

```
20  -- Retrieve the total number of orders placed.
```

```
21
```

```
22  ● SELECT
```

```
23      COUNT(order_id) AS total_orders
```


```
24  FROM
```

```
25      orders;
```

```
26
```

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

	total_orders
▶	21350

  
Result  
Grid

```
27 -- Calculate the total revenue generated from pizza sales.
28
29 • SELECT
30     ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_revenue
31 FROM
32     order_details
33 JOIN
34     pizzas
35 ON
36     pizzas.pizza_id = order_details.pizza_id;
37
```

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





	total_revenue
▶	817860.05

```
38  -- Identify the highest-priced pizza.
39
40  • SELECT
41      pizza_types.name, pizzas.price
42  FROM
43      pizza_types
44  JOIN
45      pizzas
46  ON
47      pizza_types.pizza_type_id = pizzas.pizza_type_id
48  ORDER BY
49      pizzas.price DESC
50  LIMIT 1;
51
```

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

	name	price
▶	The Greek Pizza	35.95

```
52 -- Identify the most common pizza size ordered.
53
54 • SELECT
55     pizzas.size, COUNT(order_details.order_details_id) AS order_count
56 FROM
57     pizzas
58 JOIN
59     order_details
60 ON
61     pizzas.pizza_id = order_details.pizza_id
62 GROUP BY
63     pizzas.size
64 ORDER BY
65     order_count DESC;
66
```


Result Grid   Filter Rows:  Export:  Wrap Cell Content: 


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





```
67 -- List the top 5 most ordered pizza types along with their quantities.
68
69 • SELECT
70     pizza_types.name, SUM(order_details.quantity) AS quantity
71 FROM
72     pizza_types
73 JOIN
74     pizzas
75 ON
76     pizza_types.pizza_type_id = pizzas.pizza_type_id
77 JOIN
78     order_details
79 ON
80     order_details.pizza_id = pizzas.pizza_id
81 GROUP BY
82     pizza_types.name
83 ORDER BY
84     quantity DESC
85 LIMIT 5;
86
```

Result Grid

 Filter Rows:

Export: 

Wrap Cell Content: 

Fetch rows: 

	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

```
87  -- Join the necessary tables to find the total quantity of each pizza category ordered.
88
89  • SELECT
90      pizza_types.category,
91      SUM(order_details.quantity) AS quantity
92  FROM
93      pizza_types
94  JOIN
95      pizzas
96  ON
97      pizza_types.pizza_type_id = pizzas.pizza_type_id
98  JOIN
99      order_details
100  ON
101      order_details.pizza_id = pizzas.pizza_id
102  GROUP BY
103      pizza_types.category
104  ORDER BY
105      quantity DESC;
106
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Result Grid

```
107 -- Determine the distribution of orders by hour of the day.
108
109 • SELECT
110     HOUR(order_time) AS HOUR,
111     COUNT(order_id) AS order_count
112 FROM
113     orders
114 GROUP BY
115     HOUR(order_time);
116
```

Result Grid

 Filter Rows:





Export: 

Wrap Cell Content: 

	HOUR	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468







```
117 -- Join relevant tables to find the category-wise distribution of pizzas.
118
119 • SELECT
120     category,
121     COUNT(name)
122 FROM
123     pizza_types
124 GROUP BY
125     category;
```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

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






```
127 -- Group the orders by date and calculate the average number of pizzas ordered per day.
128
129 • SELECT
130     round(AVG(quantity), 0) AS avg_pizzas_ordered_per_day
131 FROM
132     (SELECT
133         orders.order_date,
134         SUM(order_details.quantity) AS quantity
135     FROM
136         orders
137     JOIN
138         order_details
139     ON
140         orders.order_id = order_details.order_id
141     GROUP BY
142         orders.order_date) AS order_quantity;
143
```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	avg_pizzas_ordered_per_day
▶	138

```
144 -- Determine the top 3 most ordered pizza types based on revenue.
```

```
145
146 • SELECT
147     pizza_types.name,
148     SUM(order_details.quantity * pizzas.price) AS revenue
149 FROM
150     pizza_types
151 JOIN
152     pizzas
153 ON
154     pizzas.pizza_type_id = pizza_types.pizza_type_id
155 JOIN
156     order_details
157 ON
158     order_details.pizza_id = pizzas.pizza_id
159 GROUP BY
160     pizza_types.name
161 ORDER BY
162     revenue DESC
163 LIMIT 3;
164
```



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

```

165  -- Calculate the percentage contribution of each pizza type to total revenue.
166
167  • SELECT
168      pizza_types.category,
169      ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
170          ROUND(SUM(order_details.quantity * pizzas.price),
171              2) AS total_sales
172          FROM
173              order_details
174              JOIN
175                  pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
176          2) AS revenue
177  FROM
178      pizza_types
179      JOIN
180          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
181      JOIN
182          order_details ON order_details.pizza_id = pizzas.pizza_id
183  GROUP BY pizza_types.category
184  ORDER BY revenue DESC;
185

```





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

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

```

186 -- Analyze the cumulative revenue generated over time.
187
188 • SELECT
189     order_date,
190     SUM(revenue) OVER(ORDER BY order_date) AS cum_revenue
191 FROM
192     (SELECT
193         orders.order_date,
194         SUM(order_details.quantity * pizzas.price) AS revenue
195     FROM
196         order_details
197     JOIN
198         pizzas
199     ON
200         order_details.pizza_id = pizzas.pizza_id
201     JOIN
202         orders
203     ON
204         orders.order_id = order_details.order_id
205     GROUP BY
206         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

Result 12 x


Result Grid

Read Only

```

208 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
209 • SELECT
210     name, revenue
211 FROM
212     (SELECT
213         category, name, revenue,
214         RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rn
215     FROM
216         (SELECT
217             pizza_types.category, pizza_types.name,
218             SUM((order_details.quantity) * pizzas.price) AS revenue
219         FROM
220             pizza_types
221         JOIN
222             pizzas
223         ON
224             pizza_types.pizza_type_id = pizzas.pizza_type_id
225         JOIN
226             order_details
227         ON
228             order_details.pizza_id = pizzas.pizza_id
229         GROUP BY
230             pizza_types.category, pizza_types.name) AS a) AS b
231 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

Result 13 x

 Result Grid



 Read Only