SQL PROJECT

SHUBHAM SHARMA

PIZZA SALES PROJECT

- From Dough to Data — Using SQL to Deliver Fresh Insights on Pizza Sales!

SQL PROJECT

Home About Contact

HELLO EVERYONE !

I AM SHUBHAM SHARMA

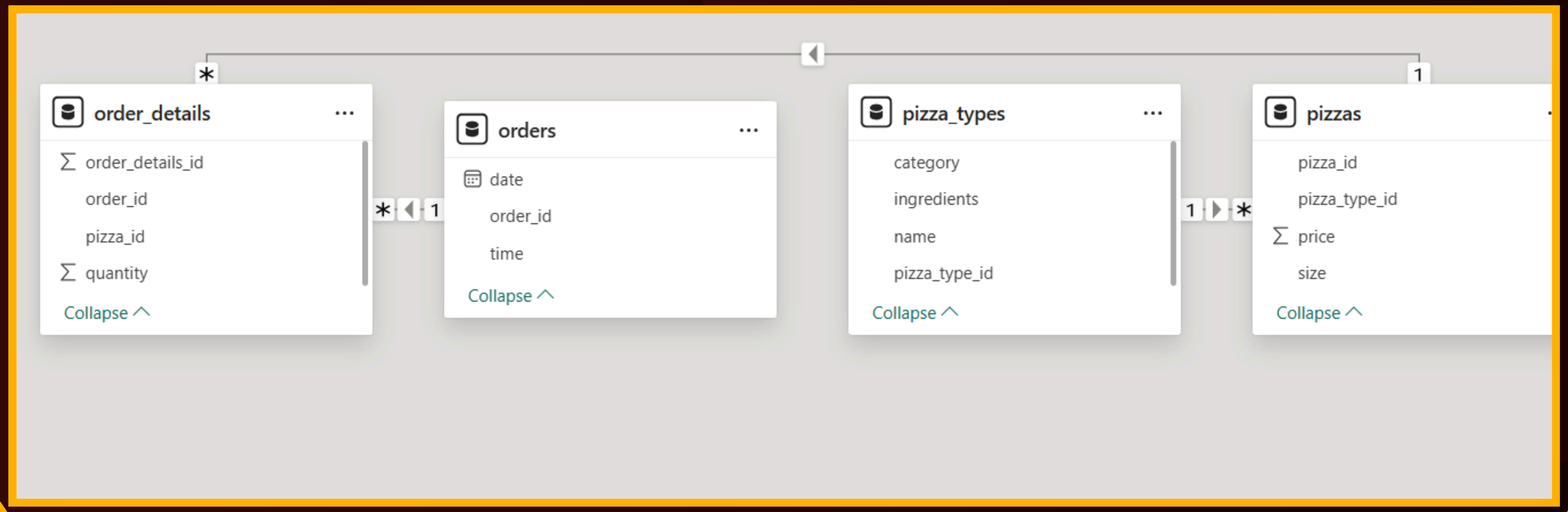
An aspiring Data Analyst. In this project, I utilized my SQL skills to write queries that analyzed pizza sales data, providing valuable business insights

SHUBHAM SHARMA

SQL PROJECT

Home About Contact

DATABASE SCHEMA



SQL PROJECT

QUERIES

Home About Contact

- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(order_id) as total_orders from orders;
```

total_orders
21350

- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
select
round(sum(order_details.quantity*pizzas.price),2) as total_sales
from order_details
join pizzas
on pizzas.pizza_id = order_details.pizza_id ;
```

Result Grid

total_sales
817860.05

SHUBHAM SHARMA

SQL PROJECT

QUERIES

- IDENTIFY THE HIGHEST-PRICED PIZZA.

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

Result Grid | Filter Rows:

name	highest_priced_pizza
The Greek Pizza	35.95

SHUBHAM SHARMA

SQL PROJECT

QUERIES

- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Result Grid

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

SQL PROJECT

QUERIES

SHUBHAM SHARMA

- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY Quantity DESC
LIMIT 5;
```

Result Grid | Filter 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

SQL PROJECT

QUERIES

SHUBHAM SHARMA

- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

Result Grid | Filter Rows:

category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

SHUBHAM SHARMA

SQL PROJECT

QUERIES

- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

hour(order_time)	count(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

SQL PROJECT

QUERIES

SHUBHAM SHARMA

- JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rc

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

- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_order_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | F

avg_order_per_day
138

SHUBHAM SHARMA

SQL PROJECT

QUERIES

- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

SHUBHAM SHARMA

SQL PROJECT

QUERIES

- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100),
            0) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Result Grid | Filt

category	revenue
Classic	27
Supreme	25
Veggie	24
Chicken	24

SHUBHAM SHARMA

SQL PROJECT

QUERIES

- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date, revenue ,
round(sum(revenue) over (order by order_date),2) as cumulative_revenue
from
(select orders.order_date,
round(sum(order_details.quantity*pizzas.price),2) 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 ;
```

Result Grid | Filter Rows:

order_date	revenue	cumulative_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
2015-01-10	2463.95	23990.35
2015-01-11	1872.3	25862.65
2015-01-12	1919.05	27781.7
2015-01-13	2049.6	29831.3
2015-01-14	2527.4	32358.7
2015-01-15	1984.8	34343.5

SHUBHAM SHARMA

SQL PROJECT

QUERIES

- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select  category,name , revenue ,ranking  from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as ranking
from
(select pizza_types.category , pizza_types.name,
round((sum(order_details.quantity*pizzas.price)),2)as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category ,  pizza_types.name) as pizza ) as pizza_b
where ranking <= 3;
```

Result Grid | Filter Rows: | Export:

category	name	revenue	ranking
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.7	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

SQL PROJECT

SHUBHAM SHARMA

THANK YOU

FOR YOUR TIME AND ATTENTION

WWW.LINKEDIN.COM/IN/SHUBHAM-SHARMA190

GITHUB.COM/SHUBHAMSHAR19

shubsharma729@gmail.com