SQL PROJECT-

Loan Management System

The project is all about loan management system.
In this project I have created three tables-

1. **Loans-This table contains all information about loans.**
2. **Borrower-This table contains information about the borrower details.**
3. **Account Details-This table data about account user.**

---

1. Display year wise loan amount from **BORROWER** table.

QUERY-SELECT YEAR, LOAN_AMOUNT FROM BORROWER;

OUTPUT-

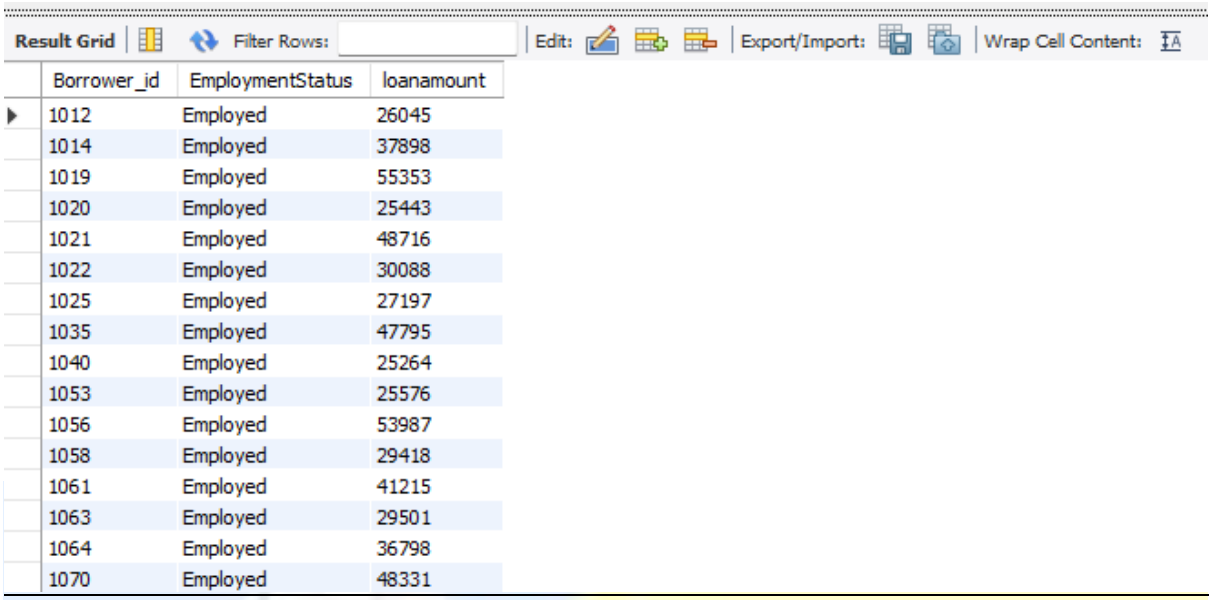
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	loan_amount
2010	50000
2012	100000
2020	150000
2021	250000
2008	200000
2000	100000
2022	50000
2024	80000
2022	60000
2010	100000
1999	55000
1995	90000
2015	115000
2005	125000
2008	150000
2002	250000
2020	80000

2. According to borrower id from the **LOANS** table. show me the records of users who are employed and their loan amount is greater than 25000.

QUERY-SELECT EMPLOYMENTSTATUS, LOANAMOUNT FROM LOANS WHERE EMPLOYMENTSTATUS='EMPLOYED' AND LOANAMOUNT > 25000;

Output-

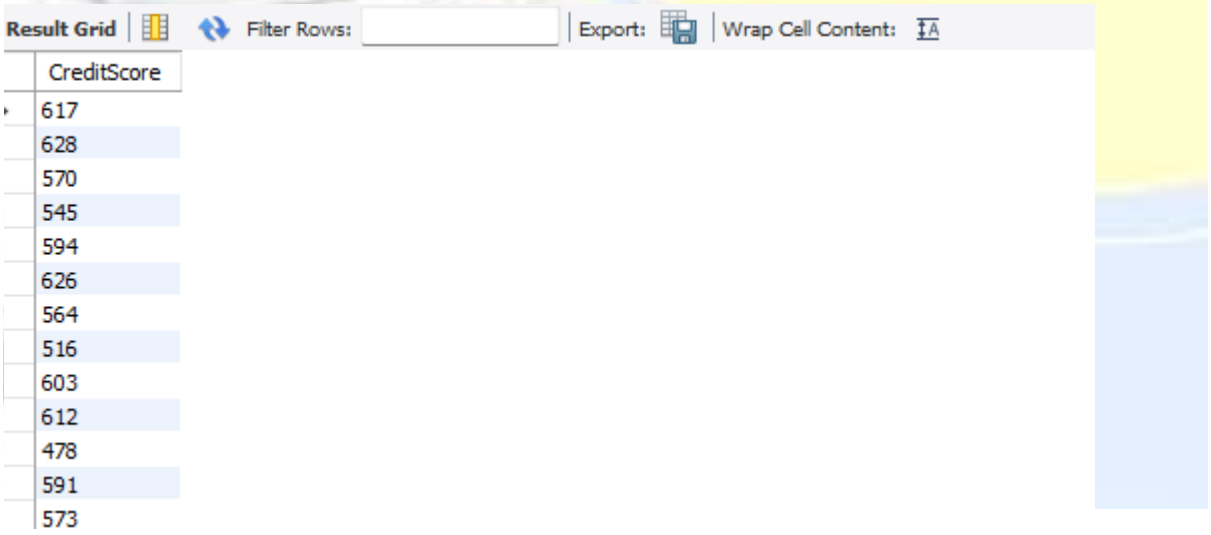


Borrower_id	EmploymentStatus	loanamount
1012	Employed	26045
1014	Employed	37898
1019	Employed	55353
1020	Employed	25443
1021	Employed	48716
1022	Employed	30088
1025	Employed	27197
1035	Employed	47795
1040	Employed	25264
1053	Employed	25576
1056	Employed	53987
1058	Employed	29418
1061	Employed	41215
1063	Employed	29501
1064	Employed	36798
1070	Employed	48331

3. Show me only Credit Score column from **LOANS** table. Do not return duplicate values.

QUERY-SELECT DISTINCT CREDITSCORE FROM LOANS;

OUTPUT-



CreditScore
617
628
570
545
594
626
564
516
603
612
478
591
573

4. Write a query to fetch records that contains Transaction type as Deposit arranged in Ascending order of Account No. Query must return Transaction id, transaction type and Account No from **ACCOUNT DETAILS** table.

QUERY- SELECT TRANSACTION_ID, TRANSACTION_TYPE, ACCOUNT_NO FROM ACCOUNT_DETAILS WHERE TRANSACTION_TYPE='DEPOSIT' ORDER BY ACCOUNT_NO;

OUTPUT-

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Transaction_id	Transaction_type	Account_No
TID-28	Deposit	65897695
TID-24	Deposit	645976994
TID-30	Deposit	658924994
TID-27	Deposit	658967994
TID-11	Deposit	658976534
TID-29	Deposit	658976933
TID-13	Deposit	658978834
TID-18	Deposit	668976912
TID-21	Deposit	678976454
TID-33	Deposit	808976980
TID-34	Deposit	858976990
TID-35	Deposit	888976988
TID-23	Deposit	888976994
TID-16	Deposit	908976979
TID-17	Deposit	958976965

5. Select all users from **ACCOUNT DETAILS** table whose first name is Vikas OR last name is Patil.

QUERY- SELECT * FROM ACCOUNT_DETAILS WHERE FNAME='VIKAS' OR LNAME='PATIL';

OUTPUT-

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Transaction_id	Transaction_type	FName	LName	Account_Balance	Account_No	IFSC_code	CVV
TID-11	Deposit	Vikas	Patil	25000	658976534	ICIC0000424	235
TID-13	Deposit	Satish	Patil	25000	658978834	ICIC0000428	354
TID-34	Deposit	Sidharth	Patil	35000	858976990	ICIC0000644	134
TID-35	Deposit	Mack	Patil	85000	888976988	ICIC0000888	712

6. Write a query to display only first 5 records from the **BORROWER** table. Query must return all the columns from the **BORROWER** table.

QUERY- SELECT * FROM BORROWER LIMIT 5;

OUTPUT-

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Borrower_name	Status	loan_term	Year	Loan_amount	Borrower_id
Ramesh	Active	6	2010	50000	1011
Rutika	Paid Off	12	2012	100000	1012
Akash	Defaulted	18	2020	150000	1013
Jay	Active	12	2021	250000	1014
Prem	Active	12	2008	200000	1015

7. Write a Query to count the total loan users from the **LOANS** table.

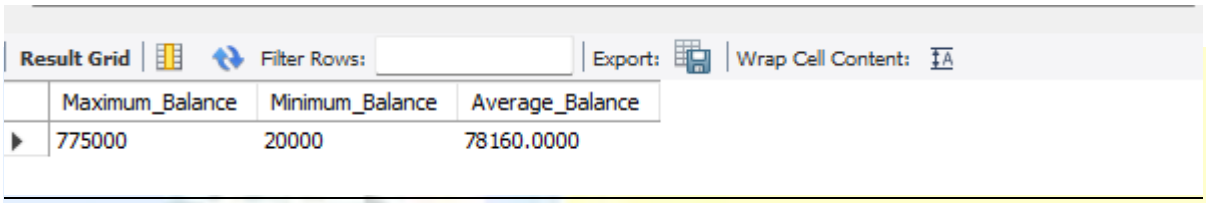
QUERY- SELECT COUNT (*) AS TOTAL_LOAN_USERS FROM LOANS;

OUTPUT-

8. Write a query to display maximum Account Balance, minimum Account Balance and average Account Balance from **ACCOUNT DETAILS** table.

QUERY- SELECT MAX(ACCOUNT_BALANCE) AS MAXIMUM_BALANCE, MIN(ACCOUNT_BALANCE) AS MINIMUM_BALANCE, AVG(ACCOUNT_BALANCE) AS AVERAGE_BALANCE FROM ACCOUNT_DETAILS;

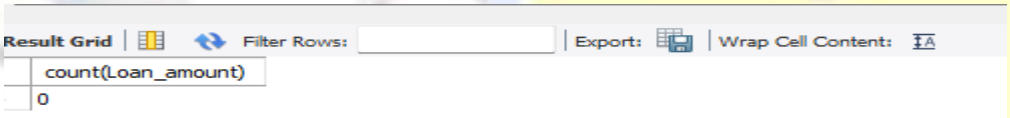
OUTPUT-



9. Count the No of loan amounts which are null in the **BORROWER** table.

QUERY- SELECT COUNT(LOAN_AMOUNT) FROM BORROWER WHERE LOAN_TERM IS NULL;

OUTPUT-



10. Replace Active with Active Account in **BORROWER** table.

QUERY- SELECT REPLACE (STATUS, "ACTIVE", 'ACTIVE ACCOUNT') AS UPDATED _RESULT FROM BORROWER;

OUTPUT-

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Updated_Result
Active Account
Paid Off
Defaulted
Active Account
Active Account
Active Account
Paid Off
Defaulted
Paid Off
Defaulted
Paid Off
Defaulted
Paid Off
Defaulted
Paid Off
Active Account
Active Account
Defaulted

11. Find the second Highest loan amount from **LOANS** table.

QUERY- SELECT MAX (LOANAMOUNT) AS SECOND_MAXIMUM FROM LOANS

WHERE LOANAMOUNT! = (SELECT MAX (LOANAMOUNT) FROM LOANS);

OUTPUT-

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Second_Maximum
66582

12. Display Employment Status wise count of loan users from **LOANS** table.

QUERY- SELECT EMPLOYMENTSTATUS, COUNT(BORROWER_ID) AS COUNT_OF_USERS FROM LOANS GROUP BY EMPLOYMENTSTATUS;

OUTPUT-

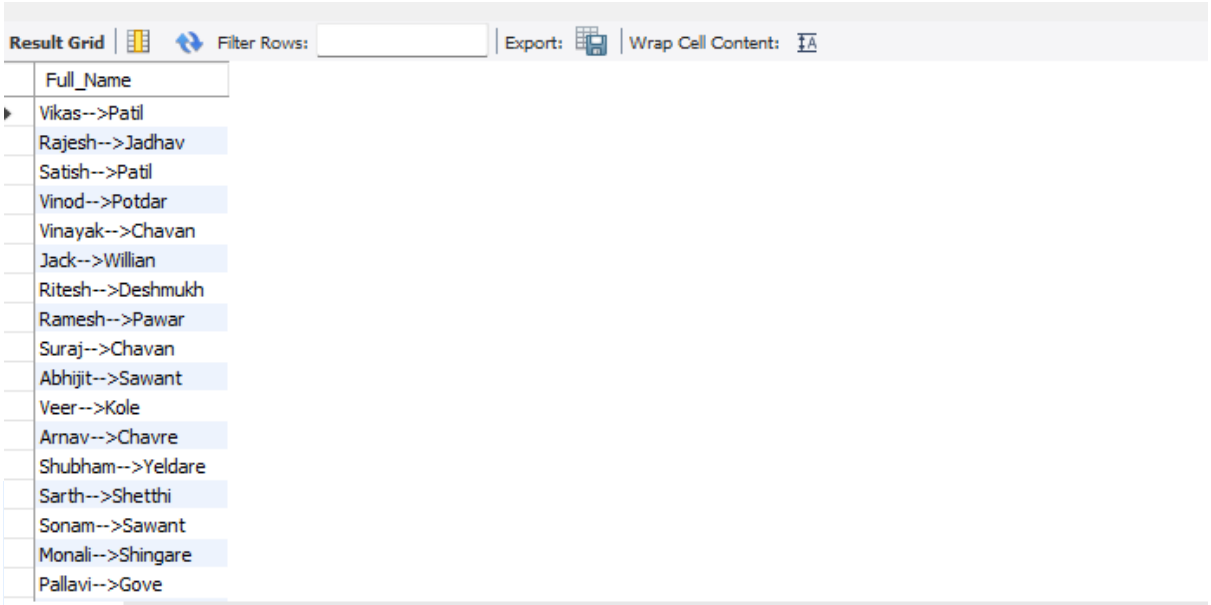
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmploymentStatus	Count_Of_Loan_Users
Employed	170
Self-Employed	16
Unemployed	12

13. Display the first name and last name of user in single column from **ACCOUNT DETAILS** table. Give Full Name as updated column name.

QUERY- SELECT CONCAT(FNAME,'-->', LNAME) AS FULL_NAME FROM ACCOUNT_DETAILS;

OUTPUT-



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Full_Name
Vikas-->Patil
Rajesh-->Jadhav
Satish-->Patil
Vinod-->Potdar
Vinayak-->Chavan
Jack-->Willian
Ritesh-->Deshmukh
Ramesh-->Pawar
Suraj-->Chavan
Abhijit-->Sawant
Veer-->Kole
Arnav-->Chavre
Shubham-->Yeldare
Sarth-->Shetthi
Sonam-->Sawant
Monali-->Shingare
Pallavi-->Gove

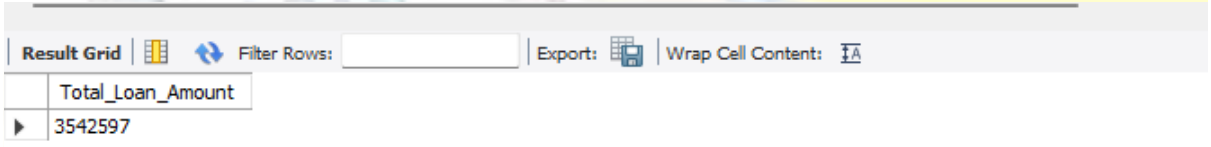
14. Write a query to calculate Total Loan Amount generated by loan amount from **LOANS** table using **INNER JOIN**. Query must return only Total Loan Amount generated.

QUERY- SELECT SUM (LOANS.LOANAMOUNT) AS TOTAL_LOAN_AMOUNT

FROM BORROWER

INNER JOIN LOANS ON BORROWER.BORROWER_ID=LOANS.BORROWER_ID;

OUTPUT-



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Total_Loan_Amount
3542597

15. Write a query to Display Marital Status and Borrower name from **LOANS AND BORROWER** table using **RIGHT JOIN.** Query must return Marital Status and Borrower Name.

QUERY- SELECT LOANS.MARITALSTATUS AS USER_MARITAL_STATUS, BORROWER.BORROWER_NAME AS BORROWER_NAME

FROM LOANS

RIGHT JOIN BORROWER ON LOANS.BORROWER_ID=BORROWER.BORROWER_ID;

OUTPUT-

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

User_Marital_Status	Borrower_name
Married	Ramesh
Single	Rutika
Married	Akash
Single	Jay
Married	Prem
Married	Prajkta
Married	Aaru
Divorced	Mahesh
Divorced	Raj
Married	Satish
Single	Arnav
Married	Shubham
Married	vikas

16. Write a query to Display Marital Status and Borrower name from **LOANS AND BORROWER** table using **LEFT JOIN.** Query must return Marital Status and Borrower Name.

QUERY- SELECT LOANS.MARITALSTATUS AS USER_MARITAL_STATUS, BORROWER.BORROWER_NAME AS BORROWER_NAME

FROM LOANS

LEFT JOIN BORROWER ON LOANS.BORROWER_ID=BORROWER.BORROWER_ID;

OUTPUT-

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

User_Marital_Status	Borrower_name
Single	Leo
Divorced	Owen
Married	Samuel
Divorced	Advik
Divorced	Arjun
Widowed	NULL
Single	NULL
Married	NULL
Single	NULL
Single	NULL
Married	Atharv
Married	Eshhan
Married	krishna
Married	Shivansh
Married	Viraj
Divorced	Aryan
Married	Kiran

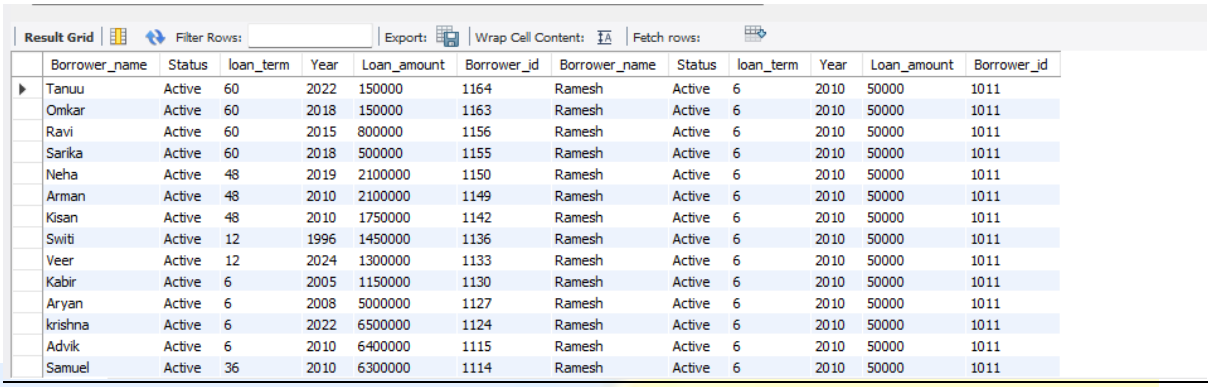
17. Write a query it matches User Borrower Id from **BORROWER** table that are from same Status using **SELF JOIN.**

QUERY- SELECT * FROM BORROWER AS A, BORROWER AS B

WHERE A. STATUS = B. STATUS

AND A. BORROWER_ID <>B.BORROWER_ID;

OUTPUT-

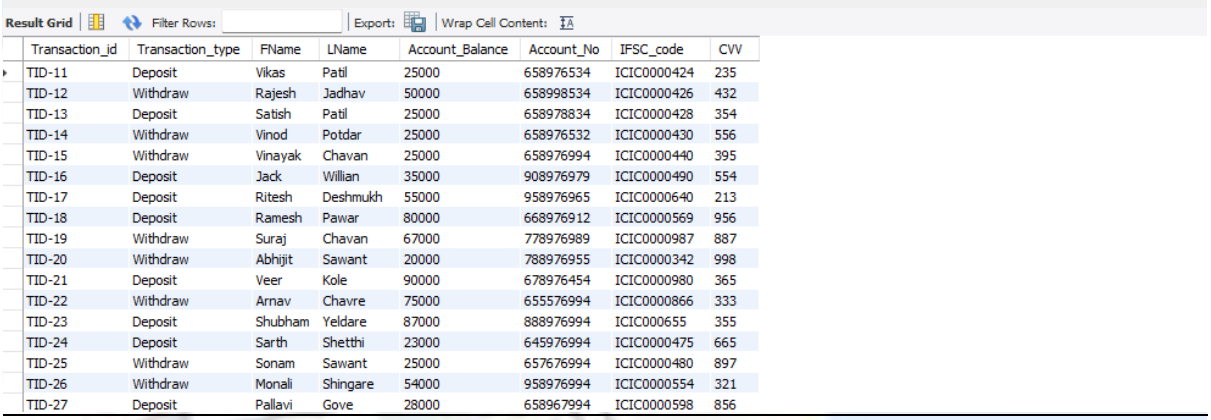


Borrower_name	Status	loan_term	Year	Loan_amount	Borrower_id	Borrower_name	Status	loan_term	Year	Loan_amount	Borrower_id
Tanuu	Active	60	2022	150000	1164	Ramesh	Active	6	2010	50000	1011
Omkar	Active	60	2018	150000	1163	Ramesh	Active	6	2010	50000	1011
Ravi	Active	60	2015	800000	1156	Ramesh	Active	6	2010	50000	1011
Sarika	Active	60	2018	500000	1155	Ramesh	Active	6	2010	50000	1011
Neha	Active	48	2019	2100000	1150	Ramesh	Active	6	2010	50000	1011
Arman	Active	48	2010	2100000	1149	Ramesh	Active	6	2010	50000	1011
Kisan	Active	48	2010	1750000	1142	Ramesh	Active	6	2010	50000	1011
Switi	Active	12	1996	1450000	1136	Ramesh	Active	6	2010	50000	1011
Veer	Active	12	2024	1300000	1133	Ramesh	Active	6	2010	50000	1011
Kabir	Active	6	2005	1150000	1130	Ramesh	Active	6	2010	50000	1011
Aryan	Active	6	2008	5000000	1127	Ramesh	Active	6	2010	50000	1011
krishna	Active	6	2022	6500000	1124	Ramesh	Active	6	2010	50000	1011
Advik	Active	6	2010	6400000	1115	Ramesh	Active	6	2010	50000	1011
Samuel	Active	36	2010	6300000	1114	Ramesh	Active	6	2010	50000	1011

18. Fetch all the records of **ACCOUNT DETAILS** table using view.

QUERY-CREATE VIEW LOAN_USER_DETAILS AS SELECT * FROM ACCOUNT_DETAILS;

OUTPUT-

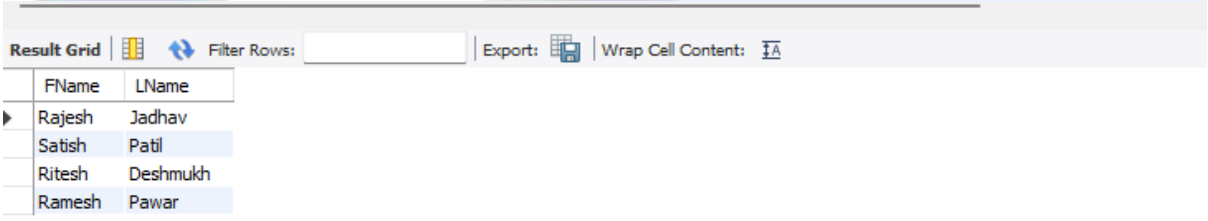


Transaction_id	Transaction_type	FName	LName	Account_Balance	Account_No	IFSC_code	CVV
TID-11	Deposit	Vikas	Patil	25000	658976534	ICIC0000424	235
TID-12	Withdraw	Rajesh	Jadhav	50000	658998534	ICIC0000426	432
TID-13	Deposit	Satish	Patil	25000	658978834	ICIC0000428	354
TID-14	Withdraw	Vinod	Potdar	25000	658976532	ICIC0000430	556
TID-15	Withdraw	Vinayak	Chavan	25000	658976994	ICIC0000440	395
TID-16	Deposit	Jack	Willian	35000	908976979	ICIC0000490	554
TID-17	Deposit	Ritesh	Deshmukh	55000	958976965	ICIC0000640	213
TID-18	Deposit	Ramesh	Pawar	80000	668976912	ICIC0000569	956
TID-19	Withdraw	Suraj	Chavan	67000	778976989	ICIC0000987	887
TID-20	Withdraw	Abhijit	Sawant	20000	788976955	ICIC0000342	998
TID-21	Deposit	Veer	Kole	90000	678976454	ICIC0000980	365
TID-22	Withdraw	Arnav	Chavre	75000	655576994	ICIC0000866	333
TID-23	Deposit	Shubham	Yeldare	87000	888976994	ICIC000655	355
TID-24	Deposit	Sarth	Shetthi	23000	645976994	ICIC0000475	665
TID-25	Withdraw	Sonam	Sawant	25000	657676994	ICIC0000480	897
TID-26	Withdraw	Monali	Shingare	54000	958976994	ICIC0000554	321
TID-27	Deposit	Pallavi	Gove	28000	658967994	ICIC0000598	856

19. Display the records of the ACCOUNT DETAILS table where First Name should end with ‘sh’. Query must return First Name column and Last Name column.

QUERY- SELECT FNAME, LNAME FROM ACCOUNT_DETAILS WHERE FNAME LIKE '%SH';

OUTPUT-



FName	LName
Rajesh	Jadhav
Satish	Patil
Ritesh	Deshmukh
Ramesh	Pawar

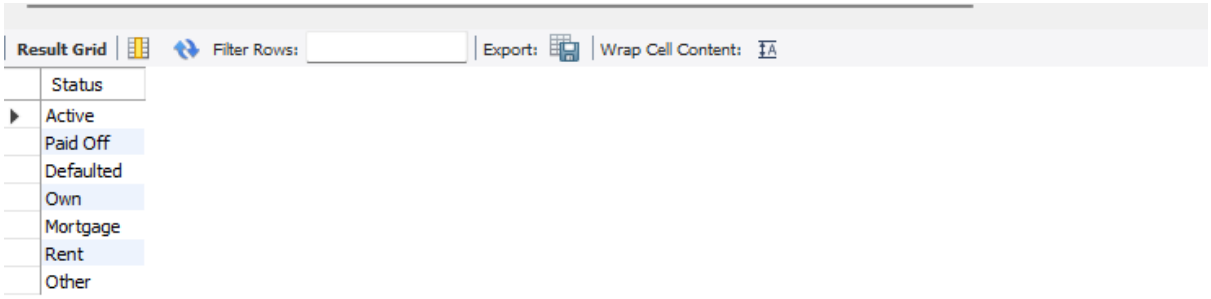
20. Write a query it returns Status from **BORROWER** table and Home Owner Ship Status from **LOANS** table using **UNION**. Query must return only distinct values.

QUERY- SELECT DISTINCT STATUS FROM BORROWER

UNION

SELECT HOMEOWNERSHIPSTATUS FROM LOANS;

OUTPUT-



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

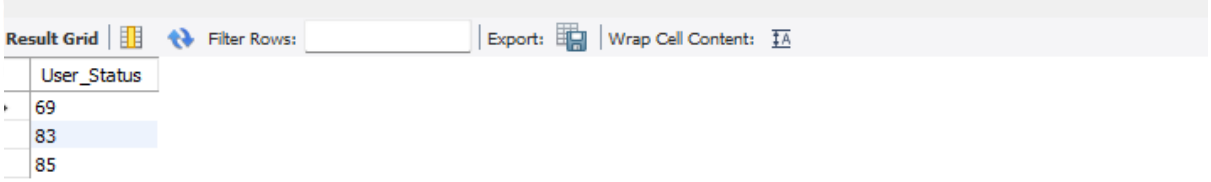
Status
Active
Paid Off
Defaulted
Own
Mortgage
Rent
Other

21. Return the ASCII value of the first character in Employment Status column from **LOAN** table. Do not return any duplicate values.

QUERY- SELECT DISTINCT ASCII (EMPLOYMENTSTATUS) AS USER_STATUS

FROM LOANS;

OUTPUT-



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

User_Status
69
83
85

22. Query the list of First Name and Last Name from **ACCOUNT DETAILS** table that do not start with vowels. result cannot contain duplicates.

QUERY-SELECT DISTINCT FNAME, LNAMEFROM ACCOUNT_DETAILS

WHERE FNAME NOT LIKE 'A%'

AND FNAME NOT LIKE 'E%'

AND FNAME NOT LIKE 'I%'

AND FNAME NOT LIKE 'O%'

AND FNAME NOT LIKE 'U%';

OUTPUT-

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FName	LName
Vikas	Patil
Rajesh	Jadhav
Satish	Patil
Vinod	Potdar
Vinayak	Chavan
Jack	Willian
Ritesh	Deshmukh
Ramesh	Pawar
Suraj	Chavan
Veer	Kole
Shubham	Yeldare
Sarth	Shetthi

23.