1) **Find all branch names and cities with assets more than 1000000. (on single table)**

```
--1
SELECT branch_name,branch_city
FROM branch
WHERE assets > 1000000;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.002 seconds

	BRANCH_NAME	BRANCH_CITY
1	Redwood	Palo Alto
2	Perryridge	Horseneck
3	Round Hill	Horseneck
4	North Town	Rye
5	Brighton	Brooklyn

2) Find all account numbers and their balance which are opened in 'Downtown' branch or which have balance in between 600 and 750. (on single table)

```
--2
SELECT account_number,balance
FROM account
WHERE branch_name = 'Downtown'
or balance >= 600 and balance <= 750;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.003 seconds

	ACCOUNT_NUMBER	BALANCE
1	A-101	500
2	A-215	700
3	A-222	700
4	A-217	750
5	A-444	625

3) Find all account numbers which are opened in a branch located in 'Rye' city. (multiple tables)

```
--3
SELECT account_number
FROM account NATURAL JOIN branch
WHERE branch_city = 'Rye';
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.009 seconds

	ACCOUNT_NUMBER
1	A-333
2	A-444

4) Find all loan numbers which have amount greater than or equal to 1000 and their customers are living in 'Harrison' city. (multiple tables)

```
--4
SELECT Loan.loan_number FROM Loan
JOIN Borrower ON Loan.loan_number=Borrower.loan_number
JOIN Customer ON Customer.customer_name=Borrower.customer_name
WHERE amount>=1000 AND customer_city='Harrison';
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.01 seconds

	LOAN_NUMBER
1	L-17
2	L-15

5) Display the account related information based on the descending order of the balance. (order by clause)

```
--5
SELECT * FROM Account
ORDER BY balance DESC;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 9 in 0.002 seconds

	ACCOUNT_NUMBER	BRANCH_NAME	BALANCE
1	A-201	Perryridge	900
2	A-333	Central	850
3	A-217	Brighton	750
4	A-215	Mianus	700
5	A-222	Redwood	700
6	A-444	North Town	625
7	A-101	Downtown	500
8	A-102	Perryridge	400
9	A-305	Round Hill	350

6) Display the customer related information in alphabetic order of customer cities. (order by clause)

```
--6
SELECT * FROM Customer
ORDER BY customer_city;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 15 in 0.004 seconds

	CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
1	Brooks	Senator	Brooklyn
2	Hayes	Main	Harrison
3	Jones	Main	Harrison
4	Johnson	Alma	Palo Alto
5	Adams	Spring	Pittsfield
6	Lindsay	Park	Pittsfield
7	Williams	Nassau	Princeton
8	Curry	North	Rye
9	McBride	Safety	Rye
10	Smith	Main	Rye
11	Majeris	First	Rye
12	Jackson	University	Salt Lake
13	Green	Walnut	Stamford
14	Turner	Putnam	Stamford
15	Glenn	Sand Hill	Woodside

7) Find all customer names who have an account as well as a loan. (intersect)

```
--7
SELECT customer_name FROM Customer
NATURAL JOIN Depositor INTERSECT
SELECT customer_name FROM Customer
NATURAL JOIN Borrower;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 3 in 0.005 seconds

	CUSTOMER_NAME
1	Hayes
2	Jones
3	Smith

8) Find all customer related information who have an account or a loan. (union)

```
--8
SELECt customer_name,customer_city,customer_street
FROM Customer NATURAL JOIN Depositor UNION
SELECT customer_name,customer_city,customer_street
FROM Customer NATURAL JOIN Borrower;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 12 in 0.015 seconds

	CUSTOMER_NAME	CUSTOMER_CITY	CUSTOMER_STREET
1	Adams	Pittsfield	Spring
2	Curry	Rye	North
3	Hayes	Harrison	Main
4	Jackson	Salt Lake	University
5	Johnson	Palo Alto	Alma
6	Jones	Harrison	Main
7	Lindsay	Pittsfield	Park
8	Majeris	Rye	First
9	McBride	Rye	Safety
10	Smith	Rye	Main
11	Turner	Stamford	Putnam
12	Williams	Princeton	Nassau

9) Find all customer names and their cities who have a loan but not an account. (minus)

```
--9
SELECt customer_name,customer_city
FROM Customer NATURAL JOIN Borrower MINUS
SELECT customer_name,customer_city
FROM Customer NATURAL JOIN Depositor;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 5 in 0.001 seconds

	CUSTOMER_NAME	CUSTOMER_CITY
1	Adams	Pittsfield
2	Curry	Rye
3	Jackson	Salt Lake
4	McBride	Rye
5	Williams	Princeton

10) Find the total assets of all branches. (aggregate function)

```
--10
SELECT SUM(assets) AS TOTAL_ASSETS
FROM Branch;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 1 in 0.002 seconds

	TOTAL_ASSETS
1	24600480

11) Find the average balance of accounts at each branch. (aggregate function)

```
--11
SELECT branch_name, AVG(balance) AS AVERAGE_BALANCE
FROM Account GROUP BY branch_name;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4

SQL | All Rows Fetched: 8 in 0.003 seconds

	BRANCH_NAME	AVERAGE_BALANCE
1	Central	850
2	Downtown	500
3	Perryridge	650
4	Mianus	700
5	North Town	625
6	Round Hill	350
7	Redwood	700
8	Brighton	750

12) Find the average balance of accounts at each branch city. (aggregate function)

```
--12
SELECT branch_city, AVG(balance) AS AVERAGE_BALANCE
FROM Account NATURAL JOIN Branch GROUP BY branch_city;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4

SQL | All Rows Fetched: 4 in 0.012 seconds

	BRANCH_CITY	AVERAGE_BALANCE
1	Palo Alto	700
2	Brooklyn	625
3	Horseneck	587.5
4	Rye	737.5

13) Find the lowest amount of loan at each branch. (aggregate function)

```
--13
SELECT branch_name, MIN(amount) AS MINIMUM_AMOUNT
FROM Loan GROUP BY branch_name;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result

SQL | All Rows Fetched: 7 in 0.005 seconds

	BRANCH_NAME	MINIMUM_AMOUNT
1	Central	570
2	Downtown	1000
3	Perryridge	1300
4	Mianus	500
5	North Town	7500
6	Round Hill	900
7	Redwood	2000

14) Find the total number of loans at each branch. (aggregate function)

```
--14
SELECT branch_name, COUNT(*) AS total_loan
FROM Loan GROUP BY branch_name;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4

SQL | All Rows Fetched: 7 in 0.004 seconds

	BRANCH_NAME	TOTAL_LOAN
1	Central	1
2	Downtown	2
3	Perryridge	2
4	Mianus	1
5	North Town	1
6	Round Hill	1
7	Redwood	1

15) Find the customer name and account number of the account which has the highest balance. (aggregate function)

```
--15
SELECT customer_name,account_number
FROM Account NATURAL JOIN Depositor
WHERE balance=(SELECT MAX(balance)FROM Account);
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query R

SQL | All Rows Fetched: 1 in 0.006 seconds

	CUSTOMER_NAME	ACCOUNT_NUMBER
1	Johnson	A-201