Smart Coin SQL Answers :

Q1.

1. SELECT (SUM(RUNS)/COUNT(MATCH\_ID)) AS AVERAGE FROM PLAYERS A INNER JOIN MATCH\_SCORES B ON A.PLAYER\_ID = B.PLAYER\_ID WHERE A.PLAYER\_ID = (SELECT PLAYER\_ID WHERE NAME LIKE ‘%ROHIT SHARMA%’);
2. SELECT COUNT(MATCH\_ID) FROM MATYCH\_SCORES WHERE MATCH\_ID < (SELECT DISTINCT(MATCH\_ID) FROM MATCH\_SCORES WHERE RUNS >= ‘200’);

Q2.

1. WITH DATE AS (SELECT DISTINCT(DISBURSED\_AT)AS FIRST FROM USER\_LOAN WHERE USER\_ID = 1 ORDER\_BY DISBURSED\_AT LIMIT 1,1

UNION ALL

SELECT DISTINCT (DISBURSED\_AT)AS SECOND FROM USER\_LOAN WHERE USER\_ID = 1 ORDER\_BY DISBURSED\_AT LIMIT =2,2

)

SELECT AVG(CASE WHEN DATEDIFF(DAY,FIRST,SECOND) FROM DATES);

2)

WITH PAYMENT\_DELAY AS (SELECT DATEDIFF(DAYS,PAYMENTDATE,PAYMENTDEADLINE) AS DELAY FROM PAYMENTS WHERE DELAY > 0

)

SELECT MAX(DELAY) AS MAX\_DELAY FROM PAYMENT\_DELAY;

3)

IF USER\_LOAN.STATUS <> ‘ACTIVE’ THEN SELECT -1

ELSE

SELECT COUNT(PAYMENTS.AMOUNTS FROM PAYMENTS

JOIN USER\_LOAN ON USER\_LOAN.USER\_ID = PAYMENTS.USER\_ID

WHERE PAYMENTS.STATUS = ‘ACTIVE’;

4)

SELECT ((USER\_LOAN.AMOUNT/PAYMENTS.AMOUNT) )\* 15 / 100) AS GND FROM USER\_LOAN

INNER JOIN PAYMENTS ON USER\_LOAN.USER\_ID = PAYMENTS.USER\_ID

WHERE USER\_LOADN.STATUS = ‘ACTIVE’ ;

5)

SELECT USER\_ID, AVG(COUNT(ID)) FROM SMS\_DATA WHERE CREATED\_AT BETWEEN USER\_LOAN.CREATE\_AT AND PAYMENT.CREADTED\_AT

GROUP BY USER\_ID

Q3)

1. Payment\_Freq : - How soon does he pays before the deadline based on which we can make a ranking
2. Cx\_Loan\_interaction : - Average gap between each sms tells about his activity with the loan
3. Best\_Cx : - On total users, who completed the balance first.