

DOMINO'S

PIZZA SALES

SQL





DOMINO'S
PIZZA SALE

ABOUT PROJECT

"Hi, I'm Krishna Pareek. Experienced in SQL, here we are solving and sorting data to unlock insights. Let's connect the dots together!"



BORCELLE
R E S T A U R A N T

ITALIAN PIZZA

More than 1000 customers
love this menu.



QUERIES

- BASIC:
- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.
- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.
- IDENTIFY THE HIGHEST-PRICED PIZZA.
- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.
- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALON



QUERIES

- INTERMEDIATE:
- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
- JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

QUERIES

ADVANCED:

- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.
- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.



- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
3 •   SELECT
4       COUNT(order_id) AS total_orders
5   FROM
6       dominoz_sales.orders;
```

	total_orders
▶	21350

- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
3 •   SELECT
4     ROUND(SUM(pizzas.price * order_details.quantity),
5           2) AS total_revenue
6   FROM
7     dominoz_sales.pizzas
8   JOIN
9     dominoz_sales.order_details ON pizzas.pizza_id = order_details.pizza_id;
```

	total_revenue
▶	817860.05

- IDENTIFY THE HIGHEST-PRICED PIZZA.

```
• SELECT pizza_types.name, pizzas.price
  FROM dominoz_sales.pizza_types
    JOIN dominoz_sales.pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
 ORDER BY price DESC
 LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
• SELECT
    pizzas.size,
    COUNT(order_details.quantity) AS common_pizza_size
FROM
    dominoz_sales.pizzas
    JOIN
    dominoz_sales.order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY common_pizza_size DESC;
```

	size	common_pizza_size
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

DOMINO'S
PIZZA SALE

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

```
3 •   SELECT
4     pizza_types.name,
5       SUM(order_details.quantity) AS quantity_details
6   FROM
7     dominoz_sales.pizza_types
8       JOIN
9     dominoz_sales.pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10      JOIN
11     dominoz_sales.order_details ON order_details.pizza_id = pizzas.pizza_id
12   GROUP BY pizza_types.name
13   ORDER BY quantity_details DESC
14   LIMIT 5;
```

	name	quantity_details
▶	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.

```
3 • SELECT
4     pizza_types.category,
5     SUM(order_details.quantity) AS quantity
6   FROM
7     dominoz_sales.pizza_types
8       JOIN
9     dominoz_sales.pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10      JOIN
11     dominoz_sales.order_details ON order_details.pizza_id = pizzas.pizza_id
12   GROUP BY category
13   ORDER BY quantity DESC;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
3 • SELECT
4      HOUR(order_time), COUNT(order_id)
5 FROM
6     orders
7 GROUP BY HOUR(order_time);
```

	HOUR(order_time)	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336

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

```
3 • SELECT
4         category, COUNT(name)
5 FROM
6     pizza_types
7 GROUP BY category;
```

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9
		9

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

```
3 • SELECT
4     ROUND(AVG(quantity), 0) as orders_perday
5 FROM
6     (SELECT
7         orders.order_date, SUM(order_details.quantity) AS quantity
8     FROM
9         orders
10    JOIN order_details ON orders.order_id = order_details.order_id
11    GROUP BY order_date) as order_quantity;
```

	orders_perday
▶	138

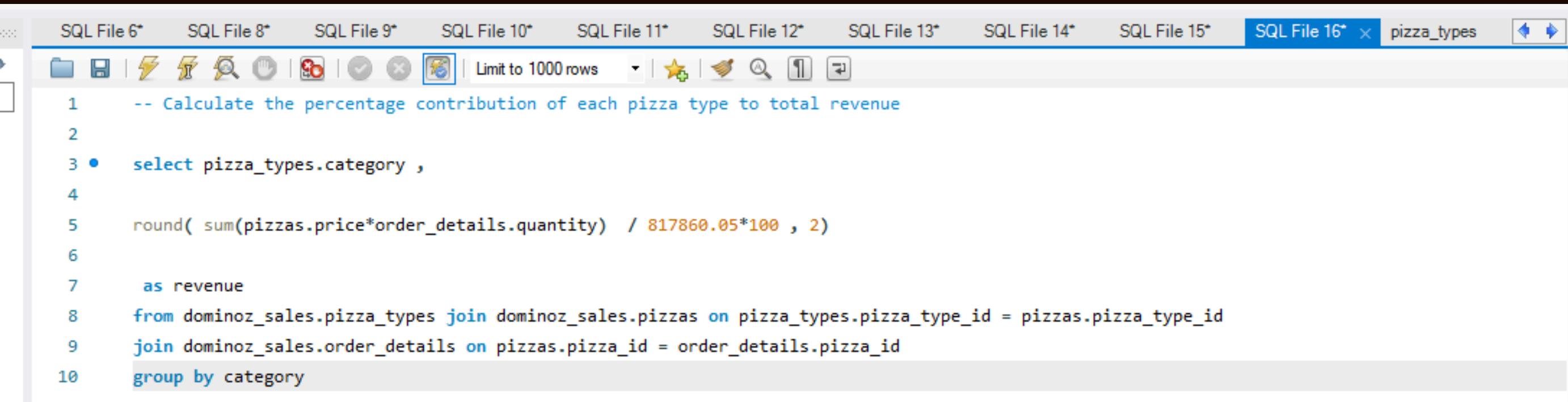
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
4 •   SELECT
5     pizza_types.name,
6     SUM(pizzas.price * order_details.quantity) AS revenue
7   FROM
8     dominoz_sales.pizza_types
9       JOIN
10    dominoz_sales.pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
11      JOIN
12    order_details ON pizzas.pizza_id = order_details.pizza_id
13   GROUP BY name
14   ORDER BY revenue DESC
15   LIMIT 3;
```

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

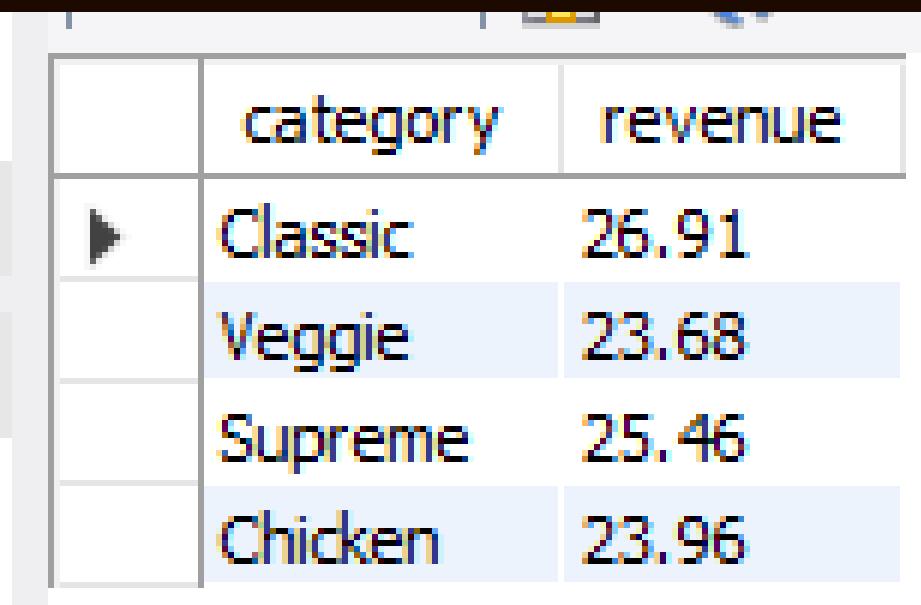
DOMINO'S PIZZA SALE

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



The screenshot shows a SQL editor window with the title bar "SQL File 16* pizza_types". The query is as follows:

```
1 -- Calculate the percentage contribution of each pizza type to total revenue
2
3 • select pizza_types.category ,
4
5 round( sum(pizzas.price*order_details.quantity) / 817860.05*100 , 2)
6
7 as revenue
8 from dominoz_sales.pizza_types join dominoz_sales.pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
9 join dominoz_sales.order_details on pizzas.pizza_id = order_details.pizza_id
10 group by category
```



	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

- METHOD 1

DOMINO'S PIZZA SALE

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

```
3 • SELECT
4     pizza_types.category,
5     ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
6             ROUND(SUM(pizzas.price * order_details.quantity),
7                 2) AS total_revenue
8         FROM
9             dominoz_sales.pizzas
10        JOIN
11            dominoz_sales.order_details ON pizzas.pizza_id = order_details.pizza_id) * 100,
12    2) AS revenue
13    FROM
14        dominoz_sales.pizza_types
15        JOIN
16            dominoz_sales.pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
17        JOIN
18            dominoz_sales.order_details ON pizzas.pizza_id = order_details.pizza_id
19    GROUP BY category
```

	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

- METHOD 2

- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
2 •   select order_date , round(sum(revenue) over(order by order_date) , 2 ) as cumulative_revenue
3   from
4
5   (select orders.order_date , SUM(pizzas.price * order_details.quantity) as revenue
6
7     from dominoz_sales.orders join dominoz_sales.order_details
8       on orders.order_id = order_details.order_id
9
10    join dominoz_sales.pizzas on order_details.pizza_id = pizzas.pizza_id
11   group by order_date ) as total_sales ;
```

	order_date	cumulative_revenue
▶	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7

DOMINO'S
PIZZA SALE

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

```
3 •   select name , revenue , rank_
4   (select name , category , revenue , rank()
5     over (partition by category order by revenue desc) as rank_
6   from
7   (select pizza_types.name , pizza_types.category , round(sum(order_details.quantity * pizzas.price) , 2 )
8     as revenue
9   from dominoz_sales.pizza_types
10  join dominoz_sales.pizzas
11  on pizza_types.pizza_type_id = pizzas.pizza_type_id
12  join order_details
13  on pizzas.pizza_id = order_details.pizza_id
14  group by pizza_types.name , pizza_types.category) as table1 ) as table2
15  where rank_ <= 3 ;
```

	name	revenue	rank_
▶	The Thai Chicken Pizza	43434.25	1
	The Barbecue Chicken Pizza	42768	2
	The California Chicken Pizza	41409.5	3
	The Classic Deluxe Pizza	38180.5	1
	The Hawaiian Pizza	32273.25	2
	The Pepperoni Pizza	30161.75	3

THANK YOU

BY:- KRISHNA PAREEK
GMAIL :- PAREEK.KRISHNA97@GMAIL.COM

