here I have utilized sql queries to solve question on pizza sales
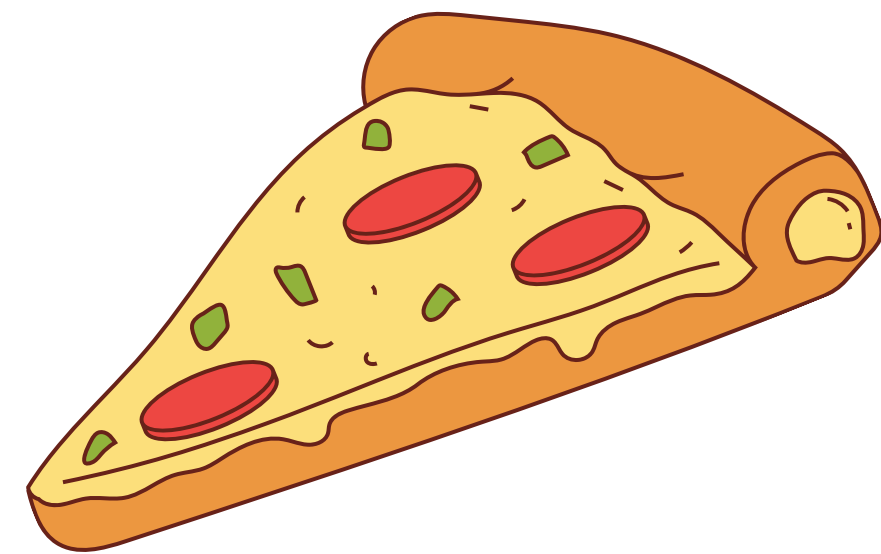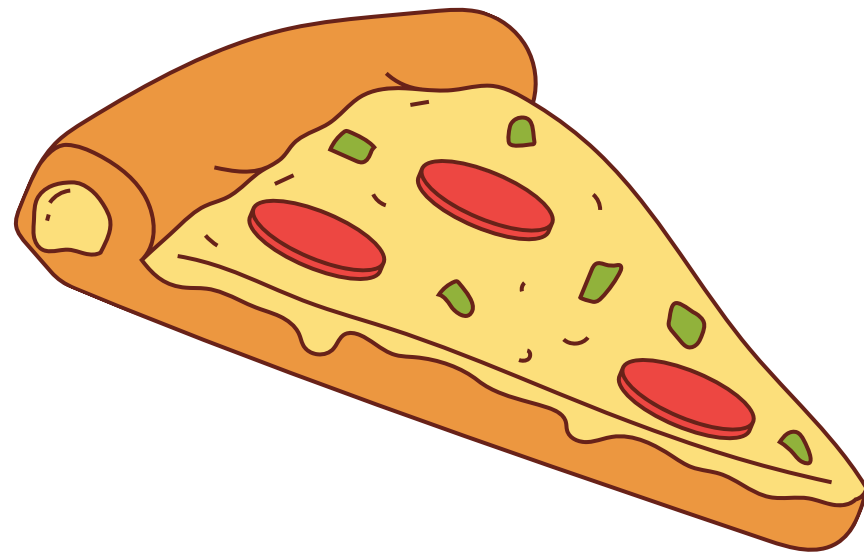
We have used 4 tables and used datbase pizzahut

tables used are:- pizza, pizza_types, orders, order_details

pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75
ckn_alfredo_s	ckn_alfredo	S	12.75

Result Grid | Filter Rows: | Export: | Wrap Cell Content: ‡A

pizza_type_id	name	category	ingredients
bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...
thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...
big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sau...

order_id	order_date	order_time
1	2015-01-01	11:38:36
2	2015-01-01	11:57:40
3	2015-01-01	12:12:28
4	2015-01-01	12:16:31
5	2015-01-01	12:21:30
6	2015-01-01	12:29:36
7	2015-01-01	12:50:37

order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1

Database:-

```sql
1   create database pizzahut;
2   use pizzahut;
3   create table orders(
4   order_id int not null,
5   order_date date not null,
6   order_time time not null,
7   primary key (order_id));
8
9   create table order_details(
10  order_details_id int not null,
11  order_id int not null,
12  pizza_id text not null,
13  quantity int not null,
14  primary key (order_details_id));
15
```

-- Retrieve the total number of orders placed.

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

	total_orders
▶	21350

Result Grid

-- Calculate the total revenue generated from pizza sales.

```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),2)
            AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid

	total_revenue
▶	817860.05

-- Identify the highest-priced pizza.

```sql
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;
```

name	price
▶ The Greek Pizza	35.95
The Greek Pizza	25.5
The Brie Carre Pizza	23.65
The Italian Vegetables Pizza	21
The Spinach Supreme Pizza	20.75

-- 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;
```

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

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS pizza_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 pizza_quantity DESC LIMIT 5;
```

Result Grid	Filter Rows:
name	pizza_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.

```sql
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;
```

Result Grid | Filter

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

-- Determine the distribution of orders by hour of the day.

```sql
select hour(order_time) as hour ,
count(order_id)
as order_count from orders
group by hour(order_time);
```

hour	order_count
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

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

```sql
select category , count(name )
 from pizza_types
group by category;
```

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_pizzas_ordered_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
```

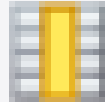|	avg_pizzas_ordered_per_day
▶	138

-- Determine the top 3 most ordered pizza types based on revenue.

```sql
select pizza_types.name,
sum(order_details.quantity*pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id=pizza_types.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	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

-- Calculate the percentage contribution of each pizza type to total revenue.

```sql
select pizza_types.category,
round(sum (order_details.quantity*pizzas.price) / (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),2)
        AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id) *100,2)
    as percent_revenue
from pizza_types join pizzas
on pizzas.pizza_type_id=pizza_types.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_types.category order by percent_revenue desc;
```

Result Grid | Filter Rows:

category	percent_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

-- Analyze the cumulative revenue generated over time.

```sql
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	2713.850000000004
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
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002

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

```sql
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 pizzas.pizza_type_id=pizza_types.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.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