

SQl PROJECT FOR PIZZA SALES

- Revenue
- Orders
- Busiest Hours
- And More



SCHEMA

24 • `SELECT * FROM pizzas;`

pizza_id	pizza_type_id	size	price
bbq_dkn_s	bbq_dkn	S	12.75
bbq_dkn_m	bbq_dkn	M	16.75
bbq_dkn_l	bbq_dkn	L	20.75

26 • `SELECT * FROM order_details;`

order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	dassic_dlx_m	1
3	2	five_cheese_l	1

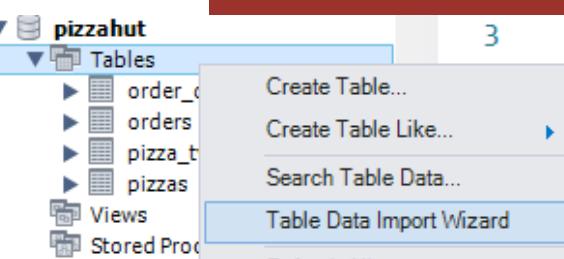
`SELECT * FROM orders;`

order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40

25 • `SELECT * FROM pizza_types;`

pizza_type_id	name	category	ingredients
bbq_dkn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Pepe...
cali_dkn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...

```
1 • CREATE DATABASE pizzahut;
2 • USE pizzahut;
3
4 • CREATE TABLE orders(
5     order_id INT NOT NULL,
6     order_date DATE NOT NULL,
7     order_time TIME NOT NULL,
8     PRIMARY KEY(order_id)
9 );
10
11 • CREATE TABLE order_details(
12     order_details_id INT NOT NULL,
13     order_id INT NOT NULL,
14     pizza_id TEXT NOT NULL,
15     quantity INT NOT NULL,
16     PRIMARY KEY(order_details_id)
17 );
18
19 • LOAD DATA INFILE 'D:/MySQL Learn/pizza_sales.csv'
20     INTO TABLE order_details
21     FIELDS TERMINATED BY ','
```



INTRODUCTION

Highlights:

- #13 identifies the top 3 pizzas by revenues in each category.**
- #12 Cumulative Revenue Generated Over Time**
- #9 Average pizza ordered every day.**
- #7 Busiest hours of the day.**
- #5 Top 5 Most ordered pizzas, their size, quantity, and revenue.**

From a sample dataset of a pizza store, we'll be using MySQL to query through data and find important business factors such as those as listed in the highlights.

1. Total numbers of orders placed.

```
36 -- 1. Total numbers of orders placed
37
38 • SELECT COUNT(order_id) AS total_orders FROM orders;
39
```

total_orders
21350

2. Total revenue from pizza sales.

```
42 • SELECT
43     ROUND(SUM(order_details.quantity * pizzas.price),
44             2) AS total_sales
45 FROM
46     order_details
47     JOIN
48         pizzas ON order_details.pizza_id = pizzas.pizza_id;
49
```

Result Grid	
total_sales	
▶	817860.05

3. Highest priced pizza. The Name of highest priced pizza.

```
52  -- 3. Highest priced pizza. The Name of highest priced pizza.  
53  
54 • SELECT  
55     t.name, p.price  
56 FROM  
57     pizza_types t  
58     JOIN  
59     pizzas p ON t.pizza_type_id = p.pizza_type_id  
60 HAVING MAX(price);  
61
```

Result Grid	
name	price
The Barbecue Chicken Pizza	12.75

4. Most common pizza size ordered.

```
90 • SELECT
91      p.size, COUNT(o.order_details_id) AS order_count
92  FROM
93      pizzas p
94      JOIN
95          order_details o ON p.pizza_id = o.pizza_id
96  GROUP BY p.size
97  ORDER BY order_count DESC;
```

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

5. List of top 5 most ordered pizza, their types, and quantities.

```
105 •  SELECT
106      pizza_types.name,
107      SUM(o.quantity) AS total_quantity_ordered
108  FROM
109      pizza_types
110  JOIN pizzas p
111      ON pizza_types.pizza_type_id = p.pizza_type_id
112  JOIN order_details o
113      ON o.pizza_id = p.pizza_id
114  GROUP BY pizza_types.name
115  ORDER BY total_quantity_ordered DESC
116  LIMIT 5;
```

Result Grid		Filter Rows:	Exp
	name	total_quantity_ordered	
▶	The Classic Deluxe Pizza	2453	
	The Barbecue Chicken Pizza	2432	
	The Hawaiian Pizza	2422	
	The Pepperoni Pizza	2418	
	The Thai Chicken Pizza	2371	

6. Total quantity of each pizza category ordered.

```
121 • SELECT
122     p.category, SUM(o.quantity) AS TotalQuantity
123     FROM
124         pizza_types p
125     JOIN pizzas
126         ON p.pizza_type_id = pizzas.pizza_type_id
127     JOIN order_details o
128         ON o.pizza_id = pizzas.pizza_id
129     GROUP BY p.category
130     ORDER BY TotalQuantity DESC;
```

	category	TotalQuantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

7. Distribution of orders by hour of the day.

Helps determine at what time of the day there are most pizza orders.

```
135 • SELECT  
136     HOUR(order_time) AS DayHour, COUNT(order_ID) AS OrderCount  
137     FROM  
138     orders  
139     GROUP BY HOUR(order_time)  
140     ORDER BY OrderCount DESC;
```

DayHour	OrderCount
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1

DayHour	OrderCount
9	1
10	8
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

8. Category wise distribution of pizzas.

Helps answer the number of pizzas in each category.

```
145 • SELECT  
146     category, COUNT(category) AS PizzasInCategory  
147     FROM  
148     pizza_types  
149     GROUP BY category  
150     ORDER BY PizzasInCategory DESC;
```

	category	PizzasInCategory
▶	Supreme	9
▶	Veggie	9
▶	Classic	8
▶	Chicken	6

9. Average number of pizzas ordered every day.

```
155 • SELECT
156     ROUND(AVG(quantity),0) AS OrdersPerDay
157     FROM
158     (SELECT
159         o.order_date, SUM(d.quantity) AS quantity
160     FROM
161     orders o
162     JOIN order_details d ON o.order_id = d.order_id
163     GROUP BY o.order_date) AS order_quantity;
```

Result Grid	
OrdersPerDay	
▶	138

```
158 (SELECT
159     o.order_date, SUM(d.quantity) AS quantity
160     FROM
161     orders o
162     JOIN order_details d ON o.order_id = d.order_id
163     GROUP BY o.order_date) AS order_quantity;
```

Result Grid	
order_date	quantity
2015-01-01	162
2015-01-02	165
2015-01-03	158
2015-01-04	106
2015-01-05	125
2015-01-06	147
2015-01-07	138
2015-01-08	173

10. Top 3 most ordered pizza types based on revenue.

```
167 • SELECT
168     t.name, SUM(p.price * d.quantity) AS revenue
169 FROM
170     pizza_types t
171     JOIN
172     pizzas p ON t.pizza_type_id = p.pizza_type_id
173     JOIN
174     order_details d ON d.pizza_id = p.pizza_id
175 GROUP BY t.name
176 ORDER BY revenue DESC
177 LIMIT 3;
```

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

11. Percentage contribution of pizza category in total revenue.

```
180 •   SELECT
181     t.category,
182     ROUND(100 * SUM(p.price * d.quantity) / (SELECT
183         SUM(order_details.quantity * pizzas.price) AS total_sales
184     FROM
185         order_details
186     JOIN pizzas
187         ON order_details.pizza_id = pizzas.pizza_id),2) AS revenue
188     FROM pizza_types t
189     JOIN pizzas p
190         ON t.pizza_type_id = p.pizza_type_id
191     JOIN order_details d
192         ON d.pizza_id = p.pizza_id
193     GROUP BY t.category
194     ORDER BY revenue DESC;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

12. Cumulative Revenue generated over time.

```
197 • SELECT order_date, revenue, ROUND(SUM(revenue) OVER (ORDER BY order_date),2) AS CumulativeRevenue  
198   FROM  
199   (SELECT  
200     o.order_date, ROUND(SUM(d.quantity * p.price),2) AS revenue  
201   FROM orders o  
202   JOIN order_details d  
203   ON o.order_id = d.order_id  
204   JOIN pizzas p  
205   ON p.pizza_id = d.pizza_id  
206   GROUP BY o.order_date) as sales;
```

	order_date	revenue	CumulativeRevenue
▶	2015-01-01	2713.85	2713.85
	2015-01-02	2731.9	5445.75
	2015-01-03	2662.4	8108.15
	2015-01-04	1755.45	9863.6
	2015-01-05	2065.95	11929.55
	2015-01-06	2428.95	14358.5
	2015-01-07	2202.2	16560.7
	2015-01-08	2838.35	19399.05
	2015-01-09	2127.35	21526.4
	2015-01-10	2463.95	23990.35
	2015-01-11	1872.3	25862.65
	2015-01-12	1919.05	27781.7
	2015-01-13	2049.6	29831.3

13. Top 3 most ordered pizza types, based on revenue, for each pizza category.

```
210 • SELECT name, category, ROUND(TotalPizzaRevenue,2) AS PizzaRevenue
211   FROM
212   (SELECT name, category, TotalPizzaRevenue, rank() OVER (PARTITION BY category ORDER BY TotalPizzaRevenue DESC) as rnk
213     FROM
214     (SELECT
215       t.name, t.category, SUM(d.quantity * p.price) AS TotalPizzaRevenue
216     FROM pizza_types t
217     JOIN pizzas p
218       ON p.pizza_type_id = t.pizza_type_id
219     JOIN order details d
220       ON d.pizza_id = p.pizza_id
221     GROUP BY t.category, t.name
222     ORDER BY t.category, TotalPizzaRevenue DESC) as RevenueTable) AS RANKTABLE
223   WHERE rnk <=3
224   ORDER BY rnk DESC;
```

	name	category	PizzaRevenue
▶	The California Chicken Pizza	Chicken	41409.5
	The Pepperoni Pizza	Classic	30161.75
	The Sicilian Pizza	Supreme	30940.5
	The Five Cheese Pizza	Veggie	26066.5
	The Barbecue Chicken Pizza	Chicken	42768
	The Hawaiian Pizza	Classic	32273.25
	The Italian Supreme Pizza	Supreme	33476.75
	The Mexicana Pizza	Veggie	26780.75
	The Thai Chicken Pizza	Chicken	43434.25
	The Classic Deluxe Pizza	Classic	38180.5
	The Spicy Italian Pizza	Supreme	34831.25
	The Four Cheese Pizza	Veggie	32265.7

A festive illustration set against a red background with a white grid. In the center, the words "THANK YOU" are written in large, bold, white capital letters. Below this, a smaller white text reads: "Feel free to reach out if you have any feedback!" To the left, a person with dark curly hair, wearing a green shirt, holds a slice of pizza with three holes. To the right, another person with glasses and a green shirt holds a piece of yellow cheese. Above them, a reindeer with a gold bell around its neck (bearing the letter 'Q') walks across the scene. The background is decorated with yellow stars and a large yellow sunburst at the bottom.

THANK YOU

Feel free to reach out if you have
any feedback!