



SQL Projects

Data Analysis



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

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

	hour	orders
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2000

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

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

Average_pizza_ordered_perday
138.57

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

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	category	Distributed_pizzas
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

```

1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2
3
4  • select category, name, revenue , rank_r
5  from
6  (select category,name,revenue,rank() over(partition by category order by revenue desc) as rank_r
7  from
8  (select pizza_types.category, pizza_types.name, sum(order_details.quantity * pizzas.price) as revenue
9  from order_details
10 join pizzas
11 on pizzas.pizza_id = order_details.pizza_id
12 join pizza_types
13 on pizza_types.pizza_type_id = pizzas.pizza_type_id
14 group by pizza_types.category, pizza_types.name) as a) as b
15 where rank_r <= 3
16

```

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

	category	name	revenue	rank_r
►	Chicken	The Barbecue Chicken Pizza	22756.25	1
	Chicken	The Thai Chicken Pizza	21638.5	2
	Chicken	The California Chicken Pizza	21303.25	3
	Classic	The Classic Deluxe Pizza	19062.5	1
	Classic	The Hawaiian Pizza	16354.75	2
	Classic	The Pepperoni Pizza	15401.5	3

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

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:



	name	price
▶	The Greek Pizza	35.95

```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2
3
4  • select category, name, revenue , rank_r
5  from
6  (select category,name,revenue,rank() over(partition by category order by revenue desc) as rank_r
7  from
8  (select pizza_types.category, pizza_types.name, sum(order_details.quantity * pizzas.price) as revenue
9  from order_details
10 join pizzas
11 on pizzas.pizza_id = order_details.pizza_id
12 join pizza_types
13 on pizza_types.pizza_type_id = pizzas.pizza_type_id
14 group by pizza_types.category, pizza_types.name) as a) as b
15 where rank_r <= 3
16
```

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

	category	name	revenue	rank_r
→	Chicken	The Barbecue Chicken Pizza	22756.25	1
	Chicken	The Thai Chicken Pizza	21638.5	2
	Chicken	The California Chicken Pizza	21303.25	3
	Classic	The Classic Deluxe Pizza	19062.5	1
	Classic	The Hawaiian Pizza	16354.75	2
	Classic	The Pepperoni Pizza	15401.5	3

Result 4 v

```
1  -- Identify the most common pizza size ordered.
2
3  • SELECT
4      pi.size, COUNT(od.order_details_id)
5  FROM
6      pizzas AS pi
7      JOIN
8      order_details AS od ON pi.pizza_id = od.pizza_id
9  GROUP BY pi.size
10 ORDER BY COUNT(od.order_details_id) DESC
11 LIMIT 1;
12
13  -- why we count order_details here?
```



Result Grid			 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 	Fetch rows: 
	size	count(od.order_details_id)				
	L	9554				


```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2
3
4  • SELECT
5  ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
6  ROUND(SUM(order_details.quantity * pizzas.price),
7  1) AS reveue
8  FROM
9  order_details
10 JOIN
11 pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
12 2) AS revenue,
13 pizza_types.category
14 FROM
15 order_details
16 JOIN
17 pizzas ON pizzas.pizza_id = order_details.pizza_id
18 JOIN
19 pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
20 GROUP BY pizza_types.category
21 ORDER BY revenue DESC
22
23
24
```

```

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

```

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

	order_date	cummulative_revenue
▶	2015-01-01	2713.9
	2015-01-02	5445.8
	2015-01-03	8108.200000000001
	2015-01-04	9863.7
	2015-01-05	11929.7
	2015-01-06	14358.7
	2015-01-07	16560.9

```

1  -- List the top 5 most ordered pizza types along with their quantities.
2
3  •  SELECT
4      pizza_types.name, SUM(od.quantity) AS quantity
5  FROM
6      pizzas AS pi
7      JOIN
8      order_details AS od ON od.pizza_id = pi.pizza_id
9      JOIN
10     pizza_types ON pizza_types.pizza_type_id = pi.pizza_type_id
11  GROUP BY pizza_types.name
12  ORDER BY SUM(od.quantity) DESC
13  LIMIT 5
14
15

```

Result Grid			Filter Rows:	Exports	Wrap Cell Contents	Fetch rows:
	name	quantity				
▶	The Barbecue Chicken Pizza	1291				
	The Pepperoni Pizza	1233				
	The Classic Deluxe Pizza	1227				
	The Hawaiian Pizza	1226				
	The California Chicken Pizza	1215				

```


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

```

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

	total_quantity	category
▶	7602	Classic
	6195	Supreme
	6050	Veggie
	5649	Chicken

```
1  -- Calculate the total revenue generated from pizza sales.
2
3  ●  SELECT
4  ●  ROUND(SUM(order_details.quantity * pizzas.price),
5  ●      1) AS reveue
6  FROM
7      order_details
8      JOIN
9      pizzas ON pizzas.pizza_id = order_details.pizza_id
10
```

Result Grid		 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	revenue			
▶	421121.7			

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

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

	Order_placed
▶	21350