UNIVERSITY OF

DEPARTMENT OF COMPUTER SCIENCE

Group #1: Online Banking Transaction

Phase 4

:Instructor

Database Design

BY

Abstract

Online banking transaction allows customers of a particular bank to perform financial transactions on a secure website maintained by the bank or credit union. Customers with a valid user id and password can perform an online transaction. This system provides the following facilities for a valid user:

- Balance Inquiry
- Deposit
- Withdrawal
- Funds transfer within and outside the bank
- Viewing monthly and annual statements

Problem Statement

In our online-banking transaction, there are users. Users are those people who have an account with the bank. Each user has an account number, account type, name, valid ID, valid password, address, joining date, phone number and email address. Users also have the option to have multiple accounts.

There are several services that users would like to get from the online-banking system, including balance checking, making deposit and withdrawal, transaction checking and viewing their information.

Regarding online transactions, every user has the option to request details of the transactions he/she has performed within a certain period. Each transaction will have a unique transaction id. Transactions include the following types:

- Deposit
- Withdraw
- Fund transfers
- Pay Bills

In order to transfer funds from a user's account, the user has to provide the following information to add an account to transfer to:

- User's account number
- Recipient’s name
- Recipient’s account number
- Recipient’s account type (Checking/Savings)
- Recipient’s zip code
- Nickname (description of the transaction). This field would be optional.

There are different types of fund transfers that can take place.

- TRANSFERRING WITHIN THE BANK

  Users can transfer funds from his account to any other account in the same bank. If the transaction is successful a notification should appear to the user, in case it is unsuccessful, a proper message should be given to the customer as to why it failed. In order for this transaction to happen, the user has to fill in the recipient’s account number, amount to transfer, and frequency of transfer (One-Time, weekly, monthly and annually).

- TRANSFERRING OUTSIDE THE BANK

  Users can transfer funds from his account to any other account in a different bank. If the transaction is successful a notification should appear to the user, in case it is unsuccessful, a proper message should be given to the customer as to why it failed. In order for this

transaction to happen, the user has to fill in the recipient's routing number(9 digit) or bank name, account number, amount to transfer, and frequency of transfer (One-Time, weekly, monthly and annually).

Users can also pay their bills (like phone or electricity bills). For this to happen, the user has to initially save the payee's information like:

- company name or individual name
- User Account number
- Payee Zip code

In order to make a pay, user has to enter:

- the amount to be paid
- the date of payment

Queries

1. Retrieve the account balances for user-id “ABC”.

2. Retrieve the transaction history for account number “123456789” from Jan 01, 2008 to Mar 15, 2008.

3. Retrieve all transactions that are either deposit or withdrawal for an account number “123456789”.

4. Retrieve all customers who have a balance greater than 100,000.

5. Retrieve the number of customers and the maximum balance for each account type.

6. For each user having at least two accounts, retrieve the name, account number and balance for each account.

7. Customer wants to transfer funds from their account to a different account within the bank.

8. Customer wants to transfer funds from their account to a different account outside the bank.

9. Customer may want to pay their bills from a particular account.

10. Customer wants to view their personal information.

11. Customer may want to add an account.

12. Customer may want to manage their account like updating their personal information.

Conceptual Design

Entities:

USERS, TAX, TRANSACTION, ACCOUNT, TRANSFER, PAYEE, CREATES, TRANSFER

Weak Entity:

STATEMENT

Attributes:

- USERS – Name, User ID, Password, Address, Phone, Email, SSN, LoginDate, Role
- STATEMENT – Account No, Start Date, End Date
- ACCOUNT – Account number, type, balance
- TRANSACTION – ID, Type, Amount, Balance, Account No, Date, PayeeID, Deliver, TransID
- TRANSFER – Transfer ID, Type, UserID, Name, Account No, ZIP, Route No, Active
- PAYEE – ID, UserID, Name, Account No, ZIP
- CREATES – Account No, UserID, JoinDate
- TAX – SSN, Tax

Relationship:

- User deposits and withdraws to/from their account.
- User transfer funds within and outside the bank.
- User adds a payee.
- User performs a transaction from a particular account.
- User requests for statements.

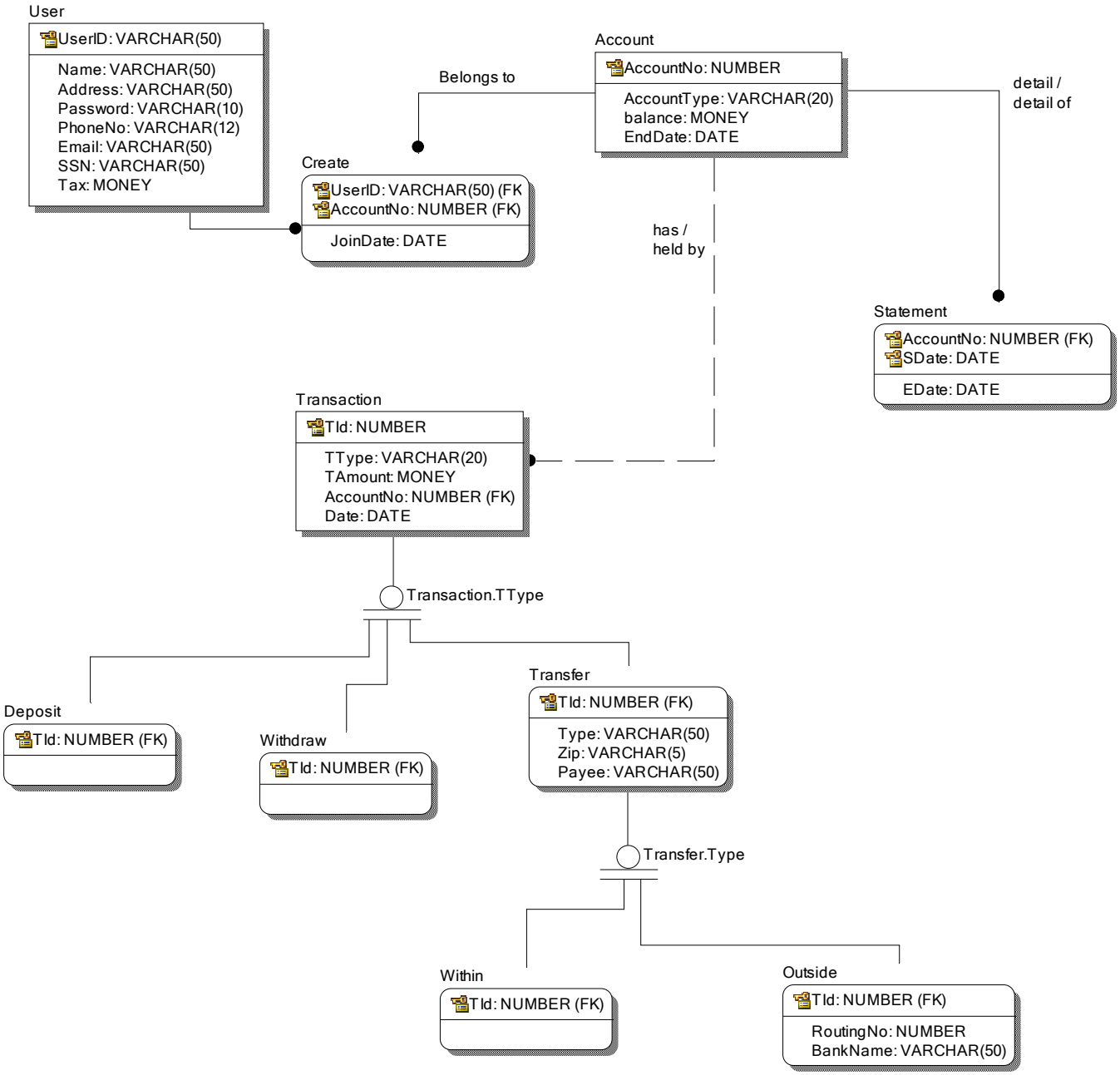
Attributes of relationship types:

- Transfer Outside Bank – Bank ID
- User manages account – Join date

Keys:

- USERS - User ID
- TAX – SSN
- ACCOUNT - Account Number
- TRANSACTION – Transaction ID
- TRANSFER – Transfer ID
- PAYEE – Payee ID
- CREATES – Account No
- STATEMENT – Account No

❖ EER Modeling



Mapping to Relations

Step 1: Mapping of Regular entity types

We create the relations USERS, TRANSACTION, ACCOUNT, PAYEE and TRANSFER in the relational schema corresponding to the regular entities in the ER diagram. UserID is taken as a primary key. It can be used in case if a customer doesn't have a SSN.

USERS

USERID	NAME	ADDRESS	PASSWORD	PHONENO	EMAIL	SSN	Tax

TRANSACTION

TID	TTYPE	TAMOUNT	ACCTNO	DATE

ACCOUNT

ACCOUNTNO	ATYPE	BALANCE

Step 2: Mapping of Weak Entity Types

We create the relation STATEMENT in this step to correspond to the weak entity type.

STATEMENT

ACCOUNTNO	SDATE	EDATE

Step 3: Mapping of Binary 1:1 Relation Types

STATEMENT is a 1:1 relationship between ACCOUNT and STATEMENT.

STATEMENT (E1)

FK		
ACCOUNTNO	SDATE	EDATE

ACCOUNT (E2): The ACCOUNT relation is unchanged from step 1

ACCOUNTNO	ACCOUNTTYPE	BALANCE

Step 4: Mapping of Binary 1:N Relationship Types.

TRANSACTION

			FK	
TID	TTYPE	TAMOUNT	ACCOUNT NO	DATE

ACCOUNT

ACCOUNTNO	ACCOUNTTYPE	BALANCE

Step 5: Mapping of Binary M:N Relationship Types

USER

USERID	NAME	ADDRESS	PASSWORD	PHONENO	EMAIL	SSN	Tax

CREATES

ACCOUNTNO	USERID	JOINDATE

ACCOUNT

ACCOUNTNO	ATYPE	BALANCE

Step 6 and Step 7:

There is neither mapping of multivalued attributes nor mapping of N-ary relationship types.

Step 8: Mapping Specialization or Generalization

(a)

TRANSACTION

TID	TTYPE	TAMOUNT	ACCTNO	DATE

DEPOSIT

TID

WITHDRAW

TID

TRANSFER

TID	TYPE	ZIP	PAYEE

(b)

TRANSFER

TID	TYPE	ZIP	PAYEE

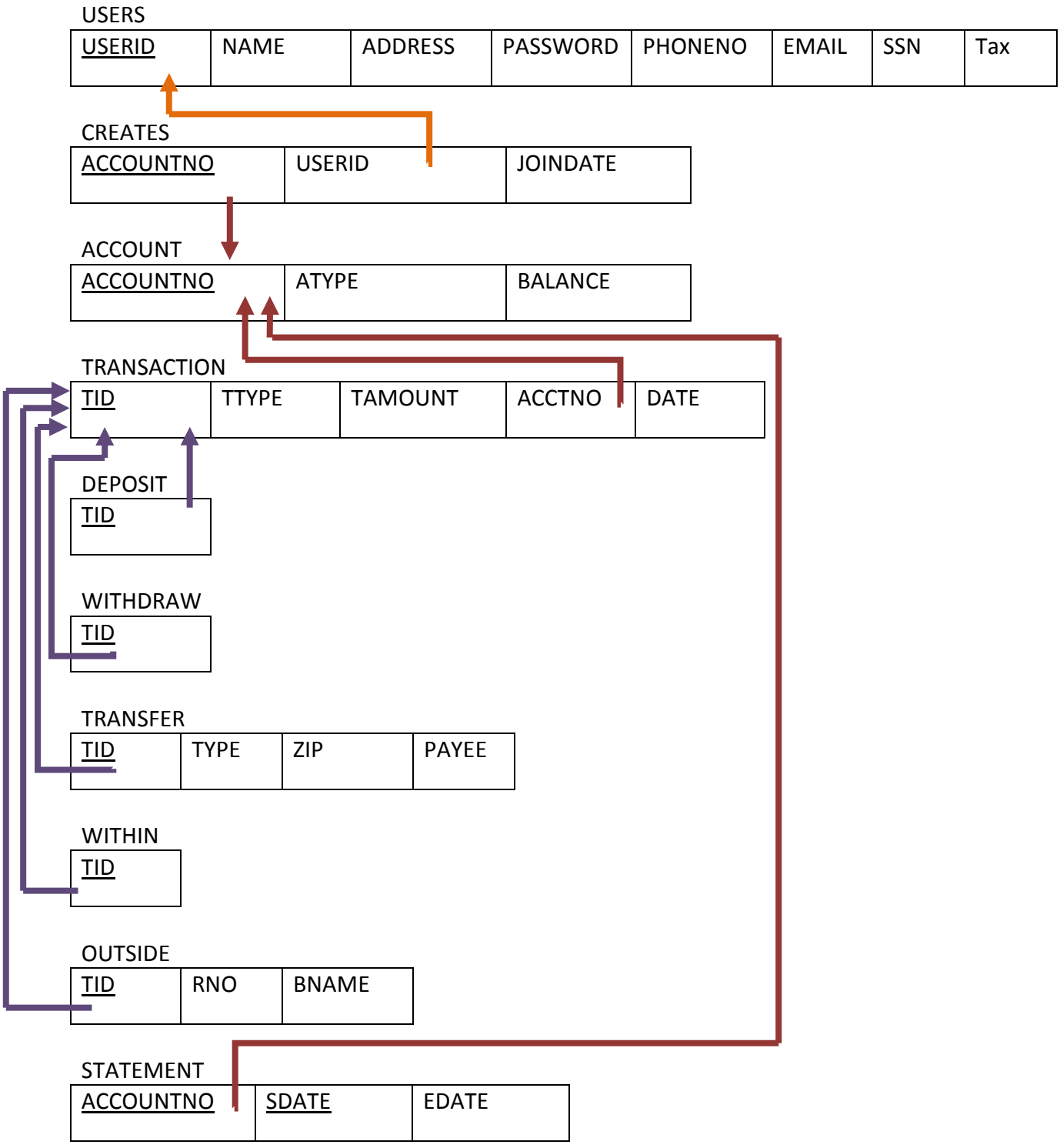
WITHIN

TID

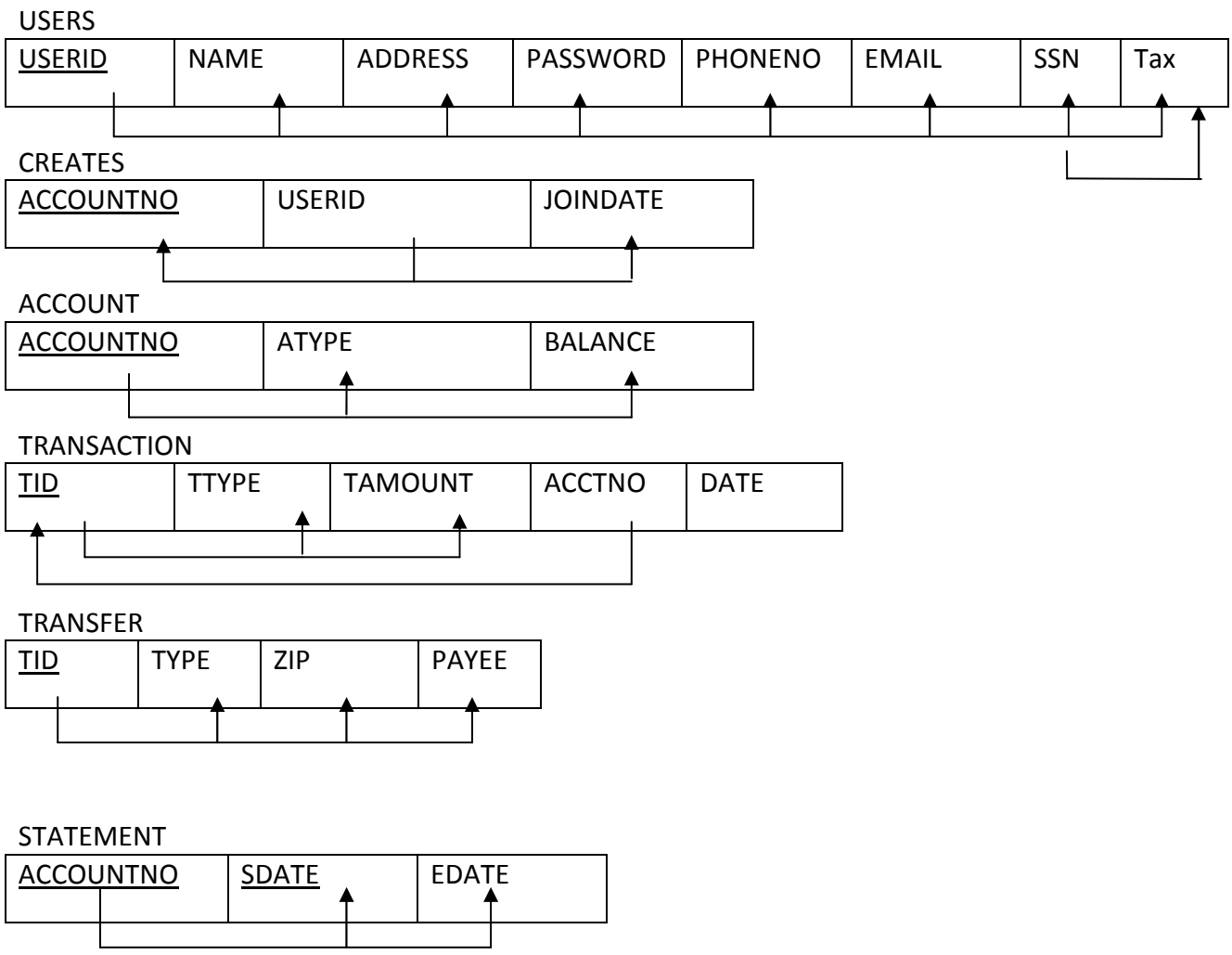
OUTSIDE

TID	RNO	BNAME

Result of mapping the ONLINE BANKINGER schema into a relational schema

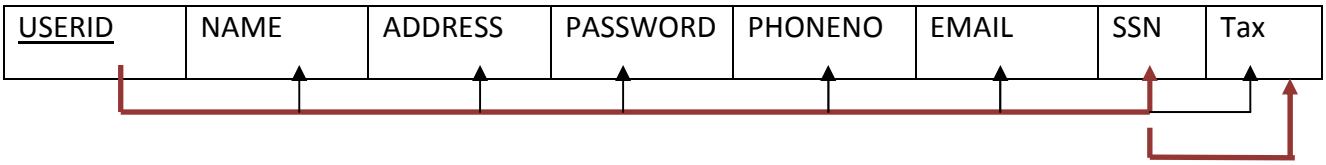


Functional Dependencies



Normalization

USER



1NF: USER is in 1NF

2NF: USER is in 2NF

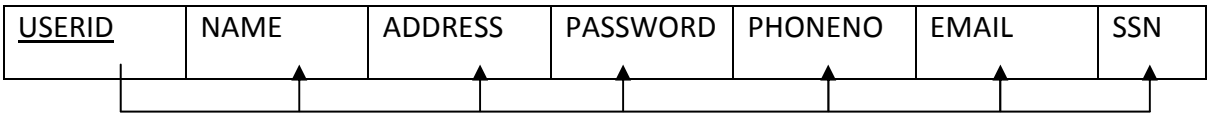
3NF: USER is in 3NF

BCNF: There is transitive dependency between USERID and Tax, since USERID -> SSN and SSN -> Tax holds.

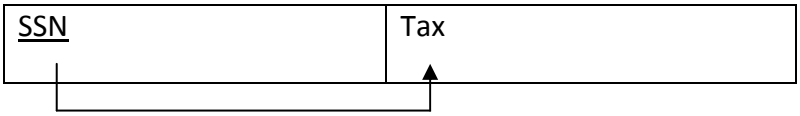
Solution to BCNF:

USER (USERID, NAME, ADDRESS, PASSWORD, PHONENO, EMAIL, SSN)

USER



TAX (SSN, Tax)



Relational Algebra

1. Retrieve the account type and balance for each account user-id "ABC" has.

$R1 \leftarrow \Pi_{AccountNo} (\sigma_{UserID='ABC'} (CREATE))$

$Result \leftarrow \Pi_{Balance} (R1 \bowtie_{R1.AccountNo = Account.AccountNo} ACCOUNT)$

2. Retrieve the transaction history, including transaction ID, type, amount of money, and date, for account number "100000000011" from Jan 01, 2000 to Mar 15, 2008.

$Result \leftarrow \sigma_{AccountNo = 100000000011 \text{ AND}} (TRANSACTION)$

Date >= Jan 01, 2000 AND

Date <= Mar 15, 2008

3. Retrieve all transactions that are either deposit or withdrawal for an account number "100000000011".

$\sigma_{AccountNo = 100000000011 \text{ AND}} (TRANSACTION) \text{ U } \sigma_{AccountNo = 100000000011} (TRANSACTION)$

TType = 'Deposit' TType = 'Withdraw'

4. Retrieve all customers who have a balance greater than 100,000.

$R1 \leftarrow \sigma_{Balance > 100000} (ACCOUNT)$

$Result \leftarrow \Pi_{UserID,} (R1 \bowtie(*) CREATES)$

Balance,

ATYPE

5. Retrieve the number of customers and the maximum balance for each account type.

$AType \text{ } F_{COUNT(UserID),} (CREATES * ACCOUNT)$

MAX (Balance)

Physical Database Design

DDL Script

```
CREATE TABLE USERS
(USERID VARCHAR2(20) NOT NULL PRIMARY KEY,
NAME VARCHAR2(25) NOT NULL,
ADDRESS VARCHAR2(30) NOT NULL,
PASSWORD VARCHAR2(15) NOT NULL,
PHONENO NUMBER(10),
EMAIL VARCHAR2(30),
SSN NUMBER(9) NOT NULL,
LOGINDATE VARCHAR2(30),
ROLE VARCHAR2(5) NOT NULL);

CREATE TABLE TAX
(SSN NUMBER(9) NOT NULL PRIMARY KEY,
TAX NUMBER(4));

CREATE TABLE CREATES
(ACCOUNTNO NUMBER(12) NOT NULL PRIMARY KEY,
USERID VARCHAR2(20) NOT NULL,
JOINDATE DATE);

CREATE TABLE ACCOUNT
(ACCOUNTNO NUMBER(12) NOT NULL PRIMARY KEY,
ATYPE VARCHAR2(10) NOT NULL,
BALANCE NUMBER NOT NULL);

CREATE TABLE TRANSACTION
(TID NUMBER(6) NOT NULL PRIMARY KEY,
TTYPE VARCHAR2(10) NOT NULL,
TAMOUNT NUMBER NOT NULL,
BALANCE NUMBER NOT NULL,
ACCTNO NUMBER(12) NOT NULL,
TDATE DATE NOT NULL
PAYEEID NUMBER
DELIVER DATE
TRANSID NUMBER );

CREATE TABLE TRANSFER
(TRANSID NUMBER(6) NOT NULL PRIMARY KEY,
TTYPE VARCHAR2(10) NOT NULL,
USERID VARCHAR2(20) NOT NULL,
NAME VARCHAR2(30) NOT NULL,
ACCTNO NUMBER(12) NOT NULL,
```

```
ZIP VARCHAR2(5),
ROUTENO NUMBER(9),
ACTIVE NUMBER NOT NULL);

CREATE TABLE PAYEE
(PAYEEID NUMBER(6) NOT NULL PRIMARY KEY,
USERID VARCHAR2(20) NOT NULL,
NAME VARCHAR2(30) NOT NULL,
ACCTNO NUMBER(12) NOT NULL,
ZIP VARCHAR2(5));

CREATE VIEW DEPOSIT AS
(SELECT *
FROM TRANSACTION
WHERE TTYPE = 'DEPOSIT');

CREATE VIEW WITHDRAW AS
(SELECT *
FROM TRANSACTION
WHERE TTYPE = 'WITHDRAW');

CREATE VIEW TRANSFER_VIEW AS
(SELECT *
FROM TRANSACTION
WHERE TTYPE = 'TRANSFER');
```

DML Script

```
INSERT INTO USERS VALUES ('SMITH3623', 'JOSEPH SMITH', '126 PARK, BRIDGEPORT,
CT','QAZ123','2035538675','JSMITH@EBANK.COM','123456789',’11/12/2004 06:10:12
PM’,’user’);
INSERT INTO USERS VALUES ('JOHNSON4225','THOMAS JOHNSON','332 BROAT,
BRIDGEPORT,CT','WSX234','2034457883','TJOHNSON@EBANK.COM','234567890',’01/02/
2006 12:10:45 PM’,’ADMIN’);
INSERT INTO USERS VALUES ('WILLIAMS8852','MARY WILLIAMS','540 ATLANTIC,
BRIDGEPORT,CT','EDC345','2034850093','MWILLIAMS@EBANK.COM','111223333',’02/23
/2007 05:10:27 AM’,’user’);
INSERT INTO USERS VALUES ('JONES3347','CHRISTOPHER JONES','15 MARINA PARK,
BRIDGEPORT, CT','RFV456',NULL,'CJONES@EBANK.COM','222334444',’11/01/1999
10:10:10 PM’,’user’);
INSERT INTO USERS VALUES ('BROWN1374','ROBERT BROWN','80 UNIVERSITY,
BRIDGEPORT,
CT','TGB567','2031252349','RBROWN@EBANK.COM','333445555',’05/05/2005 05:05:05
PM’,’user’);
INSERT INTO USERS VALUES ('DAVIS9953','MICHAEL DAVIS','337 MYRTLE,
BRIDGEPORT,
CT','YHN678','2039475834','MDAVIS@EBANK.COM','444556666',’10/10/10 06:10:12
PM’,’user’);
INSERT INTO USERS VALUES ('MILLER3467','DANIEL MILLER','295 AUSTIN,
BRIDGEPORT,
CT','UJM789','2035847366','DMILLER@EBANK.COM','555667777',’08/07/2001
04:19:45 PM’,’user’);
INSERT INTO USERS VALUES ('WILSON1234','MARK WILSON','125 WALNUT, BRIDGEPORT,
CT','IKO890','2039045748','MWILSON@EBANK.COM','666778888',’07/07/2007
07:07:07 PM’,’user’);
```

```
INSERT INTO USERS VALUES ('MOORE7034','PAUL MOORE','334 GREGORY, BRIDGEPORT,
CT','LPQ135','2039146384',NULL,'777889999','01/29/2006 03:20:18 PM','user');
INSERT INTO USERS VALUES ('TAYLOR2239','JACKSON TAYLOR','773 FAIRFIELD,
BRIDGEPORT, CT','LPG246','2031158374',NULL,'888990000','10/05/2007 10:00:00
PM','user');

INSERT INTO TAX VALUES ('123456789','26');
INSERT INTO TAX VALUES ('234567890','443');
INSERT INTO TAX VALUES ('111223333','345');
INSERT INTO TAX VALUES ('222334444','226');
INSERT INTO TAX VALUES ('333445555','845');
INSERT INTO TAX VALUES ('444556666','126');
INSERT INTO TAX VALUES ('555667777','88');
INSERT INTO TAX VALUES ('666778888','43');
INSERT INTO TAX VALUES ('777889999','87');
INSERT INTO TAX VALUES ('888990000','343');


INSERT INTO CREATES VALUES
('100000000011','SMITH3623',to_date('03/23/1994','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000012','JOHNSON4225',to_date('06/30/1993','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000013','WILLIAMS8852',to_date('08/26/1997','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000014','JONES3347',to_date('04/14/1992','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000015','BROWN1374',to_date('03/01/1991','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000016','DAVIS9953',to_date('11/06/1997','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000017','MILLER3467',to_date('02/08/1994','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000018','WILSON1234',to_date('05/15/1997','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000019','MOORE7034',to_date('05/05/1995','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000020','TAYLOR2239',to_date('08/08/1997','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000021','MOORE7034',to_date('07/07/1997','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000022','SMITH3623',to_date('01/25/1995','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000023','DAVIS9953',to_date('04/27/1991','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000024','WILLIAMS8852',to_date('09/15/1997','mm/dd/yyyy'));
INSERT INTO CREATES VALUES
('100000000025','JOHNSON4225',to_date('03/09/1992','mm/dd/yyyy'));


INSERT INTO ACCOUNT VALUES ('100000000011','CHECKING','634573');
INSERT INTO ACCOUNT VALUES ('100000000012','CHECKING','7344468');
INSERT INTO ACCOUNT VALUES ('100000000013','CHECKING','338945659');
INSERT INTO ACCOUNT VALUES ('100000000014','CHECKING','56734435');
INSERT INTO ACCOUNT VALUES ('100000000015','CHECKING','345634');
```

```
INSERT INTO ACCOUNT VALUES ('100000000016','CHECKING','9456354');
INSERT INTO ACCOUNT VALUES ('100000000017','CHECKING','345799');
INSERT INTO ACCOUNT VALUES ('100000000018','CHECKING','2223567');
INSERT INTO ACCOUNT VALUES ('100000000020','CHECKING','345637345');
INSERT INTO ACCOUNT VALUES ('100000000021','SAVING','7746345');
INSERT INTO ACCOUNT VALUES ('100000000022','SAVING','23445');
INSERT INTO ACCOUNT VALUES ('100000000023','SAVING','2345');
INSERT INTO ACCOUNT VALUES ('100000000024','SAVING','8345');
INSERT INTO ACCOUNT VALUES ('100000000025','SAVING','9943125');
INSERT INTO ACCOUNT VALUES ('100000000019','CHECKING','879965');


INSERT INTO TRANSACTION VALUES
('100001','DEPOSIT','750','100000000011',to_date('10/15/2003','mm/dd/yyyy'));
INSERT INTO TRANSACTION VALUES
('100002','WITHDRAW','150','100000000012',to_date('12/25/1998','mm/dd/yyyy'))
;
INSERT INTO TRANSACTION VALUES
('100003','WITHDRAW','320','100000000013',to_date('11/21/2005','mm/dd/yyyy'))
;
INSERT INTO TRANSACTION VALUES
('100004','DEPOSIT','335','100000000012',to_date('09/16/2008','mm/dd/yyyy'));
INSERT INTO TRANSACTION VALUES
('100005','DEPOSIT','285','100000000014',to_date('08/31/2002','mm/dd/yyyy'));
INSERT INTO TRANSACTION VALUES
('100006','WITHDRAW','550','100000000015',to_date('07/14/2005','mm/dd/yyyy'))
;
INSERT INTO TRANSACTION VALUES
('100007','WITHDRAW','1850','100000000016',to_date('06/06/1999','mm/dd/yyyy')
);
INSERT INTO TRANSACTION VALUES
('100008','WITHDRAW','2250','100000000011',to_date('08/21/2000','mm/dd/yyyy')
);
INSERT INTO TRANSACTION VALUES
('100009','DEPOSIT','8900','100000000017',to_date('01/22/1999','mm/dd/yyyy'))
;
INSERT INTO TRANSACTION VALUES
('100010','WITHDRAW','2100','100000000018',to_date('8/22/2001','mm/dd/yyyy'))
;
INSERT INTO TRANSACTION VALUES
('100012','WITHDRAW','4350','100000000020',to_date('10/03/2005','mm/dd/yyyy')
);
INSERT INTO TRANSACTION VALUES
('100013','WITHDRAW','9800','100000000019',to_date('09/27/2007','mm/dd/yyyy')
);
INSERT INTO TRANSACTION VALUES
('100014','DEPOSIT','13500','100000000019',to_date('07/21/2002','mm/dd/yyyy')
);
INSERT INTO TRANSACTION VALUES
('100015','DEPOSIT','350','100000000020',to_date('03/31/2004','mm/dd/yyyy'));
INSERT INTO TRANSACTION VALUES
('100016','DEPOSIT','75000','100000000011',to_date('07/15/2003','mm/dd/yyyy')
);
```

```
INSERT INTO TRANSACTION VALUES
('100017','TRANSFER','750','100000000020',to_date('06/29/2002','mm/dd/yyyy'))
;
INSERT INTO TRANSACTION VALUES
('100018','TRANSFER','4500','100000000013',to_date('04/14/2003','mm/dd/yyyy')
);
INSERT INTO TRANSACTION VALUES
('100019','TRANSFER','12500','100000000017',to_date('03/25/2004','mm/dd/yyyy'
));
INSERT INTO TRANSACTION VALUES
('100020','TRANSFER','620','100000000011',to_date('04/15/2005','mm/dd/yyyy'))
;
INSERT INTO TRANSACTION VALUES
('100021','TRANSFER','500','100000000012',to_date('05/22/2006','mm/dd/yyyy'))
;
INSERT INTO TRANSACTION VALUES
('100022','TRANSFER','900','100000000014',to_date('06/27/2007','mm/dd/yyyy'))
;
INSERT INTO TRANSACTION VALUES
('100023','TRANSFER','100','100000000016',to_date('07/05/2008','mm/dd/yyyy'))
;


INSERT INTO TRANSFER VALUES ('100017','WITHIN','06604','JOSEPH SMITH');
INSERT INTO TRANSFER VALUES ('100018','OUTSIDE','06603','CHRIS MOON');
INSERT INTO TRANSFER VALUES ('100019','OUTSIDE','06606','DAVID BECKHAM');
INSERT INTO TRANSFER VALUES ('100020','WITHIN','06601','THOMAS JOHNSON');
INSERT INTO TRANSFER VALUES ('100021','OUTSIDE','06602','WAYNE ROONEY');
INSERT INTO TRANSFER VALUES ('100022','WITHIN','06600','ROBERT BROWN');
INSERT INTO TRANSFER VALUES ('100023','OUTSIDE','06607','MICHAEL JORDAN');


INSERT INTO STATEMENT VALUES
('100000000011',to_date('01/01/2008','mm/dd/yyyy'),to_date('12/31/2008','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000011',to_date('01/01/2007','mm/dd/yyyy'),to_date('12/31/2007','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000011',to_date('01/01/2006','mm/dd/yyyy'),to_date('12/31/2006','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000011',to_date('01/01/2005','mm/dd/yyyy'),to_date('12/31/2005','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000012',to_date('01/01/2008','mm/dd/yyyy'),to_date('12/31/2008','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000012',to_date('01/01/2007','mm/dd/yyyy'),to_date('12/31/2007','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000012',to_date('01/01/2006','mm/dd/yyyy'),to_date('12/31/2006','mm/d
d/yyyy'));
```

```
INSERT INTO STATEMENT VALUES
('100000000013',to_date('01/01/2008','mm/dd/yyyy'),to_date('12/31/2008','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000013',to_date('01/01/2007','mm/dd/yyyy'),to_date('12/31/2007','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000013',to_date('01/01/2006','mm/dd/yyyy'),to_date('12/31/2006','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000014',to_date('01/01/2008','mm/dd/yyyy'),to_date('12/31/2008','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000014',to_date('01/01/2007','mm/dd/yyyy'),to_date('12/31/2007','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000015',to_date('01/01/2008','mm/dd/yyyy'),to_date('12/31/2008','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000015',to_date('01/01/2007','mm/dd/yyyy'),to_date('12/31/2007','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000016',to_date('01/01/2008','mm/dd/yyyy'),to_date('12/31/2008','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000017',to_date('01/01/2008','mm/dd/yyyy'),to_date('12/31/2008','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000017',to_date('01/01/2007','mm/dd/yyyy'),to_date('12/31/2007','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000018',to_date('01/01/2008','mm/dd/yyyy'),to_date('12/31/2008','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000019',to_date('01/01/2008','mm/dd/yyyy'),to_date('12/31/2008','mm/d
d/yyyy'));
INSERT INTO STATEMENT VALUES
('100000000019',to_date('01/01/2007','mm/dd/yyyy'),to_date('12/31/2007','mm/d
d/yyyy'));
```

Drop Script

```
DROP TABLE USERS;

DROP TABLE TAX;

DROP TABLE CREATES;

DROP TABLE ACCOUNT;

DROP TABLE TRANSACTION;
```

```
DROP TABLE TRANSFER;

DROP TABLE STATEMENT;

DROP VIEW TRANSFER_VIEW;

DROP VIEW VIP_ID;

DROP VIEW NAME_VIEW;

DROP VIEW TEMP;
```

SQL Script

1. Retrieve the account type and balance for each account user-id “SMITH3623” has.

```
select atype as acct_type, balance
from account
where accountno in (select accountno
                    from creates
                    where userid = 'SMITH3623');
```

2. Retrieve the transaction history, including transaction ID, type, amount of money, and date, for account number “100000000011” from Jan 01, 2000 to Mar 15, 2008.

```
select tid, ttype, tamount, tdate
from transaction
where acctno = '100000000011'
      AND tdate >= to_date('01/01/2000','mm/dd/yyyy')
      AND tdate <= to_date('03/15/2008','mm/dd/yyyy')
order by tdate;
```

3. Retrieve all transactions that are either deposit or withdrawal for an account number “100000000011”.

```
select *
from transaction
where acctno = '100000000011'
      AND (ttype = 'DEPOSIT' OR ttype = 'WITHDRAW');
```

4. Retrieve all customers who have a balance greater than 100,000.

```
create view vip_id
as select distinct userid
from creates c JOIN account a ON c.accountno = a.accountno
where a.balance >= 1000000;

select name
from users u, vip_id v
where u.userid = v.userid;
```

5. Retrieve the number of customers and the maximum balance for each account type.

```
select atype, count(distinct userid) as users, max(balance) as Max_Balance
 from (select *
 from creates c, account a
 where c.accountno = a.accountno)
 group by atype;
```

6. For each user having at least two accounts, retrieve the name, account number and balance for each account.

```
-- step1: create view to list all the name who have at least two accounts

create view name_view as
select name, userid
from users
where userid in (select c.userid
from creates c
group by c.userid
having count(*) >= 2);

-- step2: create view to list all the name, userid, accountno who have at
least two accounts

create view temp as
select n.name, n.userid, c.accountno
from name_view n, creates c
where n.userid = c.userid;

-- step3: Final solution

select t.name, t.userid, t.accountno, a.atype, a.balance
from temp t, account a
where t.accountno = a.accountno;
```

Implementation

The project was developed by keeping in mind the following specifications:

- Operating System: Windows NT
- Developmental Tool: MS Visual Studio 2005
- Programming Languages: ASP.NET
    - Server Side: C#
    - Client Side: Javascript
- Database: Oracle Client 10.2.0

Below are the screenshots of our Online Banking Website. There are two main sections called, User and Administrator.

User Sections:

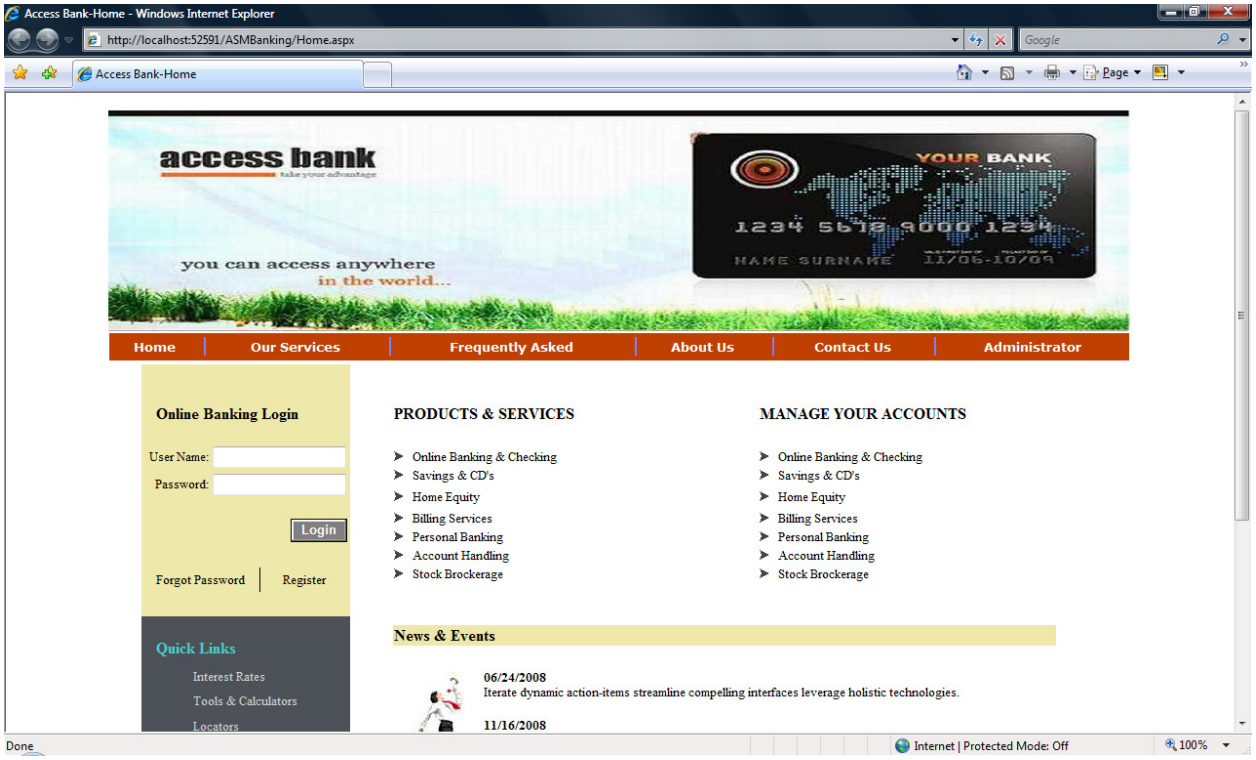


Fig: Bank Homepage

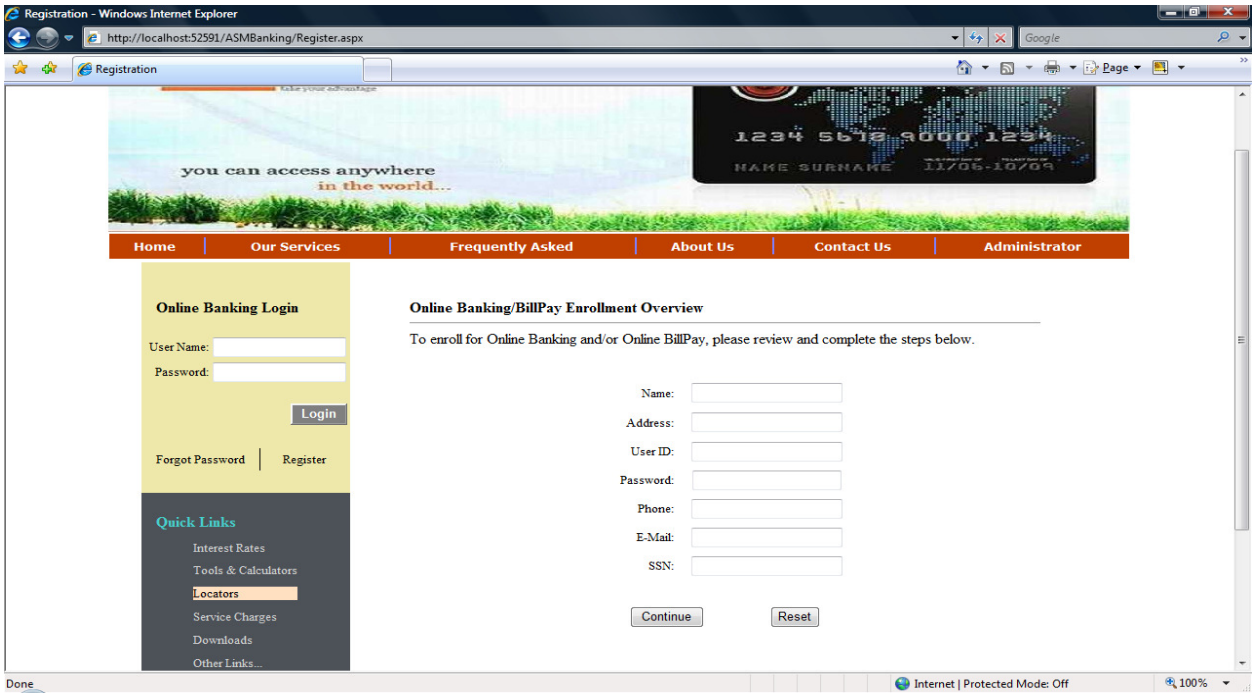


Fig: Registration page to access the online banking system

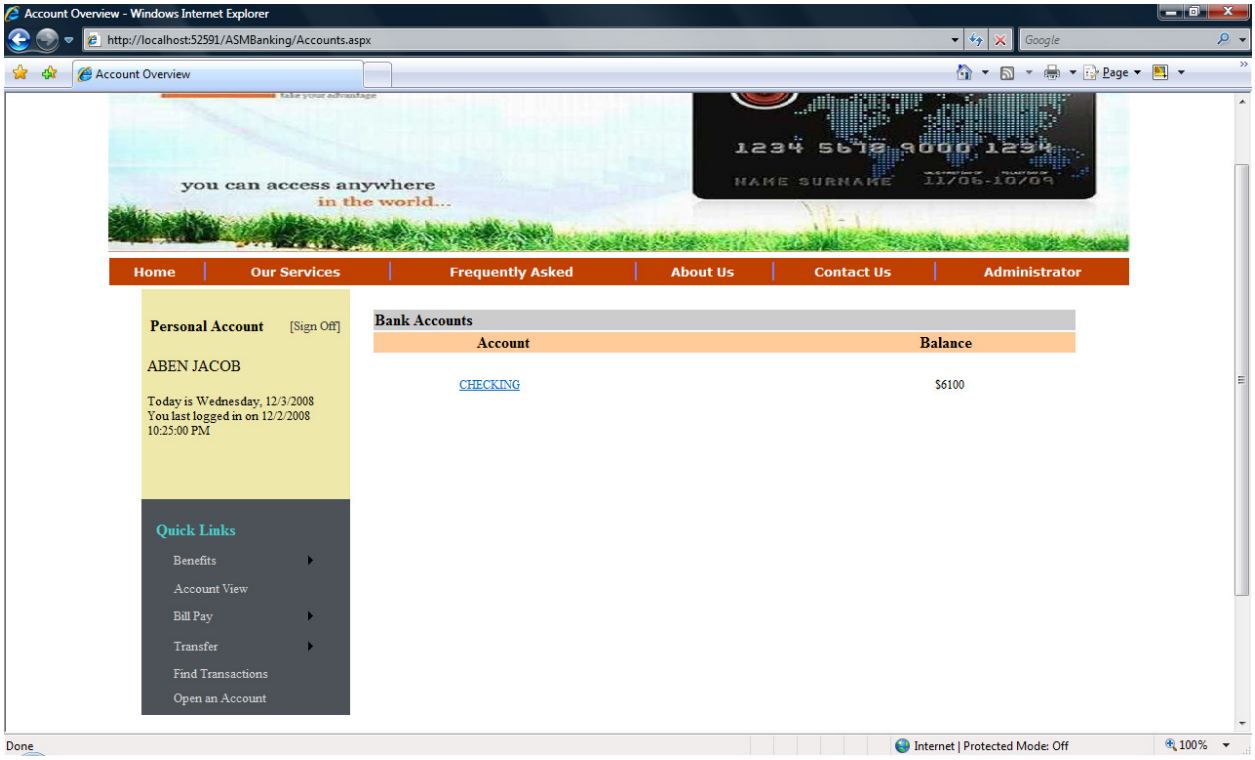


Fig: User Profile Page – Accounts

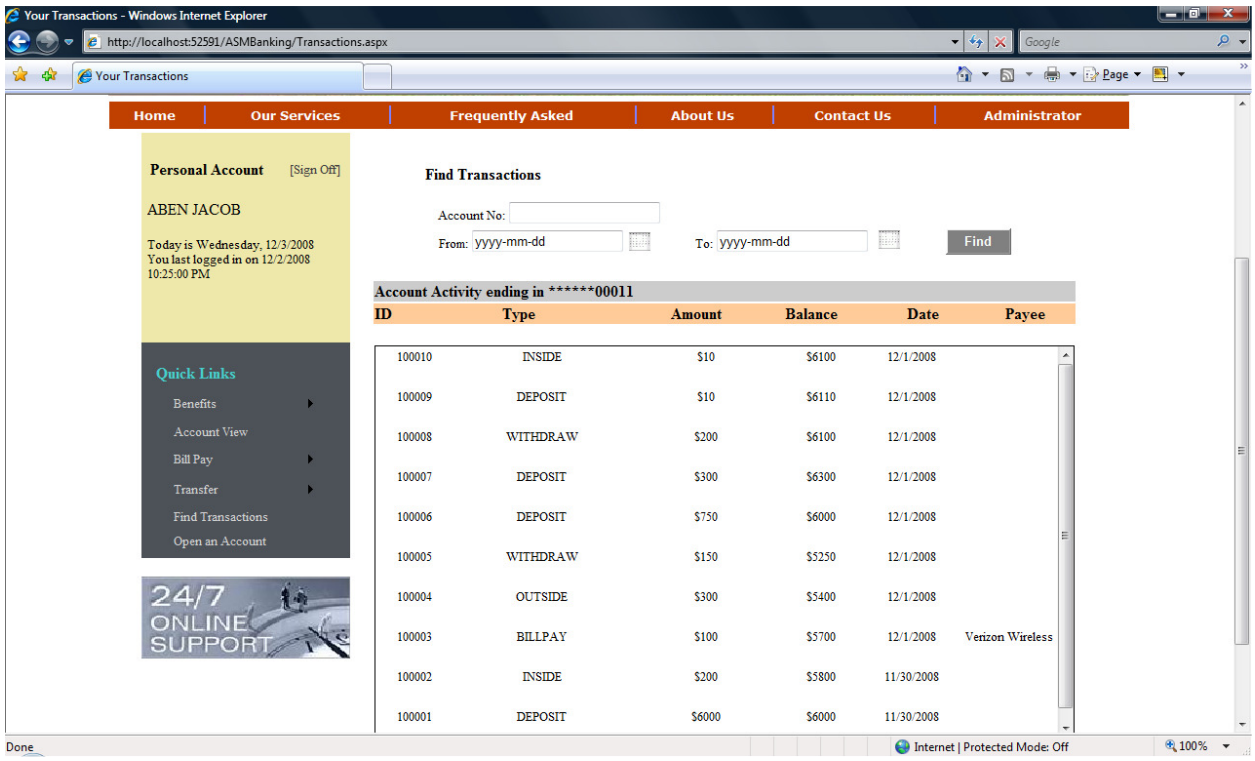


Fig: Transactions

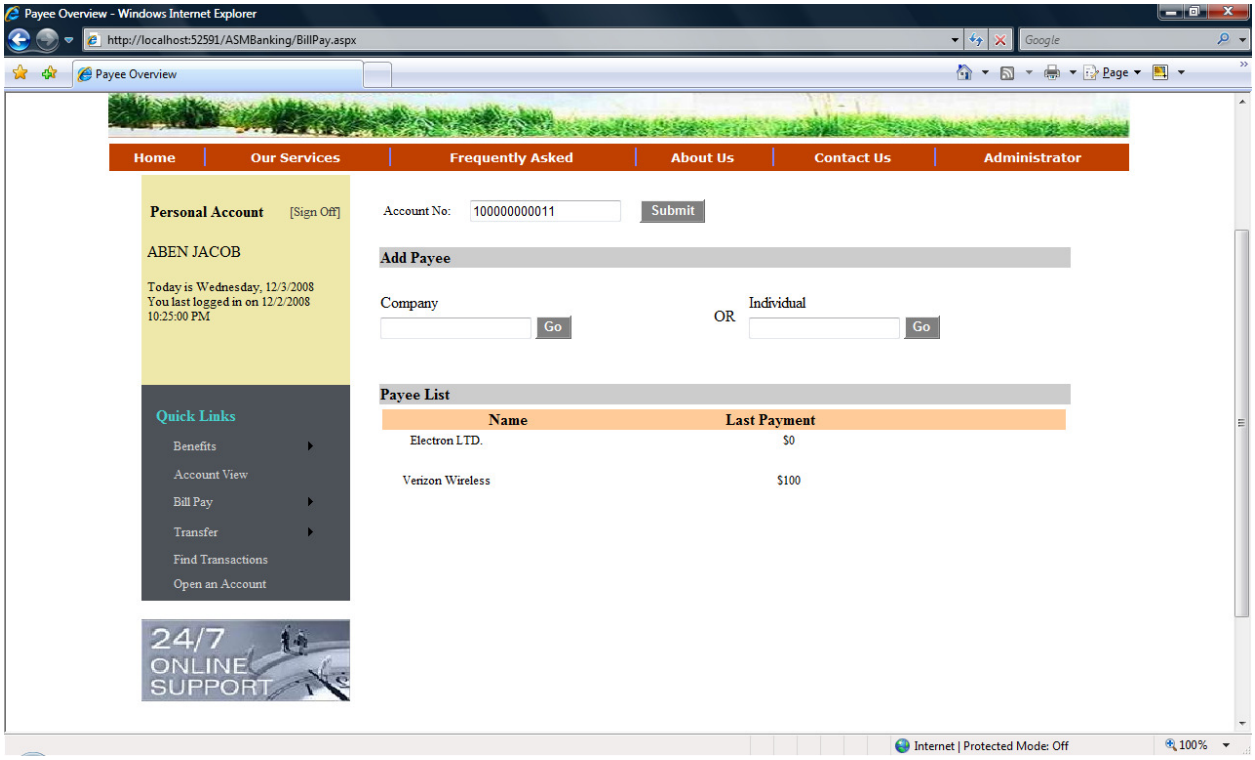


Fig: Payee List

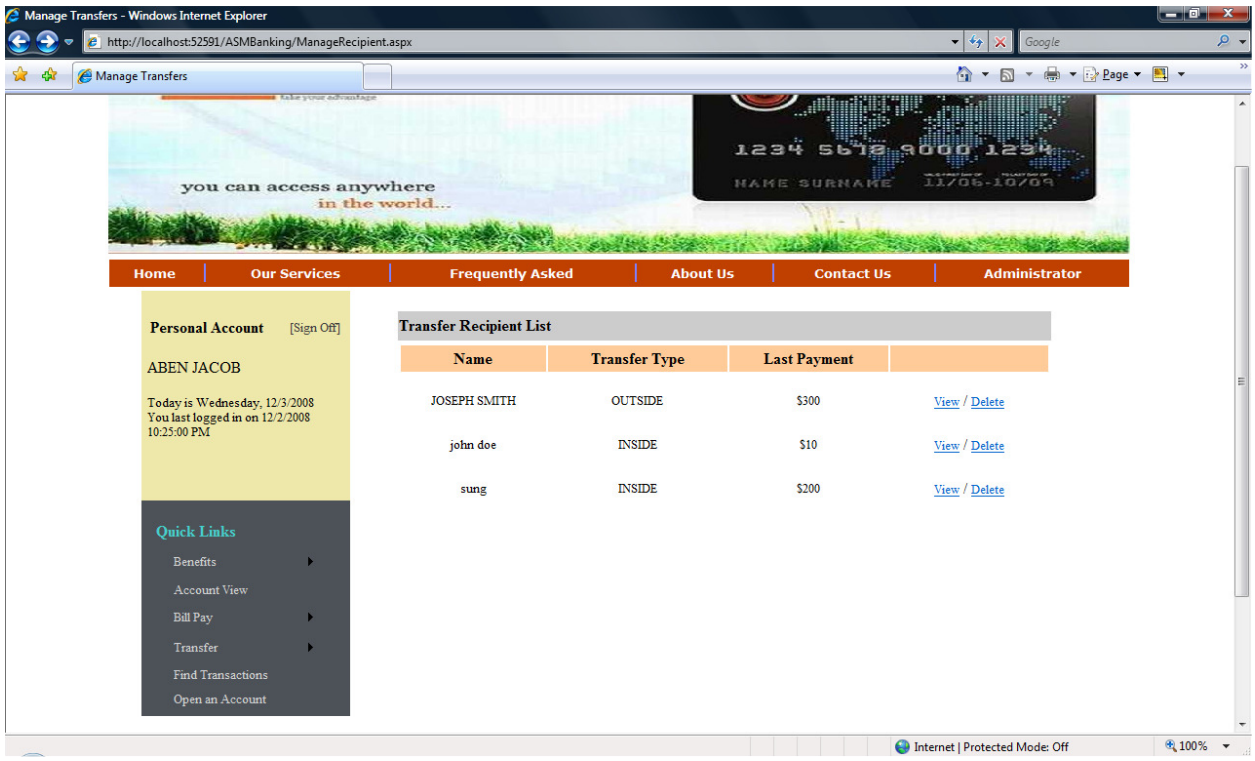


Fig: Transfer List

Administrator Sections:

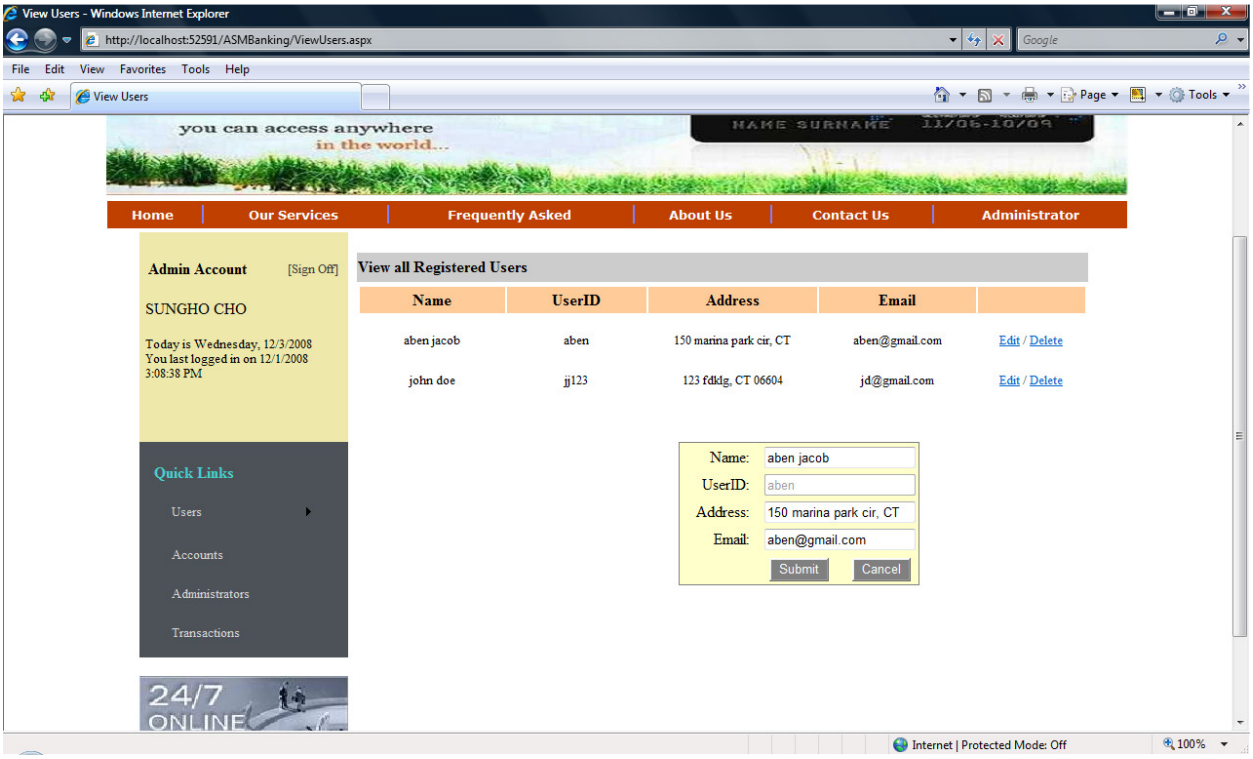


Fig: Online Users List

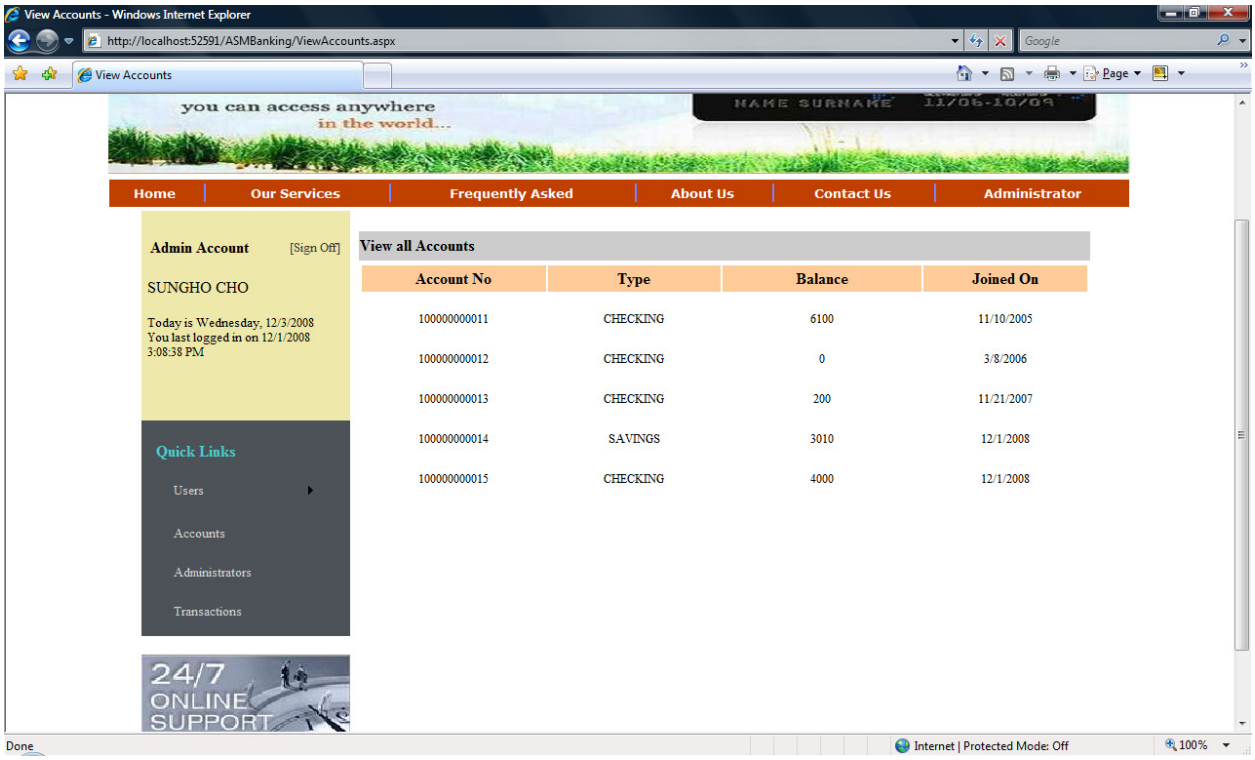


Fig: All available Accounts

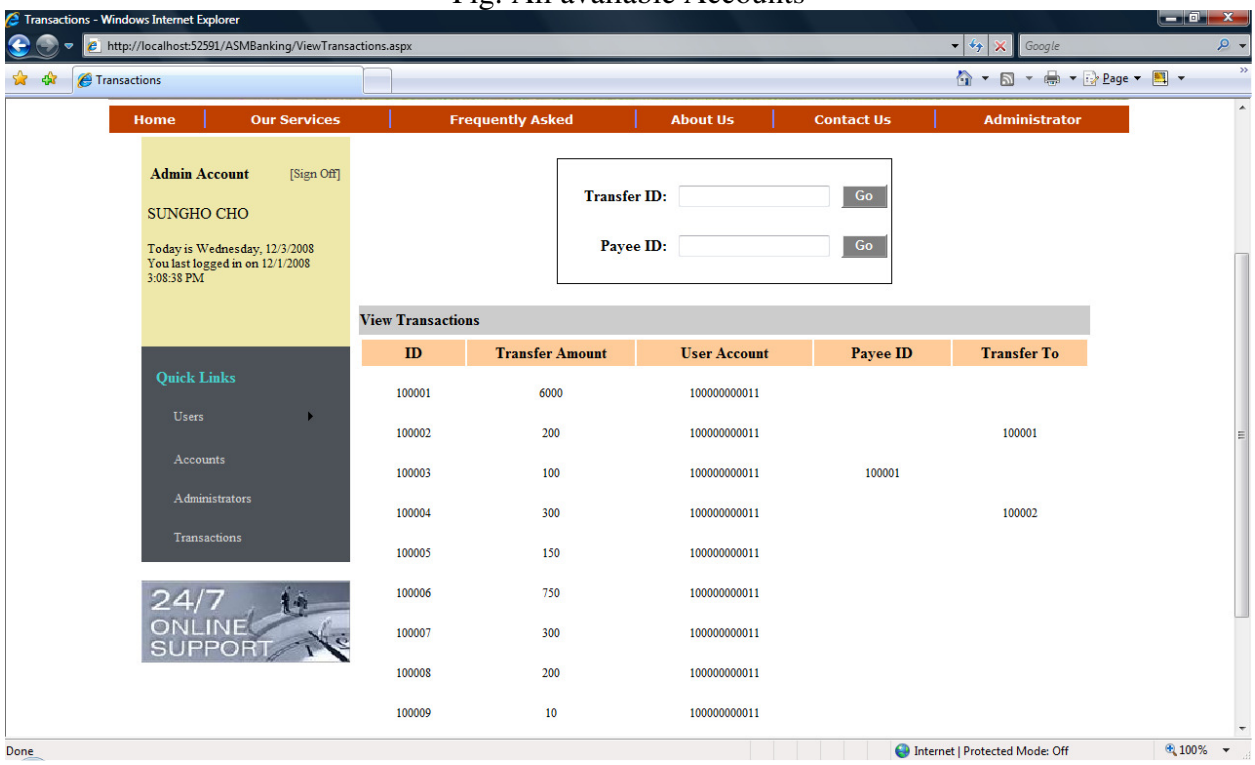


Fig: All available Transactions

Conclusion

By doing this project we were able to better understand the various concepts of any database. We learned how to create web pages. Finally, using C# we were able to figure out how to connect to .an Oracle server Database