

Create the following tables with given attributes having appropriate data type and specify the necessary primary and foreign key constraints:

Customer (Custid, Custname, Age, Phone)

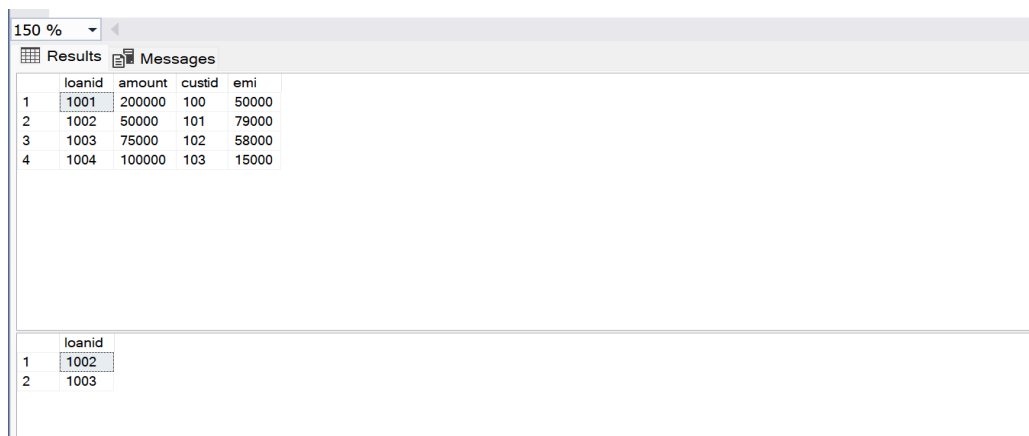
Loan (Loanid, Amount, Custid, EMI)

Account (Acno, Custid, Balance)

- List the Loanid of Loans with EMI more than Rs.50,000.
- List the EMI and number of loans with that loan amount.
- Create a view to list the total number of loans availed.
- Display the EMI amount of Customer "Smith".
- Create a procedure to print the Amount and Custid of Loanid 1001.
- Create a function to display the loan amount of customer with customerid 100.

```
create table customer(Custid int primary key,custname nvarchar(20) not null,Age int,
phone int);
create table loan(loanid int primary key,amount int not null,custid int foreign key
references customer, emi int);
create table account(acno int primary key,custid int foreign key references
customer,balance int );
```

- ```
select * from loan;
select loanid from loan where emi>50000;
```



150 %

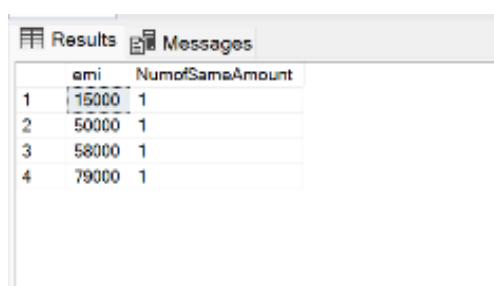
Results Messages

|   | loanid | amount | custid | emi   |
|---|--------|--------|--------|-------|
| 1 | 1001   | 200000 | 100    | 50000 |
| 2 | 1002   | 50000  | 101    | 79000 |
| 3 | 1003   | 75000  | 102    | 58000 |
| 4 | 1004   | 100000 | 103    | 15000 |

loanid

|   |      |
|---|------|
| 1 | 1002 |
| 2 | 1003 |

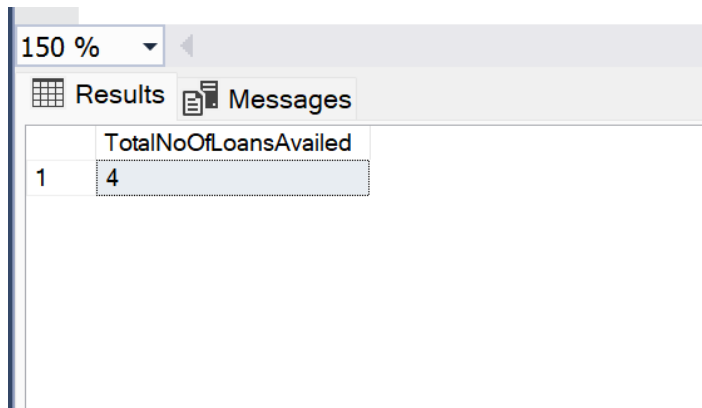
- ```
select emi,count(emi)as NumofSameAmount
from loan
group by emi
order by emi;
```



Results Messages

	emi	NumofSameAmount
1	15000	1
2	50000	1
3	58000	1
4	79000	1

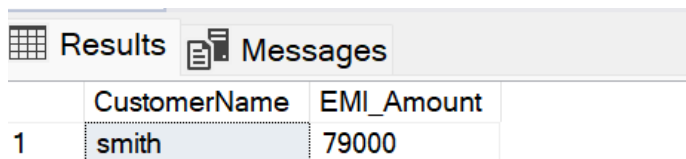
c) `select count(loanid) as TotalNoOfLoansAvailed
from loan;`



The screenshot shows a SQL Server query results window. At the top, there is a zoom level dropdown set to '150 %'. Below it are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a single row of data. The column header is 'TotalNoOfLoansAvailed' and the value in the row is '4'.

	TotalNoOfLoansAvailed
1	4

d) `select c.custname CustomerName, l.emi EMI_Amount
from customer c join loan l
on c.Custid=l.custid
where custname='smith';`

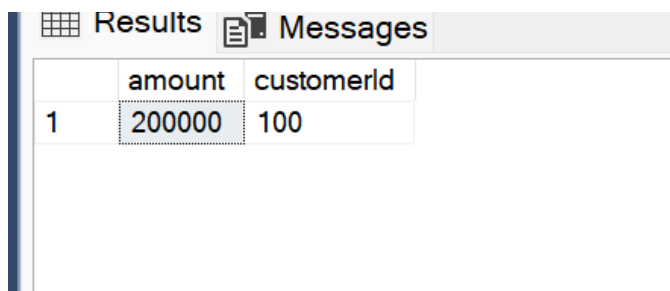


The screenshot shows a SQL Server query results window. At the top, there are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a single row of data. The column headers are 'CustomerName' and 'EMI_Amount'. The values in the row are 'smith' and '79000' respectively.

	CustomerName	EMI_Amount
1	smith	79000

e) `CREATE PROCEDURE usp_loandisplay`



```
AS  
BEGIN  
    SELECT l.amount ,c.custid customerId from customer c full outer join  
loan l  
    on c.custid=l.custid  
    where loanid=1001;  
END  
GO  
  
exec dbo.usp_loandisplay;
```



The screenshot shows a SQL Server query results window. At the top, there are two tabs: 'Results' (active) and 'Messages'. The 'Results' tab displays a single row of data. The column headers are 'amount' and 'customerId'. The values in the row are '200000' and '100' respectively.

	amount	customerId
1	200000	100

```
f) CREATE or alter FUNCTION udf_display
(
)
RETURNS TABLE
AS
RETURN
(
    SELECT amount as LoanAmount ,custid as CustomerId
    from loan
    where custid=100
)
GO
select * from udf_display();
```

<div>  Results </div> <div>  Messages </div>		
	LoanAmount	CustomerId
1	200000	100