



SCHEMAS

Filter objects

▼ pizza_sales_analysis

▼ Tables

- ▶ order_details
- ▶ orders
- ▶ pizza_types
- ▶ pizzas

Views

Stored Procedures

Functions

▶ sys

No object selected



```
1  -- 4. Identify the most common pizza size ordered.
```

```
2
```

```
3  •  SELECT
```

```
4      pizzas.size, COUNT(order_details.quantity) AS total_order
```

```
5  FROM
```

```
6      order_details
```

```
7      JOIN
```

```
8      pizzas ON order_details.pizza_id = pizzas.pizza_id
```

```
9  GROUP BY pizzas.size
```

```
10 ORDER BY total_order DESC
```

```
11 LIMIT 1
```

```
12
```



| | size | total_order |
|---|------|-------------|
| ▶ | L | 18526 |



Navigator

SCHEMAS

Filter objects

▼ pizza_sales_analysis

▼ Tables

▶ order_details

▶ orders

▶ pizza_types

▶ pizzas

Views

Stored Procedures

Functions

▶ sys

Administration Schemas

Information

No object selected

SQL File 10*

SQL File 11*

SQL File 12*

SQL File 13*

SQL File 14*

SQL File 17*

SQL File 18*

SQL File 19*



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

2

3 • select category, count(name) from pizza_types

4 group by category

5

6

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

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



Navigator

SCHEMAS

Filter objects

pizza_sales_analysis

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

sys

Administration Schemas

Information

No object selected

SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 17* SQL File 18* SQL File 19* SQL File 20*



```
1 -- 9. Group the orders by date and calculate the average number of pizzas ordered per day.
2
3 • SELECT
4     ROUND(AVG(quantity), 2) AS average_order_per_day
5 FROM
6     (SELECT
7         orders.order_date AS date,
8         SUM(order_details.quantity) AS quantity
9     FROM
10        orders
11     JOIN order_details ON orders.order_id = order_details.order_id
12     GROUP BY orders.order_date) AS new_table
```

Result Grid Filter Rows: Export: Wrap Cell Content:

| | average_order_per_day |
|---|-----------------------|
| ▶ | 138.47 |

Result 1 x



Navigator:

SCHEMAS

Filter objects

- ▼ pizza_sales_analysis
 - ▼ Tables
 - order_details
 - orders
 - pizza_types
 - pizzas
 - Views
 - Stored Procedures
 - Functions
- ▼ sys

Administration Schemas

Information:

No object selected

SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 17* SQL File 18* SQL File 19* SQL File 20*

Limit to 1000 rows

```
1  -- 10. Determine the top 3 most ordered pizza types based on revenue.
2
3  • select pizza_types.name, sum(order_details.quantity*pizzas.price) as revenue
4    from pizzas join order_details on pizzas.pizza_id = order_details.pizza_id
5    join pizza_types on pizza_types.pizza_type_id = pizzas.pizza_type_id
6    group by name
7    order by revenue desc
8    limit 3
```

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

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

Result 2



Navigator

SCHEMAS

Filter objects

▼ pizza_sales_analysis

▼ Tables

- ▶ order_details
- ▶ orders
- ▶ pizza_types
- ▶ pizzas

Views

Stored Procedures

Functions

▶ sys

Administration Schemas

Information

No object selected

SQL File 10* x SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 17*



Limit to 1000 rows

```
1  -- 1. Retrieve the total number of orders placed.
2
3  •  SELECT
4      COUNT(order_id)
5  FROM
6      orders
```

Result Grid Filter Rows: Export: Wrap Cell Content:

| | COUNT(order_id) |
|---|-----------------|
| ▶ | 21350 |

Navigator: CHEMAS

Filter objects

- ▼ pizza_sales_analysis
 - ▼ Tables
 - order_details
 - orders
 - pizza_types
 - pizzas
 - Views
 - Stored Procedures
 - Functions
- sys

Administration Schemas

Information

No object selected

SQL File 10 SQL File 11 SQL File 12 SQL File 13 SQL File 14 SQL File 17 SQL File 18 SQL File 19 SQL File 20

Limit to 1000 rows

```
1 -- 6. Join the necessary tables to find the total quantity of each pizza category ordered.
2
3 • SELECT
4     pizza_types.category,
5     SUM(order_details.quantity) AS total_quantity
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10    JOIN
11    order_details ON pizzas.pizza_id = order_details.pizza_id
12 GROUP BY pizza_types.category
```

Result Grid

| | category | total_quantity |
|---|----------|----------------|
| ▶ | Classic | 14888 |
| | Veggie | 11649 |
| | Supreme | 11987 |
| | Chicken | 11050 |

Export: Wrap Cell Content:



Navigator

SCHEMAS

Filter objects

▼ pizza_sales_analysis

▼ Tables

▶ order_details

▶ orders

▶ pizza_types

▶ pizzas

Views

Stored Procedures

Functions

▶ sys

Administration Schemas

Information

No object selected

SQL File 10*

SQL File 11*

SQL File 12*

SQL File 13*

SQL File 14* x

SQL File 17*

SQL File 18*



Limit to 1000 rows

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

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:



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



SCHEMAS

Filter objects

▼ pizza_sales_analysis

▼ Tables

- ▶ order_details
- ▶ orders
- ▶ pizza_types
- ▶ pizzas

Views

Stored Procedures

Functions

▶ sys

Administration Schemas

Information

No object selected

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

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

| | total_sales |
|---|-------------|
| ▶ | 817860.05 |



Navigator

SCHEMAS

Filter objects

▼ pizza_sales_analysis

▼ Tables

▶ order_details

▶ orders

▶ pizza_types

▶ pizzas

Views

Stored Procedures

Functions

▶ sys

Administration Schemas

Information

No object selected

SQL File 10*

SQL File 11*

SQL File 12* x

SQL File 13*

SQL File 14*

SQL File 17*

SQL F



1 -- 3. Identify the highest-priced pizza.

2

3 • SELECT

4 pizza_types.name, pizzas.price

5 FROM

6 pizza_types

7 JOIN

8 pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id

9 ORDER BY pizzas.price DESC

10 LIMIT 1

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:



| | name | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |



Navigator

SCHEMAS

Filter objects

▼ pizza_sales_analysis

▼ Tables

- ▶ order_details
- ▶ orders
- ▶ pizza_types
- ▶ pizzas

Views

Stored Procedures

Functions

▶ sys

Administration Schemas

Information

No object selected

SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 17* SQL File 18* x



```
1  -- 7. Determine the distribution of orders by hour of the day.
2
3  •  SELECT
4      HOUR(order_time) AS hour_of_the_day, COUNT(order_id) as order_count
5  FROM
6      orders
7  GROUP BY hour_of_the_day
```

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

| | hour_of_the_day | order_count |
|---|-----------------|-------------|
| ▶ | 11 | 1231 |
| | 12 | 2520 |
| | 13 | 2455 |
| | 14 | 1472 |
| | 15 | 1468 |
| | 16 | 1920 |
| | 17 | 2336 |
| | 18 | 2399 |

Result 1 x