

SQL PROJECT ON PIZZA SALES



ORDER
NOW

Start Your Slide

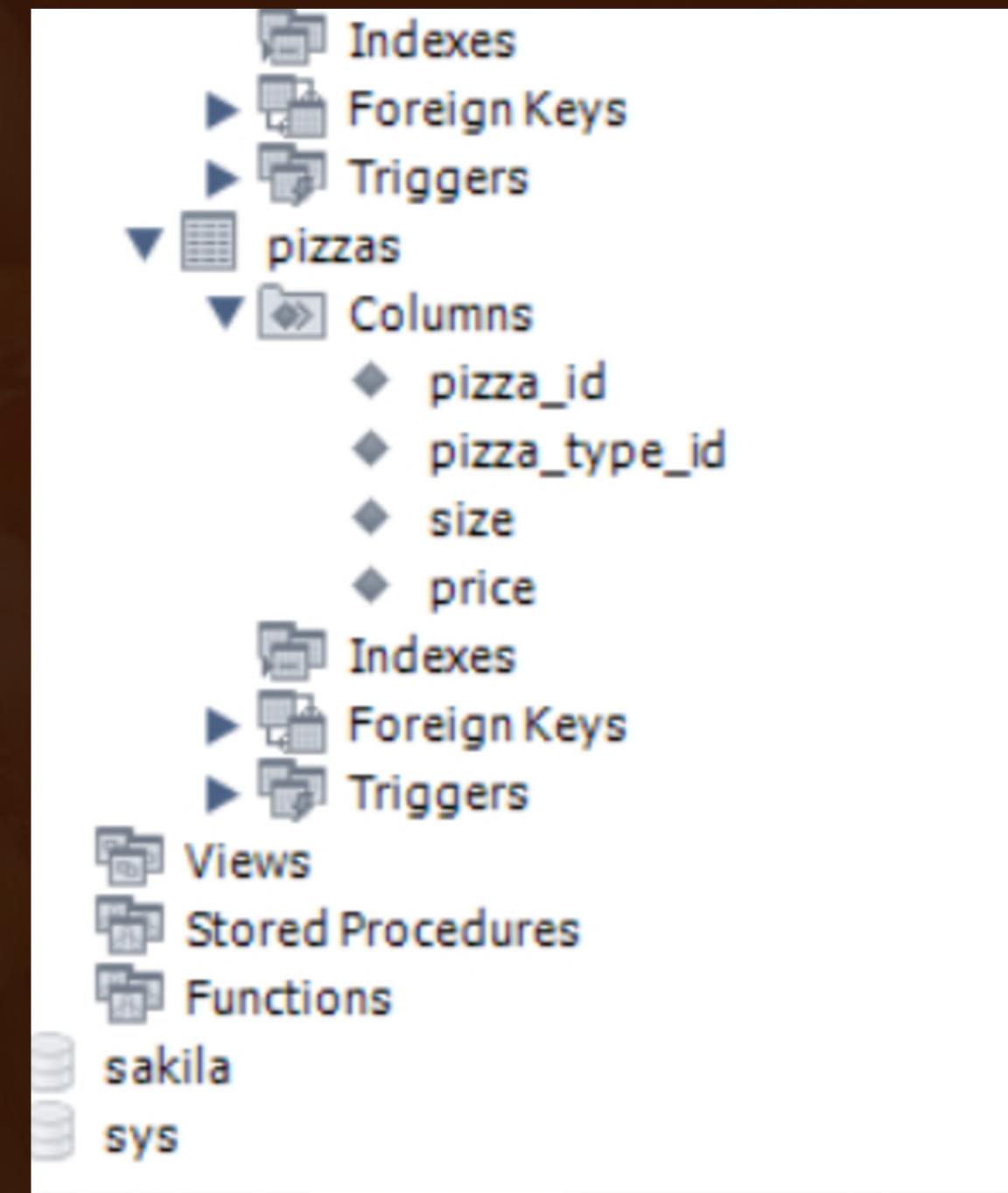
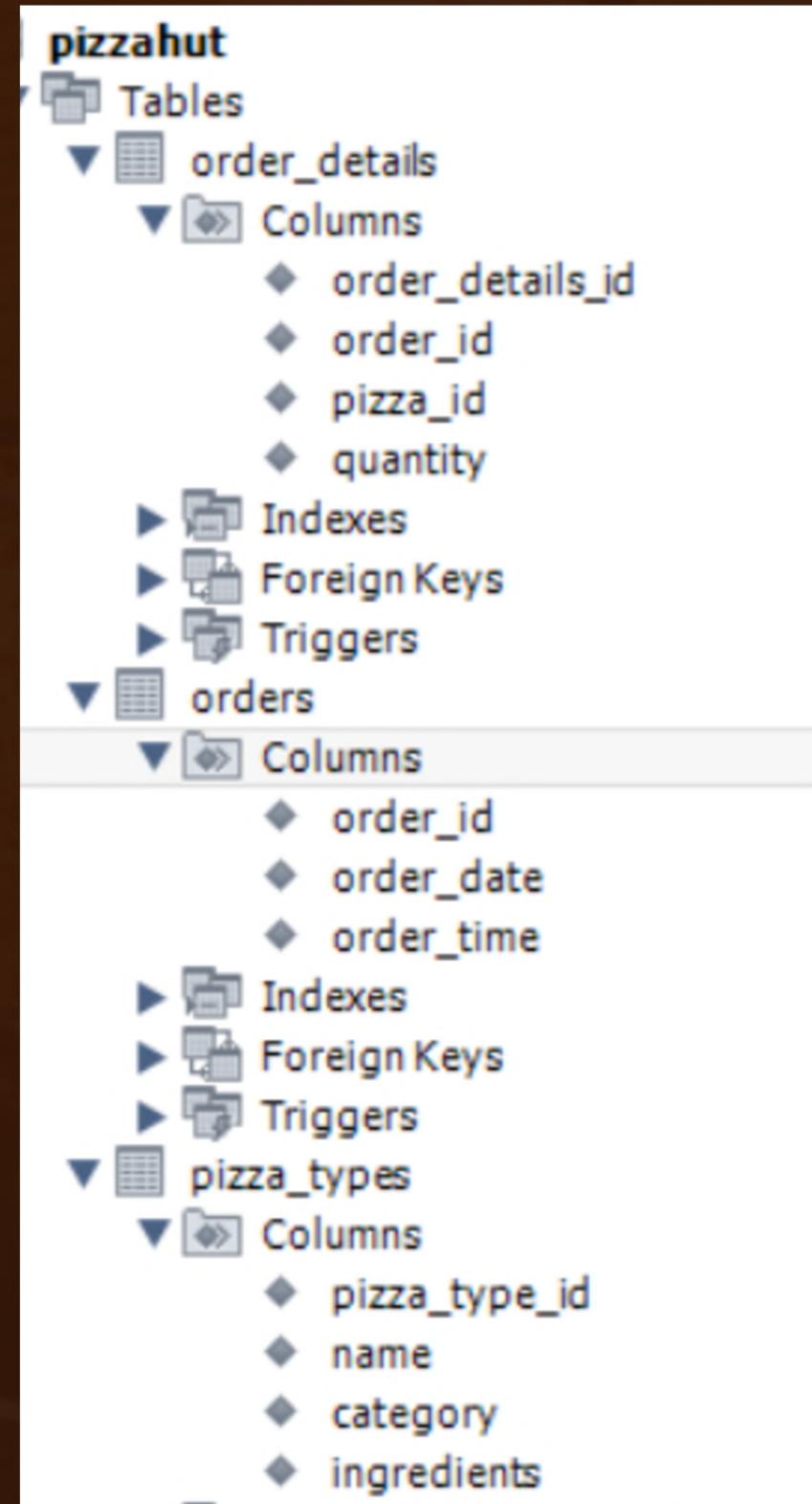


HELLO

Hello my name is Aryan Anand and in this project I have utilize SQL Queries to solve questions that were related to pizza sales.



SCHEMAS



Retrieve the total number of orders placed



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

Result Grid |

	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

Identify the highest-priced pizza

```
SELECT
    pizza_types.name, pizzas.price
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	price
▶	The Greek Pizza	35.95

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



```
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

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

```
SELECT  
    pizza_types.category,  
    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 pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.category  
ORDER BY quantity DESC;
```

Result Grid | Filter Row

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day



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

Result Grid | Filter Row

	HOURS	Orders
▶	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

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



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

Result Grid | Filter Rows

	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

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

	Average_pizza_per_day
▶	138

Determine the top 3 most ordered pizza types based on revenue

```
SELECT
    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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY Revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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

Calculate the percentage contribution of each pizza type to total revenue

```
SELECT pizza_types.category AS category ,  
CONCAT(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 order_details.pizza_id = pizzas.pizza_id  
)) * 100,2),"%) 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 category  
ORDER BY Revenue DESC;
```

Result Grid | Filter Rows:

	category	Revenue
▶	Classic	26.91%
	Supreme	25.46%
	Chicken	23.96%
	Veggie	23.68%

Analyze the cumulative revenue generated over time

```
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;
```

Result Grid | Filter Rows:

	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55

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

```
SELECT name , Revenue
FROM
(SELECT category , name , Revenue,
RANK() OVER(PARTITION by category
ORDER BY Revenue DESC ) AS rn
FROM
(SELECT pizza_types.category , pizza_types.name,
SUM(order_details.quantity * pizzas.price) 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.category , pizza_types.name) AS a) AS b
WHERE rn <= 3;
```

	name	Revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

THANK YOU FOR ATTENTION

See You Next