

## Learn with Ankit Bansal



**100 Coding Problems**



**Akash Mahindrakar**

**Data Engineer**

**akashsjce8050@gmail.com**

**100 DAY CODING PROBLEMS**

### Step - 1 : Problem Statement

#### Problem Statement:

#### 52 - LOAN REPAYMENT

You're working for a large financial institution that provides various types of loans to customers. Your task is to analyze loan repayment data to assess credit risk and improve risk management strategies.

Write an SQL to create 2 flags for each loan as per below rules. Display loan id, loan amount , due date and the 2 flags.

1- `fully_paid_flag`: 1 if the loan was fully repaid irrespective of payment date else it should be 0.

2- `on_time_flag` : 1 if the loan was fully repaid on or before due date else 0.

 **Difficult Level : MEDIUM**

# NAMASTE SQL - DAY 7

## Step - 2 : Identifying The Input Data And Expected

### INPUT

| loans   |             |             |            |
|---------|-------------|-------------|------------|
| LOAN_ID | CUSTOMER_ID | LOAN_AMOUNT | DUE_DATE   |
| 1       | 1           | 5000        | 2023-01-15 |
| 2       | 2           | 8000        | 2023-02-20 |
| 3       | 3           | 10000       | 2023-03-10 |
| 4       | 4           | 6000        | 2023-04-05 |
| 5       | 5           | 7000        | 2023-05-01 |

| payments   |         |              |             |
|------------|---------|--------------|-------------|
| PAYMENT_ID | LOAN_ID | PAYMENT_DATE | AMOUNT_PAID |
| 1          | 1       | 2023-01-10   | 2000        |
| 2          | 1       | 2023-02-10   | 1500        |
| 3          | 2       | 2023-02-20   | 8000        |
| 4          | 3       | 2023-04-20   | 5000        |
| 5          | 4       | 2023-03-15   | 2000        |
| 6          | 4       | 2023-04-02   | 4000        |
| 7          | 5       | 2023-04-02   | 4000        |
| 8          | 5       | 2023-05-02   | 3000        |

### OUTPUT

| LOAN_ID | LOAN_AMOUNT | DUE_DATE   | FULLY_PAID_FLAG | ON_TIME_FLAG |
|---------|-------------|------------|-----------------|--------------|
| 1       | 5000        | 2023-01-15 | 0               | 0            |
| 3       | 10000       | 2023-03-10 | 0               | 0            |
| 4       | 6000        | 2023-04-05 | 1               | 0            |
| 5       | 7000        | 2023-05-01 | 1               | 1            |
| 2       | 8000        | 2023-02-20 | 1               | 1            |

# NAMASTE SQL - DAY 7

## Step - 3 : Writing the sql query to solve the

```
1
2 WITH CTE
3 AS (
4     SELECT L.loan_id
5           ,L.loan_amount
6           ,L.due_date
7           ,sum(amount_paid) AS total_amount
8           ,max(payment_date) AS last_payment_date
9     FROM payments P
10    JOIN loans L ON P.loan_id = L.loan_id
11   GROUP BY L.loan_id
12           ,L.loan_amount
13           ,L.due_date
14 )
15 SELECT LOAN_ID
16       ,LOAN_AMOUNT
17       ,DUE_DATE
18       ,CASE
19           WHEN LOAN_AMOUNT = TOTAL_AMOUNT
20             THEN 1
21           ELSE 0
22           END AS fully_paid_flag
23       ,CASE
24           WHEN LOAN_AMOUNT = TOTAL_AMOUNT
25             AND DUE_DATE ≤ last_payment_date
26             THEN 1
27           ELSE 0
28           END AS on_time_flag
29 FROM CTE
30
```



Save

**Was it  
helpful?**  
follow for more!



**Akash Mahindrakar**

Data Engineer

akashsjce8050@gmail.com



Comment

**SHARE YOUR THOUGHTS  
IN COMMENT BELOW**



Share