

Question B:

- (i) Write a CTE 'Histogram' that counts the number of loans for each risk score.
- Write a simple query to display the results of the CTE in descending order of the count.

The screenshot shows the SQL Developer interface with a query window titled 'A.sql'. The query is as follows:

```
WITH HISTOGRAM (RISK_SCORE, NUM_OF_LOAN) AS
(SELECT L.RISK_SCORE, L.AMOUNT FROM LOAN L)
SELECT L.RISK_SCORE, COUNT(*) AS NUM_OF_LOAN
FROM LOAN L
GROUP BY L.RISK_SCORE
ORDER BY L.RISK_SCORE;
```

The 'Script Output' pane shows the results of the query, fetched in 0.14 seconds. The results are displayed in a table with two columns: RISK_SCORE and NUM_OF_LOAN.

RISK_SCORE	NUM_OF_LOAN
1	0
2	412
3	419
4	429
5	444
6	450
7	454
8	460
9	461
10	463
11	468
12	472
13	473
14	474
15	475
16	479

The 'SQL History' pane at the bottom shows the query was executed on 06-OCT-... at 0.14 seconds.

- (ii) Spot the risk score(s) with the minimum number of loans. Call this CTE as RiskScoresWithMinLoans.

The screenshot shows the SQL Developer interface with a query window titled 'B.sql'. The query is as follows:

```
--(ii) Spot the risk score(s) with the minimum number of loans. Call this CTE as RiskScoresWithMinLoans.
WITH RiskScoreWithLoans (RISK_SCORE, MIN_NUM_OF_LOAN) AS
(SELECT L.RISK_SCORE, COUNT(*)
FROM LOAN L
GROUP BY L.RISK_SCORE
ORDER BY L.RISK_SCORE), RiskScoreWithMinLoans (MIN_LOANS) AS
(SELECT MIN(MIN_NUM_OF_LOAN)
FROM RiskScoreWithLoans
ORDER BY RISK_SCORE)
SELECT L.RISK_SCORE, R.MIN_LOANS
FROM RiskScoreWithLoans L, RiskScoreWithMinLoans R
WHERE L.MIN_NUM_OF_LOAN = R.MIN_LOANS;
```

The 'Script Output' pane shows the results of the query, fetched in 0.356 seconds. The results are displayed in a table with two columns: RISK_SCORE and MIN_LOANS.

RISK_SCORE	MIN_LOANS
1	412
2	419
3	429
4	444
5	450
6	460
7	461
8	463
9	468
10	472
11	473

The 'SQL History' pane at the bottom shows the query was executed on 06-OCT-... at 0.136 seconds.

--(iii) Using your Histogram and RiskScoresWithMinLoans CTE,
 --write a SQL query to find a description of loans which have a risk score spotted in (ii).

The screenshot displays the Oracle SQL Developer environment. The 'Connections' pane on the left shows 'Oracle Connections' with 'Adarsh' selected. The 'Worksheet' pane contains a SQL query that uses two Common Table Expressions (CTEs) to filter loans based on risk score and minimum loan count. The 'Query Result' pane shows the output of the query, which is a list of 11 loans with their risk scores and titles. The 'SQL History' pane at the bottom shows the executed query and its details.

SQL Query:

```

WITH HISTOGRAM(RISK_SCORE,NO_OF_LOANS) AS
( SELECT RISK_SCORE, COUNT(*)
  FROM LOAN
 GROUP BY RISK_SCORE
 ORDER BY RISK_SCORE),
RiskScoreWithMinLoans(MIN_LOANS) AS
( SELECT MIN(NO_OF_LOANS)
  FROM HISTOGRAM
 ORDER BY RISK_SCORE)
SELECT H.RISK_SCORE,L.LOAN_TITLE
FROM LOAN L, HISTOGRAM H, RiskScoreWithMinLoans R
WHERE H.NO_OF_LOANS = R.MIN_LOANS AND L.RISK_SCORE = H.RISK_SCORE;
  
```

Query Result:

RISK_SCORE	LOAN_TITLE
1	703 Consolidating Debt
2	715 Want to consolidate my debt
3	645 Trinfiniti
4	700 For Justin.
5	573 tingerst
6	710 need to consolidate
7	688 bmoore5110
8	708 Recent College Grad Wants to Pay Off CCs
9	685 FoundersCafe.com
10	712 twbmc
11	686 Phuocpnn

SQL History:

Connect...	TimeSt...	Type	Executed	Duratio...
Adarsh	06-OCT...	SQL	2	0.099