PIZZA SALES SQL PROJECT FOR DATA ANALYSIS

hello my name is chaitanya dhuri, and this project i have utilized sql query to solve the questions that related to pizza sales . all table show in the git hub

chaitanyadhuri

1) RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

QUERY:

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

OUTPUT:

	count bigint
1	21350

2) CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

QUERY:

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

OUTPUT:

	total_sales numeric
1	817860.05

3) IDENTITY THE HIGHEST-PRICED PIZZA.

QUERY:

```
select
    pizza_type.name , pizzas.price
from
    pizza_type
    join
    pizzas on pizza_type.pizza_type_id = pizzas.pizza_type_id
order by pizzas.price desc
limit 1;
```

OUTPUT:

	name character varying (100)	price numeric (10,2)
1	The Greek Pizza	35.95

4) IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

QUERY:

```
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;
```

OUTPUT:

	size character varying (30)	order_count bigint
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

5) LIST THE TOP 5 MOST ORDERED PIZZA TYPE ALONG WITH THEIR QUANTITIES

QUERY:

```
select pizza_type.name,sum( order_details.quantity ) as most_pizzas
   from pizza_type
join pizzas
   on pizza_type.pizza_type_id = pizzas.pizza_type_id
join order_details
   on order_details.pizza_id = pizzas.pizza_id
group by pizza_type.name
order by most_pizzas desc
limit 5;
```

OUTPUT:

	name character varying (100)	most_pizzas 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

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

QUERY:

```
select pizza_type.category,
sum(order_details.quantity) as quantity
from pizza_type join pizzas
on pizza_type.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_type.category order by quantity desc;
```

OUTPUT:

	category character varying (30)	quantity bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

7) DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

QUERY:

```
SELECT
    EXTRACT(HOUR FROM TO_TIMESTAMP(time, 'HH24:MI:SS')) AS hour , count(order_id) as order_count
FROM
    orders
group by  hour ;
```

OUTPUT:

	hour 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

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

QUERY:

```
select category, count(name) from pizza_type
group by category
```

OUTPUT:

	category character varying (30)	count bigint
1	Supreme	9
2	Classic	8
3	Veggie	9
4	Chicken	6

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

QUERY:

```
select round(avg(quantity),2) as avg_pizza_per_day_sales from
(select orders.date,  sum(order_details.quantity) as quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.date) as order_quantity ;
```

OUTPUT:

	avg_pizza_per_day_sales numeric
1	138.47

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

QUERY:

```
select pizza_type.name,
sum(order_details.quantity * pizzas.price) as revenue
from pizza_type join pizzas
on pizzas.pizza_type_id = pizza_type.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_type.name
order by revenue desc
limit 3;
```

OUTPUT:

	name character varying (100)	revenue numeric
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768.00
3	The California Chicken Pizza	41409.50

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

QUERY:

```
select pizza_type.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) * 100,2) as revenue
from pizza_type join pizzas
on pizza_type.pizza_type_id =pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_type.category
order by revenue desc;
```

OUTPUT:

	category character varying (30)	revenue numeric
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

12) ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

QUERY:

```
select date,
sum(revenue) over(order by date) as cum_revenue
from
(select orders.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.date) as sales;
```

OUTPUT:

	date character varying (30)	cum_revenue numeric
1	01-01-2015	2713.85
2	01-02-2015	5903.05
3	01-03-2015	7501.60
4	01-04-2015	9678.45
5	01-05-2015	12250.40
6	01-06-2015	15318.15
7	01-07-2015	17549.65
8	01-08-2015	19990.20
9	01-09-2015	22343.05
10	01-10-2015	25545.20
11	01-11-2015	27531.85
12	01-12-2015	29608.55
13	02-01-2015	32340.45
14	02-02-2015	34669.05
15	02-03-2015	37048.10
16	02-04-2015	39595.25
17	02-05-2015	41995.45
18	02-06-2015	44445.40

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

QUERY:

```sql
SELECT category, name, revenue
FROM (
    SELECT pizza_type.category, pizza_type.name,
           SUM(order_details.quantity * pizzas.price) AS revenue,
           RANK() OVER (PARTITION BY pizza_type.category ORDER BY SUM(order_details.quantity * pizzas.price) DESC) AS rn
    FROM pizza_type
    JOIN pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
    JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
    GROUP BY pizza_type.category, pizza_type.name
) AS ranked_revenue
WHERE rn <= 3;
```

OUTPUT:

	category character varying (30)	name character varying (100)	revenue 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