BANKING DATABASE PROJECT REPORT

Abhijeet Deshmukh || Akhil Prasad || Akshay Chougule

PROBLEM STATEMENT

Creating a banking database system where a customer opens a bank account providing all the details required by the bank for account creation. The banks find it hard to retrieve customer information based on only basic parameters like customer name, account number, etc. By creating relations between entities we can eliminate redundant data as well as handling of similar information. Identification of customers is another issue faced. By using more number of parameters we can ease the identification of customers.

OBJECTIVES OF THE PROJECT-

- Specify relations between entities involved in a banking management system. One-to-many, many-to-one, etc.

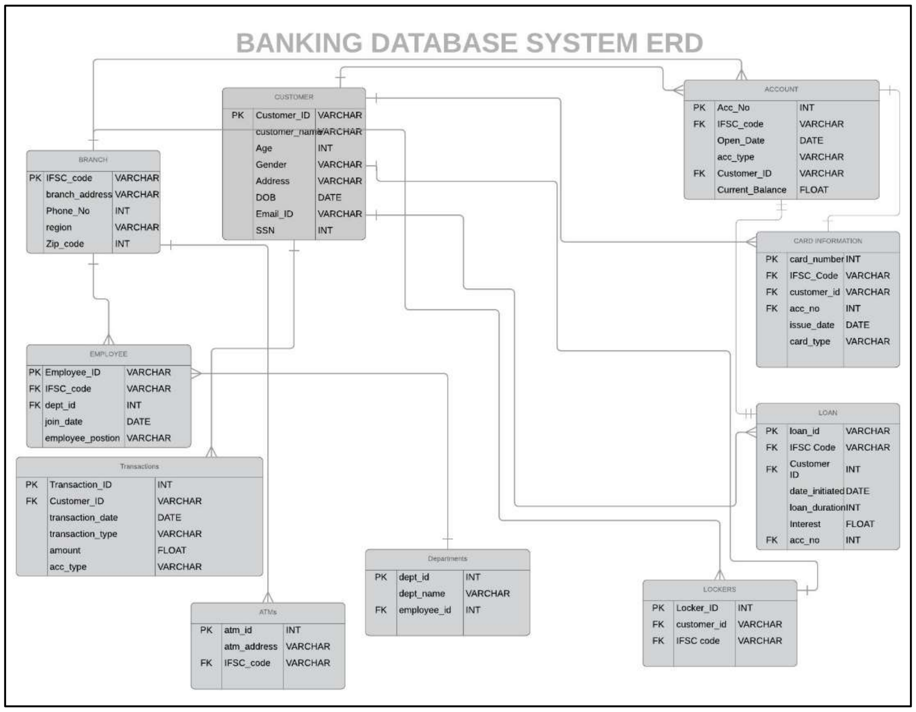
- Understand how the banking systems work.

- Storing and operating on the information using SQL queries.

- Determining the active and inactive customers based on transaction dates.

- Understand the attributes involved in a banking system like account types, transaction history, etc.

E-R DIAGRAM

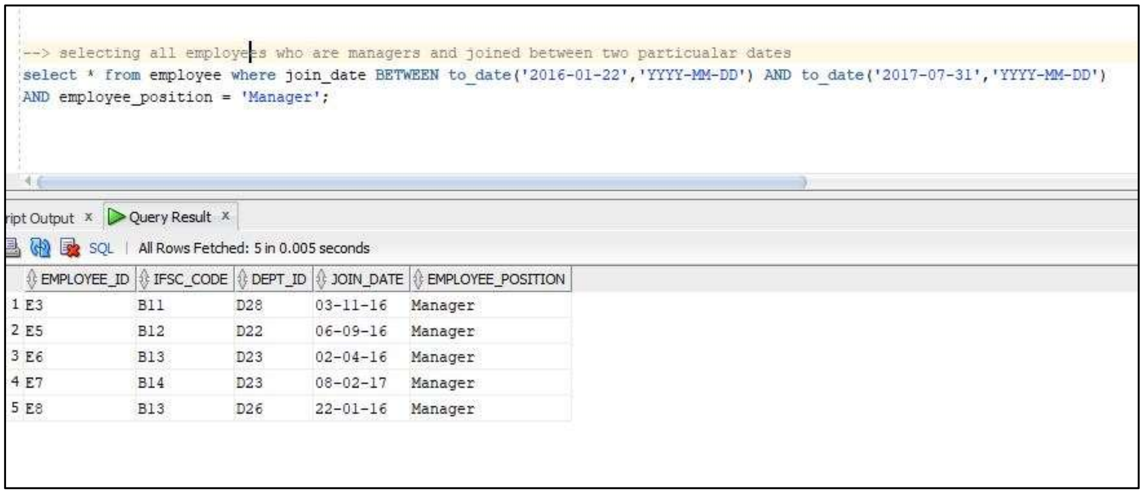


SQL QUERIES & OUTPUTS

The Following Queries Retrieve Important Information In Real World Banking System

1. **Selecting all employees who joined between two particular dates and employee position is manager**

*SELECT * from employee where join_date BETWEEN to_date('2016-01-22','YYYY-MM-DD') AND to_date('2017-07-31','YYYY-MM-DD') AND employee_position = 'Manager';*



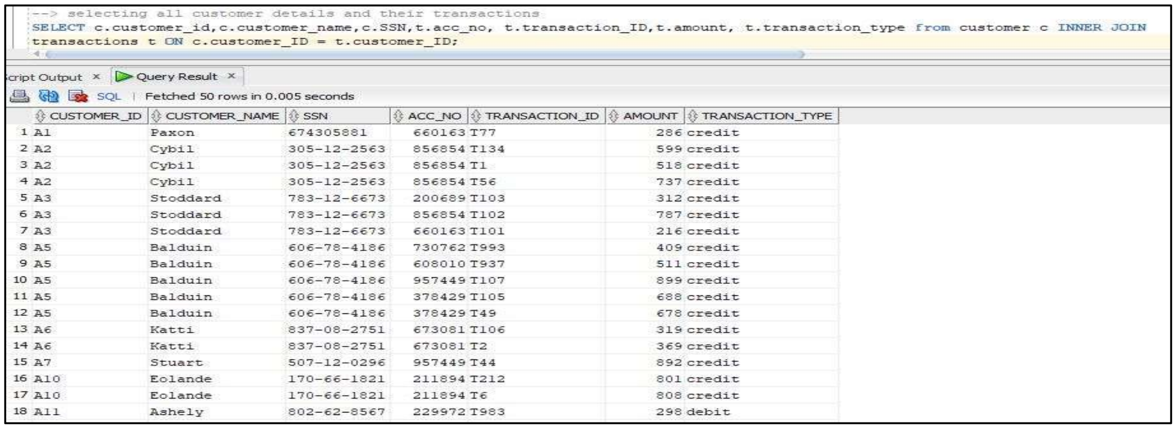
```
--> selecting all employees who are managers and joined between two particualar dates
select * from employee where join_date BETWEEN to_date('2016-01-22','YYYY-MM-DD') AND to_date('2017-07-31','YYYY-MM-DD')
AND employee_position = 'Manager';
```

All Rows Fetched: 5 in 0.005 seconds

	EMPLOYEE_ID	IFSC_CODE	DEPT_ID	JOIN_DATE	EMPLOYEE_POSITION
1	E3	B11	D28	03-11-16	Manager
2	E5	B12	D22	06-09-16	Manager
3	E6	B13	D23	02-04-16	Manager
4	E7	B14	D23	08-02-17	Manager
5	E8	B13	D26	22-01-16	Manager

2. **Selecting all customer details and their transactions**

SELECT c.customer_id,c.customer_name,c.SSN,t.acc_no, t.transaction_ID,t.amount, t.transaction_type from customer c INNER JOIN transactions t ON c.customer_ID = t.customer_ID



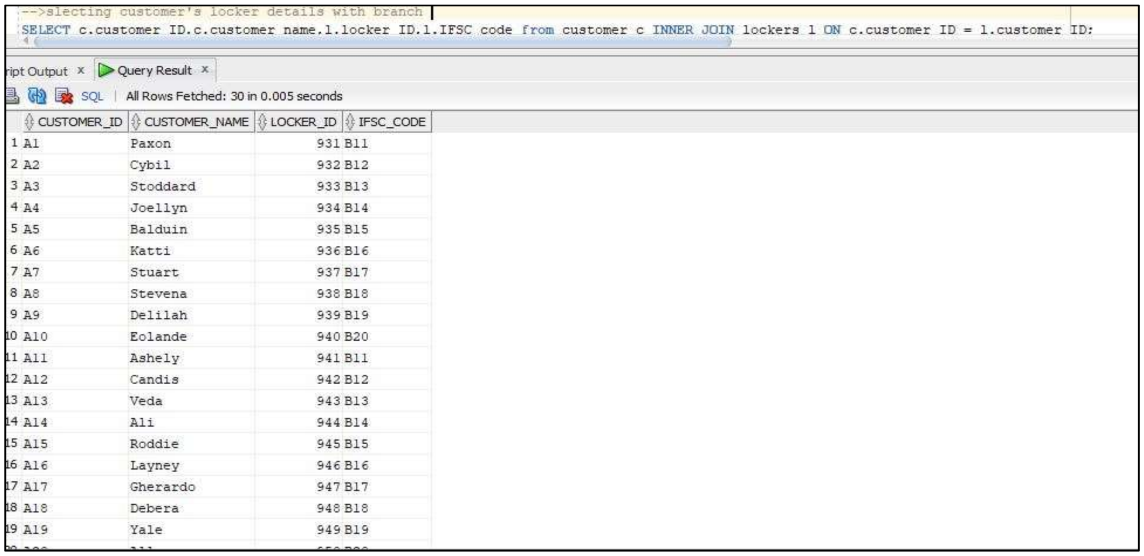
```
--> selecting all customer details and their transactions
SELECT c.customer_id,c.customer_name,c.SSN,t.acc_no, t.transaction_ID,t.amount, t.transaction_type from customer c INNER JOIN
transactions t ON c.customer_ID = t.customer_ID;
```

Fetched 50 rows in 0.005 seconds

	CUSTOMER_ID	CUSTOMER_NAME	SSN	ACC_NO	TRANSACTION_ID	AMOUNT	TRANSACTION_TYPE
1	A1	Paxon	674305881	660163	T77	286	credit
2	A2	Cybil	305-12-2563	856854	T134	599	credit
3	A2	Cybil	305-12-2563	856854	T1	518	credit
4	A2	Cybil	305-12-2563	856854	T56	737	credit
5	A3	Stoddard	783-12-6673	200689	T103	312	credit
6	A3	Stoddard	783-12-6673	856854	T102	787	credit
7	A3	Stoddard	783-12-6673	660163	T101	216	credit
8	A5	Balduin	606-78-4186	730762	T993	409	credit
9	A5	Balduin	606-78-4186	608010	T937	511	credit
10	A5	Balduin	606-78-4186	957449	T107	899	credit
11	A5	Balduin	606-78-4186	378429	T105	688	credit
12	A5	Balduin	606-78-4186	378429	T49	678	credit
13	A6	Katti	837-08-2751	673081	T106	319	credit
14	A6	Katti	837-08-2751	673081	T2	369	credit
15	A7	Stuart	507-12-0296	957449	T44	892	credit
16	A10	Eolande	170-66-1821	211894	T212	801	credit
17	A10	Eolande	170-66-1821	211894	T6	808	credit
18	A11	Ashely	802-62-8567	229972	T983	298	debit

3. **Selecting customer's locker details with branch**

SELECT c.customer_ID,c.customer_name,l.locker_ID,l.IFSC_code from customer c INNER JOIN lockers l ON c.customer_ID = l.customer_ID;



```
-->slecting customer's locker details with branch
SELECT c.customer ID.c.customer name.l.locker ID.l.IFSC code from customer c INNER JOIN lockers l ON c.customer ID = l.customer ID:
```

ript Output × Query Result ×

SQL | All Rows Fetched: 30 in 0.005 seconds

	CUSTOMER_ID	CUSTOMER_NAME	LOCKER_ID	IFSC_CODE
1	A1	Paxon	931	B11
2	A2	Cybil	932	B12
3	A3	Stoddard	933	B13
4	A4	Joellyn	934	B14
5	A5	Balduin	935	B15
6	A6	Katti	936	B16
7	A7	Stuart	937	B17
8	A8	Stevena	938	B18
9	A9	Delilah	939	B19
10	A10	Eolande	940	B20
11	A11	Ashely	941	B11
12	A12	Candis	942	B12
13	A13	Veda	943	B13
14	A14	Ali	944	B14
15	A15	Roddie	945	B15
16	A16	Layney	946	B16
17	A17	Gherardo	947	B17
18	A18	Debera	948	B18
19	A19	Yale	949	B19

4. **Selecting top 5 customers with highest number of transactions**

*SELECT * FROM (*

SELECT transactions.customer_ID,customer.customer_name, COUNT(transactions.customer_ID) AS Number_of_transactions

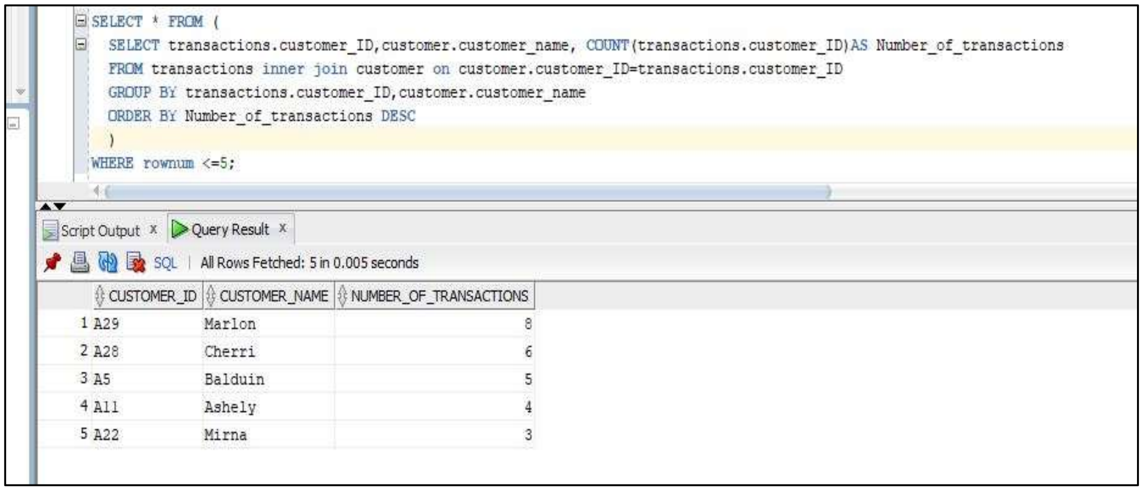
FROM transactions

GROUP BY customer_ID

ORDER BY COUNT(*) DESC

)

WHERE rownum < 5;



```
SELECT * FROM (
  SELECT transactions.customer_ID,customer.customer_name, COUNT(transactions.customer_ID)AS Number_of_transactions
  FROM transactions inner join customer on customer.customer_ID=transactions.customer_ID
  GROUP BY transactions.customer_ID,customer.customer_name
  ORDER BY Number_of_transactions DESC
  )
WHERE rownum <=5;
```

Script Output × Query Result ×

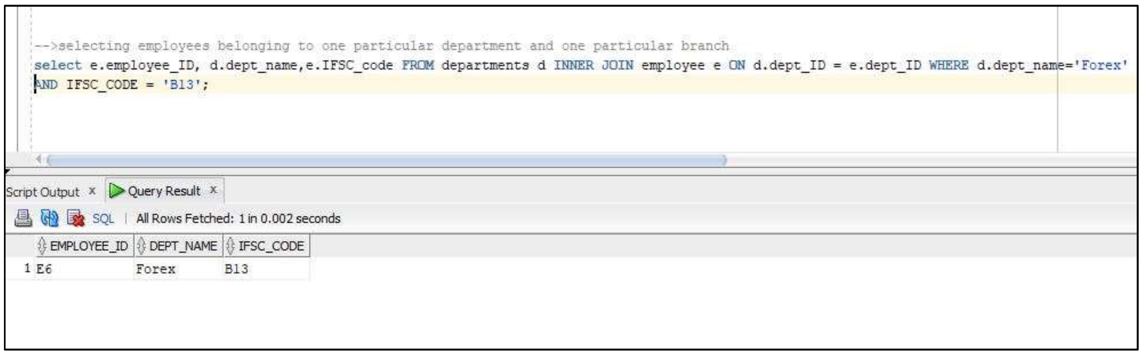
SQL | All Rows Fetched: 5 in 0.005 seconds

	CUSTOMER_ID	CUSTOMER_NAME	NUMBER_OF_TRANSACTIONS
1	A29	Marlon	8
2	A28	Cherri	6
3	A5	Balduin	5
4	A11	Ashely	4
5	A22	Mirna	3

5. **Selecting employees belonging to one particular department and one particular branch(Here we have chosen forex department and branch B13)**

SELECT e.employee_ID, d.dept_name,e.IFSC_code

FROM departments d INNER JOIN employee e ON d.dept_ID = e.dept_ID WHERE d.dept_name='Forex' AND IFSC_CODE = 'B13';



```
-->selecting employees belonging to one particular department and one particular branch
select e.employee_ID, d.dept_name,e.IFSC_code FROM departments d INNER JOIN employee e ON d.dept_ID = e.dept_ID WHERE d.dept_name='Forex'
AND IFSC_CODE = 'B13';
```

All Rows Fetched: 1 in 0.002 seconds

	EMPLOYEE_ID	DEPT_NAME	IFSC_CODE
1	E6	Forex	B13

6. **Customers with balance amount below 5000000 should raise their accounts and above 8000000 are gold members of the bank**

SELECT customer_ID, current_balance, acc_no,

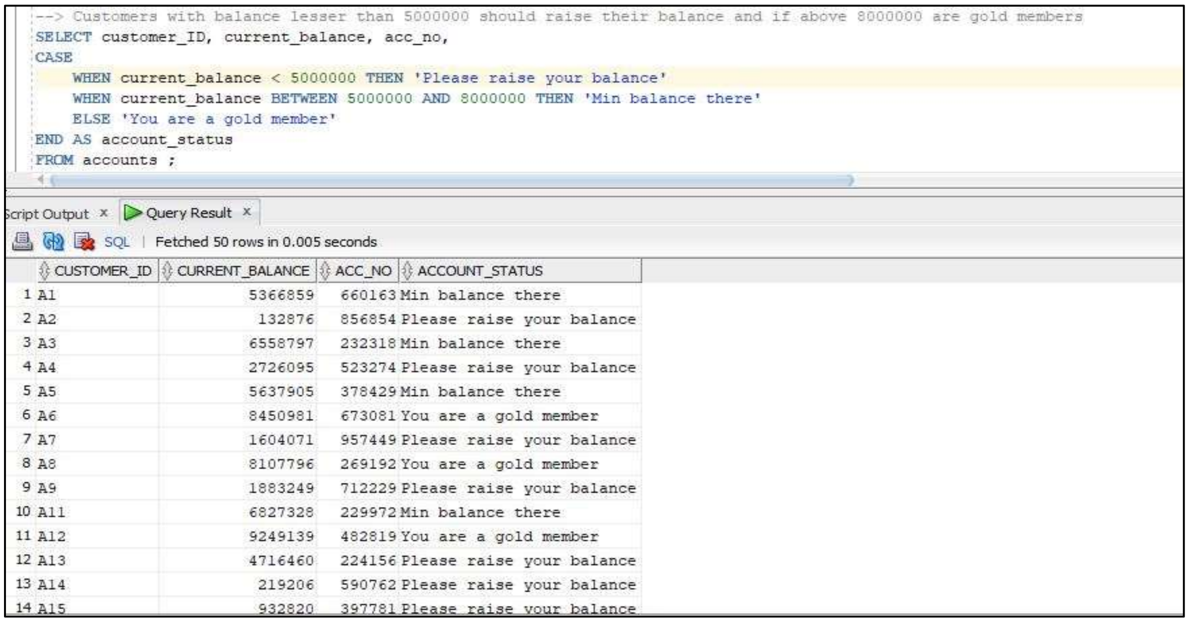
CASE

WHEN current_balance < 5000000 THEN 'Please raise your balance'

WHEN current_balance BETWEEN 5000000 AND 8000000 THEN 'Min balance there'

ELSE 'You are a gold member'

END AS account_status FROM accounts ;



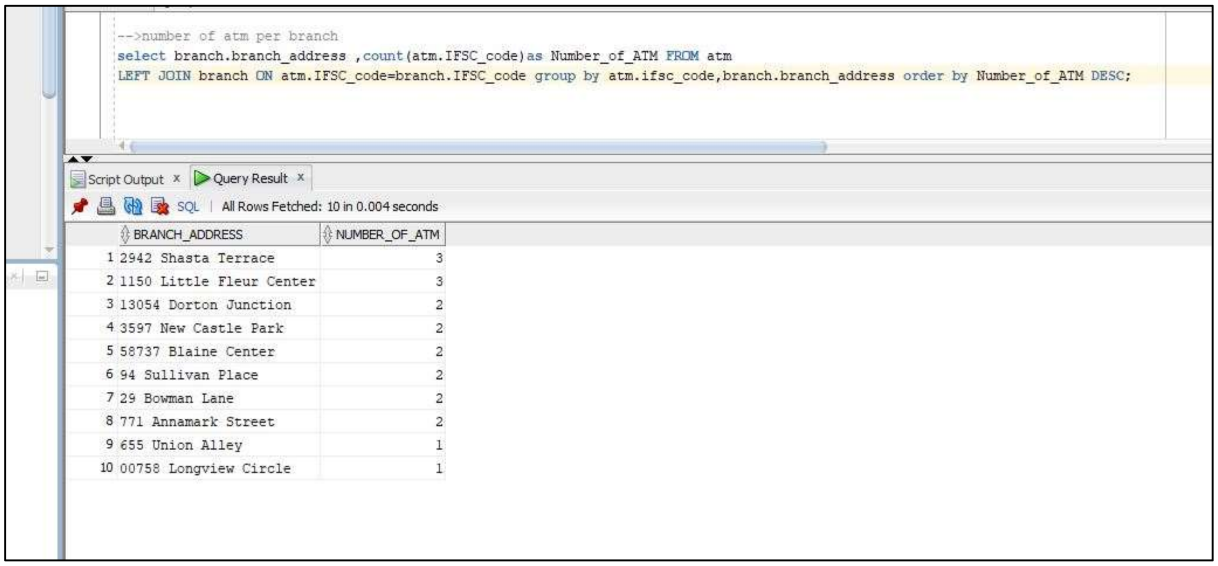
```
--> Customers with balance lesser than 5000000 should raise their balance and if above 8000000 are gold members
SELECT customer_ID, current_balance, acc_no,
CASE
    WHEN current_balance < 5000000 THEN 'Please raise your balance'
    WHEN current_balance BETWEEN 5000000 AND 8000000 THEN 'Min balance there'
    ELSE 'You are a gold member'
END AS account_status
FROM accounts ;
```

Fetched 50 rows in 0.005 seconds

	CUSTOMER_ID	CURRENT_BALANCE	ACC_NO	ACCOUNT_STATUS
1	A1	5366859	660163	Min balance there
2	A2	132876	856854	Please raise your balance
3	A3	6558797	232318	Min balance there
4	A4	2726095	523274	Please raise your balance
5	A5	5637905	378429	Min balance there
6	A6	8450981	673081	You are a gold member
7	A7	1604071	957449	Please raise your balance
8	A8	8107796	269192	You are a gold member
9	A9	1883249	712229	Please raise your balance
10	A11	6827328	229972	Min balance there
11	A12	9249139	482819	You are a gold member
12	A13	4716460	224156	Please raise your balance
13	A14	219206	590762	Please raise your balance
14	A15	932820	397781	Please raise your balance

7. **Determining number of atms for every branch**

SELECT branch.branch_address , COUNT(atm.IFSC_code) as Number_of_ATM
FROM
atm
LEFT JOIN
branch
ON
atm.IFSC_code=branch.IFSC_code
GROUP BY atm.ifsc_code,branch.branch_address
ORDER BY Number_of_ATM DESC;



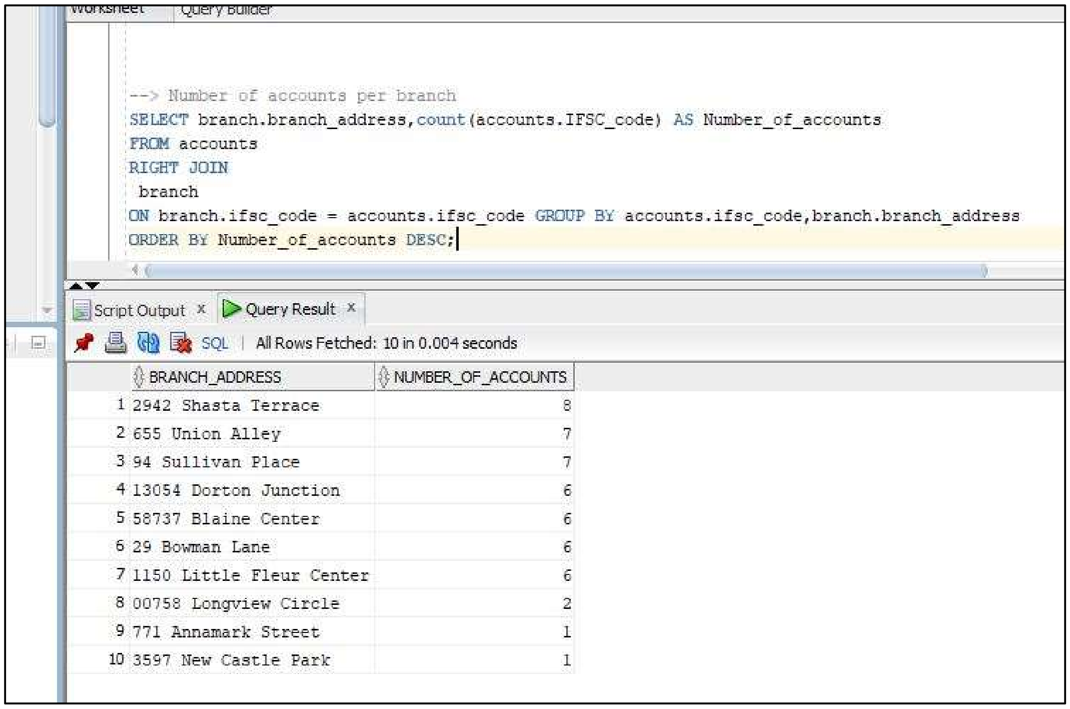
```
-->number of atm per branch
select branch.branch_address ,count(atm.IFSC_code)as Number_of_ATM FROM atm
LEFT JOIN branch ON atm.IFSC_code=branch.IFSC_code group by atm.ifsc_code,branch.branch_address order by Number_of_ATM DESC;
```

All Rows Fetched: 10 in 0.004 seconds

	BRANCH_ADDRESS	NUMBER_OF_ATM
1	2942 Shasta Terrace	3
2	1150 Little Fleur Center	3
3	13054 Dorton Junction	2
4	3597 New Castle Park	2
5	58737 Blaine Center	2
6	94 Sullivan Place	2
7	29 Bowman Lane	2
8	771 Annamark Street	2
9	655 Union Alley	1
10	00758 Longview Circle	1

8. **Determining number of accounts in each branch(descending order)**

SELECT branch.branch_address,count(accounts.IFSC_code) AS Number_of_accounts
FROM accounts
RIGHT JOIN
branch
ON branch.ifsc_code = accounts.ifsc_code GROUP BY
accounts.ifsc_code,branch.branch_address
ORDER BY Number_of_accounts DESC;



```
--> Number of accounts per branch
SELECT branch.branch_address,count(accounts.IFSC_code) AS Number_of_accounts
FROM accounts
RIGHT JOIN
 branch
ON branch.ifsc_code = accounts.ifsc_code GROUP BY accounts.ifsc_code,branch.branch_address
ORDER BY Number_of_accounts DESC;
```

	BRANCH_ADDRESS	NUMBER_OF_ACCOUNTS
1	2942 Shasta Terrace	8
2	655 Union Alley	7
3	94 Sullivan Place	7
4	13054 Dorton Junction	6
5	58737 Blaine Center	6
6	29 Bowman Lane	6
7	1150 Little Fleur Center	6
8	00758 Longview Circle	2
9	771 Annamark Street	1
10	3597 New Castle Park	1

9. Gathering customer details, account information and card details of all customers

SELECT c.customer_id, c.customer_name, c.email_id, c.SSN, a.acc_no, a.acc_type, a.current_balance,

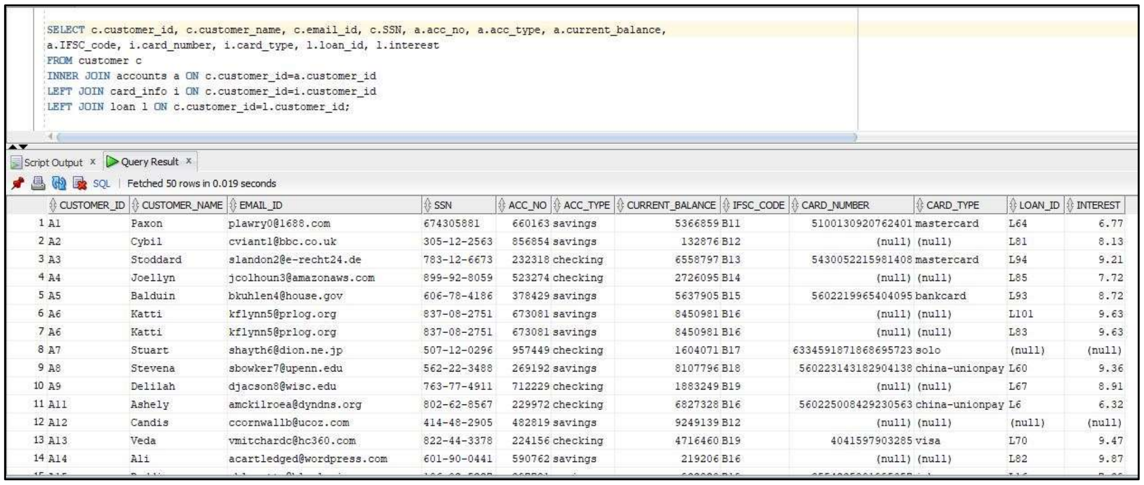
a.IFSC_code, i.card_number, i.card_type, l.loan_id, l.interest

FROM customer c

INNER JOIN accounts a ON c.customer_id=a.customer_id

LEFT JOIN card_info i ON c.customer_id=i.customer_id

LEFT JOIN loan l ON c.customer_id=l.customer_id;



```sql
SELECT c.customer_id, c.customer_name, c.email_id, c.SSN, a.acc_no, a.acc_type, a.current_balance,
a.IFSC_code, i.card_number, i.card_type, l.loan_id, l.interest
FROM customer c
INNER JOIN accounts a ON c.customer_id=a.customer_id
LEFT JOIN card_info i ON c.customer_id=i.customer_id
LEFT JOIN loan l ON c.customer_id=l.customer_id;
```

Script Output × Query Result ×

SQL | Fetched 50 rows in 0.019 seconds

	CUSTOMER_ID	CUSTOMER_NAME	EMAIL_ID	SSN	ACC_NO	ACC_TYPE	CURRENT_BALANCE	IFSC_CODE	CARD_NUMBER	CARD_TYPE	LOAN_ID	INTEREST
1	A1	Paxon	plawry0@1688.com	674305881	660163	savings	5366859	B11	5100130920762401	mastercard	L64	6.77
2	A2	Cybil	cviant1@bbc.co.uk	305-12-2563	856854	savings	132876	B12	(null)	(null)	L81	8.13
3	A3	Stoddard	slandon2@e-recht24.de	783-12-6673	232318	checking	6558797	B13	5430052215981408	mastercard	L94	9.21
4	A4	Joellyn	jcolhoun3@amazonaws.com	899-92-8059	523274	checking	2726095	B14	(null)	(null)	L85	7.72
5	A5	Balduin	bkuhlen4@house.gov	606-78-4186	378429	savings	5637905	B15	5602219965404095	bankcard	L93	8.72
6	A6	Katti	kflynn5@prlog.org	837-08-2751	673081	savings	8450981	B16	(null)	(null)	L101	9.63
7	A6	Katti	kflynn5@prlog.org	837-08-2751	673081	savings	8450981	B16	(null)	(null)	L83	9.63
8	A7	Stuart	shayth6@dion.ne.jp	507-12-0296	957449	checking	1604071	B17	6334591871868695723	solo	(null)	(null)
9	A8	Stevena	sbowker7@upenn.edu	562-22-3488	269192	savings	8107796	B18	560223143182904138	china-unionpay	L60	9.36
10	A9	Delilah	djacson8@wisc.edu	763-77-4911	712229	checking	1883249	B19	(null)	(null)	L67	8.91
11	A11	Ashely	amckilroea@dyndns.org	802-62-8567	229972	checking	6827328	B16	560225008429230563	china-unionpay	L6	6.32
12	A12	Candis	ccornwallb@ucoz.com	414-48-2905	482819	savings	9249139	B12	(null)	(null)	(null)	(null)
13	A13	Veda	vmitchardc@hc360.com	822-44-3378	224156	checking	4716460	B19	4041597903285	visa	L70	9.47
14	A14	Ali	acartledged@wordpress.com	601-90-0441	590762	savings	219206	B16	(null)	(null)	L82	9.87