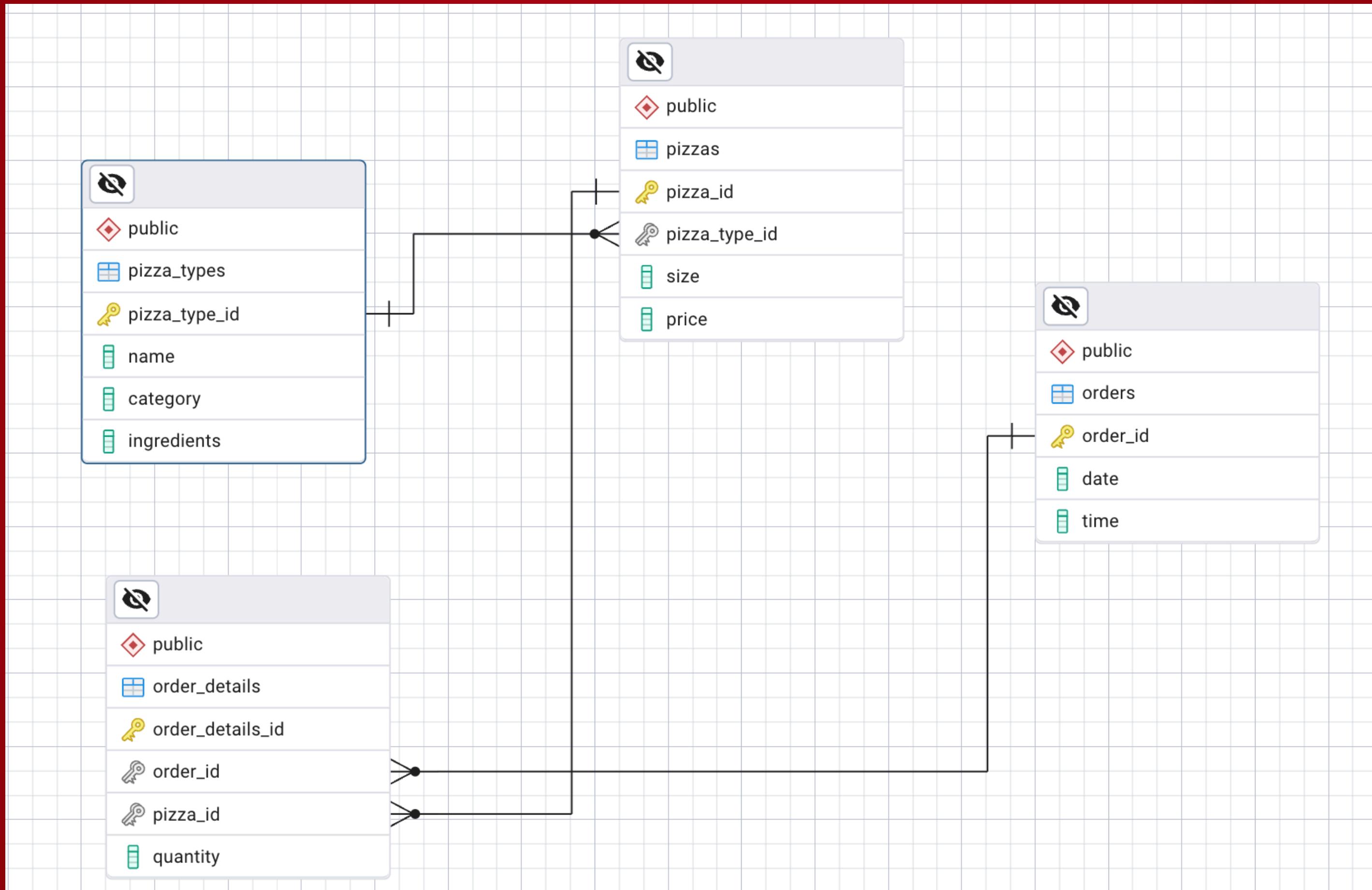


ANALYSIS OF PIZZA SALES DATA



ENTITY RELATIONSHIP DIAGRAM



The total number of orders placed

```
SELECT COUNT(order_id) AS total_orders  
FROM orders;
```

	total_orders	bigint
1		21350

The total revenue generated from pizza sales

```
SELECT  
    ROUND(  
        CAST(SUM(od.quantity * p.price) AS NUMERIC), 2  
    ) AS total_revenue  
FROM  
    pizzas p  
JOIN  
    order_details od  
ON p.pizza_id = od.pizza_id;
```



total_revenue	numeric
1	817860.05



Identifying the highest-priced pizza

```
SELECT
    pt.name,
    p.price AS max_price
FROM
    pizzas p
JOIN
    pizza_types pt
    ON pt.pizza_type_id = p.pizza_type_id
ORDER BY
    max_price DESC
LIMIT 1;
```

	name character varying (300)	max_price double precision
1	The Greek Pizza	35.95

Identifying the most common pizza size ordered

```
SELECT
    p.size AS pizza_size,
    SUM(quantity) AS total_quantity
FROM
    pizzas p
JOIN
    order_details od
    ON od.pizza_id = p.pizza_id
GROUP BY
    p.size
ORDER BY
    total_quantity DESC;
```

	pizza_size character varying	total_quantity bigint
1	L	18956
2	M	15635
3	S	14403
4	XL	552
5	XXL	28

Top 5 most ordered pizza types along with their quantities

```
SELECT
    pt.name AS pizza_type,
    SUM(od.quantity) AS total_quantities
FROM
    order_details od
JOIN
    pizzas p
    ON p.pizza_id = od.pizza_id
JOIN
    pizza_types pt
    ON p.pizza_type_id = pt.pizza_type_id
GROUP BY
    pt.name
ORDER BY
    total_quantities DESC
LIMIT 5;
```

	pizza_type character varying (300)	total_quantities bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

Find the total quantity of each pizza category ordered

```
SELECT
    pt.category AS pizza_category,
    SUM(od.quantity) AS total_quantity
FROM
    pizza_types pt
JOIN
    pizzas p
    ON p.pizza_type_id = pt.pizza_type_id
JOIN
    order_details od
    ON od.pizza_id = p.pizza_id
GROUP BY
    pt.category
ORDER BY
    total_quantity DESC;
```

	pizza_category character varying (50)	total_quantity bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

Determine the distribution of orders by hour of the day

```
SELECT  
    EXTRACT(HOUR FROM time) AS hour,  
    COUNT(order_id) AS total_orders  
FROM  
    orders  
GROUP BY  
    EXTRACT(HOUR FROM time)  
ORDER BY  
    hour;
```

	hour numeric	total_orders bigint
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009

Find the category-wise distribution of pizzas

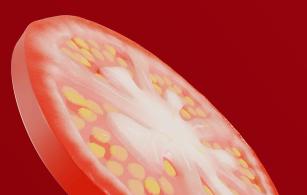
```
SELECT
    category,
    COUNT(name) AS total_pizza_types
FROM
    pizza_types
GROUP BY
    category
ORDER BY
    total_pizza_types;
```

	category character varying (50)	total_pizza_types bigint
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

Calculate the average number of pizzas ordered per day

```
SELECT  
    FLOOR(AVG(quantity)) AS avg_pizzas_per_day  
FROM (  
    SELECT  
        o.date,  
        SUM(od.quantity) AS quantity  
    FROM  
        orders o  
    JOIN  
        order_details od  
        ON o.order_id = od.order_id  
    GROUP BY  
        o.date  
    ORDER BY  
        o.date  
) AS sub1;
```

	avg_pizzas_per_day	locked
1	138	locked



Determine the top 3 most ordered pizza types based on revenue

```
SELECT
    ROUND(SUM(p.price * od.quantity)::NUMERIC, 2) AS revenue,
    pt.name AS pizza_type
FROM
    order_details od
JOIN
    pizzas p
    ON p.pizza_id = od.pizza_id
JOIN
    pizza_types pt
    ON pt.pizza_type_id = p.pizza_type_id
GROUP BY
    pt.name
ORDER BY
    revenue DESC
LIMIT 3;
```

	revenue numeric	pizza_type character varying (300)
1	43434.25	The Thai Chicken Pizza
2	42768.00	The Barbecue Chicken Pizza
3	41409.50	The California Chicken Pizza

Calculate the percentage contribution of each pizza type to total revenue

```
WITH total_revenue AS (
    SELECT
        SUM(p.price * od.quantity) AS total_revenue
    FROM
        pizzas p
    JOIN
        order_details od
    ON p.pizza_id = od.pizza_id
)

SELECT
    pt.name AS pizza_type,
    ROUND(
        (SUM(p.price * od.quantity) * 100 / (SELECT total_revenue FROM total_revenue))::NUMERIC, 3
    ) AS percent_contribution
FROM
    order_details od
JOIN
    pizzas p
ON p.pizza_id = od.pizza_id
JOIN
    pizza_types pt
ON pt.pizza_type_id = p.pizza_type_id
GROUP BY
    pt.name
ORDER BY
    percent_contribution DESC;
```

Calculate the percentage contribution of each pizza type to total revenue

	pizza_type character varying (300)	percent_contribution numeric
1	The Thai Chicken Pizza	5.311
2	The Barbecue Chicken Pizza	5.229
3	The California Chicken Pizza	5.063
4	The Classic Deluxe Pizza	4.668
5	The Spicy Italian Pizza	4.259
6	The Southwest Chicken Pizza	4.243
7	The Italian Supreme Pizza	4.093
8	The Hawaiian Pizza	3.946
9	The Four Cheese Pizza	3.945
10	The Sicilian Pizza	3.783
11	The Pepperoni Pizza	3.688
12	The Greek Pizza	3.479
13	The Mexicana Pizza	3.274
14	The Five Cheese Pizza	3.187

Analyze the cumulative revenue generated over time

```
WITH cte AS (
    SELECT
        o.date,
        ROUND(SUM(p.price * od.quantity)::NUMERIC, 2) AS total_rev_per_day
    FROM
        orders o
    JOIN
        order_details od
        ON od.order_id = o.order_id
    JOIN
        pizzas p
        ON p.pizza_id = od.pizza_id
    GROUP BY
        o.date
)
SELECT
    date,
    SUM(total_rev_per_day) OVER (ORDER BY date) AS cumulative_revenue
FROM
    cte;
```

Analyze the cumulative revenue generated over time

	date date	cumulative_revenue numeric
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.60
5	2015-01-05	11929.55
6	2015-01-06	14358.50
7	2015-01-07	16560.70
8	2015-01-08	19399.05
9	2015-01-09	21526.40
10	2015-01-10	23990.35
11	2015-01-11	25862.65
12	2015-01-12	27781.70
13	2015-01-13	29831.30

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

```
WITH cte AS (
    SELECT
        pt.name,
        pt.category,
        ROUND(SUM(od.quantity * p.price)::NUMERIC, 2) AS revenue
    FROM
        pizza_types pt
    JOIN
        pizzas p
    ON p.pizza_type_id = pt.pizza_type_id
    JOIN
        order_details od
    ON od.pizza_id = p.pizza_id
    GROUP BY
        pt.name, pt.category
)

SELECT name, category, revenue
FROM (
    SELECT
        name,
        category,
        revenue,
        RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS ranks
    FROM
        cte
) AS sub
WHERE
    ranks <= 3;
```

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

	name character varying (300) 	category character varying (50) 	revenue numeric 
1	The Thai Chicken Pizza	Chicken	43434.25
2	The Barbecue Chicken Pizza	Chicken	42768.00
3	The California Chicken Pizza	Chicken	41409.50
4	The Classic Deluxe Pizza	Classic	38180.50
5	The Hawaiian Pizza	Classic	32273.25
6	The Pepperoni Pizza	Classic	30161.75
7	The Spicy Italian Pizza	Supreme	34831.25
8	The Italian Supreme Pizza	Supreme	33476.75
9	The Sicilian Pizza	Supreme	30940.50
10	The Four Cheese Pizza	Veggie	32265.70
11	The Mexicana Pizza	Veggie	26780.75
12	The Five Cheese Pizza	Veggie	26066.50



ANALYSIS BY SACHIN YADAV

🌐 <https://github.com/sachinyadav461/pizza-sales-sql.git>

THANK YOU!

