


SQL PROJECT FOR PIZZA SALES

BY SWADEEP GAJBHIYE



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
SELECT count(order_id) as Total_orders FROM orders;
```

	total_orders bigint 
1	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```
SELECT sum(order_details.quantity * pizzas.price) as total_sales  
FROM order_details JOIN pizzas  
ON order_details.pizza_id = pizzas.pizza_id;
```

	total_sales numeric
1	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;
```

	name text	price numeric (10,2)
1	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT pizzas.size, count(order_details.order_details_id) as total_sales
FROM pizzas JOIN order_details
ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size ORDER BY total_sales DESC;
```

	size text 🔒	total_sales bigint 🔒
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28



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;
```

	name text	quantity 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

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

	category  text	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 order_time) as Hour_value, count(order_id) as order_count
FROM orders
GROUP BY Hour_value ;
```

	hour_value numeric 🔒	order_count bigint 🔒
1	11	1231
2	23	28
3	18	2399
4	19	2009
5	15	1468
6	9	1
7	21	1198
8	17	2336
9	20	1642
10	13	2455
11	10	8
12	16	1920
13	22	663
14	12	2520
15	14	1472


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

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

	category text	count bigint
1	Supreme	9
2	Chicken	6
3	Classic	8
4	Veggie	9

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

```
SELECT ROUND(avg(quantity),0) 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;
```

	round numeric 
1	138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

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

	name text	revenue numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
SELECT pizza_types.category,  
round((sum(order_details.quantity * pizzas.price) / (select sum(order_details.quantity * pizzas.price)  
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 order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category ORDER BY revenue ;
```

	category text	revenue numeric
1	Veggie	23.68
2	Chicken	23.96
3	Supreme	25.46
4	Classic	26.91

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;
```

	order_date date	cum_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
14	2015-01-14	32358.70
15	2015-01-15	34343.50
16	2015-01-16	36937.65
17	2015-01-17	39001.75
18	2015-01-18	40978.60
19	2015-01-19	43365.75
20	2015-01-20	45763.65
21	2015-01-21	47804.20

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

```
select category, 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 order_details JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id
JOIN pizza_types
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category, pizza_types.name) as a ) as b
where rn <=3;
```

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

THANK YOU

