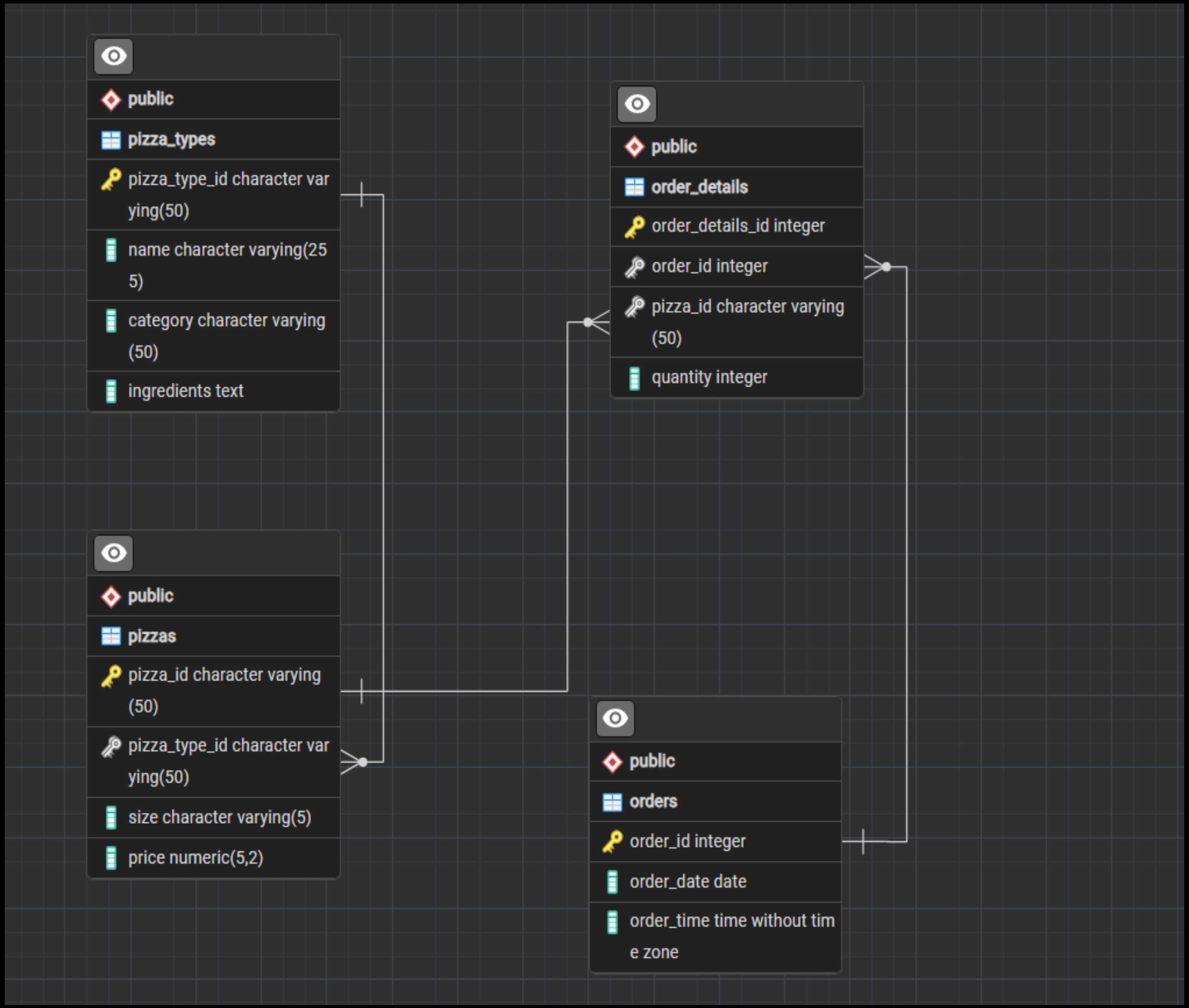
Introduction

Welcome to the Pizza Sales Analysis Project! This project aims to provide a comprehensive analysis of pizza sales data for a fictional pizza restaurant. Using SQL and data analytics techniques, I explored various aspects of sales trends, customer preferences, and revenue generation.

Project Highlights

•**Sales Trends**: Analyzed monthly sales patterns and cumulative revenue growth.
•**Customer Insights**: Identified peak sales time and popular pizza types.
•**Revenue Analysis**: Determined top-performing products driving the most revenue.

ER-Diagram Model



public
pizza_types
pizza_type_id character varying(50)
name character varying(255)
category character varying(50)
ingredients text

public
pizzas
pizza_id character varying(50)
pizza_type_id character varying(50)
size character varying(5)
price numeric(5,2)

public
order_details
order_details_id integer
order_id integer
pizza_id character varying(50)
quantity integer

public
orders
order_id integer
order_date date
order_time time without time zone

Q1. Retrieve the total no. of orders placed

Code

```
SELECT
        COUNT(ORDER_ID) AS "total orders"
FROM ORDERS;
```

Answer Set

```
“Total orders"
21350
```

Q.2 Calculate the total revenue generated from pizza sales.

Code

```
SELECT
        SUM(PIZZAS.PRICE * ORDER_DETAILS.QUANTITY) AS "Revenue“
FROM
        PIZZAS
JOIN ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID;
```

Answer Set

```
"Revenue"
817860.05
```

Q3. Identify the highest-priced pizza.

Code

```
SELECT
     NAME,
     MAX(PRICE) AS "Max price“
FROM
     PIZZAS
     JOIN PIZZA_TYPES ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.PIZZA_TYPE_ID
GROUP BY
     NAME
ORDER BY
     2 DESC
LIMIT
     1;
```

Answer Set

"name"	"Max price"
"The Greek Pizza"	35.95

Q4. Identify the most common pizza size ordered.

Code

```
SELECT
    PIZZAS.SIZE,
    COUNT(ORDER_DETAILS.ORDER_DETAILS_ID)
FROM
    PIZZAS
   JOIN ORDER_DETAILS ON PIZZAS.PIZZA_ID =
ORDER_DETAILS.PIZZA_ID
GROUP BY
   PIZZAS.SIZE
ORDER BY
   2 DESC;
```

Answer Set

"size"“	count"
"L"	18526
"M"	15385
"S"	14137
"XL"	544
"XXL“	28

Q5. List the top 5 most ordered pizza types along with their quantities

Code

```
SELECT
    SUM(ORDER_DETAILS.QUANTITY) AS "Total quantity",
    PIZZA_TYPES.name as "Pizza“
FROM
    PIZZAS
    JOIN PIZZA_TYPES ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
    JOIN ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
GROUP BY
   PIZZA_TYPES.name
ORDER BY
   1 DESC
limit 5;
```

Answer Set

```
"Total quantity" |          "Pizza"
2453             |   "The Classic Deluxe Pizza"
2432             |   "The Barbecue Chicken Pizza"
2422             |    "The Hawaiian Pizza"
2418             |    "The Pepperoni Pizza"
2371             |    "The Thai Chicken Pizza"
```

Q6. Join the necessary tables to find the total quantity of each pizza category

Code

```
SELECT
    SUM(ORDER_DETAILS.QUANTITY) AS "Total quantity",
    PIZZA_TYPES.CATEGORY
FROM
   PIZZAS
   JOIN PIZZA_TYPES ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
   JOIN ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
GROUP BY
  PIZZA_TYPES.CATEGORY
ORDER BY
  1 DESC;
```

Answer Set

"Total quantity“	"category"
14888	"Classic"
11987	"Supreme"
11649	"Veggie"
11050	"Chicken"

Q7. Determine the distribution of orders by hour of the day.

Code

```
SELECT
    COUNT(ORDER_DETAILS.ORDER_ID) AS "Count",
    EXTRACT(
        HOUR
        FROM
       ORDER_TIME
    ) AS "Order hour“
FROM
    ORDER_DETAILS
    JOIN ORDERS ON ORDER_DETAILS.ORDER_ID = ORDERS.ORDER_ID
GROUP BY
   "Order hour“
 ORDER BY
   "Count" DESC;
```

Answer Set

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

Q8. Join relevant tables to find the category-wise distribution of pizzas.

Code

```
SELECT
    CATEGORY,
    COUNT(NAME)
FROM
    PIZZA_TYPES
GROUP BY
   CATEGORY;
```

Answer Set

```
"category“  | "count"
"Supreme"   |      9
"Classic"   |      8
"Veggie“    |      9
"Chicken"   |      6
"category“  |      1
```

Q9. Group the orders by date and calculate the average number of pizzas ordered per day.

Code

```
SELECT
        TO_CHAR(AVG(QUANTITY), '999,999.00') AS "Avg pizza order”
 FROM
       (
        SELECT
               SUM(QUANTITY) AS "quantity",
               ORDER_DATE
        FROM
              ORDER_DETAILS
              JOIN ORDERS ON ORDER_DETAILS.ORDER_ID = ORDERS.ORDER_ID
       GROUP BY
             ORDER_DATE )
 AS "order quantity";
```

Answer Set

```
"Avg pizza order"
          "138.47"
```

Q10. Determine the top 3 most ordered pizza types based on revenue.

Code

```
SELECT
        PIZZA_TYPES.NAME,
        SUM(PIZZAS.PRICE * ORDER_DETAILS.QUANTITY) AS "Revenue“
FROM
        PIZZA_TYPES
        JOIN PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
        JOIN ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
GROUP BY
       PIZZA_TYPES.NAME
ORDER BY
      "Revenue" DESC
LIMIT
3;
```

Answer Set

"name"	"Revenue"
"The Thai Chicken Pizza"	43434.25
"The Barbecue Chicken Pizza“	42768.00
"The California Chicken Pizza“	41409.50

Q11. Analyze the cumulative revenue generated over time.

Code

```
SELECT
        order_date,
        SUM(Revenue) OVER (ORDER BY order_date) AS Cum_revenue
FROM
      (
       SELECT
              orders.order_date,
              SUM(order_details.quantity * pizzas.price) AS Revenue
       FROM
             order_details
       JOIN
             pizzas
       ON
            order_details.pizza_id = pizzas.pizza_id
      JOIN
           orders
      ON
          orders.order_id = order_details.order_id
     GROUP BY
          orders.order_date) AS Sales
 ORDER BY
          order_date;
```

Answer Set

"order_date"	"cum_revenue"
"2015-01-01"	2713.85
"2015-01-02"	5445.75
"2015-01-03"	8108.15
"2015-01-04"	9863.60
"2015-01-05"	11929.55
"2015-01-06"	14358.50
"2015-01-07"	16560.70
"2015-01-08"	19399.05
"2015-01-09"	21526.40
"2015-01-10"	23990.35
"2015-01-11"	25862.65
"2015-01-12"	27781.70
"2015-01-13"	29831.30
"2015-01-14"	32358.70
"2015-01-15"	34343.50
"2015-01-16"	36937.65
"2015-01-17"	39001.75
"2015-01-18"	40978.60
"2015-01-19"	43365.75
"2015-01-20"	45763.65