

Report on Credit Card Management System

Database Management and Database Design

Spring 2018

By

Bhagyashree Dinkar Patil

INDEX

Sr. No.	Topic	Page No.
1	ER-DIAGRAM FOR CREDIT CARD TRANSACTION SYSTEM	3
2	CREDIT CARD TRANSACTION SYSTEM	4
3	VIEW	5
4	STORED PROCEDURES	7
5	TRIGGER FOR EMAIL VALIDATION	9
6	TRIGGER TO TRACK OF DELETED CREDIT CARDS	10
7	TRANSACTION	12
8	ACCESS PRIVILAGES	14
9	MYSQL TABLES	16

ER DIAGRAM ON CREDIT CARD MANAGEMENT SYSTEM

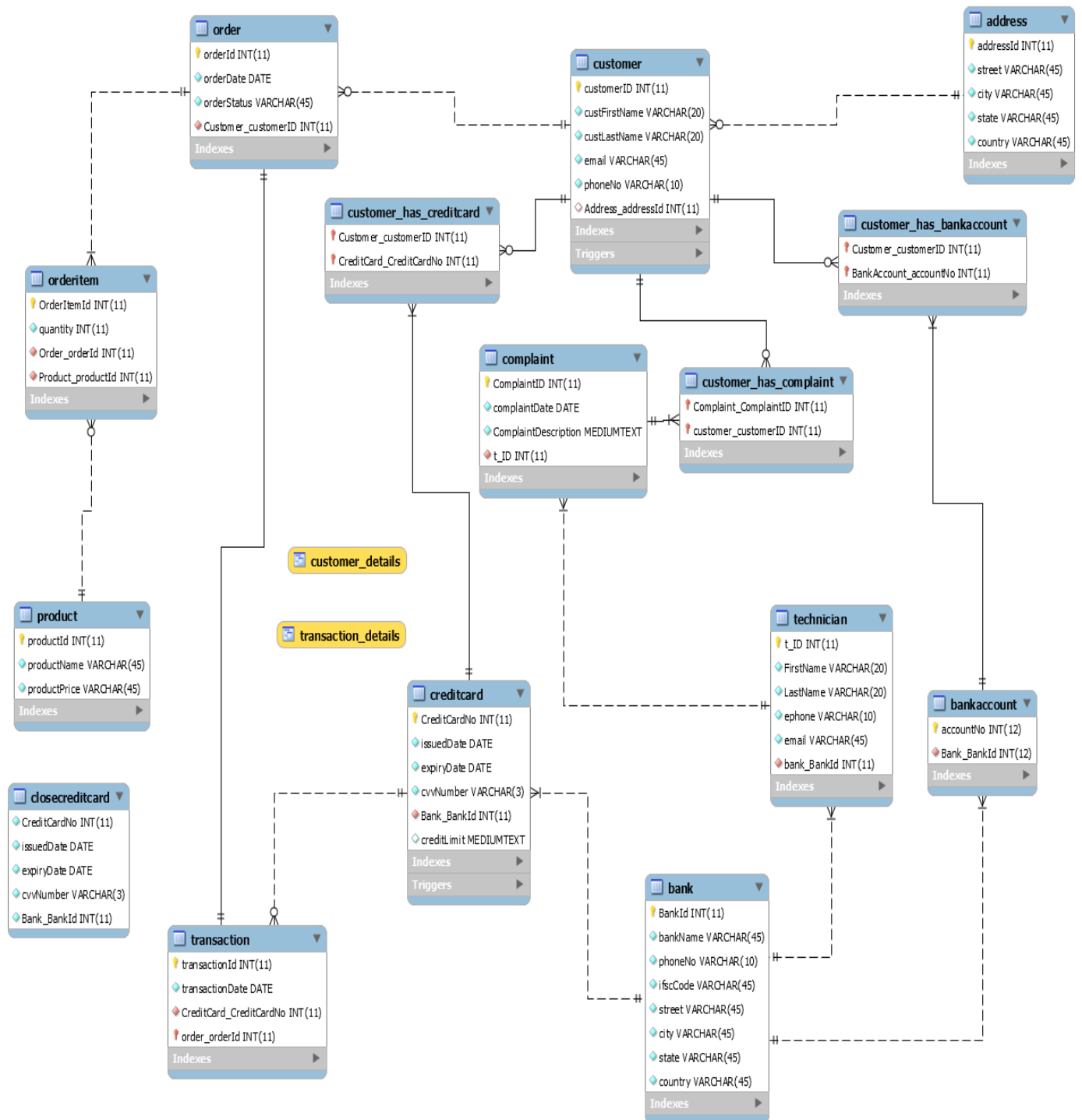


Fig1.ER diagram of credit card transaction system

CREDIT CARD MANGEMENT SYSTEM

Problem Statement: Create Database for credit card transaction system and to show relationships between different entity involved in it.

Approach: In credit card transaction system different Bank has many credit cards. Bank and creditcard has one-to-many cardinality. Customer can have zero or more credit cards of different banks. Each credit card has zero or more transactions. Each transaction contains Order. Each Order contains OrderItem. Each orderItem contains Product details. Bank has also technicians who solves customers complaints about creditcard.

Creditcard transaction system has following entity type

1. BANK
2. BANKACCOUNT
3. CREDITCARD
4. CUSTOMER
5. ADDRESS
6. CUSTOMER_HAS_BANKACCOUNT (Associative entity)
7. CUSTOMER_HAS_CREDITCARD (Associative entity)
8. TRANSACTION
9. ORDER
10. ORDERITEM
11. PRODUCT
12. TECHNICIAN
13. COMPLAINT
14. CUSTOMER_HAS_COMPLAINT (Associative entity)

.

VIEW

Delimiter \$\$

CREATE VIEW **Transaction_Details**

AS

select customerid, custFirstName, custLastName, email, customer.phoneNo, address.city,
bankName,

customer_has_creditcard.CreditCard_CreditCardNo as

creditCardNo,creditlimit,transactionDate,transaction.Order_orderId as

orderid,productName,productPrice,quantity,orderStatus

from customer inner join address on customer.customerID=address.addressId inner join

customer_has_creditcard on

customer.customerID=customer_has_creditcard.Customer_customerID inner join

creditcard on customer_has_creditcard.CreditCard_CreditCardNo=creditcard.CreditCardNo
inner join

bank on creditcard.Bank_BankId=bank.BankId inner join

`transaction` on creditcard.CreditCardNo=`transaction`.CreditCard_CreditCardNo inner join

`order` on customer.customerID=`order`.Customer_customerID inner join orderitem on

`order`.orderId=orderitem.Order_orderId

inner join product on orderitem.Product_productId=product.productId;





\$\$

```
SELECT * FROM Transaction_Details;
```

12

13 • `SELECT * FROM Transaction_Details;`

<

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	customerid	custFirstName	custLastName	email	phoneNo	city	bankName	creditCardNo	creditlimit	transactionDate	orderid	productName	productPrice	quantity	orderStatus
	2	Pankaj	Salunkhe	pi@yahoo.com	5513503639	Boston	Bank of America	56676767	8000	2017-01-01	11	mobile	500	2	Deliver
	3	Ashish	Patil	ashish@yahoo.com	8867564534	NewJersev	Santander	78899087	9500	2018-01-01	22	tv	100	5	Delivered
	4	Sonali	Patel	soni@gmail.com	5513583639	Boston	TDBank	45566778	10000	2018-03-03	33	camera	200	1	Delivered

STORED PROCEDURES

DELIMITER \$\$

```
CREATE procedure ADDRESS_UPDATE(IN street1 VARCHAR(45),IN city1 varchar(45), IN  
state1 varchar(45), IN country1 varchar(45),IN id int)
```

```
BEGIN
```

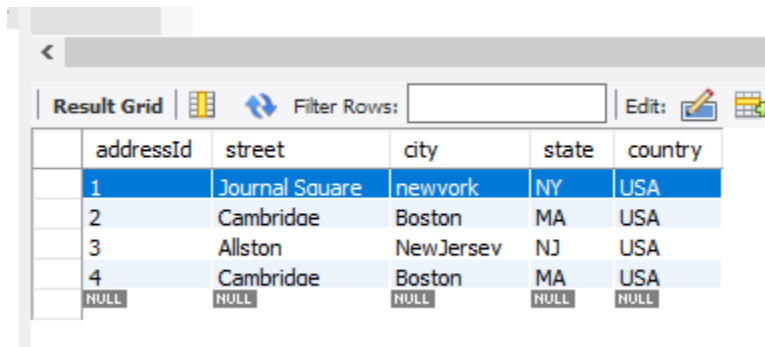
```
UPDATE Address SET street=street1 AND city=city1 AND state=state1 AND  
country=country1 where addressId=id;
```

```
END;
```

```
$$
```

-----Address table before procedure call-----

```
select * from address;
```



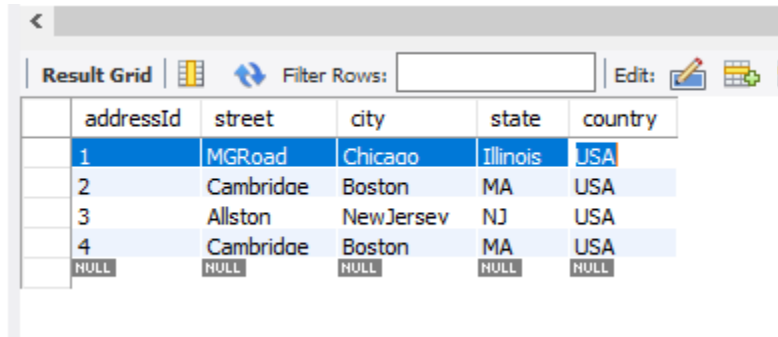
	addressId	street	city	state	country
1	1	Journal Square	newvork	NY	USA
2	2	Cambridge	Boston	MA	USA
3	3	Allston	NewJersev	NJ	USA
4	4	Cambridge	Boston	MA	USA
	NULL	NULL	NULL	NULL	NULL

----- Address table after procedure call-----

call ADDRESS_UPDATE ('MGRoad','Chicago','Illinois','USA',1);

select * from address;

(On execution of address_update() procedure address is updated where is equal to 1)




	addressId	street	city	state	country
1	MGRoad	Chicago	Illinois	USA	
2	Cambridge	Boston	MA	USA	
3	Allston	NewJersev	NJ	USA	
4	Cambridge	Boston	MA	USA	
	NULL	NULL	NULL	NULL	NULL

TRIGGER FOR EMAIL VALIDATION

```
DELIMITER $$  
CREATE TRIGGER email_validation  
BEFORE INSERT  
ON Customer  
FOR EACH ROW  
BEGIN  
    IF new.email NOT LIKE '%_@%_.__%'  
    THEN  
        SIGNAL SQLSTATE VALUE '45000'  
        SET MESSAGE_TEXT = 'Invalid email id';  
    END IF;  
END;  
$$
```

```
insert into customer (custFirstName, custLastName, email, phoneNo, address_addressid)  
values('Karishma','Patil','676767@ @ @ @','8867564534',2);
```

	10 20:23:11 insert into customer (custFirstName, custLastName, email, phoneNo, address_addressid) values('... Error Code: 1644. Invalid email id
---	--

TRIGGER TO TRACK OF DELETED CREDIT CARDS

```
CREATE TABLE CloseCreditCard (  
    CreditCardNo INT NOT NULL,  
    issuedDate DATE NOT NULL,  
    expiryDate DATE NOT NULL,  
    cvvNumber VARCHAR(3) NOT NULL,  
    Bank_BankId INT NOT NULL);
```

DELIMITER \$\$

```
CREATE TRIGGER CreditCard_Delete_Backup
```

```
AFTER DELETE
```

```
ON creditcard
```

```
FOR EACH ROW
```

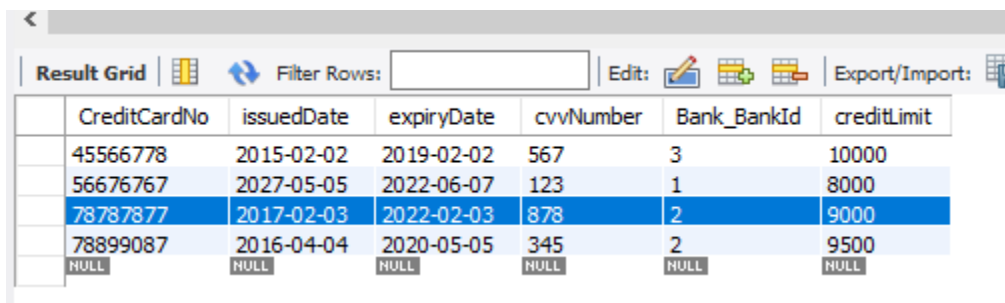
```
BEGIN
```

```
    insert into CloseCreditCard  
    values(old.CreditCardNo,old.issuedDate,old.expiryDate,old.cvvNumber,old. Bank_BankId);
```

```
END;
```

```
$$
```

```
select * from creditcard;
```

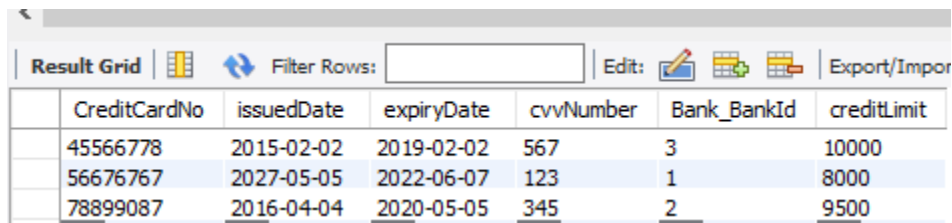


The screenshot shows a database application interface with a 'Result Grid' tab. The grid contains a table with 7 columns: CreditCardNo, issuedDate, expiryDate, cvvNumber, Bank_BankId, and creditLimit. The data is as follows:

CreditCardNo	issuedDate	expiryDate	cvvNumber	Bank_BankId	creditLimit
45566778	2015-02-02	2019-02-02	567	3	10000
56676767	2027-05-05	2022-06-07	123	1	8000
78787877	2017-02-03	2022-02-03	878	2	9000
78899087	2016-04-04	2020-05-05	345	2	9500
NULL	NULL	NULL	NULL	NULL	NULL

delete from creditcard where cvvNumber=878;

(Record having cvv number 878 deleted from **creditcard** table)

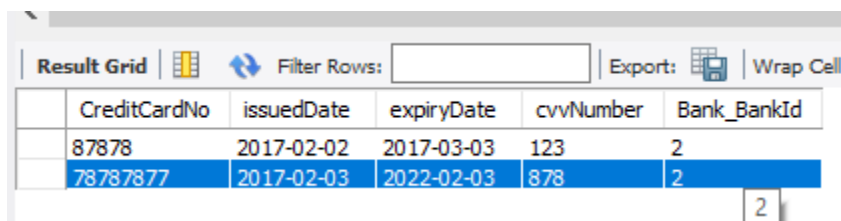


A screenshot of a database application's 'Result Grid'. The grid has a toolbar at the top with icons for 'Filter Rows', 'Edit', and 'Export/Import'. Below the toolbar is a table with the following data:

	CreditCardNo	issuedDate	expiryDate	cvvNumber	Bank_BankId	creditLimit
	45566778	2015-02-02	2019-02-02	567	3	10000
	56676767	2027-05-05	2022-06-07	123	1	8000
	78899087	2016-04-04	2020-05-05	345	2	9500

select * from CloseCreditCard;

(Record deleted from **creditcard** table is inserted into CloseCreditCard table)



A screenshot of a database application's 'Result Grid'. The grid has a toolbar at the top with icons for 'Filter Rows', 'Export', and 'Wrap Cell'. Below the toolbar is a table with the following data:

	CreditCardNo	issuedDate	expiryDate	cvvNumber	Bank_BankId
	87878	2017-02-02	2017-03-03	123	2
	78787877	2017-02-03	2022-02-03	878	2

TRANSACTION T1

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

SET SQL_SAFE_UPDATES=0;

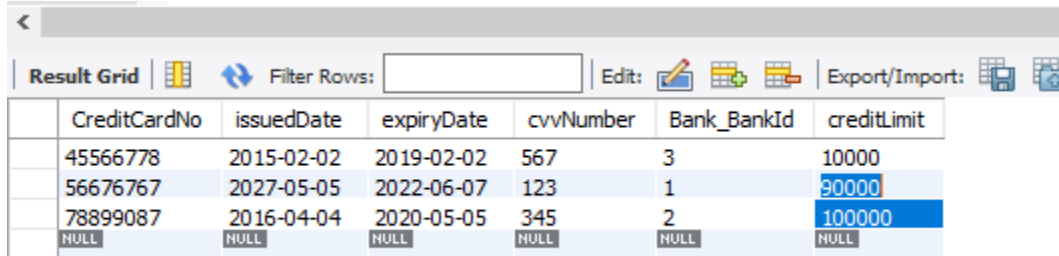
START transaction;

update Transaction_Details set creditlimit=creditlimit-(quantity*productprice) where orderid
in(select orderid from `order` inner join customer on

```
customer.customerid=`order`.customer_customerID);
DO SLEEP(20);
COMMIT;
```

-----credit card table before transaction update-----

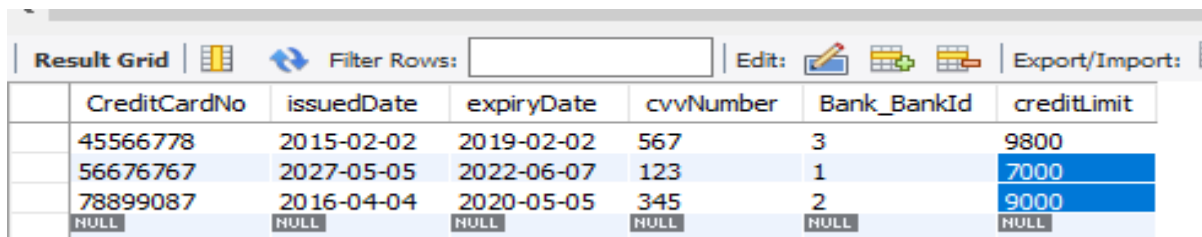
```
SELECT * FROM CREDITCARD;
```



	CreditCardNo	issuedDate	expiryDate	cvvNumber	Bank_BankId	creditLimit
	45566778	2015-02-02	2019-02-02	567	3	10000
	56676767	2027-05-05	2022-06-07	123	1	90000
	78899087	2016-04-04	2020-05-05	345	2	100000
	NULL	NULL	NULL	NULL	NULL	NULL

----- credit card table after transaction update-----

(creditlimit in **creditcard** table updated after T1 execution)



	CreditCardNo	issuedDate	expiryDate	cvvNumber	Bank_BankId	creditLimit
	45566778	2015-02-02	2019-02-02	567	3	9800
	56676767	2027-05-05	2022-06-07	123	1	7000
	78899087	2016-04-04	2020-05-05	345	2	9000
	NULL	NULL	NULL	NULL	NULL	NULL

TRANSACTION T2

```
SET TRANSACTION ISOLATION LEVEL serializable;
SET SQL_SAFE_UPDATES=0;
START transaction;
```

```
SELECT CREDITLIMIT FROM CREDITCARD;  
COMMIT;
```

-----T2 is waiting for completion of T1-----

Action Output				
	#	Time	Action	Message
✓	5	20:46:05	START transaction	0 row(s) affected
4	6	20:46:05	SELECT CREDITLIMIT FROM CREDITCARD LIMIT 0, 1000	Running...

Insert privileges to customer in customer table

- create a new user

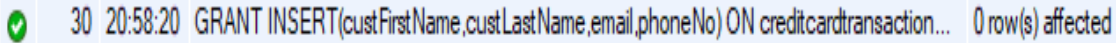
```
create user 'customer3'@'localhost' identified by 'customer3';
```

-- to check all the users existing in db

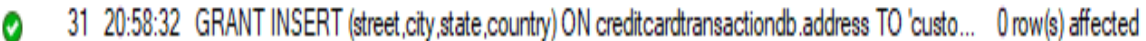
select user from mysql.user;

-- login as newly created user, no databases shown because of privileges

GRANT INSERT (custFirstName, custLastName, email, phoneNo) ON
creditcardtransactiondb.customer TO 'customer3'@'localhost' identified by 'customer3';

A screenshot of a MySQL command execution result. It shows a green checkmark icon, the command ID '30', the time '20:58:20', the command text 'GRANT INSERT(custFirstName,custLastName,email,phoneNo) ON creditcardtransaction...', and the result '0 row(s) affected'.

GRANT INSERT (street, city, state, country) ON creditcardtransactiondb.address TO
'customer3'@'localhost' identified by 'customer3';

A screenshot of a MySQL command execution result. It shows a green checkmark icon, the command ID '31', the time '20:58:32', the command text 'GRANT INSERT (street,city,state,country) ON creditcardtransactiondb.address TO 'custo...', and the result '0 row(s) affected'.

Select access privilege to admin

create user 'BankAdmin'@'localhost' identified by 'BankAdmin';

select user from mysql.user;

GRANT select ON creditcardtransactiondb.* TO 'BankAdmin'@'localhost' identified by
'BankAdmin';

32 21:18:20 GRANT select ON creditcardtransactiondb.* TO 'BankAdmin'@'localhost' identified by 'Ba... 0 row(s) affected

select access privilege to technician

```
create user 'technician'@'localhost' identified by 'technician';
```

```
select user from mysql.user;
```

```
GRANT select (complaintID,complaintDate,complaintDescription) ON  
creditcardtransactiondb.complaint
```

```
TO 'technician'@'localhost' identified by 'technician';
```

47 00:23:06 GRANT select (complaintID,complaintDate,complaintDescription) ON creditcardtransactio... 0 row(s) affected

MySQL Database

Address table

```
"addressId";"street";"city";"state";"country"
```

```
1;"Journal Square";"newyork";"NY";"USA"
```

```
2;"Cambridge";"Boston";"MA";"USA"
```

3;"Allston";"NewJersey";"NJ";"USA"
4;"Cambridge";"Boston";"MA";"USA"
5;"Allston";"Chicago";"Illinoise";"USA"
7;"PorterStreet";"Boston";"MA";"USA "

Bank Table

"BankId";"bankName";"phoneNo";"ifscCode";"street";"city";"state";"country"
1;"Bank of America";"9850445975";"23456";"Allston";"Boston";"MA";"USA"
2;"Santander";"8878888888";"78676";"Cambridge";"Boston";"MA";"USA "
3;"TDBank";"8989898989";"6756";"Journal square";"New Jersey";"NJ";"USA "

BankAccount Table

"accountNo";"Bank_BankId"
8989898;1
6677889;2
5555555;3

CloseCreditCard Table

"CreditCardNo";"issuedDate";"expiryDate";"cvvNumber";"Bank_BankId"
87878;"2017-02-02";"2017-03-03";"123";2
78787877;"2017-02-03";"2022-02-03";"878";2
98798798;"2012-05-05";"2020-06-06";"987";3

Complaint Table

"ComplaintID";"complaintDate";"ComplaintDescription";"t_ID"
201;"2018-02-02";"unauthorize use";111

Creditcard Table

"CreditCardNo";"issuedDate";"expiryDate";"cvvNumber";"Bank_BankId";"creditLimit"
45566778;"2015-02-02";"2019-02-02";"567";3;"8800"
56676767;"2027-05-05";"2022-06-07";"123";1;"7000"
78899087;"2016-04-04";"2020-05-05";"345";2;"9500"

Customer Table

"customerID";"custFirstName";"custLastName";"email";"phoneNo";"Address_addressId"
1;"Bhagyashree";"Patil";"56565";"9850445975";1
2;"Pankaj";"Salunkhe";"pj@yahoo.com";"5513503639";1
3;"Ashish";"Patil";"ashish@yahoo.com";"8867564534";2
4;"Sonali";"Patel";"son@gmail.com";"5513583639";3
24;"Minal";"Jadhav";"min@yahoo.com";"8787878778";3
26;"Pinu";"Shinde";"p@gmail.com";"9878677889";0

Customer_has_BankAccount Table

"Customer_customerID";"BankAccount_accountNo"
4;5555555
3;6677889
2;8989898

Customer_has_Complaint Table

"Complaint_ComplaintID";"customer_customerID"
201;2

Customer_has_Creditcard Table

"Customer_customerID";"CreditCard_CreditCardNo"
4;45566778
2;56676767
3;78899087

Order Table

"orderId";"orderDate";"orderStatus";"Customer_customerID"
11;"2017-01-01";"Deliver";2
22;"2018-02-02";"Delivered";3
33;"2016-02-02";"Delivered";4

OrderItem Table

"OrderItemId";"quantity";"Order_orderId";"Product_productId"
111;2;11;101
222;5;22;102
333;1;33;103

Product Table

"productId";"productName";"productPrice"
101;"mobile";"500"
102;"tv";"100"
103;"camera";"200"

Transaction

"transactionId";"transactionDate";"CreditCard_CreditCardNo";"Order_orderId"
1;"2017-01-01";56676767;11
2;"2018-01-01";78899087;22
3;"2018-03-03";45566778;33

Technician Table

"t_ID";"FirstName";"LastName";"ephone";"email";"bank_BankId"
111;"Rashmi";"Patel";"9898989898";"r@gmail.com";1