```
create database pizzas.pizzas;
use pizzas;
select * from pizzas;
select * from pizza_types;
create table orders
(order_id int not null,
 order_date date not null,
 order_time time not null,
 primary key (order_id));
 desc orders;
 select * from orders;
 create table orders_details
(order_details_id
 int not null,
 order_id int not null,
 pizza_id text not null,
 quantity int not null,
 primary key (order_details_id));
 select * from orders_details;
```

```
 -- (2) retrived the total number of order placed
 select * from orders;
 select count(order_id) as total_order from orders;    -- (21350)

--  (3) calculate the total revenue generated from pizza sales
 select * from pizzas;
 select sum(price) as revenue from pizzas;    -- (1578)

 select round(sum((orders_details.quantity * pizzas.price)),0) as total_sales
-- (total_sales= 817890)
 from orders_details join pizzas on pizzas.pizza_id= orders_details.pizza_id;

 SELECT
    (orders_details.quantity * pizzas.price) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id;

select * from pizza_types;
```

```
select * from pizzas;


   -- (4) identify he highest-priced  pizza

select max(price)  from pizzas;

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


-- --(5) identify the most common pizza ordered


select size,count(pizza_id) from pizzas group by size;  -- (size(s)=32)


select * from orders_details;

select * from pizzas;

SELECT

    pizzas.size, COUNT(orders_details.order_details_id)

FROM

    pizzas

        JOIN

    orders_details ON pizzas.pizza_id = orders_details.pizza_id
```

```
GROUP BY pizzas.size order by COUNT(orders_details.order_details_id) desc
limit 1 ;    -- L=18526

-- (6) list the top 5 most order pizzas types along with their quantities

select * from pizzas;
select * from orders_details;
select * from orders;
select * from pizza_types;

SELECT
    pizza_types.name, SUM(orders_details.quantity) AS totalq
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY totalq DESC
```

```
LIMIT 5;


-- (7) Join the necessary tables to find the total quantity of each pizza
category ordered.
select * from pizzas;
select * from orders_details;
select * from pizza_types;


SELECT
    pizza_types.category, SUM(orders_details.quantity) as quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;



-- (8) Determine the distribution of orders by hour of the day
```

```
select * from orders_details;

select * from orders;


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


-- (9) join relevant tables to find the category-wise distribution of pizzas

select * from pizza_types;


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



-- (10) Group the orders by date and calculate the average number of
pizzas ordered per day


select * from pizzas;

select * from orders_details;

select * from orders;


SELECT
```

```
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_perday
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders_details.order_id = orders.order_id
    GROUP BY order_date) AS order_quant;

-- (11) Determine the top 3 most ordered pizza types based on revenue.
select * from pizzas;
select * from orders_details;
select * from pizza_types;

SELECT
    pizza_types.name,
   sum( (orders_details.quantity * pizzas.price) )AS revenue
FROM
    pizzas
        JOIN
```

```
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;

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

select pizza_types.category,
 round(sum(orders_details.quantity * pizzas.price) / (select
round(sum(orders_details.quantity * pizzas.price),0)
 As total_sales from orders_details  join pizzas on
 pizzas.pizza_id=orders_details.pizza_id )*100,2) as revenue
 from pizza_types join pizzas on
 pizzas.pizza_type_id=pizza_types.pizza_type_id
 join orders_details on
orders_details.pizza_id= pizzas.pizza_id group by pizza_types.category;
```

```
(select round(sum(orders_details.quantity * pizzas.price),0)
 As total_sales from orders_details  join pizzas on
 pizzas.pizza_id=orders_details.pizza_id );

select pizza_types.category,
 round(sum(orders_details.quantity * pizzas.price) / 817860*100,2) as
revenue
 from pizza_types join pizzas on
 pizzas.pizza_type_id=pizza_types.pizza_type_id
 join orders_details on
orders_details.pizza_id= pizzas.pizza_id group by pizza_types.category;

-- (13) Analyze the cumulative revenue generated over time.

select order_date, sum(revenue) over(order by order_date)as cum_value
from (select orders.order_date, sum(orders_details.quantity * pizzas.price) as
revenue from orders_details join pizzas
on orders_details.pizza_id= pizzas.pizza_id join orders on
orders.order_id=orders_details.order_id group by
orders.order_date) as sales ;
```

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

```
select name, category, revenue, rank() over(partition by category order by revenue) as rn
from
(select pizza_types.name, pizza_types.category,
sum((orders_details.quantity) * pizzas.price) as revenue from pizzas join orders_details on
orders_details.pizza_id= pizzas.pizza_id join pizza_types on
pizza_types.pizza_type_id= pizzas.pizza_type_id group by pizza_types.name, pizza_types.category) as a ;
```

```
select name,  revenue from
(select name, category, revenue, rank() over(partition by category order by revenue) as rn
from
(select pizza_types.name, pizza_types.category,
```

sum((orders_details.quantity) * pizzas.price) as revenue from pizzas join orders_details on

orders_details.pizza_id= pizzas.pizza_id join pizza_types on

pizza_types.pizza_type_id= pizzas.pizza_type_id group by pizza_types.name, pizza_types.category) as a) as b where rn <=3;