PIZZA SALES DATA

```
create database pizzahut;
use pizzahut;

# created table "orders, order_details" to import all rows as they are big dataset
create table orders(
order_id int primary key not null,
date date not null,
time time not null);

create table order_details(
order_details_id int primary key not null,
order_id int not null,
pizza_id varchar(20),
quantity int,
foreign key(order_id) references orders(order_id) );

select * from orders;
select * from order_details;
select * from pizzas;
select * from pizza_types;

#total number of orders placed
select count(order_id) as Total_orders from orders;

#Total revenue generated on pizza sales
```

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

#identify highest priced pizza
select * from pizzas
where price in (select max(price) from pizzas);

#identify most common pizza size ordered
select size, count(size) as Total_orders from order_details as od
join pizzas
on od.pizza_id = pizzas.pizza_id
group by size
order by Total_orders desc
limit 1;

#Top 5 most ordered pizza types along with their quantities
select pizza_id, count(pizza_id) as T_order, sum(quantity) as T_quantity
from order_details
group by pizza_id
order by T_order desc
limit 5;

#Join the necessary tables to find the total quantity of each pizza category ordered.
select pt.category, sum(quantity) as total_quantity from pizza_types as pt
join pizzas as p
on pt.pizza_type_id = p.pizza_type_id
join order_details as od
```

```
on od.pizza_id = p.pizza_id
group by pt.category
order by total_quantity desc;

#determine the distribution of orders by hour of the day
select hour(time) as Hours, count(order_id) as Total_orders from orders
group by Hours
order by Hours;

#join relevant tables to find the category-wise distribution of pizzas
select category, count(name) as number_of_Variety
from pizza_types
group by category;

#group the orders by date and calculate the avg number of pizzas ordered per day
select avg(Total) from
(select date, sum(quantity) as Total from orders
join order_details as od
on orders.order_id = od.order_id
group by date) as total_data;

#determine the top 3 most ordered pizza types based on revenue
select name, pt.pizza_type_id, sum(quantity*price) as revenue from pizza_types as pt
join pizzas
on pt.pizza_type_id = pizzas.pizza_type_id
join order_details as od
on od.pizza_id = pizzas.pizza_id
group by name, pizza_type_id
```

```sql
order by revenue desc
limit 3;

#calculate the percentage contribution of each pizza type to total revenue
-- STEP 1 : Made a view of joins to get particluar data for further use and code don't look messy if used in sub query
create view all_sales as
select pt.pizza_type_id, pt.name, pt.category, pizzas.price, od.* from pizza_types as pt
join pizzas
on pt.pizza_type_id = pizzas.pizza_type_id
join order_details as od
on pizzas.pizza_id = od.pizza_id;

-- STEP 2 : Created a CTE to get revenue per type and total revenue and then percentage
with tt_sales as(
select name,
sum(quantity*price) as Each_Revenue
from all_sales
group by name)
select *,
sum(each_revenue) over() as Total_sales,
concat(round((each_revenue/sum(each_revenue) over()) * 100,2),"%") as Percentage
from tt_sales
order by percentage desc;

#Analyse the cumulative revenue generated over time
select date, sum(sales) over(order by date) as cumulative_revenue
from
```

```sql
(select orders.date, sum(od.quantity*pizzas.price) as sales from pizzas
join order_details as od
on pizzas.pizza_id = od.pizza_id
join orders
on orders.order_id = od.order_id
group by date) as date_vs_price;

#Determine the top 3 most ordered pizza types based on revenue for each pizza category;
-- using already created view "all_sales"
select * from all_sales;
with cat_sales as
(select category, name, sum(price * quantity) as revenue,
row_number() over(partition by category order by sum(price * quantity) desc) as row_no
from all_sales
group by category, name)
select category, name, revenue
from cat_sales
where row_no <= 3;
```