

Cheese Please Pizza Joint brings you

PIZZA ALL YOU CAN FRIDAYS

Buy One, Take One!
Twice the Goodness, Twice the Fun!

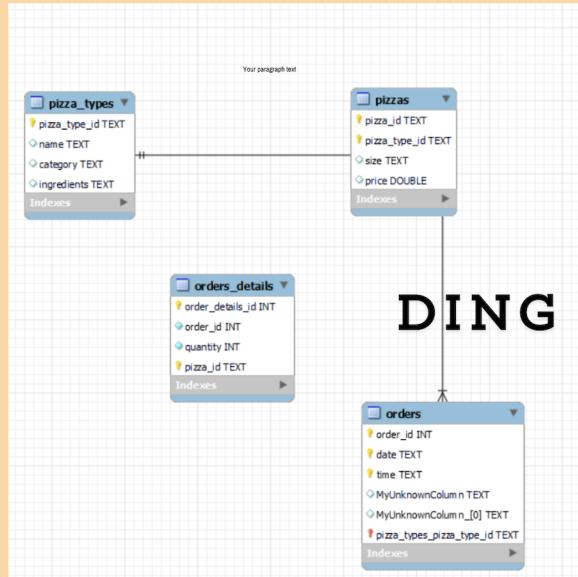
Promo runs on all Fridays of March.
For more information, visit
www.reallygreatsite.com



hello

Hello, I'm Vidushi Sharma, and I'm thrilled to introduce myself as a member of the Pizza Hut SQL project . With a passion for data analysis and a love for pizza, I bring a unique blend of technical expertise and culinary enthusiasm to the table. Throughout this project, I'll be leveraging my skills in SQL to uncover valuable insights that will drive Pizza Hut's sales strategies and enhance customer satisfaction.

ER DIAGRAM



-- Retreieve the total number of orders placed--

```
SELECT |  
        COUNT(order_id) AS total_sales  
FROM  
      orders;
```

Result Grid	
	total_sales
▶	21350

-- calculate the total revenue generated from pizza sales--

```
SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
        2) AS total_sales
FROM
    orders_details
    JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid	
	total_sales
▶	817860.05

– Identify the highest priced pizzas–

```
Select pizza_types.name, sum(orders_details.quantity) as quantity
from pizza_types
join
pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join
orders_details on orders_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by quantity
DESC limit 5;
```

Result Grid	
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

– join the necessary tables to find the total quantity of each pizza category ordered–

```
• SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

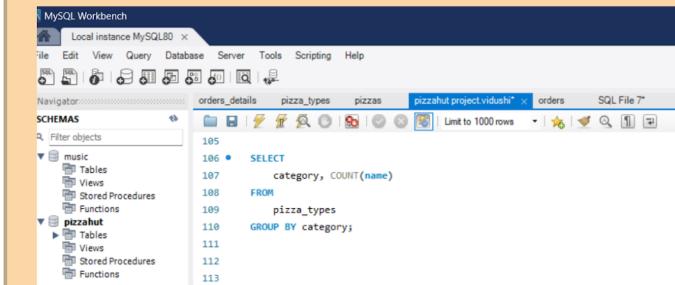
Result Grid		
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

– determine the distribution of orders by hour of the day –

```
• SELECT  
    HOUR(TIME) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(TIME);
```

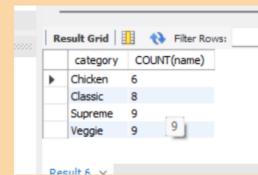
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
..	..
	Result 5

--join relevant tables to find the category-wise distribution of pizzas--



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu is a toolbar with various icons. The left sidebar, titled 'Navigator', shows 'SCHEMAS' expanded, revealing 'music' and 'pizzahut' schemas, each containing Tables, Views, Stored Procedures, and Functions. The main area displays a SQL query in the 'SQL File 7*' tab:

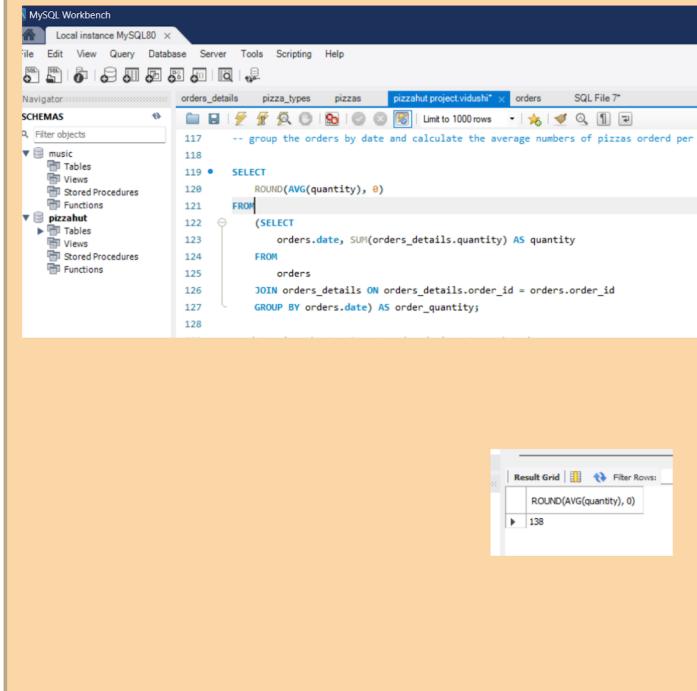
```
105
106 •   SELECT
107     category, COUNT(name)
108   FROM
109     pizza_types
110   GROUP BY category;
```



The screenshot shows the 'Result Grid' window from MySQL Workbench. It displays the output of the SQL query. The grid has two columns: 'category' and 'COUNT(name)'. The data is as follows:

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

- group the orders by date and calculate the average numbers of pizzas ordered per day-



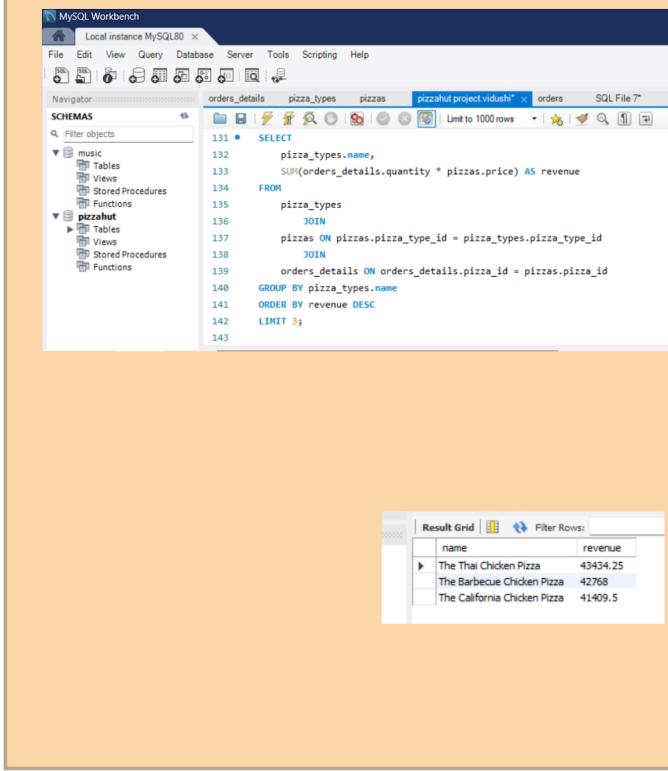
The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following query:

```
117 -- group the orders by date and calculate the average numbers of pizzas ordered per day
118
119 • SELECT
120     ROUND(AVG(quantity), 0)
121     FROM
122     (SELECT
123         orders.date, SUM(orders_details.quantity) AS quantity
124     FROM
125     orders
126     JOIN orders_details ON orders_details.order_id = orders.order_id
127     GROUP BY orders.date) AS order_quantity;
128
```

The Result Grid shows the output of the query:

ROUND(AVG(quantity), 0)
138

— determine the top 3 most ordered pizza types based on revenue—



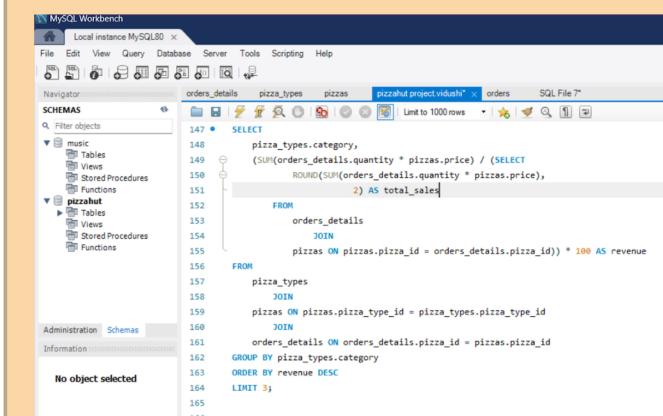
The screenshot shows the MySQL Workbench interface. The left pane displays the Navigator with two schemas: 'music' and 'pizzahut'. The 'pizzahut' schema is expanded, showing tables like 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The right pane contains a SQL editor window with the following query:

```
131 • SELECT
132     pizza_types.name,
133     SUM(orders_details.quantity * pizzas.price) AS revenue
134   FROM
135     pizza_types
136   JOIN
137     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
138   JOIN
139     orders_details ON orders_details.pizza_id = pizzas.pizza_id
140 GROUP BY pizza_types.name
141 ORDER BY revenue DESC
142 LIMIT 3;
```

Below the SQL editor is a 'Result Grid' window showing the results of the query:

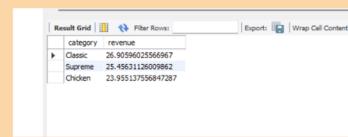
	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

-- calculate the percentage contribution of each pizza type of total revenue--



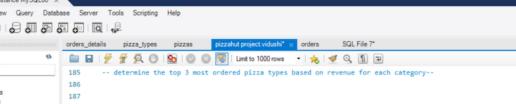
The screenshot shows the MySQL Workbench interface with a query editor window. The query is as follows:

```
147 • SELECT
148     pizza_types.category,
149     (SUM(orders_details.quantity * pizzas.price) / (SELECT
150         ROUND(SUM(orders_details.quantity * pizzas.price),
151             2) AS total_sales))
152     FROM
153     orders_details
154     JOIN
155     pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100 AS revenue
156     FROM
157     pizza_types
158     JOIN
159     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
160     JOIN
161     orders_details ON orders_details.pizza_id = pizzas.pizza_id
162     GROUP BY pizza_types.category
163     ORDER BY revenue DESC
164     LIMIT 3;
```



category	revenue
Classic	26.90596025566967
Supreme	25.49511260099862
Chicken	23.935127556847287

-- determine the top 3 most ordered pizza types based on revenue for each category--



```
File Edit View Query Database Server Tools Scripting Help
orders_details pizza_types pizzas plazahut project.violations orders SQL File ?*
185 -- determine the top 3 most ordered pizza types based on revenue for each category...
186
187
188 *+ select name,revenue from(select category,name,revenue,
189 rank() over (partition by category order by revenue desc) as rn
190 from
191 (select pizza_types.category,pizza_types.name,
192 sum(orders_details.quantity * pizzas.price)*revenue
193 from pizza_types join pizzas
194 on pizzas.pizza_type_id = pizza_types.pizza_type_id
195 join orders_details
196 on orders_details.pizza_id = pizzas.pizza_id
197 group by pizza_types.category,pizza_types.name) as a) as b
198 where rn=1
```

Result Grid		Filtr Rows:	Exports	Wrap Cell Content:
	name	revenue		
▶	The California Chicken Pizza	41490.5		
	The Southwest Chicken Pizza	34075.75		
	The Chicken Alfredo Pizza	16900.25		
	The Chicken Pesto Pizza	16711.75		
	The Pepperoni Pizza	30131.15		
	The Greek Pizza	28456.000000000013		
	The Italian Capuccino Pizza	25094		
	The Napolitana Pizza	24087		
	▶ The Big Megal Strips	17648		



THANK YOU!