

SQL Pizza Sales Analysis Project

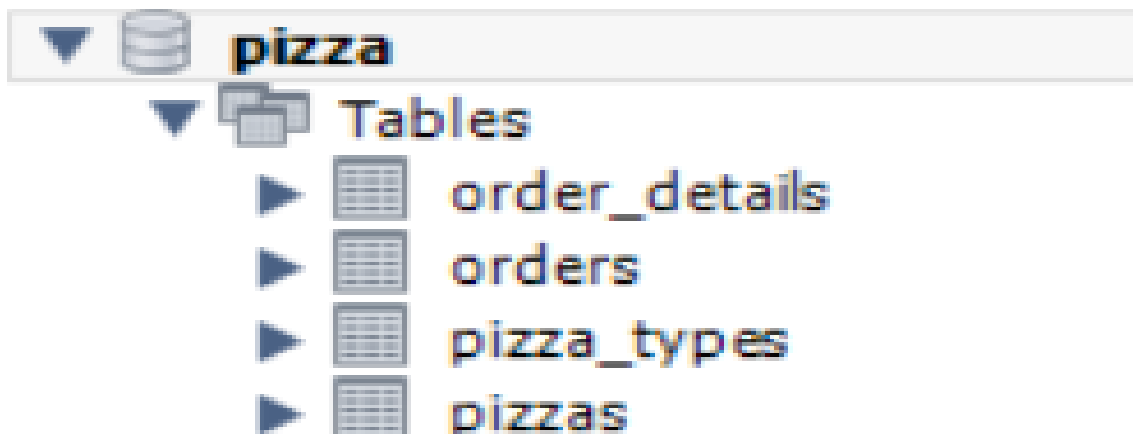
```
CREATE DATABASE pizza;
```

```
USE pizza;
```

```
CREATE TABLE orders(  
  order_id int not null,  
  order_date date not null,  
  order_time time not null,  
  primary key(order_id));
```

```
CREATE TABLE order_details(  
  order_details_id int not null,  
  order_id int not null,  
  pizza_id text not null,  
  quantity int not null,  
  primary key(order_details_id));
```

Here I am creating one pizza name database. And in this database I am adding 4 tables – order_details, orders, pizzas, pizza_type.



- Below are the 13 questions that I analyze through SQL.

```

21  -- Retrieve the total number of orders placed.
22  •  SELECT
23      COUNT(order_id) AS total_orders
24  FROM
25      orders;

```



Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	total_orders
▶	21350

```

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

```


Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	Total_revenue
▶	55113.1

```

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

```





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

	name	price
▶	The Greek Pizza	35.95

```

47  -- Identify the most common pizza size ordered.
48  • SELECT
49      pizzas.size,
50      COUNT(order_details.order_details_id) AS order_count
51  FROM
52      pizzas
53      JOIN
54      order_details ON pizzas.pizza_id = order_details.pizza_id
55  GROUP BY pizzas.size
56  ORDER BY order_count DESC;

```






Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	size	order_count
▶	L	1256
	M	1021
	S	971
	XL	40

```

59  -- List the top 5 most ordered pizza types along with their quantities.
60  • SELECT
61      pizza_types.name, SUM(order_details.quantity) AS Quantity
62  FROM
63      pizza_types
64      JOIN
65      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
66      JOIN
67      order_details ON order_details.pizza_id = pizzas.pizza_id
68  GROUP BY pizza_types.name
69  ORDER BY Quantity DESC
70  LIMIT 5;
71

```

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

	name	Quantity
▶	The Pepperoni Pizza	195
	The Thai Chicken Pizza	164
	The Barbecue Chicken Pizza	162
	The California Chicken Pizza	161
	The Classic Deluxe Pizza	152

```

72  -- Join the necessary tables to find the total quantity of each pizza category ordered.
73  • SELECT
74      pizza_types.category,
75      SUM(order_details.quantity) AS Quantity
76  FROM
77      pizza_types
78      JOIN
79      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
80      JOIN
81      order_details ON order_details.pizza_id = pizzas.pizza_id
82  GROUP BY pizza_types.category
83  ORDER BY Quantity DESC;
84

```

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

	category	Quantity
▶	Classic	992
	Supreme	825
	Veggie	805
	Chicken	731

```

85  -- Determine the distribution of orders by hour of the day.
86  • SELECT
87      HOUR(order_time) AS hour, COUNT(order_id) AS order_count
88  FROM
89      orders
90  GROUP BY HOUR(order_time);

```

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

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

```

92  -- Join relevant tables to find the category-wise distribution of pizzas.
93  • SELECT
94      category, COUNT(name)
95  FROM
96      pizza_types
97  GROUP BY category;
98
99

```

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

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

```

100 -- Group the orders by date and calculate the average number of pizzas ordered per day.
101 • SELECT
102     ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
103 FROM
104     (SELECT
105         orders.order_date, SUM(order_details.quantity) AS Quantity
106     FROM
107         orders
108     JOIN order_details ON orders.order_id = order_details.order_id
109     GROUP BY orders.order_date) AS order_quantity;
110
111

```

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

avg_pizza_ordered_per_day
134

```

112 -- Determin the top 3 most ordered pizza types based on revenue.
113 • SELECT
114     pizza_types.name,
115     SUM(order_details.quantity * pizzas.price) AS revenue
116 FROM
117     pizza_types
118     JOIN
119     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
120     JOIN
121     order_details ON order_details.pizza_id = pizzas.pizza_id
122 GROUP BY pizza_types.name
123 ORDER BY revenue DESC
124 LIMIT 3;
125
126

```


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

name	revenue
The Thai Chicken Pizza	3003
The Barbecue Chicken Pizza	2841.5
The California Chicken Pizza	2756.75

```

127 -- Calculate the percentage contribution of each pizza type to total revenue
128 • SELECT
129     pizza_types.category,
130     ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
131         ROUND(SUM(order_details.quantity * pizzas.price),
132             2) AS Total_sales
133     FROM
134         order_details
135     JOIN
136         pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
137     2) AS revenue
138 FROM
139     pizza_types
140     JOIN
141     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
142     JOIN
143     order_details ON order_details.pizza_id = pizzas.pizza_id
144 GROUP BY pizza_types.category
145 ORDER BY revenue DESC;

```

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

	category	revenue
▶	Classic	26.48
	Supreme	25.64
	Veggie	24.5
	Chicken	23.38

```

147 -- Analyze the cumulative revenue generated over time.
148 • SELECT order_date,
149     SUM(revenue) over(order by order_date) AS CUM_revenue
150 FROM
151     (SELECT orders.order_date,
152         SUM(order_details.quantity * pizzas.price) AS revenue
153     FROM order_details JOIN pizzas
154     ON order_details.pizza_id = pizzas.pizza_id
155     JOIN orders
156     ON orders.order_id = order_details.order_id
157     GROUP BY orders.order_date) AS sales;
158

```




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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.350000000002

```

159 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
160
161 • SELECT name, revenue
162 FROM (
163     SELECT category, name, revenue,
164            RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rn
165     FROM (
166         SELECT pizza_types.category,
167                pizza_types.name,
168                SUM(order_details.quantity * pizzas.price) AS revenue
169         FROM pizza_types
170         JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
171         JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
172         GROUP BY pizza_types.category, pizza_types.name
173     ) AS a
174 ) AS b
175 WHERE rn <= 3;
176
177

```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	name	revenue
▶	The Thai Chicken Pizza	3003
	The Barbecue Chicken Pizza	2841.5
	The California Chicken Pizza	2756.75
	The Pepperoni Pizza	2454
	The Classic Deluxe Pizza	2367
	The Greek Pizza	1946
	The Italian Supreme Pizza	2516
	The Sicilian Pizza	2313.5
	The Spicy Italian Pizza	2272.5
	The Five Cheese Pizza	2164.5