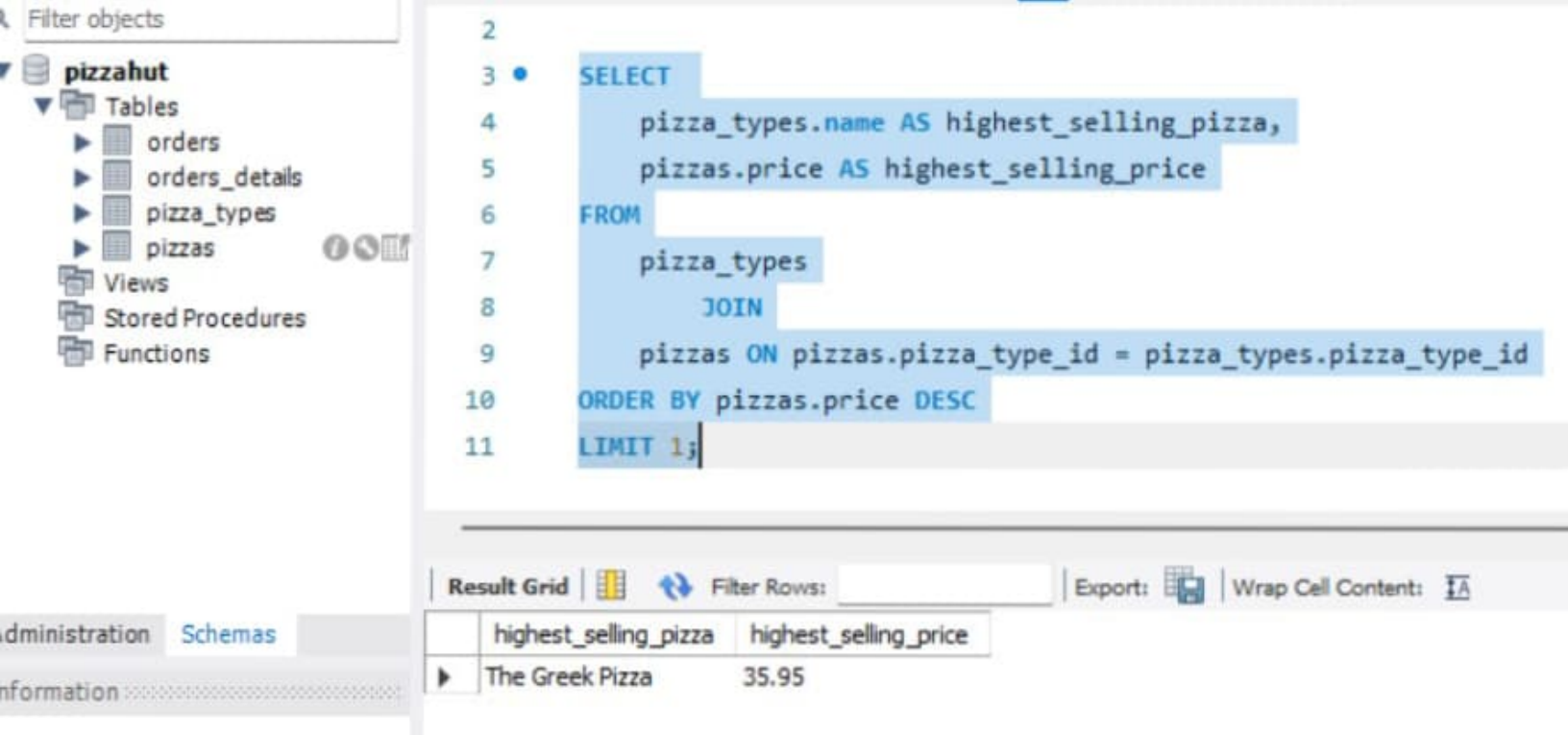
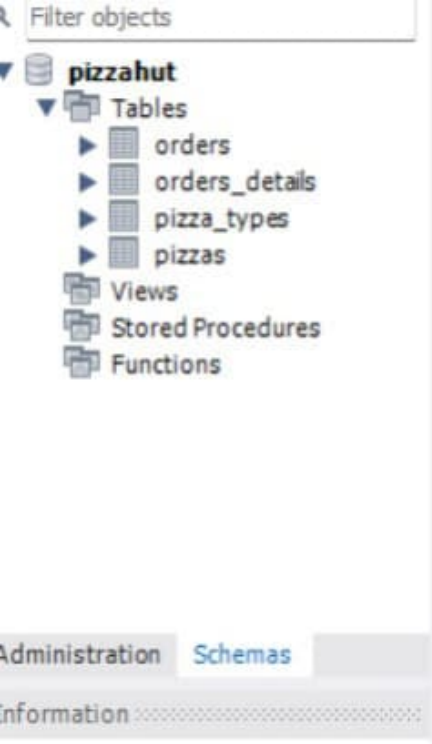

Filter objects
pizzahut
Tables
orders
orders_details
pizza_types
pizzas
Views
Stored Procedures
Functions
1 -- Calculate the ren=venue generated from pizza sales
2
3 SELECT
4 ROUND(SUM(orders_details.quantity * pizzas.price),
5 2) AS total_revenue
6 FROM
7 orders_details
8 JOIN
9 pizzas ON pizzas.pizza_id = orders_details.pizza_id
10
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
total_revenue
817860.05
Administration
Schemas
Information

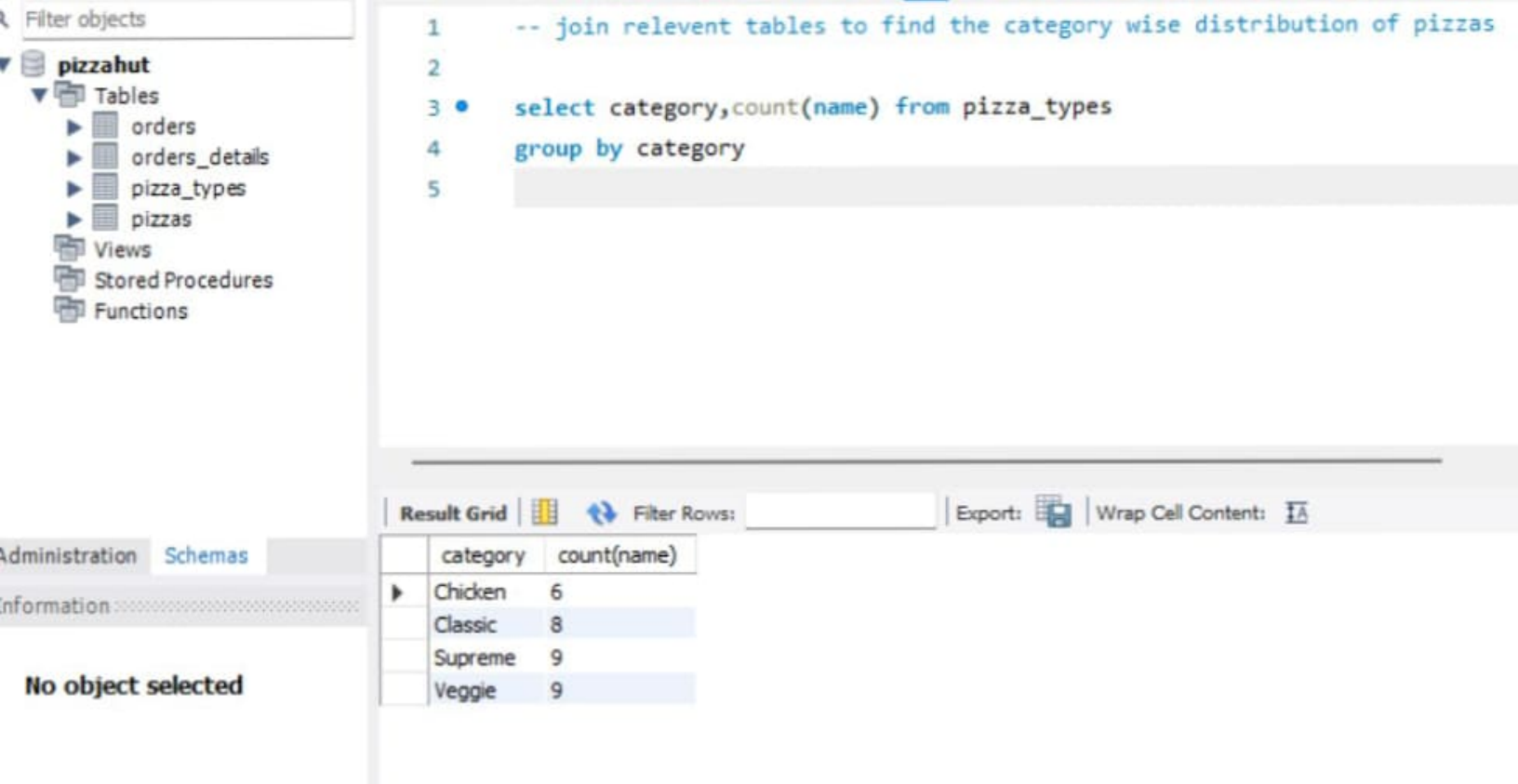

Filter objects
pizzahut
Tables
orders
orders_details
pizza_types
pizzas
Views
Stored Procedures
Functions
Administration
Schemas
Information
2
3
4
5
6
7
8
9
10
11
SELECT
pizza_types.name AS highest_selling_pizza,
pizzas.price AS highest_selling_price
FROM
pizza_types
JOIN
pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
highest_selling_pizza
highest_selling_price
The Greek Pizza
35.95

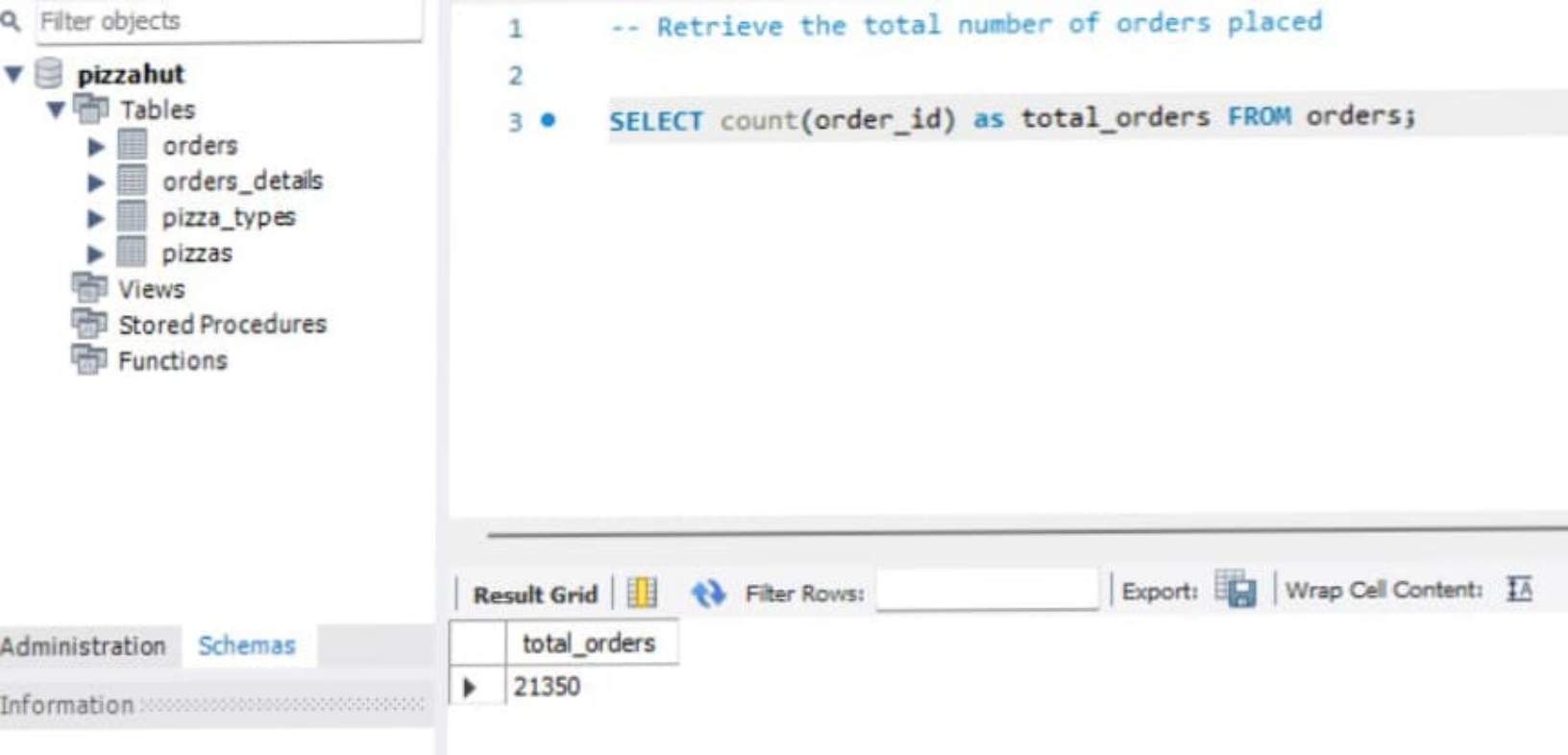


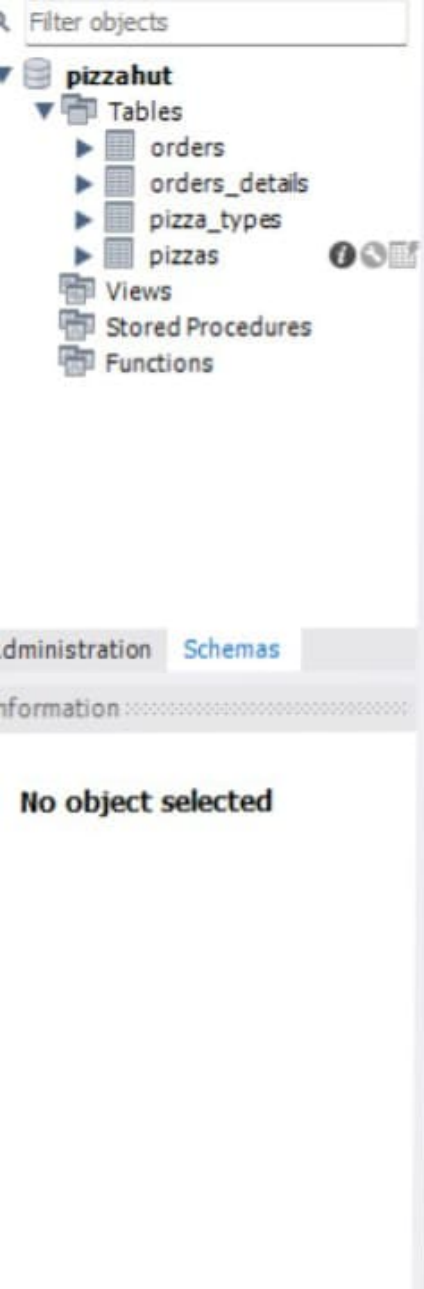

```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

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

size	order_count
L	18526


Filter objects
pizzahut
Tables
orders
orders_details
pizza_types
pizzas
Views
Stored Procedures
Functions
1 -- join relevent tables to find the category wise distribution of pizzas
2
3 select category,count(name) from pizza_types
4 group by category
5
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
category
count(name)
Chicken 6
Classic 8
Supreme 9
Veggie 9
Administration
Schemas
Information
No object selected

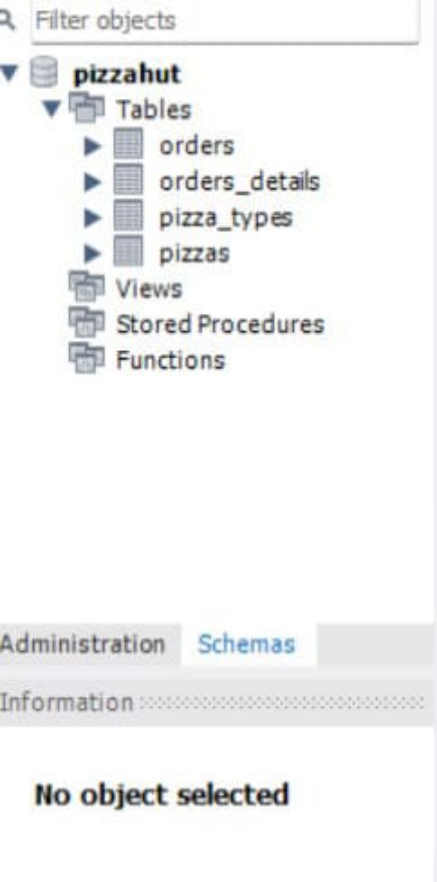

Filter objects
pizzahut
Tables
orders
orders_details
pizza_types
pizzas
Views
Stored Procedures
Functions
Administration
Schemas
Information
1 -- Retrieve the total number of orders placed
2
3 SELECT count(order_id) as total_orders FROM orders;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
total_orders
21350




```sql
-- Analyze the cumulative revenue generated over time
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id=pizzas.pizza_id
join orders
on orders.order_id=orders_details.order_id
group by orders.order_date) as sales;
```

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

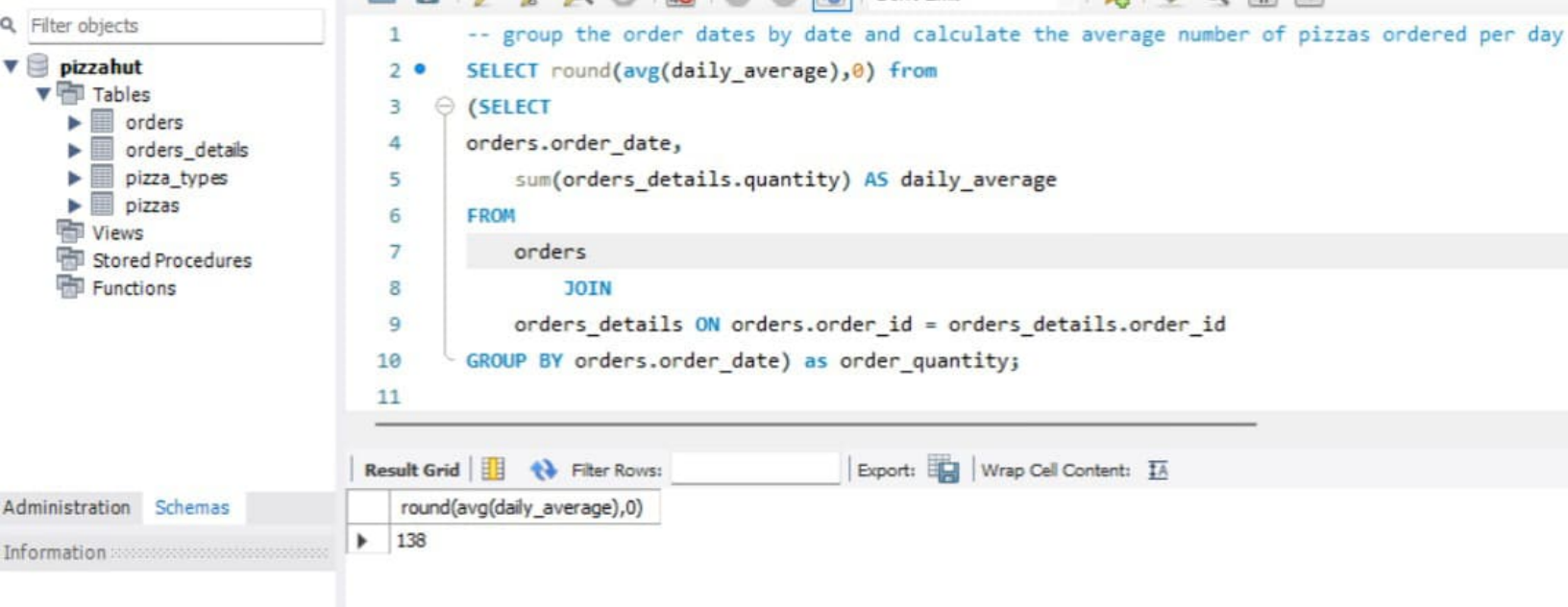
order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001

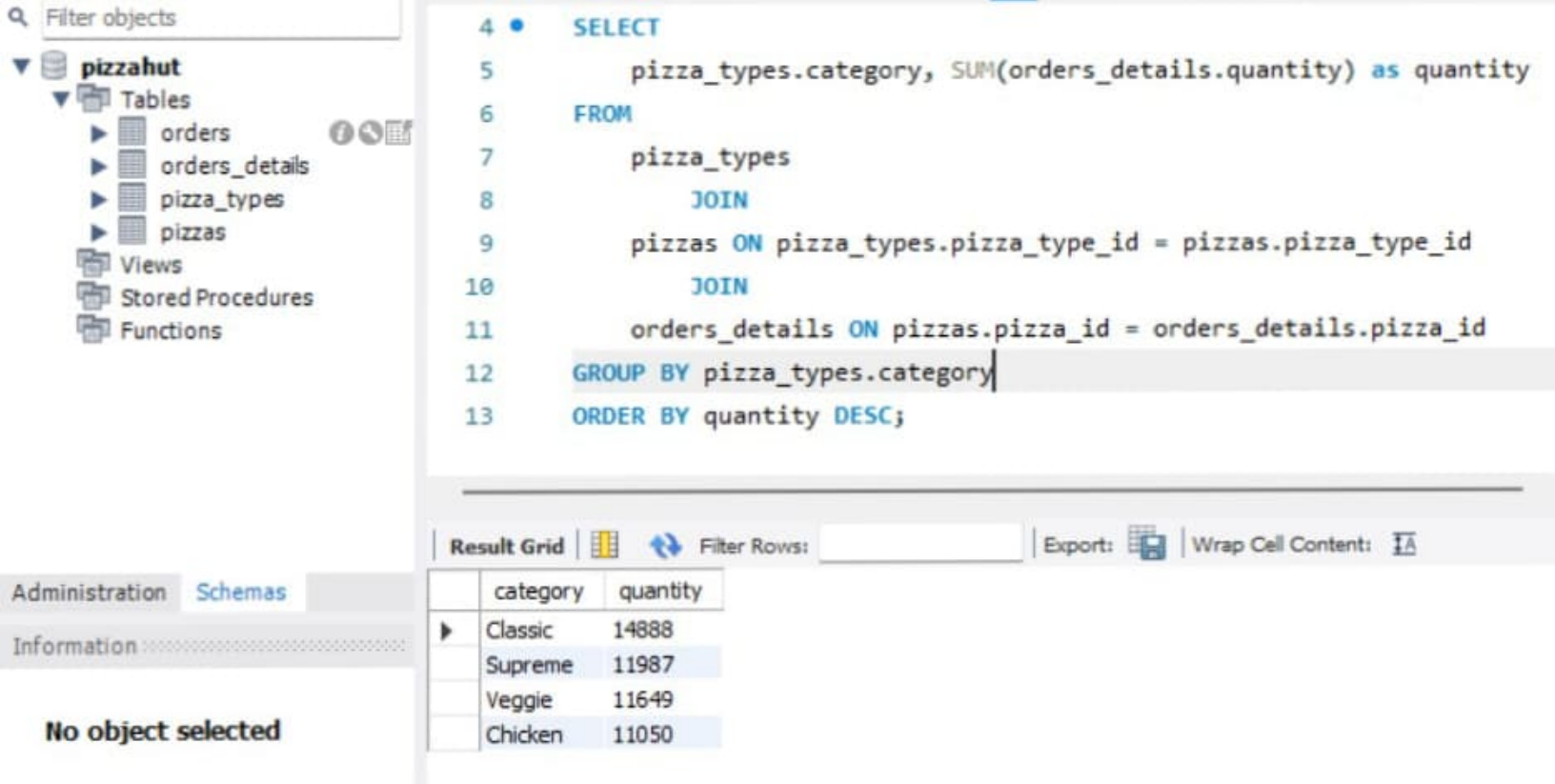


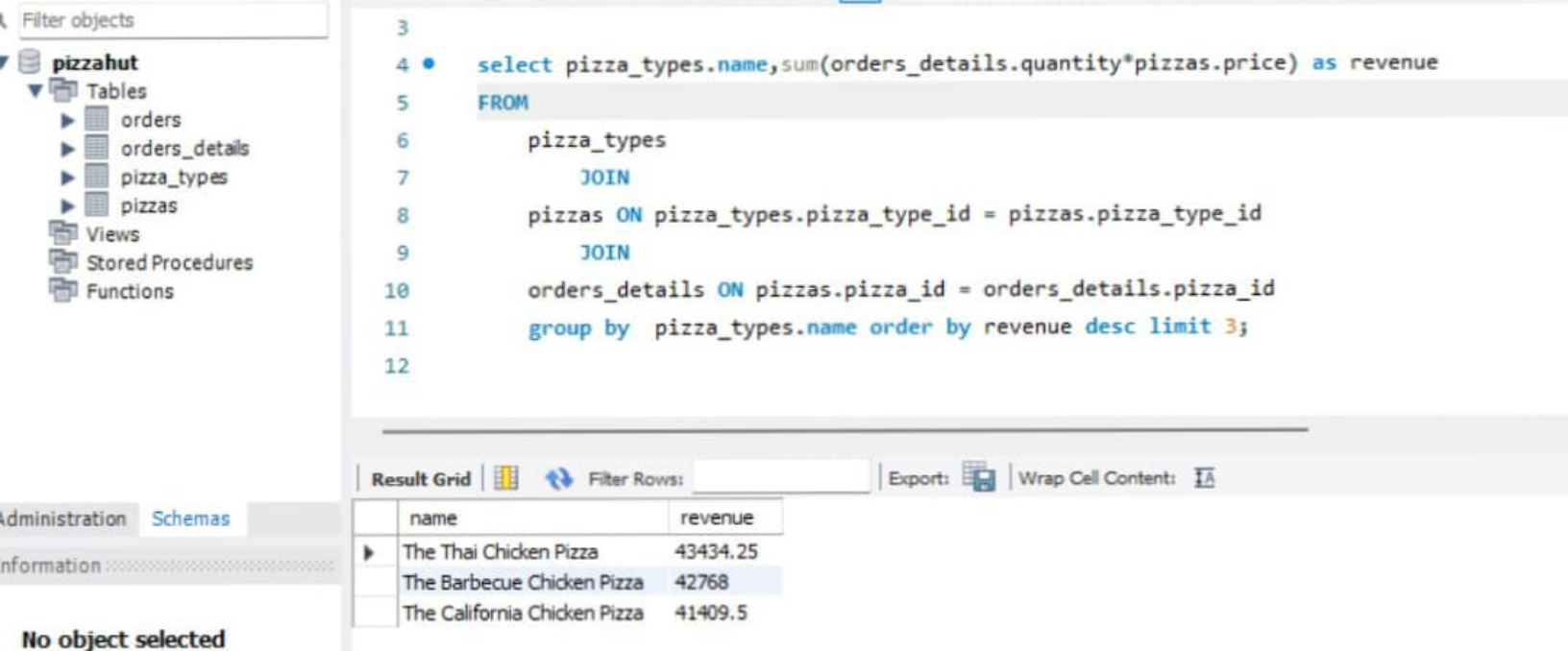

```sql
    pizza_types.name, SUM(orders_details.quantity) as quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

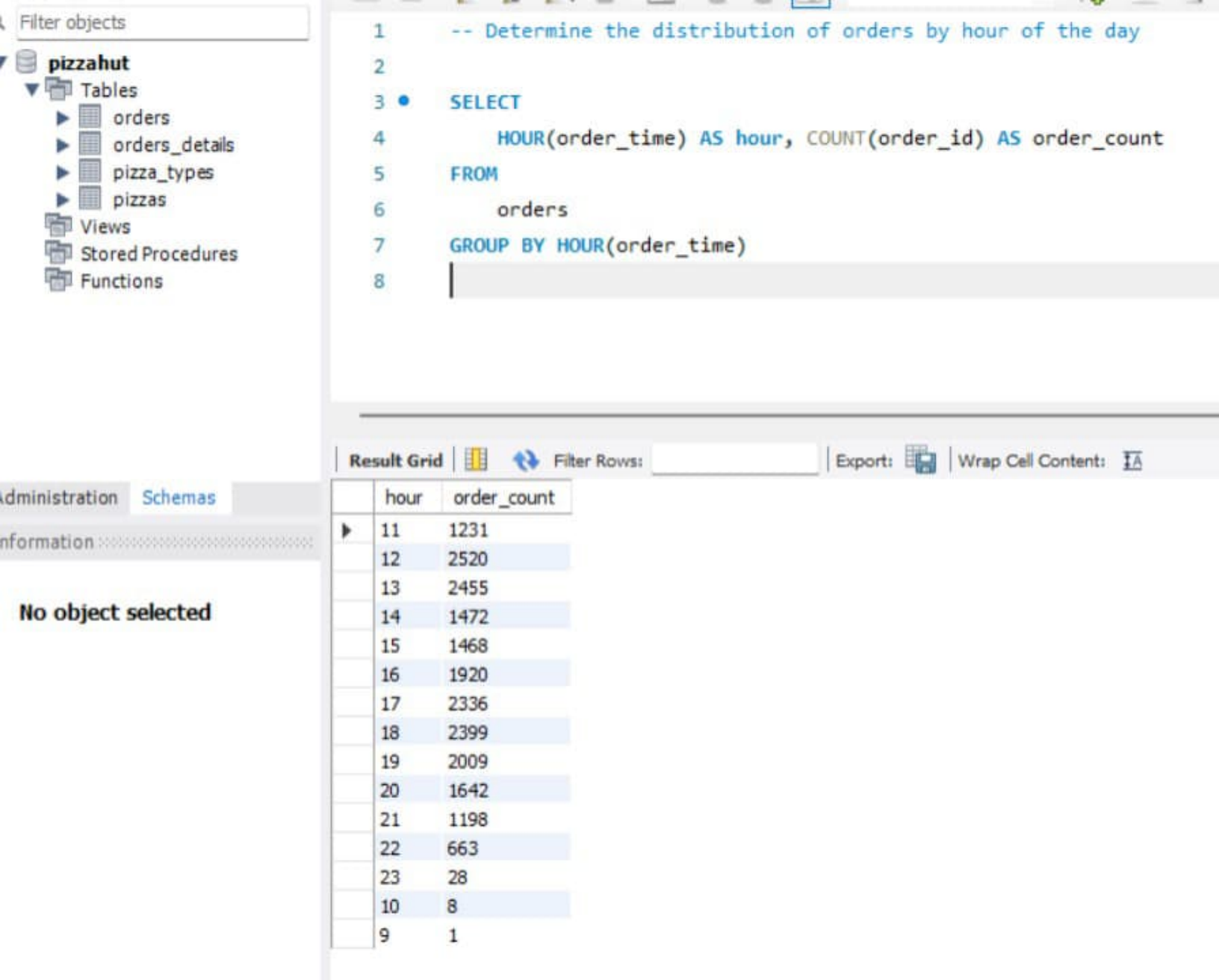
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

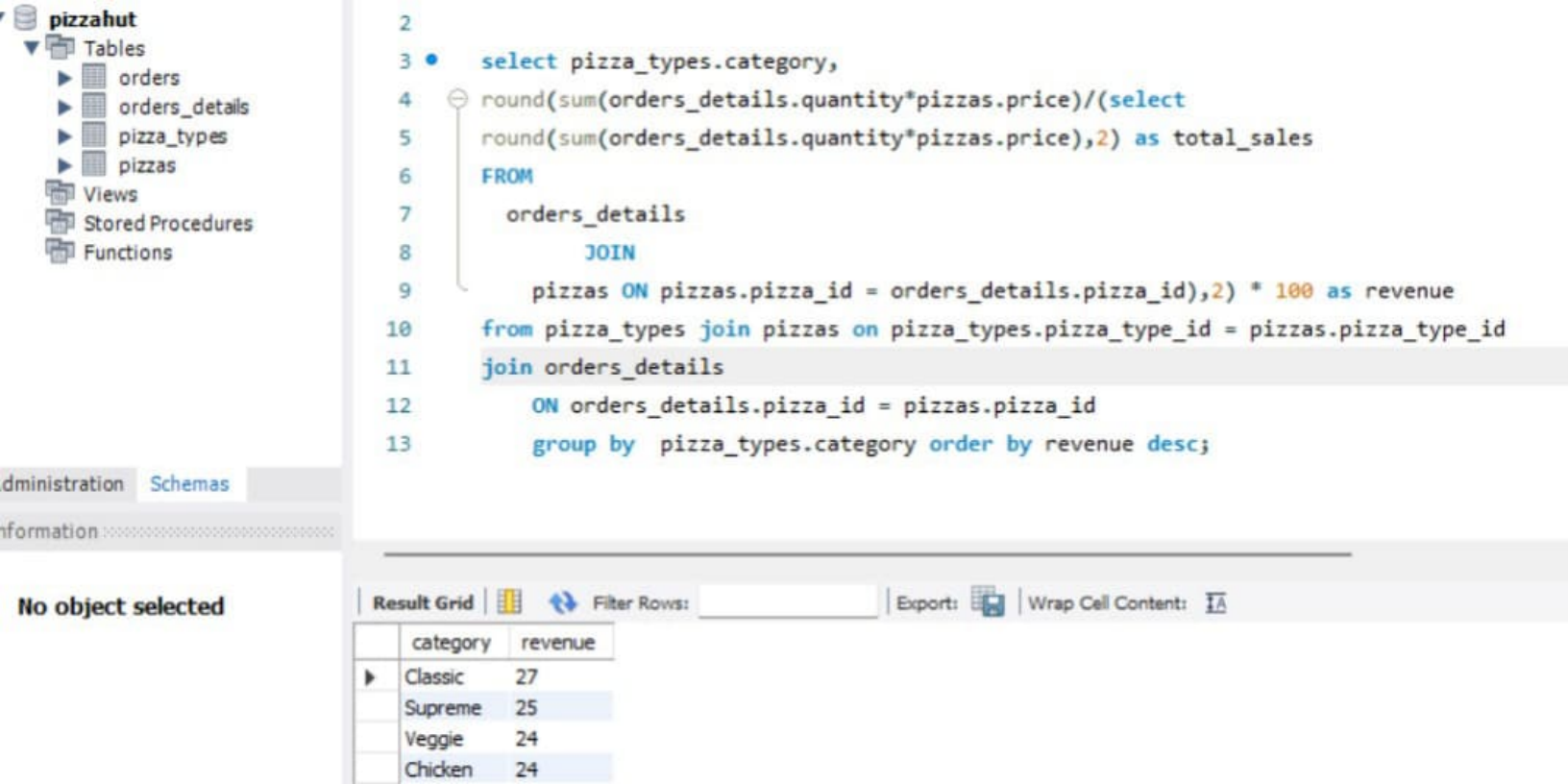
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371


Filter objects
pizzahut
Tables
orders
orders_details
pizza_types
pizzas
Views
Stored Procedures
Functions
Administration
Schemas
Information
1 -- group the order dates by date and calculate the average number of pizzas ordered per day
2 SELECT round(avg(daily_average),0) from
3 (SELECT
4 orders.order_date,
5 sum(orders_details.quantity) AS daily_average
6 FROM
7 orders
8 JOIN
9 orders_details ON orders.order_id = orders_details.order_id
10 GROUP BY orders.order_date) as order_quantity;
11
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
round(avg(daily_average),0)
138


Filter objects
pizzahut
Tables
orders
orders_details
pizza_types
pizzas
Views
Stored Procedures
Functions
Administration
Schemas
Information
No object selected
4 SELECT
5 pizza_types.category, SUM(orders_details.quantity) as quantity
6 FROM
7 pizza_types
8 JOIN
9 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10 JOIN
11 orders_details ON pizzas.pizza_id = orders_details.pizza_id
12 GROUP BY pizza_types.category
13 ORDER BY quantity DESC;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
category quantity
Classic 14888
Supreme 11987
Veggie 11649
Chicken 11050


Filter objects
pizzahut
Tables
orders
orders_details
pizza_types
pizzas
Views
Stored Procedures
Functions
3
4 • select pizza_types.name,sum(orders_details.quantity*pizzas.price) as revenue
5 FROM
6 pizza_types
7 JOIN
8 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9 JOIN
10 orders_details ON pizzas.pizza_id = orders_details.pizza_id
11 group by pizza_types.name order by revenue desc limit 3;
12
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
name
revenue
The Thai Chicken Pizza
43434.25
The Barbecue Chicken Pizza
42768
The California Chicken Pizza
41409.5
Administration
Schemas
Information
No object selected


Filter objects
pizzahut
Tables
orders
orders_details
pizza_types
pizzas
Views
Stored Procedures
Functions
1 -- Determine the distribution of orders by hour of the day
2
3 SELECT
4 HOUR(order_time) AS hour, COUNT(order_id) AS order_count
5 FROM
6 orders
7 GROUP BY HOUR(order_time)
8
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
18 2399
19 2009
20 1642
21 1198
22 663
23 28
10 8
9 1
Administration
Schemas
Information
No object selected


pizzahut
Tables
orders
orders_details
pizza_types
pizzas
Views
Stored Procedures
Functions
dministration
Schemas
nformation
No object selected
2
3
4
5
6
7
8
9
10
11
12
13
select pizza_types.category,
round(sum(orders_details.quantity*pizzas.price)/(select
round(sum(orders_details.quantity*pizzas.price),2) as total_sales
FROM
orders_details
JOIN
pizzas ON pizzas.pizza_id = orders_details.pizza_id),2) * 100 as revenue
from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
ON orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.category order by revenue desc;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
category
revenue
Classic
27
Supreme
25
Veggie
24
Chicken
24