PIZZA SALES ANALYSIS :

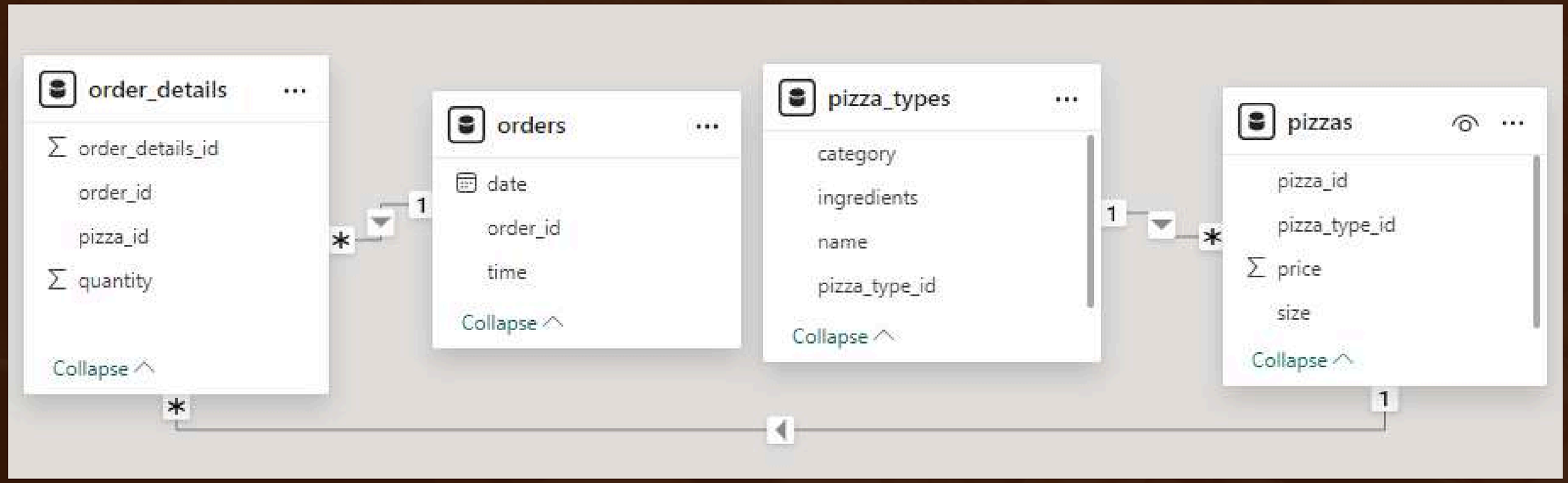
A SQL-BASED DATA ANALYSIS PROJECT

Hi, I'm Sabah Khan. In this project, I've used SQL queries to explore and answer key questions about pizza sales, demonstrating how data can provide valuable insights.

Start Your Slide

Database Structure Overview

This database is designed to track pizza sales, including details about orders, pizzas, and their categories. It provides the necessary structure to analyze sales performance and customer preferences effectively.



Total numbers of orders placed.

```
SELECT
    COUNT(ORDER_ID) AS TOTAL_ORDERS
FROM
    ORDERS
```

total_orders bigint
21350

Total revenue generated from pizza sales.

```
SELECT
    SUM(OD.QUANTITY * PIZZAS.PRICE) AS TOTAL_REVENUE
FROM
    ORDER_DETAILS AS OD
    JOIN PIZZAS ON OD.PIZZA_ID = PIZZAS.PIZZA_ID
```

Calculate the highest-priced pizza.

```sql
SELECT
    PT.PIZZA_NAME,
    PIZZAS.PRICE
FROM
    PIZZAS
    JOIN PIZZA_TYPES AS PT ON PIZZAS.PIZZA_TYPE_ID = PT.PIZZA_TYPE_ID
ORDER BY 2 DESC
LIMIT 1;
```

pizza_name character varying (50)	price numeric (5,2)
The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
SELECT
    P.SIZE,
    SUM(OD.QUANTITY) AS TOTAL_QUANTITY_ORDERED
FROM
    PIZZAS AS P
    JOIN ORDER_DETAILS AS OD ON P.PIZZA_ID = OD.PIZZA_ID
GROUP BY
    P.SIZE
ORDER BY
    TOTAL_QUANTITY_ORDERED DESC;
```

size character varying (3)	total_quantity_ordered bigint
L	18956
M	15635
S	14403
XL	552
XXL	28

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

```sql
SELECT
    PT.PIZZA_NAME,
    SUM(QUANTITY) AS ORDER_QUANTITY
FROM
    ORDER_DETAILS AS OD
    JOIN PIZZAS ON OD.PIZZA_ID = PIZZAS.PIZZA_ID
    JOIN PIZZA_TYPES AS PT ON PIZZAS.PIZZA_TYPE_ID = PT.PIZZA_TYPE_ID
GROUP BY
    PT.PIZZA_NAME
ORDER BY
    2 DESC
LIMIT
    5;
```

pizza_name character varying (50)	order_quantity bigint
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.

```sql
SELECT
    PT.CATEGORY,
    SUM(OD.QUANTITY) AS TOTAL_QUANTITY
FROM
    ORDER_DETAILS AS OD
    JOIN PIZZAS ON OD.PIZZA_ID = PIZZAS.PIZZA_ID
    JOIN PIZZA_TYPES AS PT ON PIZZAS.PIZZA_TYPE_ID = PT.PIZZA_TYPE_ID
GROUP BY
    1
ORDER BY
    2;
```

category character varying (50)	total_quantity bigint
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

Determine the distribution of orders by hour of the day.

```
SELECT
    EXTRACT(
        HOUR
        FROM
            TIME
    ) AS HOUR_OF_DAY,
    COUNT(ORDER_ID) AS ORDER_COUNT
FROM
    ORDERS
GROUP BY
    1
ORDER BY
    1;
```

Find the category-wise distribution of pizzas.

```
SELECT
    CATEGORY,
    COUNT(PIZZA_NAME) AS PIZZA
FROM
    PIZZA_TYPES
GROUP BY
    CATEGORY;
```

category character varying (50)	pizza bigint
Supreme	9
Classic	8
Veggie	9
Chicken	6

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

```sql
WITH
    ORDER_BY_DATE AS (
        SELECT
            ORDERS.DATE,
            SUM(OD.QUANTITY) AS ORDER_COUNT
        FROM
            ORDERS
            JOIN ORDER_DETAILS AS OD ON ORDERS.ORDER_ID = OD.ORDER_ID
        GROUP BY
            DATE
        ORDER BY
            DATE
    )
SELECT
    ROUND(AVG(ORDER_COUNT)) AS AVERAGE_ORDER_PER_DAY
FROM
    ORDER_BY_DATE;
```

avg_order_per_day numeric
138

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

```sql
SELECT
    PT.PIZZA_NAME,
    SUM(OD.QUANTITY * PIZZAS.PRICE) AS REVENUE
FROM
    ORDER_DETAILS AS OD
    JOIN PIZZAS ON OD.PIZZA_ID = PIZZAS.PIZZA_ID
    JOIN PIZZA_TYPES AS PT ON PIZZAS.PIZZA_TYPE_ID = PT.PIZZA_TYPE_ID
GROUP BY
    1
ORDER BY
    2 DESC
LIMIT
    3;
```

pizza_name character varying (50)	revenue numeric
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768.00
The California Chicken Pizza	41409.50

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

```sql
WITH TotalRevenue AS (
    SELECT SUM(od.quantity * p.price) AS total_revenue
    FROM order_details od
    JOIN pizzas p ON od.pizza_id = p.pizza_id
),
CategoryRevenue AS (
    SELECT pt.category, SUM(od.quantity * p.price) AS category_revenue
    FROM pizza_types pt
    JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN order_details od ON p.pizza_id = od.pizza_id
    GROUP BY pt.category
)
SELECT cr.category, ROUND((cr.category_revenue * 100.0 / tr.total_revenue), 2) AS percentage_revenue
FROM CategoryRevenue cr
CROSS JOIN TotalRevenue tr
ORDER BY percentage_revenue DESC;
```

category character varying (50)	percentage_revenue numeric
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```sql
WITH REVENUE_BY_DATE AS (
    SELECT ORDERS.DATE, SUM(OD.QUANTITY * PIZZAS.PRICE) AS REVENUE
    FROM ORDERS
    JOIN ORDER_DETAILS AS OD ON ORDERS.ORDER_ID = OD.ORDER_ID
    JOIN PIZZAS ON OD.PIZZA_ID = PIZZAS.PIZZA_ID
    GROUP BY 1
    ORDER BY 1
)
SELECT RD.DATE, SUM(REVENUE) OVER (ORDER BY DATE) AS CUM_REVENUE
FROM REVENUE_BY_DATE AS RD;
```

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

```sql
WITH CategoryRevenue AS (
    SELECT pt.category, pt.pizza_name, SUM(od.quantity * p.price) AS revenue,
    ROW_NUMBER() OVER (PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) as rn
    FROM pizza_types pt
    JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN order_details od ON p.pizza_id = od.pizza_id
    GROUP BY pt.category, pt.pizza_name
)
SELECT category, pizza_name, revenue
FROM CategoryRevenue
WHERE rn <= 3
ORDER BY category, revenue DESC;
```

category character varying (50)	pizza_name character varying (50)	revenue numeric
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768.00
Chicken	The California Chicken Pizza	41409.50
Classic	The Classic Deluxe Pizza	38180.50
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.50
Veggie	The Four Cheese Pizza	32265.70
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.50

INSIGHTS AND RECOMMENDATIONS:

1. Financial Performance: $817,860 total revenue from 21,350 orders.
2. High-Revenue Categories: "Classic" & "Supreme" drive significant revenue; focus marketing efforts here.
3. Peak Hours (12-2 PM): Optimize staffing during peak lunch hours for efficient service.
4. Popular Sizes (M & L): Prioritize Medium & Large pizza bundles/offers.
5. September Revenue Dip: Investigate & address the September revenue decline with targeted promotions.
6. Top Performers: Ensure consistent availability of ingredients for top-selling pizzas like "Classic Deluxe."
7. Customer Feedback: Regularly gather feedback for continuous improvement.

THANK YOU FOR ATTENTION