BORCELLE PIZZA

20% OFF

Our Best Seller

SPICY PIZZA

Order from our website to enjoy a special 20% discount.

www.reallygreatsite.com

About me

MY NAME IS MONIKA KUMARI, AND I HAVE RECENTLY COMPLETED MY M.TECH IN COMPUTER SCIENCE. WITH A STRONG FOUNDATION IN FRONTEND DEVELOPMENT AND A GROWING EXPERTISE IN DATA SCIENCE, I HAVE DEVELOPED A KEEN INTEREST IN DATA ANALYSIS AND MANIPULATION USING VARIOUS TOOLS AND TECHNOLOGIES.

About my project

For this project, I worked on the "Pizza Quantity Analyzer," where I utilized MySQL to analyze and manage pizza order data. The project involved creating a database, importing data, and running queries to gain insights into customer preferences, order quantities, and other relevant metrics. This project not only strengthened my SQL skills but also provided valuable experience in working with real-world data scenarios.

PROBLEM NO:1- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT COUNT(ORDER_ID) AS TOTAL_ORDERS FROM ORDERS;
```

PROBLEM NO:2 - CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

-- --ANLYSISNG PIZZA_TYPES WHERE PIZZA_ID IS PRESENT AND CHECKING ORDER_DETAILS TABLE WHERE PIZZA_ID ALSO PRESENT.

```
SELECT
ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),
2) AS TOTAL_SALES
FROM
ORDER_DETAILS
JOIN
PIZZAS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID;
```

Problem no: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;
```

PROBLEM NO:4- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
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;
```

PROBLEM NO: 5- 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;
```

-- MEDIUM LEVELS

PROBLEM NO:6- 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;
```

PROBLEM NO:7- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT
*
FROM
ORDERS;
SELECT
HOUR(TIME) AS HOUR, COUNT(ORDER_ID) AS ORDER_COUNT
FROM
ORDERS
GROUP BY HOUR(TIME);
```

PROBLEM NO:8- JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
SELECT CATEGORY, COUNT(NAME) FROM PIZZA_TYPES
GROUP BY CATEGORY;
```

PROBLEM NO:9- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
SELECT
*
FROM
ORDERS;
SELECT
ROUND(AVG(QUANTITY), 0) AS AVERAGE_PIZZA_ORDERED_PER_DAY
FROM
(SELECT
ORDERS.DATE, SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
FROM
ORDERS
JOIN ORDER_DETAILS ON ORDERS.ORDER_ID =
ORDER_DETAILS.ORDER_ID
GROUP BY ORDERS.DATE) AS ORDER_QUANTITY;
```

PROBLEM NO:10- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
SELECT PIZZA_TYPES.NAME,
SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENU
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 REVENU DESC LIMIT 3;
```

-- ADVANCED:

PROBLEM NO:11- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT PIZZA_TYPES.CATEGORY,
ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) / (SELECT
ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE), 2) AS TOTAL_SALES
FROM
ORDER_DETAILS
JOIN PIZZAS ON PIZZAS.PIZZA_TYPE_ID = ORDER_DETAILS.PIZZA_ID)*100,2)
AS REVENU
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 REVENU DESC;
```

PROBLEM NO:12-ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT DATE,
SUM(REVENU) OVER(ORDER BY DATE) AS CUM_REVENU
FROM(SELECT ORDERS.DATE,
SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENU
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.DATE) AS  SALES;
```

PROBLEM NO:13- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
SELECT NAME, REVENU FROM
(SELECT CATEGORY, NAME, REVENU,
RANK() OVER(PARTITION BY CATEGORY ORDER BY REVENU DESC) AS RN
FROM
(SELECT PIZZA_TYPES.CATEGORY, PIZZA_TYPES.NAME,
SUM((ORDER_DETAILS.QUANTITY) * PIZZAS.PRICE) AS REVENU
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, PIZZA_TYPES.NAME) AS A) AS B
WHERE RN <=3;
```