

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

name	price
The Greek Pizza	35.95

```
1 -- Retrieve the total number of orders placed.  
2  
3 • SELECT  
4     COUNT(order_id) AS total_orders  
5 FROM  
6     orderss;  
7
```

Result Grid	
<input type="button" value="Filter Rows:"/>	<input type="button" value="Export:"/>
total_orders	21350

```
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;  
10
```

Result Grid	
	Filter Rows:
total_sales	817860.05

```
1
2  -- Identify the most common pizza size ordered.
3 • SELECT
4      pizzas.size,
5      COUNT(order_details.order_details_id) AS order_count
6  FROM
7      pizzas
8      JOIN
9          order_details ON pizzas.pizza_id = order_details.pizza_i
10     GROUP BY pizzas.size
11     ORDER BY order_count DESC;
12
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	size	order_count		
▶	L	18526		
	M	15385		
	S	14137		
	XL	544		
	XXL	28		

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

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

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

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

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
category	quantity			
Classic	14888			
Supreme	11987			
Veggie	11649			
Chicken	11050			

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

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

```
1
2    -- Join relevant tables to find the
3    -- category-wise distribution of pizzas.
4
5 • SELECT
6      category, COUNT(name)
7  FROM
8      pizza_types
9  GROUP BY category;
```

category		COUNT(name)
►	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

```
1
2  -- Group the orders by date and calculate the average
3  -- number of pizzas ordered per day
4
5 • SELECT
6      ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
7  FROM
8    (SELECT
9      orderss.order_date, SUM(order_details.quantity) AS quantity
10     FROM
11       orderss
12      JOIN order_details ON orderss.order_id = order_details.order_id
13      GROUP BY orderss.order_date) AS order_quantity;
14
15
```

Result Grid	
Filter Rows:	Export: Wrap Cell Content:
avg_pizza_ordered_per_da	138

```
1
2  -- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.
3
4 • SELECT
5      pizza_types.name,
6      SUM(order_details.quantity * pizzas.price) AS revenue
7  FROM
8      pizza_types
9      JOIN
10     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
11     JOIN
12     order_details ON order_details.pizza_id = pizzas.pizza_id
13   GROUP BY pizza_types.name
14   ORDER BY revenue DESC
15   LIMIT 3;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	name	revenue		
▶	The Thai Chicken Pizza	43434.25		
▶	The Barbecue Chicken Pi...	42768		
▶	The California Chicken Pi...	41409.5		

```
1      -- Calculate the percentage contribution of each
2      -- pizza type to total revenue.
3
4 •  SELECT
5      pizza_types.category,
6      ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
7          ROUND(SUM(order_details.quantity * pizzas.price),
8              2) AS total_sales
9      FROM
10         order_details
11         JOIN
12             pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
13      2) revenue
14  FROM
15      pizza_types
16      JOIN
17          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
18      JOIN
19          order_details ON order_details.pizza_id = pizzas.pizza_id
20  GROUP BY pizza_types.category
21  ORDER BY revenue DESC;
22
23
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
category	revenue			
▶ Classic	26.91			
Supreme	25.46			
Chicken	23.96			
Veggie	23.68			

```
1
2  -- Analyze the cumulative revenue generated over time.
3
4 • SELECT order_date,
5   SUM(revenue) OVER(ORDER BY order_date) AS CUM_revenue
6   FROM
7   (SELECT orderss.order_date,
8    sum(order_details.quantity * pizzas.price) AS revenue
9    FROM order_details JOIN pizzas ON
10   order_details.pizza_id = pizzas.pizza_id
11   JOIN orderss
12   ON orderss.order_id = order_details.order_id
13   GROUP BY orderss.order_date) as sales;
```

Result Grid	
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

```
2 -- Determine the top 3 most ordered pizza types
3 -- based on revenue for each pizza category.
4
5 • SELECT name, revenue FROM
6   (SELECT category, name, revenue,
7    RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rn
8   FROM
9   (SELECT pizza_types.category, pizza_types.name,
10    sum((order_details.quantity) * pizzas.price) AS revenue
11   FROM pizza_types JOIN pizzas
12   ON pizza_types.pizza_type_id = pizzas.pizza_type_id
13   JOIN order_details
14   ON order_details.pizza_id = pizzas.pizza_id
15   GROUP BY pizza_types.category, pizza_types.name) AS a)AS b
16 WHERE rn <=3;
```

Result Grid	
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pi...	42768
The California Chicken Pi...	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