Pizza Sales Analysis Report for 'Oven Story'

1. Database and Table Overview

The 'Oven Story' database contains the following structure:

1. `orders`: Stores order IDs, order dates, and times.

2. `order_details`: Links each order to pizzas, including quantities.

3. `pizzas` and `pizza_types`: Contain pizza pricing, sizes, and type information.

2. Analysis Queries

1. Total number of orders placed

```
SELECT COUNT(order_id) FROM orders;
```

2. Total revenue generated from pizza sales

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

3. Highest-priced pizza

```
SELECT pizza_types.name, pizzas.price FROM pizza_types INNER JOIN pizzas ON
pizza_types.pizza_type_id = pizzas.pizza_type_id ORDER BY pizzas.price DESC LIMIT 1;
```

4. Most common pizza size ordered

```
SELECT pizzas.size, COUNT(*) AS order_count FROM order_details INNER JOIN pizzas ON
order_details.pizza_id = pizzas.pizza_id GROUP BY pizzas.size ORDER BY order_count DESC
LIMIT 1;
```

3. Appendix: Full SQL Script

```
create database OvenStory;

create table orders(
order_id int NOT null,

order_date date not null,

order_time time not null,

primary key(order_id));


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));


-- Retrieve the total number of orders placed.


select count(order_id) from orders;


-- Calculate the total revenue generated from pizza sales.


SELECT

    ROUND(SUM(order_details.quantity * pizzas.price),

            2) AS Total_Sales

FROM

    order_details

        INNER JOIN

    pizzas ON order_details.pizza_id = pizzas.pizza_id;


--    Identify the highest-priced pizza.
```

```
    select

  pizza_types.name , pizzas.price

        from

        pizza_types inner join pizzas

        on

   pizza_types.pizza_type_id = pizzas.pizza_type_id

            order by pizzas.price desc limit 1;



--   Identify the most common pizza size ordered.

SELECT

    pizzas.size,

    COUNT(order_details.order_details_id) AS order_count

FROM

    pizzas

        INNER JOIN

    order_details ON order_details.pizza_id = pizzas.pizza_id

GROUP BY pizzas.size

ORDER BY order_count DESC

LIMIT 1;




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

SELECT

    pizza_types.name, 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.name
ORDER BY Quantity DESC
LIMIT 5;


-- Join the necessary tables to find the total quantity of each pizza category ordered.
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;


-- Determine the distribution of orders by hour of the day.
select
 hour(order_time)as Hour, count(order_id)
from
 orders
group by Hour;
```

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


SELECT

    category, COUNT(pizza_type_id)

FROM

    pizza_types

GROUP BY category;


-- Group the orders by date and calculate the average number of pizzas ordered per day.


SELECT

    AVG(Quantity)

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;



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


SELECT

    pizza_types.name,

    SUM(pizzas.price * order_details.quantity) AS revenue

FROM
```

```
    order_details

        JOIN

    pizzas ON pizzas.pizza_id = order_details.pizza_id

        JOIN

    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id

GROUP BY pizza_types.name

ORDER BY revenue DESC

LIMIT 3;




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



SELECT

    pizza_types.category,

    ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT

                    ROUND(SUM(pizzas.price * order_details.quantity),

                                2)

                FROM

                    pizzas

                        JOIN

                    order_details ON order_details.pizza_id = pizzas.pizza_id) * 100,

            2) AS revenue

FROM

    order_details

        JOIN

    pizzas ON pizzas.pizza_id = order_details.pizza_id

        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id

GROUP BY pizza_types.category

ORDER BY revenue DESC;




-- Analyze the cumulative revenue generated over time.


select

 order_date,

    sum(revenue) over(order by order_date) as Cum_revenue

    from

(select

 orders.order_date,

    round(sum(order_details.quantity * pizzas.price),2) 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

order by revenue) as sales;
```

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



select category, name , revenue

from

(select category, name, revenue,

rank() over(partition by category order by revenue desc) as ranking

from

(select

 pizza_types.category,

    pizza_types.name,

    sum(pizzas.price * order_details.quantity) as revenue

from

 pizzas

  join

 pizza_types 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, pizza_types.category) as a) as b

where ranking <=3;
```