



Navigator

**SCHEMAS**

Filter objects

- pizzahut**
  - Tables
    - order\_details
    - orders
    - pizza\_types
    - pizzas
  - Views
  - Stored Procedures
  - Functions
- sys

Administration Schemas

Query 1 SQL File 3\* x SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\*

Limit to 1000 rows

```
1 -- Retrieve the total number of orders placed.  
2  
3 • select count(order_id) as total_orders from orders;
```

Result Grid

	total_orders
▶	21350

Navigator

SCHEMAS

Filter objects

pizzahut

- Tables
  - order\_details
  - orders
  - pizza\_types
  - pizzas
- Views
- Stored Procedures
- Functions

sys

Query 1 SQL File 3 SQL File 4 x SQL File 5 SQL File 6 SQL File 7

Limit to 1000 rows

```
1 -- 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 pizzas.pizza_id = order_details.pizza_id
```

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

	total_sales
▶	817860.05

Navigator

SCHEMAS

Filter objects

- ▼ pizzahut
  - ▼ Tables
    - ▶ order\_details
    - ▶ orders
    - ▶ pizza\_types
    - ▶ pizzas
  - Views
  - Stored Procedures
  - Functions
- ▶ sys

Administration Schemas

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* x SQL File 6\* SQL File 7\*

Limit to 1000 rows

```
1 -- Identify the highest-priced pizza.
2
3 • select pizza_types.name, pizzas.price
4 from pizza_types join pizzas
5 on pizza_types.pizza_type_id = pizzas.pizza_type_id
6 order by pizzas.price desc limit 1;
```

Result Grid

	name	price
▶	The Greek Pizza	35.95

Navigator

SCHEMAS

Filter objects

▼ pizzahut

- ▼ Tables
  - order\_details
  - orders
  - pizza\_types
  - pizzas
- Views
- Stored Procedures
- Functions

▼ sys

Administration Schemas

Information

Table: **order\_details**

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* x SQL File 7\* SQL File 8\*

Limit to 1000 rows

```
1 -- Identify the most common pizza size ordered.
2
3 • select pizzas.size, count(order_details.order_details_id) as order_count
4 from pizzas join order_details
5 on pizzas.pizza_id = order_details.pizza_id
6 group by pizzas.size order by order_count desc;
```

Result Grid

Filter Rows:

Export: Wrap Cell Content:

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28





Navigator

SCHEMAS

Filter objects

▼ pizzahut

- ▼ Tables
  - order\_details
  - orders
  - pizza\_types
  - pizzas
- Views
- Stored Procedures
- Functions

▼ sys

Administration Schemas

Information

Table: order\_details

Query 1 SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* x

Limit to 1000 rows

```
1 -- Determine the distribution of orders by hour of the day.
2
3 • select hour(order_time), count(order_id) as order_count from orders
4 group by hour(order_time);
```

Result Grid

	hour(order_time)	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

Filter Rows:

Export: Wrap Cell Content: [IA](#)

**Columns:**



Navigator

Filter objects

**SCHEMAS**

- pizzahut**
  - Tables
    - order\_details
    - orders
    - pizza\_types
    - pizzas
  - Views
  - Stored Procedures
  - Functions
- sys

Query 1    SQL File 3\*    SQL File 4\*    SQL File 5\*    SQL File 6\*    SQL File 7\*    SQL File 8\*    SQL File 9\*

Limit to 1000 rows

```

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(order_details.quantity * pizzas.price) as revenue
7      from order_details join pizzas
8      on order_details.pizza_id = pizzas.pizza_id
9      join orders
10     on orders.order_id = order_details.order_id
11     group by orders.order_date) as sales;

```

Administration    Schemas

Information

Table: **order\_details**

Columns:

<b>order_details id</b>	int
order_id	PK
pizza_id	int
quantity	te; int

Result Grid    Filter Rows:    Export:    Wrap Cell Content: [I](#)

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