

#SYCOA302 Ritish Shelke

Problem Statement:

Account(Acc_no, branch_name, balance)

branch(branch_name, branch_city, assets)

customer(cust_name, cust_street, cust_city)

Depositor(cust_name, acc_no)

Loan(loan_no, branch_name, amount)

Borrower(cust_name, loan_no)

select * from account;

Acc_no	branch_name	balance
901	Akurdi	200000
902	Deccan	1000.5
903	Kothrud	15785
904	Akurdi	20369
905	Akurdi	80123.6
906	Pimpri	10000
907	Chinchwad	1000000
908	Pimpri	6532.87
909	Akurdi	85500
910	Pimpri	6500

10 rows in set (0.03 sec)

select * from branch;

branch_name	branch_city
Akurdi	Pune
Chinchwad	Pune
Deccan	Pune
Kothrud	Pune
Pimpri	Pune

5 rows in set (0.01 sec)

select * from customer;

cust_name	cust_street	cust_city
Charisse Rolin	Mallory	Pune
Poppy Landis	Corben	Pune
Ronni Dinis	Kennedy	Pune
Levin Scambler	Elmside	Pune
Patti Berge	Weeping Birch	Pune
Koo Kennerley	Di Loreto	Pune
Travus Donke	Graedel	Pune
Bink Skellen	Grim	Pune
Rhianna Walhedd	Superior	Pune
Pace Bumphries	Karstens	Pune

10 rows in set (0.01 sec)

select * from depositor;

cust_name	acc_no
Charisse Rolin	901
Poppy Landis	902
Ronni Dinis	903
Levin Scambler	904
Patti Berge	905
Koo Kennerley	906
Travus Donke	907
Bink Skellen	908
Rhianna Walhedd	909
Pace Bumphries	910

10 rows in set (0.01 sec)

select * from loan;

loan_no	branch_name	amount
99901	Akurdi	200000
99902	Deccan	100050
99903	Kothrud	1578500
99904	Deccan	203609
99905	Akurdi	8012360
99906	Pimpri	1000000
99907	Chinchwad	999990
99908	Pimpri	653270
99909	Deccan	1400

9 rows in set (0.00 sec)

select * from borrower;

cust_name	loan_no
Charisse Rolin	99901
Poppy Landis	99902
Ronni Dinis	99903
Levin Scambler	99904
Patti Berge	99905
Koo Kennerley	99906
Travus Donke	99907
Travus Donke	99908

8 rows in set (0.01 sec)

Q1. Find the names of all branches in loan relation.

select distinct branch_name from loan;

```
+-----+
| branch_name |
+-----+
| Akurdi      |
| Akurdi      |
| Chinchwad   |
| Deccan      |
| Deccan      |
| Kothrud     |
| Pimpri      |
| Pimpri      |
+-----+
8 rows in set (0.03 sec)
```

Q2. Find all loan numbers for loans made at Akurdi Branch with loan amount > 12000.

select loan_no from loan where branch_name="Akurdi" and amount> 12000;

```
+-----+
| loan_no |
+-----+
| 99901   |
| 99905   |
+-----+
2 rows in set (0.00 sec)
```

Q3. Find all customers who have a loan from bank. Find their names, loan_no and loan amount.

select b.cust_name, l.loan_no, l.amount from borrower b inner join loan l on b.loan_no=l.loan_no;

```
+-----+-----+-----+
| cust_name | loan_no | amount |
+-----+-----+-----+
| Charisse Rolin | 99901 | 200000 |
| Poppy Landis   | 99902 | 100050 |
| Ronni Dinis    | 99903 | 1578500 |
| Levin Scambler | 99904 | 203609 |
| Patti Berge    | 99905 | 8012360 |
| Koo Kennerley  | 99906 | 1000000 |
| Travus Donke   | 99907 | 999990 |
| Travus Donke   | 99908 | 653270 |
+-----+-----+-----+
8 rows in set (0.00 sec)
```

Q4. List all customers in alphabetical order who have loan from Akurdi branch.

select cust_name from borrower where loan_no in (select loan_no from loan where branch_name="Akurdi") order by cust_name;

```
+-----+
| cust_name |
+-----+
| Charisse Rolin |
| Patti Berge |
+-----+
2 rows in set (0.00 sec)
```

Q5. Find all customers who have an account or loan or both at bank.

select cust_name from depositor union select cust_name from borrower;

```
+-----+
| cust_name |
+-----+
| Charisse Rolin |
| Poppy Landis |
| Ronni Dinis |
| Levin Scambler |
| Patti Berge |
| Koo Kennerley |
| Travus Donke |
| Bink Skellen |
| Rhianna Walhedd |
| Pace Bumphries |
+-----+
10 rows in set (0.00 sec)
```

Q6. Find all customers who have both account and loan at bank.

select cust_name from depositor intersect select cust_name from borrower;

```
+-----+
| cust_name |
+-----+
| Charisse Rolin |
| Poppy Landis |
| Ronni Dinis |
| Levin Scambler |
| Patti Berge |
| Koo Kennerley |
| Travus Donke |
+-----+
7 rows in set (0.00 sec)
```

Q7. Find all customer who have account but no loan at the bank.

select cust_name from depositor minus select cust_name from borrower;

```
+-----+
| cust_name |
+-----+
| Bink Skellen |
| Rhianna Walhedd |
| Pace Bumphries |
+-----+
3 rows in set (0.00 sec)
```

Q8. Find average account balance at Akurdi branch.

select avg(balance) from Account where branch_name="Akurdi";

```
+-----+
| avg(balance) |
+-----+
| 96498.14453125 |
+-----+
1 row in set (0.00 sec)
```

Q9. Find the average account balance at each branch

select branch_name,avg(balance) from Account group by branch_name;

```
+-----+-----+
| branch_name | avg(balance) |
+-----+-----+
| Akurdi      | 96498.14453125 |
| Chinchwad   | 1000000      |
| Deccan      | 1000.5        |
| Kothrud     | 15785         |
| Pimpri      | 7677.623372395833 |
+-----+-----+
5 rows in set (0.00 sec)
```

Q10. Find no. of depositors at each branch.

```
select a.branch_name, count(*) from Account a inner join depositor b on a.acc_no=b.acc_no
group by a.branch_name;
```

branch_name	count(*)
Akurdi	4
Chinchwad	1
Deccan	1
Kothrud	1
Pimpri	3

5 rows in set (0.00 sec)

Q11. Find the branches where average account balance > 12000.

```
select branch_name from Account group by branch_name having avg(balance)>12000;
```

branch_name
Akurdi
Chinchwad
Kothrud

3 rows in set (0.01 sec)

Q12. Find number of tuples in customer relation.

```
select count(*) from customer;
```

count(*)
10

1 row in set (0.01 sec)

Q13. Calculate total loan amount given by bank.

```
select sum(amount) as Total_Loan_amount from loan;
```

Total_Loan_amount
12747777

1 row in set (0.00 sec)

Q14. Delete all loans with loan amount between 1300 and 1500.

delete from loan where amount between 1300 and 1500;

loan_no	branch_name	amount
99901	Akurdi	200000
99902	Deccan	100050
99903	Kothrud	1578500
99904	Deccan	203609
99905	Akurdi	8012360
99906	Pimpri	1000000
99907	Chinchwad	999990
99908	Pimpri	653270

8 rows in set (0.00 sec)

Q15. Delete all tuples at every branch located in Nigdi.

DELETE FROM branch WHERE branch_name='Nigdi';

Q.16. Create synonym for customer table as cust.

CREATE SYNONYM cust FOR Customer;

Q.17. Create sequence roll_seq and use in student table for roll_no column.

create sequence roll_sequence

start with 1

increment by 1

minvalue 1

maxvalue 100

nocycle;

SQL> insert into student values(roll_sequence.nextval, 'Ritish');

1 row created.