```
create database d_manoj_pizza **create database

use d_manoj_pizza; **use databse

create table orders (
order_id int not null primary key ,
order_date date not null,
order_time  time not null);

 create table order_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 order_details;
 select * from orders;
 select * from pizza_types;
 select * from pizzas;



-- 🍕 Pizza Sales Analysis Notebook
-- ================================

**BASICS**

-- 1 Retrieve the total number of orders placed
SELECT COUNT(order_id) AS total_orders
FROM orders;
-- Result: 21350


-- 2 Calculate the total revenue generated from pizza sales

-- Step 1: View detailed revenue per order
SELECT
    order_details.order_id,
    order_details.pizza_id,
    order_details.quantity,
    pizzas.price
FROM order_details
JOIN pizzas
    ON order_details.pizza_id = pizzas.pizza_id;
```

```sql
-- Step 2: Get total revenue
SELECT
    SUM(order_details.quantity * pizzas.price) AS total_revenue
FROM order_details
JOIN pizzas
    ON order_details.pizza_id = pizzas.pizza_id;
-- Result: 817860.05


-- 3 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;
-- Result: The Greek Pizza – 35.95


-- 4 Identify the most common pizza size ordered
SELECT
    COUNT(pizzas.size) AS most_common,
    pizzas.size
FROM order_details
JOIN pizzas
    ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY most_common DESC
LIMIT 1;
-- Result: L (Large) – 18526 orders


-- 5 List the top 5 most ordered pizza types with their quantities
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_quantity
FROM order_details
JOIN pizzas
    ON order_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 total_quantity DESC
LIMIT 5;
-- Results:
-- 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


-- Pizza Sales Analysis (Intermediate)
-- ====================================

-- 6 Total quantity of each pizza type ordered
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_quantity
FROM order_details
JOIN pizzas
    ON order_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 total_quantity DESC;

-- Result:
-- The Greek Pizza                          1420
-- The Five Cheese Pizza                     1409
-- The Pepperoni, Mushroom, and Peppers Pizza  1359
-- The Green Garden Pizza                     997
-- The Chicken Alfredo Pizza                  987
-- The Italian Vegetables Pizza               981
-- The Chicken Pesto Pizza                    973
-- The Spinach Pesto Pizza                    970
-- The Soppressata Pizza                      961
-- The Spinach Supreme Pizza                  950
-- The Calabrese Pizza                        937
-- The Mediterranean Pizza                    934
-- The Brie Carre Pizza                       490


-- 7 .Total quantity of each pizza category
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM order_details
JOIN pizzas
    ON order_details.pizza_id = pizzas.pizza_id
JOIN pizza_types
    ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category
ORDER BY total_quantity DESC;

-- Result:
-- Classic    14888
```

```
-- Supreme    11987
-- Veggie     11649
-- Chicken    11050


-- 8. Distribution of orders by hour of the day (per date)
SELECT
    COUNT(orders.order_id) AS order_count,
    orders.order_date,
    HOUR(order_time) AS order_hour
FROM orders
GROUP BY orders.order_date, order_hour
ORDER BY orders.order_date ASC, order_count DESC;

-- Sample Result:
-- 10   2015-01-01   13
-- 8    2015-01-01   17
-- 8    2015-01-01   18
-- 7    2015-01-01   12
-- 7    2015-01-01   14
-- 7    2015-01-01   15
-- 6    2015-01-01   19
-- 5    2015-01-01   20
-- 4    2015-01-01   16
-- 3    2015-01-01   22


-- 9. Distribution of orders by hour (overall)
SELECT
    COUNT(order_id) AS order_count,
    HOUR(order_time) AS order_hour
FROM orders
GROUP BY order_hour
ORDER BY order_count DESC;

-- Sample Result:
-- 12  2520
-- 13  2455
-- 18  2399
-- 17  2336
-- 19  2009
-- 16  1920
-- 20  1642
-- 14  1472
-- 15  1468
-- 11  1231
-- 21  1198
-- 22  663
-- 23  28
-- 10  8
```

```
-- 9   1


-- 10. Category-wise distribution of pizzas
SELECT
    COUNT(pizza_types.name) AS type_count,
    pizza_types.category
FROM pizza_types
GROUP BY pizza_types.category;

-- Result:
-- Chicken    6
-- Classic    8
-- Supreme    9
-- Veggie     9


-- 11. Average number of pizzas ordered per day
SELECT
    ROUND(AVG(quantity), 2) AS avg_pizza_per_day
FROM (
    SELECT
        SUM(order_details.quantity) AS quantity,
        orders.order_date
    FROM order_details
    JOIN orders
        ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date
) AS order_quantity;

-- Result:
-- 138.47


-- 12. Top 3 most ordered pizza types based on revenue
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) AS total_revenue
FROM order_details
JOIN pizzas
    ON order_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 total_revenue DESC
LIMIT 3;

-- Result:
-- The Thai Chicken Pizza       42332.25
-- The Barbecue Chicken Pizza   41683
```

```
-- The California Chicken Pizza 40166.5


-- Advanced SQL Analysis
-- ==============================

-- 13. Percentage contribution of each pizza type to total revenue
SELECT
    pizza_types.name,
    ROUND(SUM(pizzas.price * order_details.quantity), 2) AS total_revenue,
    CONCAT(
        ROUND(
            ROUND(SUM(pizzas.price * order_details.quantity), 2) /
            (SELECT ROUND(SUM(pizzas.price * order_details.quantity), 2)
             FROM order_details
             JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id), 4
        ) * 100, '%'
    ) AS revenue_percentage
FROM order_details
JOIN pizzas
    ON order_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 total_revenue DESC;

-- Example Output:
-- The Thai Chicken Pizza       43434.25     5.31%
-- The Barbecue Chicken Pizza   42768        5.23%
-- The California Chicken Pizza 41409.5      5.06%
-- The Classic Deluxe Pizza     38180.5      4.67%
-- The Spicy Italian Pizza      34831.25     4.26%

-- 14 .Cumulative revenue generated over time
SELECT
    order_date,
    SUM(revenue) OVER (ORDER BY order_date) AS cumulative_revenue
FROM (
    SELECT
        orders.order_date,
        SUM(order_details.quantity * pizzas.price) AS revenue
    FROM order_details
    JOIN orders
        ON orders.order_id = order_details.order_id
    JOIN pizzas
        ON order_details.pizza_id = pizzas.pizza_id
    GROUP BY orders.order_date
) AS sales
ORDER BY order_date;
```

```
-- Example Output:
-- 2015-01-01   2713.85
-- 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


-- 15. Top 3 most ordered pizza types based on revenue for each category
SELECT
    name,
    total_revenue,
    category
FROM (
    SELECT
        category,
        name,
        total_revenue,
        RANK() OVER (PARTITION BY category ORDER BY total_revenue DESC) AS rn
    FROM (
        SELECT
            pizza_types.name,
            pizza_types.category,
            SUM(pizzas.price * order_details.quantity) AS total_revenue
        FROM order_details
        JOIN pizzas
            ON order_details.pizza_id = pizzas.pizza_id
        JOIN pizza_types
            ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        GROUP BY pizza_types.name, pizza_types.category
    ) AS a
) AS b
WHERE b.rn <= 3
ORDER BY category, total_revenue DESC;

-- Example Output:
-- Chicken Category:
-- The Thai Chicken Pizza       43434.25
-- The Barbecue Chicken Pizza   42768
-- The California Chicken Pizza 41409.5

-- Classic Category:
-- The Classic Deluxe Pizza     38180.5
-- The Hawaiian Pizza           32273.25
-- The Pepperoni Pizza          30161.75

-- Supreme Category:
-- The Spicy Italian Pizza      34831.25
```

```
-- The Italian Supreme Pizza    33476.75
-- The Sicilian Pizza           30940.5

-- Veggie Category:
-- The Four Cheese Pizza        32265.7
-- The Mexicana Pizza           26780.75
-- The Five Cheese Pizza        26066.5
```