



# AAKASH

P O R T F O L I O   P R O J E C T

*Pizza Data Analysis Project  
using SQL*





AAKASH

PORTFOLIO PROJECT

# ABOUT ME

Data & Business Intelligence Analyst with over a year of experience at Amdocs, specializing in data warehousing and BI Engineering. Successfully migrated critical projects from Teradata to GCP, ensuring 100% data integrity.

Proficient in SQL, Excel, Big Query, and GCP for comprehensive data analysis, enhancing data and code accuracy by 80%. Leveraged AI tools like Copilot for automation, reducing time and effort by 50%. Certified in Data Analysis and Generative AI by Microsoft and LinkedIn.

Skilled in Power BI, Python, and Apache Airflow DAGs. Looking to leverage my data analysis skills and business acumen to drive data-driven decision making.





AAKASH

P O R T F O L I O P R O J E C T

# INTRODUCTION

Hi Everyone,

Thanks for taking out time to review my portfolio project.

Any suggestion and feedbacks are welcomed.

Feel free to contact and connect with me for any discussions.

Contact Details End of Slides.

Thank you!





AAKASH

PORTFOLIO PROJECT

Retrieve the total number of orders placed.

```
1   -- Retrieve the total number of orders placed.  
2 •  SELECT  
3     COUNT(*)  
4 FROM  
5   orders;
```

The screenshot shows a MySQL query editor interface. At the top, there is a code editor window containing the SQL query. Below the code editor is a toolbar with several buttons: 'Result Grid' (selected), 'Filter Rows:', 'Export:', and 'Wrap Cell Content:'. The main area displays the results of the query in a grid format. The grid has two columns: the first column contains an arrow pointing right, and the second column contains the value '21350'. The entire interface is set against a dark background.

	COUNT(*)
▶	21350



## Calculate the total revenue generated from pizza sales.

```
7      -- Calculate the total revenue generated from pizza sale
8 •   SELECT
9     ROUND(SUM(order_details.quantity * pizzas.price),
10           2) AS revenue
11
12    FROM
13      order_details
14      JOIN
15        pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

The screenshot shows a MySQL query editor interface. At the top, there is a code editor window containing the provided SQL query. Below the code editor is a toolbar with several buttons: 'Result Grid' (selected), 'Filter Rows:', 'Export:' (with a file icon), and 'Wrap Cell Content'. The result grid below the toolbar displays a single row with two columns. The first column is labeled 'revenue' and the second column contains the value '817860.05'.

revenue	817860.05
---------	-----------



AAKASH

PORTFOLIO PROJECT

Identify the highest-priced pizza.

```
--  
16      -- Identify the highest-priced pizza  
17 •  SELECT  
18      pizza_types.name, pizzas.price  
19      FROM  
20      pizza_types  
21          JOIN  
22      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
23      ORDER BY pizzas.price DESC  
24      LIMIT 1;  
25  
  
Result Grid | Filter Rows: [ ] | Export: | Wrap Cell Content: | Fetch rows:  


|   | name            | price |
|---|-----------------|-------|
| ▶ | The Greek Pizza | 35.95 |


```



AAKASH

PORTFOLIO PROJECT

Identify the most common pizza size ordered.

```
26      -- Identify the most common pizza size ordered.  
27 •  SELECT  
28      pizzas.size,  
29      COUNT(order_details.order_details_id) AS order_count  
30  FROM  
31      order_details  
32          JOIN  
33      pizzas ON order_details.pizza_id = pizzas.pizza_id  
34  GROUP BY pizzas.size  
35  ORDER BY order_count DESC;
```

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query editor contains the SQL code provided above. The result grid displays the output of the query:

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



AAKASH

PORTFOLIO PROJECT

List the top 5 most ordered pizza types along with their quantities.

```
--  
37      -- List the top 5 most ordered pizza types along with their quantities.  
38 •  SELECT  
39      pizza_types.name,  
40      ROUND(SUM(order_details.quantity), 2) AS quantity  
41  FROM  
42      pizza_types  
43      JOIN  
44      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
45      JOIN  
46      order_details ON pizzas.pizza_id = order_details.pizza_id  
47  GROUP BY pizza_types.name  
48  ORDER BY quantity DESC  
49  LIMIT 5;
```

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



AAKASH

PORTFOLIO PROJECT

Join the necessary tables to find the total quantity of each pizza category ordered.

```
51      -- Join the necessary tables to find the total quantity of each pizza category ordered.  
52 •  SELECT  
53      pizza_types.category,  
54      ROUND(SUM(order_details.quantity), 0) AS quantity  
55  FROM  
56      pizza_types  
57      JOIN  
58      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
59      JOIN  
60      order_details ON pizzas.pizza_id = order_details.pizza_id  
61  GROUP BY pizza_types.category  
62  ORDER BY quantity DESC;
```

The screenshot shows a MySQL query editor interface with the following details:

- Toolbar:** Includes "Result Grid" (selected), "Filter Rows:" (with an input field), "Export:" (with a file icon), and "Wrap Cell Content:" (with a text icon).
- Result Grid:** A table displaying the results of the executed SQL query. The table has two columns: "category" and "quantity". The data is as follows:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



AAKASH

PORTFOLIO PROJECT

## Determine the distribution of orders by hour of the day.

```
64      -- Determine the distribution of orders by hour of the day.  
65  
66 •  SELECT  
67      HOUR(orders.time) AS hourOfDay,  
68      COUNT(order_details.order_details_id) AS orderByDay  
69  FROM  
70      orders  
71      JOIN  
72      order_details ON orders.order_id = order_details.order_id  
73  GROUP BY hourOfDay  
74  ORDER BY orderByDay DESC;
```

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

	hourOfDay	orderByDay
▶	12	6543
	13	6203
	18	5359
	17	5143
	19	4350
	16	4185
	14	3521
	20	3487
	15	3170
	11	2672
	21	2528
	22	1370
	23	68
	10	17
	9	4



AAKASH

PORTFOLIO PROJECT

Join relevant tables to find the category-wise distribution of pizzas.

```
76      -- Join relevant tables to find the category-wise distribution of pizzas.
77 •  SELECT
78      category, COUNT(category) AS countDistribution
79  FROM
80      pizza_types
81  GROUP BY category;
```

Result Grid | Filter Rows: [ ] | Export: | Wrap Cell Content: [A]

	category	countDistribution
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



AAKASH

PORTFOLIO PROJECT

Group the orders by date and calculate the average number of pizzas ordered per day.

```
82  
83      -- Group the orders by date and calculate the average number of pizzas ordered per day.  
84 •  SELECT  
85          AVG(sumOfQty) AS AvgOrdersPerDAY  
86      FROM  
87          (SELECT  
88              (orders.date) AS orderdate,  
89                  ROUND(SUM((order_details.quantity)), 0) AS sumOfQty  
90              FROM  
91                  orders  
92              JOIN order_details ON orders.order_id = order_details.order_id  
93              GROUP BY orderdate) AS salesData;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	AvgOrdersPerDAY			
▶	138.4749			



AAKASH

PORTFOLIO PROJECT

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

```
95      -- Determine the top 3 most ordered pizza types based on revenue.  
96  •  SELECT  
97      pizza_types.name,  
98      SUM((pizzas.price * order_details.quantity)) AS revenue  
99  FROM  
100     order_details  
101        JOIN  
102         pizzas ON order_details.pizza_id = pizzas.pizza_id  
103        JOIN  
104         pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
105    GROUP BY pizza_types.name  
106    ORDER BY revenue DESC  
107    LIMIT 3;  
108
```

< [REDACTED]

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

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



AAKASH

PORTFOLIO PROJECT

Calculate the percentage contribution of each pizza type to total revenue.

```
109      -- Calculate the percentage contribution of each pizza type to total revenue.
110 •  SELECT
111      pizza_types.category,
112      round(SUM(pizzas.price * order_details.quantity / (SELECT
113          SUM(pizzas.price * order_details.quantity) AS TOTALSALES
114      FROM
115          order_details
116          JOIN
117              pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100),2) AS revenue
118  FROM
119      pizzas
120          JOIN
121              order_details ON order_details.pizza_id = pizzas.pizza_id
122          JOIN
123              pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
124      GROUP BY pizza_types.category
125      order by revenue desc;
```

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

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



AAKASH

PORTFOLIO PROJECT

## Analyze the cumulative revenue generated over time.

```
128      -- Analyze the cumulative revenue generated over time.  
129 •   select order_date, revenue ,  
130     round(sum(revenue) over (order by order_date),2) as cummulative_revenue  
131   from  
132   (SELECT  
133     orders.date as order_date,  
134     round(SUM((pizzas.price * order_details.quantity)),2) AS revenue  
135   FROM  
136     order_details  
137       JOIN  
138       pizzas ON order_details.pizza_id = pizzas.pizza_id  
139       JOIN  
140       orders ON orders.order_id = order_details.order_id  
141   GROUP BY orders.date ) as sales ;
```

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

	order_date	revenue	cummulative_revenue
▶	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



AAKASH

PORTFOLIO PROJECT

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
144 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
145 • select
146 CATEGORY , revenue ,NAME,rn
147 from
148 ( select CATEGORY , revenue ,NAME,
149      RANK() OVER (partition by category ORDER BY revenue DESC) as rn
150      from
151      (SELECT
152          pizza_types.category as CATEGORY,
153          pizza_types.name as NAME,
154          SUM((pizzas.price * order_details.quantity)) AS revenue
155        FROM
156          order_details
157          JOIN
158          pizzas ON order_details.pizza_id = pizzas.pizza_id
159          JOIN
160          pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
161        GROUP BY CATEGORY , NAME
162        ORDER BY revenue DESC
163        ) as Total_Sales
164    ) as RankResults
165    where rn <= 3;
166
```

	CATEGORY	revenue	NAME	rn
▶	Chicken	43434.25	The Thai Chicken Pizza	1
	Chicken	42768	The Barbecue Chicken Pizza	2
	Chicken	41409.5	The California Chicken Pizza	3
	Classic	38180.5	The Classic Deluxe Pizza	1
	Classic	32273.25	The Hawaiian Pizza	2
	Classic	30161.75	The Pepperoni Pizza	3
	Supreme	34831.25	The Spicy Italian Pizza	1
	Supreme	33476.75	The Italian Supreme Pizza	2
	Supreme	30940.5	The Sicilian Pizza	3
	Veggie	32265.70000000065	The Four Cheese Pizza	1
	Veggie	26780.75	The Mexicana Pizza	2
	Veggie	26066.5	The Five Cheese Pizza	3

Note: Resource files and Code is uploaded on my GitHub. Link to GitHub in next slide



AAKASH

P O R T F O L I O P R O J E C T

# PLEASE CONNECT

Email ID: aakashbhatia.work24@gmail.com

LinkedIn : <https://www.linkedin.com/in/aakash-bhatia-5204a318a/>

GitHub: <https://github.com/aakash2404/Data-Business-Intelligence-Analyst>

Contact: 7448244151 / 8669544924





# AAKASH

P O R T F O L I O P R O J E C T

# THANK YOU

