



Borcalle
PIZZA RESTAURANT

Special Delicious

PIZZA MADE IN HEAVEN



Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL SQL i + + + + + + + +

Navigator SQL File 7* × SQL File 8* SQL File 9* SQL File 10* SQL File 10* SQL File 11* SQL File

SCHEMAS Filter objects

pizzahut

- Tables
 - orders
 - orders_details
 - pizza_types
 - pizzas
- Views
- Stored Procedures
- Functions

sys

SQL File 7* × SQL File 8* SQL File 9* SQL File 10* SQL File 10* SQL File 11* SQL File

1 -- Retrive total number of orders placed.

2

3 • SELECT count(order_id)as total_orders FROM pizzahut.orders;

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

	total_orders
▶	21350

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL SQL i + f0

Navigator

SCHEMAS

Filter objects

pizzahut

- Tables
 - orders
 - orders_details
 - pizza_types
 - pizzas
- Views
- Stored Procedures
- Functions

sys

SQL File 7* SQL File 8* × SQL File 9* SQL File 10* SQL File 10* SQL File 11* SQL File 12*

1 -- Calculate the total revenue generated from pizza sales.
2 • SELECT
3 ROUND(SUM(orders_details.quantity * pizzas.price),
4 2) AS total_sales
5 FROM
6 orders_details
7 JOIN
8 pizzas ON pizzas.pizza_id = orders_details.pizza_id

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

	total_sales
▶	817860.05



Navigator

SCHEMAS

Filter objects

pizzahut

Tables

- ▶ orders
- ▶ orders_details
- ▶ pizza_types
- ▶ pizzas

Views

Stored Procedures

Functions

sys

SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 10* SQL File 11* SQL File 10*



```
1 -- Identify the highest price pizza.  
2 • SELECT  
3     pizza_types.name, (pizzas.price)  
4 FROM  
5     pizza_types  
6     JOIN  
7         pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
8 ORDER BY pizzas.price DESC  
9 LIMIT 1;  
10  
11
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:

Fetch rows:



	name	price
▶	The Greek Pizza	35.95

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL SQL SQL SQL SQL SQL SQL

Navigator SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 10* SQL File 11* SQL File 10*

SCHEMAS Filter objects

▼ pizzahut

- Tables
 - orders
 - orders_details
 - pizza_types
 - pizzas
- Views
- Stored Procedures
- Functions

► sys

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

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

	size	order_count
▶ L		18526

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQl SQL i + fo SQL Navigator SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 10* × SQL File 11* SQL File 10* SQL File 11

SCHEMAS

Filter objects

▼ pizzahut

- Tables
 - orders
 - orders_details
 - pizza_types
 - pizzas
- Views
- Stored Procedures
- Functions

► sys

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

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

	name	SUM(orders_details.quantity)
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL SQL+ Database Import Export Find All Open Recent

Navigator

SCHEMAS

Filter objects

pizzahut

- Tables
 - orders
 - orders_details
 - pizza_types
 - pizzas
- Views
- Stored Procedures
- Functions

sys

SQL File 9* SQL File 10* SQL File 10* SQL File 11* × SQL File 10* SQL File 11* SQL File 14*

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

2 • SELECT pizza_types.category, SUM(orders_details.quantity)as quantity

3 FROM

4 pizza_types

5 JOIN

6 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

7 JOIN

8 orders_details ON orders_details.pizza_id = pizzas.pizza_id

9 GROUP BY pizza_types.category

10 ORDER BY SUM(orders_details.quantity) DESC

11 LIMIT 5;

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

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL Navigator SQL File 9* SQL File 10* SQL File 10* SQL File 11* SQL File 10* × SQL File 11* SQL File 14* SQL File 12* SQL File 13* SQL

SCHEMAS

Filter objects

pizzahut

- Tables
 - orders
 - orders_details
 - pizza_types
 - pizzas
- Views
- Stored Procedures
- Functions

sys

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

1
2
3 • SELECT
4 HOUR(order_time) AS hour, COUNT(order_id)
5 FROM
6 orders
7 GROUP BY HOUR(order_time)
8 ORDER BY COUNT(order_id) DESC;

9
10

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

	hour	COUNT(order_id)
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198

Administration Schemas

Information

No object selected

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL SQL+ Database Navigator

SCHEMAS Filter objects

pizzahut

- Tables
 - orders
 - orders_details
 - pizza_types
 - pizzas
- Views
- Stored Procedures
- Functions

sys

SQL File 9* SQL File 10* SQL File 10* SQL File 11* SQL File 10* SQL File 11* × SQL File 14* SQL File 12*

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

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

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL SQL Navigator

SCHEMAS Filter objects

pizzahut

- Tables
 - orders
 - orders_details
 - pizza_types
 - pizzas
- Views
- Stored Procedures
- Functions

sys

SQL File 9* SQL File 10* SQL File 10* SQL File 11* SQL File 10* SQL File 11* SQL File 12* × SQL File 13*

1 -- Group the order by date and calculate the average number of pizzas ordered per day
2 • SELECT
3 ROUND(AVG(quantity), 0) as avg_pizzas_ordered_per_day
4 FROM
5 (SELECT
6 orders.order_date, SUM(orders_details.quantity) AS quantity
7 FROM
8 orders
9 JOIN 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:

	avg_pizzas_ordered_per_day
▶	138



Navigator:

SCHEMAS

 Filter objects

▼ pizzahut

- ▶  orders
 - ▶  orders_details
 - ▶  pizza_types
 - ▶  pizzas

Views

-  Stored Procedures
-  Functions

sys

```
1 -- Determine the top 3 most ordered pizza types based on revenue
2 • SELECT pizza_types.name,
3         SUM(orders_details.quantity * pizzas.price) AS revenue
4     FROM pizza_types
5             JOIN
6         pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
7             JOIN
8         orders_details ON orders_details.pizza_id = pizzas.pizza_id
9     GROUP BY pizza_types.name
10    ORDER BY revenue DESC
11    LIMIT 3;
```

Result Grid

Filter Row

Export:

W

Wrap Cell Content:

Fetch rows

1

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

SCHEMAS

Filter objects

pizzahut

Tables

orders

orders_details

pizza_types

pizzas

Views

Stored Procedures

Functions

sys

```

1      -- Calculate the percentage contribution of each pizza type to total revenue.
2 •  SELECT pizza_types.category,
3   round(  ( SUM(orders_details.quantity * pizzas.price) / (SELECT
4   ROUND(SUM(orders_details.quantity * pizzas.price),
5   2) AS total_sales
6   FROM orders_details
7   JOIN
8   pizzas ON pizzas.pizza_id = orders_details.pizza_id))*100,2)as revenue
9   FROM pizza_types
10  JOIN
11  pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
12  JOIN
13  orders_details ON orders_details.pizza_id = pizzas.pizza_id
14  GROUP BY pizza_types.category
15  ORDER BY revenue DESC
16  LIMIT 3;
17

```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:



	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	22.26

Administration Schemas

Information



Navigator

SCHEMAS

Filter objects

pizzahut

Tables

- orders
- orders_details
- pizza_types
- pizzas

Views

Stored Procedures

Functions

sys

SQL File 9* SQL File 10* SQL File 10* SQL File 11* SQL File 10* SQL File 11* SQL File 12* S



```
1      -- Analyze the cumulative revenue generated over time.  
2  •  select order_date,  
3      sum(revenue) over(order by order_date) as cum_revenue  
4      from  
5      (select orders.order_date,  
6          sum(orders_details.quantity * pizzas.price) as revenue  
7          from orders_details join pizzas  
8          on orders_details.pizza_id = pizzas.pizza_id  
9          join orders  
10         on orders.order_id = orders_details.order_id  
11      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	14258.5

Filter objects

pizzahut

Tables

- ▶ orders
- ▶ orders_details
- ▶ pizza_types
- ▶ pizzas

Views

Stored Procedures

Functions

sys

```

1   -- Determine the top 3 most ordered pizza types based on revenue for each pizza category
2 •  select name, revenue from
3   (select category,name,revenue,
4    rank() over(partition by category order by revenue desc) as rn
5   from
6   (select pizza_types.category,pizza_types.name,sum(orders_details.quantity * pizzas.price)as revenue
7    from pizza_types join pizzas
8    on pizza_types.pizza_type_id = pizzas.pizza_type_id
9    join orders_details
10   on orders_details.pizza_id = pizzas.pizza_id
11   group by pizza_types.category,pizza_types.name) as a ) as b
12   where rn <=3;

```

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
	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	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5