KIT-Kalaignarkarunanidhi Institute of Technology

(An Autonomous Institution, Approved by AICTE & Affiliated to Anna University, Chennai)
Coimbatore-641 402

DEPARTMENT OF COMPUTER APPLICATIONS

Name	
Roll No.	
Register No.	
Class	

M23CAP201 DATABASE MANAGEMENT SYSTEMS

LABORATORY

RECORD NOTE BOOK

KIT-Kalaignarkarunanidhi Institute of Technology

(An Autonomous Institution, Approved by AICTE & Affiliated to Anna University, Chennai)
Coimbatore – 641 402

Department of

………

Record Work of …………………………………………………………………………………….*Laboratory*

Certified that this record is the bona fide work done by

Name:………………………………………………………………………………………………

Class:…………………………………………..   *Roll No:*………………………………………

Branch……………………………………………………………………………………………..

Place: KIT,CBE   *Faculty In-Charge*   *HOD*

Date:

University Register No……………………………………………

Submitted for the University Practical Examination held on

………………………………………………………

Internal Examiner   *External Examiner*

Practical Record Book Index Page

Sl. No.	Date	Name of the Experiment	Page Number	Program (20 Marks)	Observation (25 Marks)	Inference (10 Marks)	Viva-Voce (20 Marks)	Total (75 Marks)	Signature of the Faculty Member
01		CREATING BASE TABLE AND VIEW							
02		DATA MANIPULATION, SUBQUERY AND JOINS							
03		DATA CONTROL COMMANDS							
04		PL/SQL PROGRAMS USING PROCEDURE							
05		USE OF CURSOR, PROCEDURE & FUNCTION							
06		MONGODB CRUD OPERATIONS							
07		MONGODB INDEXING AND SHARDING							
08		XML DATABASE AND TABLE CREATIONS							
09		DESIGN AND IMPLEMENTATION OF PRODUCT MANAGEMENT							
10		DESIGN AND IMPLEMENTATION OF ACCOUNT MANAGEMENT							

Total Marks:______/75

Signature of the Faculty Member

Ex.No:1 **Date:**	**CREATING BASE TABLE AND VIEW**

AIM:

To create Base table & View Table in Oracle

ALGORITHM:

1.Connect to scott in oracle

2.Create the table employee with eno,ename,dept,sex,maritial status,age,education,designation,salary.

3.Insert records in employee table.

4.Create the table account with acno,bank,ac_type,branch,eno

5.Insert records in account table

6.Display the table

7.View the table by single table view and double table view acno,eno is primary key.

8.View the single table view with eno,ename,education,salary

9.View the double view with eno,ename,salary,bank,ac_type.

1.CREATE BASE TABLE:

Employee:

```
SQL> create table employee(eno number(5) PRIMARY KEY,ename
varchar(10),dept varchar(10),sex var
char(6),martialstaus varchar(10),age number(5),education
varchar(5),designation varchar(10),salary number(10));

Table created.
```

Account:

```
SQL> create table account(acno number(5) PRIMARY KEY,bank
varchar(10),ac_type varchar(10),branch varchar(10),eno
number(5),FOREIGN KEY(eno) REFERENCES employee(eno));

Table created.
```

```
SQL> desc employee;

 Name            Null?          Type
 -----------------------------------------
 ENO         NOT NULL       NUMBER(5)

 ENAME                      VARCHAR2(10)

 DEPT                       VARCHAR2(10)

 SEX                        VARCHAR2(6)

 MARTIALSTAUS               VARCHAR2(10)

 AGE                        NUMBER(5)

 EDUCATION                  VARCHAR2(5)

 DESIGNATION                VARCHAR2(10)

 SALARY                     NUMBER(10)


SQL> desc account

 Name             Null?          Type
 -----------------------------------------
 ACNO             NOT NULL     NUMBER(5)

 BANK                          VARCHAR2(10)

 AC_TYPE                       VARCHAR2(10)

 BRANCH                        VARCHAR2(10)

 ENO                           NUMBER(5)
```

Insert into Base table:

Employee:

```
SQL> insert into employee
values('&eno','&ename','&dept','&sex','&martialstatus','&age
','&education','&designation','&salary');
Enter value for eno: 1
Enter value for ename: ram

Enter value for dept: EEE

Enter value for sex: male

Enter value for martialstatus: single

Enter value for age: 22

Enter value for education: BE

Enter value for designation: staff

Enter value for salary: 20000

old   1: insert into employee
values('&eno','&ename','&dept','&sex','&martialstatus','&age','&educ
ation','designation','salary')

new   1: insert into employee
values('1','ram','EEE','male','single','22','BE','staff','20000')

1 row created.

SQL> /

Enter value for eno: 2

Enter value for ename: raja

Enter value for dept: ECE

Enter value for sex: male

Enter value for martialstatus: single

Enter value for age: 26

Enter value for education: ME

Enter value for designation: staff

Enter value for salary: 40000

old   1: insert into employee
values('&eno','&ename','&dept','&sex','&martialstatus','&age','&educ
ation','designation','salary')
```

```
new   1: insert into employee
values('2','raja','ECE','male','single','26','ME','staff','40000')

1 row created.

SQL> /

Enter value for eno: 3

Enter value for ename: sham

Enter value for dept: CSE

Enter value for sex: male

Enter value for martialstatus: single

Enter value for age: 25

Enter value for education: BE

Enter value for designation: staff

Enter value for salary: 25000

old   1: insert into employee
values('&eno','&ename','&dept','&sex','&martialstatus','&age','&
education’,’designation’,’salary’)

new   1: insert into employee
values('3','sham','CSE','male','single','25','BE','staff','25000')

1 row created.

SQL> /

Enter value for eno: 4

Enter value for ename: sony

Enter value for dept: IT

Enter value for sex: female

Enter value for martialstatus: single

Enter value for age: 23

Enter value for education: BE
Enter value for designation: programmer

Enter value for salary: 50000

old   1: insert into employee
values('&eno','&ename','&dept','&sex','&martialstatus','&age','&
education’,’designation’,’salary’)

new   1: insert into employee
values('4','sony','IT','female','single','23','prgm','','')

1 row created.

SQL> select * from employee;

ENO ENAME  DEPT SEX MARTIALSTATUS AGE EDUCATION DESIGNATION  SALARY

---------- ---------- ---------- ------ -------- ------------ -----
1    ram   EEE  male   single       22    BE      staff       20000

2    raja  ECE  male   single       26    ME      staff       40000

3    sham  CSE  male   single       25    BE      staff       25000

4    sony  IT   female single       23    BE      programmer  50000
```

Account:

```
SQL> insert into account
values('&acno','&bank','&ac_type','&branch','&eno');
SQL> /

Enter value for acno: 12345

Enter value for bank: ICICI

Enter value for ac_type: CURRENT

Enter value for branch: CBE

Enter value for eno: 2

old   1: insert into account
values('&acno','&bank','&ac_type','&branch','&eno')

new   1: insert into account
values('12345','ICICI','CURRENT','CBE','2')
```

```
1 row created.

SQL> /

Enter value for acno: 23451

Enter value for bank: SBI

Enter value for ac_type: SAVINGS

Enter value for branch: CBE

Enter value for eno: 3

old   1: insert into account
values('&acno','&bank','&ac_type','&branch','&eno')

new   1: insert into account
values('23451','SBI','SAVINGS','CBE','3')

1 row created.

SQL> /

Enter value for acno: 25361

Enter value for bank: STATE BANK

Enter value for ac_type: DEPOSIT

Enter value for branch: CBE

Enter value for eno: 1

old   1: insert into account
values('&acno','&bank','&ac_type','&branch','&eno')

new   1: insert into account values('25361','STATE
BANK','DEPOSIT','CBE','1')

1 row created.

SQL> /
Enter value for acno: 15632

Enter value for bank: FBI

Enter value for ac_type: FIXED
Enter value for branch: CBE

Enter value for eno: 4

old   1: insert into account
values('&acno','&bank','&ac_type','&branch','&eno')

new   1: insert into account values('15632','FBI','FIXED','CBE','4')

1 row created.

SQL> select * from account;

      ACNO   BANK         AC_TYPE    BRANCH     ENO
    ------ ---------- ---------- ---------- --------
     12345   ICICI        CURRENT    CBE          2

     23451   SBI          SAVINGS    CBE          3

     25361   STATE BANK  DEPOSIT    CBE          1

     15632   FBI          FIXED      CBE          4
```

2. CREATING VIEW TABLE:

```
SQL> create view emp_details as select eno,ename,education,salary
from employee;

View created.

SQL> select * from emp_details;

     ENO  ENAME    EDUCATION     SALARY
    ------- ---------- ----- ----------
      1    ram        BE         20000

      2    raja       ME         40000

      3    sham       BE         25000

      4    sony       BE         50000
```

```
SQL> create view emp_acc_details as select
e.eno,e.ename,e.salary,a.bank,a.acno,a.branch from employee
e,account a where e.eno=a.eno;

View created.
```

```
SQL> select * from emp_acc_details;

     ENO   ENAME      SALARY     BANK           ACNO   BRANCH
---------- ---------- ---------- ---------- ---------- ----------
      1     ram       20000    STATE BANK     25361   CBE
      2     raja      40000    ICICI          12345   CBE
      3     sham      25000    SBI            23451   CBE
      4     sony      50000    FBI            15632   CBE
```

RESULT

The database objects are created and executed successfully.

Ex.No:2
Date:

DATA MANIPULATION, SUBQUERY AND JOINS

AIM

To implement Data manipulation Command, Sub queries & Joins in Oracle

SQL COMMANDS

1. CREATE

Employee:

```
SQL>create table employee_join(employee_id number(5),last_name
varchar(10),first_name varchar(10),age number(2),did number(5));

Table created.
```

Department:

```
SQL>create table department_join(did number(5),department_name
varchar(10),employee_id number(5));

Table created.
```

2. INSERT

Employee:

```
SQL> insert into employee_join values('&employee_id','&last_name',

    '&first_name','&age', '&did');

Enter value for employee_id: 12

Enter value for last_name: tom

Enter value for first_name: jerry

Enter value for age: 22

Enter value for did: 1

old   1: insert into employee_join
values('&employee_id','&last_name', '&first_name','&age', '&did')

new   1: insert into employee_join values('12','tom', 'jerry','22',
'1')
```

```
1 row created.

SQL> /

Enter value for employee_id: 13

Enter value for last_name: ram

Enter value for first_name: mohan

Enter value for age: 26

Enter value for did: 2

old   1: insert into employee_join
values('&employee_id','&last_name', '&first_name','&age', '&did')

new   1: insert into employee_join values('13','ram', 'mohan','26',
'2')

1 row created.

SQL> /

Enter value for employee_id: 14

Enter value for last_name: saran

Enter value for first_name: raj

Enter value for age: 24

Enter value for did: 3

old   1: insert into employee_join
values('&employee_id','&last_name', '&first_name','&age', '&did')

new   1: insert into employee_join values('14','saran', 'raj','24',
'3')

1 row created.

SQL> /

Enter value for employee_id: 15

Enter value for last_name: sam

Enter value for first_name: kamelash
Enter value for age: 23

Enter value for did: 2

old   1: insert into employee_join
values('&employee_id','&last_name', '&first_name','&age', '&did')

new   1: insert into employee_join values('15','sam',
'kamelash','23', '2')

1 row created.

SQL> /

Enter value for employee_id: 16

Enter value for last_name: anu

Enter value for first_name: raj

Enter value for age: 22

Enter value for did: 1

old   1: insert into employee_join
values('&employee_id','&last_name', '&first_name','&age', '&did')

new   1: insert into employee_join values('16','anu', 'raj','22',
'1')

1 row created.
```

Department:

```
SQL> insert into department_join values('&did', '&department_name',

    '&employee_id');

Enter value for did: 1

Enter value for department_name: admin

Enter value for employee_id: 12

old   1: insert into department_join values('&did',
'&department_name', '&employee_id')

new   1: insert into department_join values('1', 'admin', '12')
```

```
1 row created.

SQL> /

Enter value for did: 2

Enter value for department_name: operation

Enter value for employee_id: 13

old   1: insert into department_join values('&did',
'&department_name', '&employee_id')

new   1: insert into department_join values('2', 'operation', '13')

1 row created.

SQL> /

Enter value for did: 3

Enter value for department_name: sales

Enter value for employee_id: 14

old   1: insert into department_join values('&did',
'&department_name', '&employee_id')

new   1: insert into department_join values('3', 'sales', '14')

1 row created.

SQL> /

Enter value for did: 4

Enter value for department_name: marketing

Enter value for employee_id:

old   1: insert into department_join values('&did',
'&department_name', '&employee_id')

new   1: insert into department_join values('4', 'marketing', '')

1 row created.

SQL> /

Enter value for did: 5
Enter value for department_name: analysis

Enter value for employee_id:

old   1: insert into department_join values('&did',
'&department_name', '&employee_id')

new   1: insert into department_join values('5', 'analysis', '')

1 row created.
```

3. SELECT

```
SQL> Select * from employee_join;

EMPLOYEE_ID LAST_NAME   FIRST_NAME    AGE       DID
 12           Tom        Jerry        22         1
 13           Ram        Mohan        26         2
 14           Saran      Raj          24         3
 15           Sam        Kamelash     23         2
 16           Anu        Raj          22         1
SQL> Select * from department_join;

DID          DEPARTMENT             EMPLOYEE_ID
----------   ----------             -----------
1            admin                         12
2            operation                     13
3            sales                         14
4            marketing
5            analyses
```

4. JOIN

4. a) Left outer join:

```
    SQL> select e.last_name,d.department_name As department from
    employee_join e NATURAL LEFT OUTER JOIN  department_join d;
LAST_NAME       DEPARTMENT
----------      ----------
Tom             admin
Ram             operation
Saran           sales
```

```
Anu
Sam
```

4. b).Right outer join:

```
SQL> Select e.last_name,e.first_name,d.department_name As
department from employee_join e NATURAL RIGHT OUTER JOIN
department_join d;

LAST_NAME      FIRST_NAME        DEPARTMENT
----------     ----------        ----------
Tom            jerry             admin
Ram            Mohan             operation
Saran          raj               sales
                                 Analysis
                                 Marketing
```

4. c) Full outer join:

```
SQL> Select e.last_name,d.department_name As department from
employee_join e NATURAL FULL OUTER JOIN department_join d;

LAST_NAME       DEPARTMENT
----------      ----------
Tom             admin
Ram             operation
Saran           sales
Anu
Sam
                Marketing
                Analysis
7 rows selected.
```

4. d) Inner join:

```
SQL> Select e.last_name,d.department_name As department from
employee_join e NATURAL INNER JOIN department_join d;

LAST_NAME    DEPARTMENT
```

```
----------   ----------
Tom          admin
Ram          operation
Saran        sales
```

4. e) Self join:

```
SQL> Create table employee_join(employee_id number(5),last_name
varchar(10),first_name varchar(10),manager varchar(10),mid
number(5));

Table created.

SQL>Insert into employeee_jovalues(‘&employee_id’,‘&last_name’,
‘&first_name’,‘&manager’,‘&mid’);
Enter value for employee_id: 12

Enter value for last_name: tom

Enter value for first_name: jerry

Enter value for manager: x

Enter value for mid: 12

old   1:  Insert into employee_join
values('&employee_id','&last_name', '&first_name','&manager','

new 1:  Insert into employee_join values('12','tom',
'jerry','12','12')

1 row created.

SQL>/

Enter value for employee_id: 13

Enter value for last_name: Scooby

Enter value for first_name: dobby

Enter value for manager: y

Enter value for mid: 13
```

```
old   1: Insert into employee_join
values('&employee_id','&last_name', '&first_name','&manager','

new 1: Insert into employee_join values('13','scooby',
'dooby','13','13')

1 row created.

SQL> /

Enter value for employee_id: 14

Enter value for last_name: martin

Enter value for first_name: ceriman

Enter value for manager: z

Enter value for mid: 14

old   1: Insert into employee_join
values('&employee_id','&last_name', '&first_name','&manager','

new 1: Insert into employee_join values('14','martin',
'cerman','14','14')

1 row created.

SQL> /

Enter value for employee_id: 15

Enter value for last_name: arun

Enter value for first_name: changer

Enter value for manager:

Enter value for mid:

old   1: Insert into employee_join
values('&employee_id','&last_name', '&first_name','&manager','

new 1: Insert into employee_join values('15','arun',
'chander','','')

1 row created.

SQL> /
```

```
Enter value for employee_id: 16

Enter value for last_name: jerry

Enter value for first_name: loops

Enter value for manager:

Enter value for mid:

old   1: Insert into employee_join
values('&employee_id','&last_name', '&first_name','&manager','

new 1: Insert into employee_join values('16','jerry', 'loops','','')

1 row created.

SQL>

1 row created.

SQL> Select * from employee;

EMPLOYEE_ID    LAST_NAME    FIRST_NAME    Manager   MID

 12            Tom           Jerry         x         12

 13            Scooby        dobby         y         13

 14            martin        ceriman       z         14

 15            arun          changer

 16            jerry         loops

SQL>Select m.manager || 'work for'||e.last_name||' '||e.first_name
As "employee and their manager " FROM employee m.employee e where
e.eid=m.mid;

EMPLOYEE AND THEIR MANAGER

X work for tom jerry

Y work for Scooby dobby

Z work for martin ceriman
```

5. Sub queries / Nested queries:

Account:

Account number	Customer name	balance
1	Abi	20000
2	Anil	30000
3	Banu	80000
4	Devi	90000

Loan:

Loan number	Customer name	Amount
1	Abi	10000
2	Anand	60000
3	Anu	50000
4	Banu	70000

5. A) IN:

```
SQL> Select distinct customer_name from loan where
customer_name IN (select customer_name from account);

CNAME
-----------
Abi

banu
```

5. B) NOT IN:

```
SQL>Select distinct customer_name from loan where customer_name
NOT_IN (select customer_name from account);

CNAME
----------
Anand

Anu
```

5. C) SOME / ANY AND ALL:

Branch Name	City	Asset

Ram Nagar	Coimbatore	70000
T Nagar	Chennai	90000
Town street	Salem	37000
Back street	Trichy	30000
Ram Nagar	Karur	21000
Peelamedu	Coimbatore	20000

SOME ANY:

```
SQL> Select branch_name, city from branch where asset>some (select
asset from branch where city='Coimbatore');

BNAME           CITY

----------     ----------

Ramnagar          cbe

T.nagar          chennai

Town street      salem

Back street      trichy

Ramnagar         karur
```

RESULT:

Thus the above data manipulation joins and sub queries are created successfully.

Ex.No:3

Date:

DATA CONTROL COMMANDS

AIM

To implement data control commands in Oracle.

ALGORITHM

1. Connect to scott in oracle.
2. Create a table balance with cus_name,ano,bal and insert records.
3. Connect to system/manager.
4. Create new user with session privilege.
5. Connect to the new user and check the privilege we can select,insert,update database.
6. Connect to system/manager and give privilege to the new user for accessing the balance table of scott.
7. Connect to new user and type the select query to access balance table of scott.
8. Connect to system/manger and give DBA privilege to new user.
9. Connect to new user and check the DBA privilege.
10. Create a user from new user with DBA privilege, grant privilege etc.
11. Connect to system/manager and revoke all privilege using REVOKE command and Check.

SQL COMMANDS:

```
SQL> create table balance(cname varchar(10),ano number, bal number);

Table created.

SQL> insert into balance values('&cname','&ano','&bal');

Enter value for cname: soni

Enter value for ano: 123
Enter value for bal: 1000
old   1: insert into balance values('&cname','&ano','&bal')
new   1: insert into balance values('soni','123','1000')
1 row created.
Enter value for cname: sasi
Enter value for ano: 144
Enter value for bal: 20000
old   1: insert into balance values('&cname','&ano','&bal')
new   1: insert into balance values('sasi','144','20000')
1 row created.
Enter value for cname: sangi
Enter value for ano: 122
Enter value for bal: 20000
old   1: insert into balance values('&cname','&ano','&bal')
new   1: insert into balance values('sangi','122','20000')
1 row created.
SQL> select * from balance;
CNAME             ANO        BAL
---------- ---------- ----------
soni              123       1000
sasi              144      20000
sangi             122      20000
```

DATA CONTROL COMMANDS

```
SQL> connect system/manager;
Connected.
SQL> create user abcde identified by sangi;
```

```
User created.
SQL>Connect
Enter user-name: abcde
Enter password: *****
ERROR:
ORA-01045: user SANGEE lacks CREATE SESSION privilege; logon denied
SQL>connect system/manager;
Connected.
SQL>grant create session to abcde;
Grant succeeded.
SQL> connect
Enter user-name: abcde
Enter password: *****
Connected.
SQL> select * from scott.balance;
select * from balance
              *
ERROR at line 1:
ORA-00942: table or view does not exist
SQL> connect
Enter user-name: scott
Enter password: *****
Connected.
SQL> grant select,insert on balance to abcde;
Grant succeeded.
SQL> connect
```

```
Enter user-name: abcde

Enter password: *****

Connected.

SQL> select * from scott.balance;

CNAME             ANO        BAL

---------- ---------- ----------

soni              123       1000

sasi              144      20000

sangi             122      20000

SQL> connect

Enter user-name: scott

Enter password: *****

Connected.

SQL> revoke select on balance from abcde;

Revoke succeeded.

SQL> connect

Enter user-name: abcde

Enter password: *****

Connected.

SQL> select * from scott.balance;

select * from scott.balance

                    *

ERROR at line 1:

ORA-01031: insufficient privileges

SQL> connect system/manager;

Connected.
```

```
SQL> grant dba to abcde;

Grant succeeded.

SQL> connect

Enter user-name: abcde

Enter password: *****

Connected.

SQL> select * from scott.balance;

CNAME                 ANO        BAL

---------- ---------- ----------

soni                  123       1000

sasi                  144      20000

sangi                 122      20000

SQL> create table bal(accno number,bal number);

Table created.
```

RESULT

The data control commends using SQL are created and executed successfully.

Ex.No:4 **DATE:**	**PL/SQL PROGRAMS USING PROCEDURE**

1) BANK TRANSACTION

AIM

To create a pl/sql program for bank transaction.

ALGORITHM

1. Start the process.
2. Create a account table[name,cur_bal,accno].
3. Insert the values into account table.
4. Write a pl/sql program for account table.
5. If the current balance is less than 1000 deduct 100rupees from The current balance.
6. Execute the pl/sql program.
7. Execute the program.
8. View the account table.
9. Stop the process.

PROGRAM

```
SQL> create table acct(name varchar(15),curbal number,acno number);
     Table created.

SQL> insert  into acct values('&name','&curbal','&acno');

SQL>insert into  acct values (‘jasmine’,’7000’,’11);

SQL>insert into acct values (‘sakthi’,’2000’,’12’);

SQL> insert into acct values(hari’,’900’,’33);

SQL>insert into scct values (‘jackson’,’600’,’47’);

SQL> insert into acct values(‘dhanush’,’4900’,’69’);
SQL> select * from acct;

NAME                    CURBAL       ACNO
--------------- ---------- ----------
jasmine                   7000         11
sakthi                    2000         12
hari                      900          33
jacksonon                 600          47
dhanush                   4900         69

SQL> set serveroutput on
SQL> ed bank.sql
SQL> ed bank;
     declare
            mano number(7);
            mcb number(6,2);
            minibal  constant number(6,2) :=1000.00;
            fine number(5,2) := 100.00;
     begin
           mano:=&mano;
           select curbal into mcb from acct  where acno=mano;
           if mcb<minibal then
           update acct set curbal= curbal-fine where acno=mano;
           end if;
     end;
SQL> select * from acct;
```

```
NAME                CURBAL       ACNO

--------------- ---------- ----------

jasmine               7000         11

sakthi                2000         12

hari                   900         33

jackson                600         47

dhanush               4900         69

SQL> ed bank;

SQL> select * from acct;

NAME                CURBAL       ACNO

--------------- ---------- ----------

jasmine               7000         11

sakthi                2000         12

hari                   900         33

jackson                600         47

dhanush               4900         69


SQL> ed bank;

SQL> @ bank;

 13  /

Enter value for mano: 33

old   7:    mano:=&mano;

new   7:    mano:=33;
```

```
PL/SQL procedure successfully completed.

SQL> select * from acct;

NAME                CURBAL       ACNO

--------------- ---------- ----------

jasmine               7000         11

sakthi                2000         12

hari                   800         33

jackson                600         47

dhanush               4900         69

SQL> @ bank;

Enter value for mano: 47

old   7:    mano:=&mano;

new   7:    mano:=47;

PL/SQL procedure successfully completed.

SQL> select * from acct;

NAME                CURBAL       ACNO
--------------- ---------- ----------
jasmine               7000         11

sakthi                2000         12

hari                   800         33

jackson                500         47

dhanush               4900         69
```

2)Electricity bill:

AIM:

To create a pl/sql program for electricity bill.

ALGORITHM:

1.Start the process.
2.Create a electricity table[sno,name,address,units,total]
3.Insert the values for sno,name,address.
4.Create pl/sql program declare the variables sno,name,address,units,total.
5.Using if statement to calculate the total based of the units consumed.
6.Using update statement update the unit and total field in the electricity table.
7.Execute the pl/sql program.
8.View the electricity table.
9.Stop the process.

PROGRAM:

```
SQL> set serveroutput on
SQL> ed electricity
SQL> @ electricity
SQL> /
Enter value for sno: 205
old   7:        sno:=&sno;
new   7:        sno:=205;
Enter value for name: Divya
old   8:        name:='&name';
new   8:        name:='Divya';
Enter value for units: 90
old   9:        units:=&units;
new   9:        units:=90;
Total amount:90
SQL> /
```

```
Enter value for sno: 562
old   7:        sno:=&sno;
new   7:        sno:=562;
Enter value for name: Raj
old   8:        name:='&name';
new   8:        name:='Raj';
Enter value for units: 50
old   9:        units:=&units;
new   9:        units:=50;
Total amount:25
SQL> /
Enter value for sno: 654
old   7:        sno:=&sno;
new   7:        sno:=654;
Enter value for name: Giri
old   8:        name:='&name';
new   8:        name:='Giri';
Enter value for units: 130
old   9:        units:=&units;
new   9:        units:=130;
Total amount:260
PL/SQL procedure successfully completed.
```

PL/SQL PROGRAM:

```
declare
      sno number;
      name varchar(15);
      units number;
      total number;
begin
       sno:=&sno;
       name:='&name';
       units:=&units;
       if(units<=50) then
        total:=units*0.50;
       else if(units>50 and units<=100) then
         total:=units*1.00;
       else if(units>100 and units<=150) then
         total:=units*2.00;
       else if(units>150 and units<=200) then
         total:=units*2.50;
       end if;
       end if;
       end if;
       end if;
       dbms_output.put_line('Total amount:'||total);
end;
```

3)Student marklist:

AIM:

To create a pl/sql program for student marklist.

ALGORITHM:

1.Start the process.
2.Create a student table
  [rollno,name,mark1,mark2,total,average,result,grade].
3.Insert the values for rollno,name,mark1,mark2.
4.Create pl/sql program declare the variables for rollno,name,
  mark1,mark2,total,average,result,grade.
5.Using if statement calculate the result and based on the marks
  scored.
6.Using update statement update the total,result and grade in
  Student table.
7.Execute the pl/sql program.
8.View the student table.
9.Stop the process.

PROGRAM:

```
SQL> create table  student(Rollno number,Name varchar(15),Mark1
number,Mark2 number,Total number,Average number,Result
varchar(5),Grade varchar(25));
Table created.

SQL> insert into student
values('&Rollno','&Name','&Mark1','&Mark2','&Total','&Average','&Res
ult','&Grade');
Enter value for rollno: 10
Enter value for name: Janani
Enter value for mark1: 56
Enter value for mark2: 29
Enter value for total:
Enter value for average:
Enter value for result:
Enter value for grade:
old   1: insert into student
values('&Rollno','&Name','&Mark1','&Mark2','&Total','&Average','&Res
ult’,'&Grade');
new   1: insert into student
values('10','Janani','56','29','','','','')

1 row created.

SQL> /
Enter value for rollno: 15
Enter value for name: Nithya
Enter value for mark1: 96
Enter value for mark2: 87
Enter value for total:
Enter value for average:
Enter value for result:
Enter value for grade:
old   1: insert into student
values('&Rollno','&Name','&Mark1','&Mark2','&Total','&Average','&Res
ult’,'&Grade');
new   1: insert into student
values('15','Nithya','96','87','','','','')

1 row created.

SQL> /
Enter value for rollno: 12
Enter value for name: Arun
Enter value for mark1: 23
Enter value for mark2: 35
Enter value for total:
Enter value for average:
Enter value for result:
Enter value for grade:
old   1: insert into student
values('&Rollno','&Name','&Mark1','&Mark2','&Total','&Average','&Res
ult’,'&Grade');
```

```
new   1: insert into student
values('12','Arun','23','35','','','','')

1 row created.

SQL> /
Enter value for rollno: 25
Enter value for name: Preethi
Enter value for mark1: 98
Enter value for mark2: 97
Enter value for total:
Enter value for average:
Enter value for result:
Enter value for grade:
old   1: insert into student
values('&Rollno','&Name','&Mark1','&Mark2','&Total','&Average','&Res
ult,'&Grade');
new   1: insert into student
values('25','Preethi','98','97','','','','')

1 row created.

SQL> /
Enter value for rollno: 20
Enter value for name: John
Enter value for mark1: 65
Enter value for mark2: 54
Enter value for total:
Enter value for average:
Enter value for result:
Enter value for grade:
old   1: insert into student
values('&Rollno','&Name','&Mark1','&Mark2','&Total','&Average','&Res
ult,'&Grade');
new   1: insert into student
values('20','John','65','54','','','','')

1 row created.

SQL> select * from student;

ROLLNO  NAME  MARK1      MARK2      TOTAL    AVERAGE RESULT GRADE
```

```
---------- --------------- ---------- ---------- ---------- ----
  10  Janani   56         29
  15  Nithya   96         87
  12  Arun     23         35
  25  Preethi  98         97
  20  John     65         54
```

PL/SQL:

```
declare
  cursor student is select * from student;
begin
  for i in student
  loop
     i.total:=i.mark1+i.mark2;
     i.average:=i.total/2;
     update student set total=i.total,average=i.average where
     rollno=i.rollno;
     if(i.mark1>=40)and(i.mark2>=40) then
        update student set result='Pass' where rollno=i.rollno;
     else
        update student set result='Fail' where rollno=i.rollno;
     end if;
     if(i.average>95)then
        update student set grade='1st class with distinct' where
        rollno=i.rollno;
     end if;
     if(i.average>90)and(i.average<=95)then
        update student set grade='1st class' where rollno=i.rollno;
     end if;
     if(i.average>75)and(i.average<=90) then
        update student set grade='2nd class' where rollno=i.rollno;
     end if;
     if(i.average>40)and(i.average<=75) then
        update student set grade='3rd class' where rollno=i.rollno;
     end if;
     if(i.average<40) then
        update student set grade='Fail' where rollno=i.rollno;
    end if;
```

```
   end loop;
end;
```

```
SQL> set serveroutput on
SQL> ed stud
SQL> @ stud

PL/SQL procedure successfully completed.

SQL> select * from student;

ROLLNO NAME MARK1 MARK2 TOTAL AVERAGE RESULT GRADE
---------- --------------- ---------- ---------- ----------
10 Janani   56    29     85    42.5    Fail  3rd class
15 Nithya   96    87     183   91.5    Pass  1st class
12 Arun     23    35     58    29      Fail  Fail
25 Preethi  98    97     195   97.5    Pass  1st class with distinct
20 John     65    54     119   59.5    Pass  3rd class
```

RESULT

The database objects are created and executed successfully.

Ex.No:5 Date:	USE OF CURSOR, PROCEDURE & FUNCTION

AIM:

To implement CURSOR,PROCEDURE,FUNCTION for Employee in Oracle.

ALGORITHM:

1.Connect to scott in oracle
2.Create the table employee with eno,ename,dept,sex,maritial status,age,education,designation,salary.
3.Insert records in employee table.
4.Create the table account with acno,bank,ac_type,branch,eno
5.Insert records in account table
6.Display the table
7.View the table by single table view and double table view acno,eno is primary key.
8.View the single table view with eno,ename,education,salary
9.View the double view with eno,ename,salary,bank,ac_type.

Create table:

```
SQL> create table tax(name char(15),basicpay number,netpay
number,grosspay number,
allowancenumber,deduction number,incometax number);

Table created.
```

Insert table:

```
SQL> insert into
SQL> insert into tax
values('&name','&basicpay','&netpay','&grosspay','&allowance',
'&deduction','&incometax');
Enter value for name: dhivya
Enter value for basicpay: 100000
Enter value for netpay:
Enter value for grosspay:
Enter value for allowance:
Enter value for deduction:
Enter value for incometax:
old   1: insert into tax
values('&name','&basicpay','&netpay','&grosspay','&allowance','&dedu
ction',
new   1: insert into tax values('dhivya','100000','','','','','')

1 row created.

SQL> /
Enter value for name: priya
Enter value for basicpay: 200000
Enter value for netpay:
Enter value for grosspay:
Enter value for allowance:
Enter value for deduction:
Enter value for incometax:
old   1: insert into tax
values('&name','&basicpay','&netpay','&grosspay','&allowance','&dedu
ction',
new   1: insert into tax values('priya','200000','','','','','')

1 row created.

SQL> /
Enter value for name: hari
Enter value for basicpay: 250000
Enter value for netpay:
Enter value for grosspay:
Enter value for allowance:
Enter value for deduction:
Enter value for incometax:
old   1: insert into tax
values('&name','&basicpay','&netpay','&grosspay','&allowance','&dedu
ction',
new   1: insert into tax values('hari','250000','','','','','')

1 row created.
```

```
SQL> select * from tax;

NAME  BASICPAY   NETPAY  GROSSPAY   ALLOWANCE   DEDUCTION  INCOMETAX
--------------- ---------- ---------- ---------- ---------- --------
dhivya 100000
priya  200000
hari   250000
```

FUNCTION

```
SQL> set serveroutput on;
SQL> ed tax.sql
create or replace function taxing(n in number,base in number)
return number is net number(10);
begin
  net:=n*12;
  if(net<100000)then
    return 0;
  else if((net>=100000)and(net<200000))then
    return(base*0.3);
  else
    return(base*0.5);
  end if;
  end if;
end;

SQL> @ tax.sql
     /
Function created.
```

PROCEDURE

```
SQL> ed  p1
create or replace procedure p1 is cursor c is select * from tax;
begin
 for i in c
 loop
     i.allowance:=i.basicpay*0.2;
     i.deduction:=i.basicpay*0.1;
     i.grosspay:=i.basicpay+i.allowance;
     i.netpay:=i.grosspay-i.deduction;
     i.incometax:=taxing(i.netpay,i.basicpay);
     update tax set allowance=i.allowance,deduction=i.deduction,
     grosspay=i.grosspay,netpay=i.netpay,incometax=i.incometax
     where name=i.name;
 end loop;
end;

SQL> @ P1
     /
Procedure created.

SQL> declare
  2  begin
  3  p1;
  4  end;
     /
PL/SQL procedure successfully completed.

SQL> select * from tax;

NAME   BASICPAY  NETPAY  GROSSPAY   ALLOWANCE  DEDUCTION  INCOMETAX
------------------------------------------------------- -------------
dhivya 100000     110000     120000      20000      10000      50000
priya  200000     220000     240000      40000      20000     100000
hari   250000     275000     300000      50000      25000     125000
```

RESULT

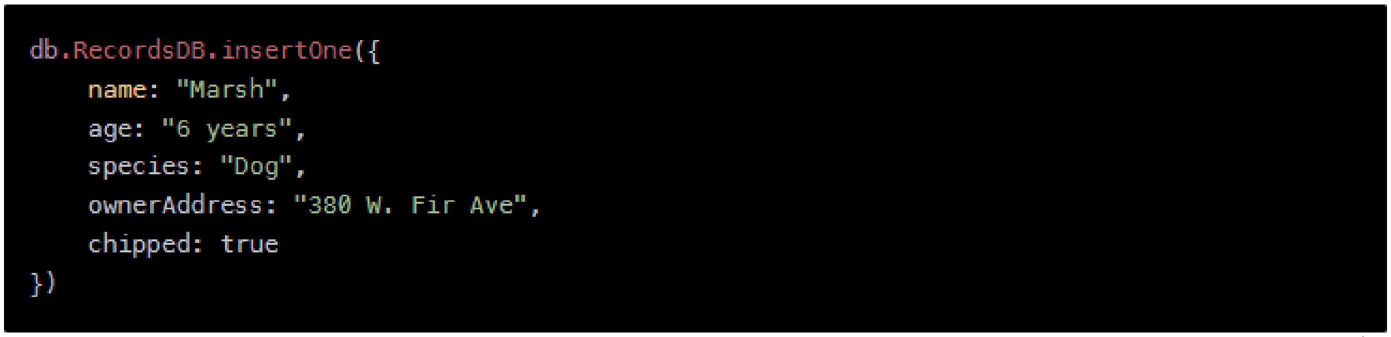
The database objects using cursor, procedure and function has been created and executed successfully.

Ex.No: 6	**MONGODB CRUD OPERATIONS**
Date:	

AIM : To perform CURD operations using MongoDB

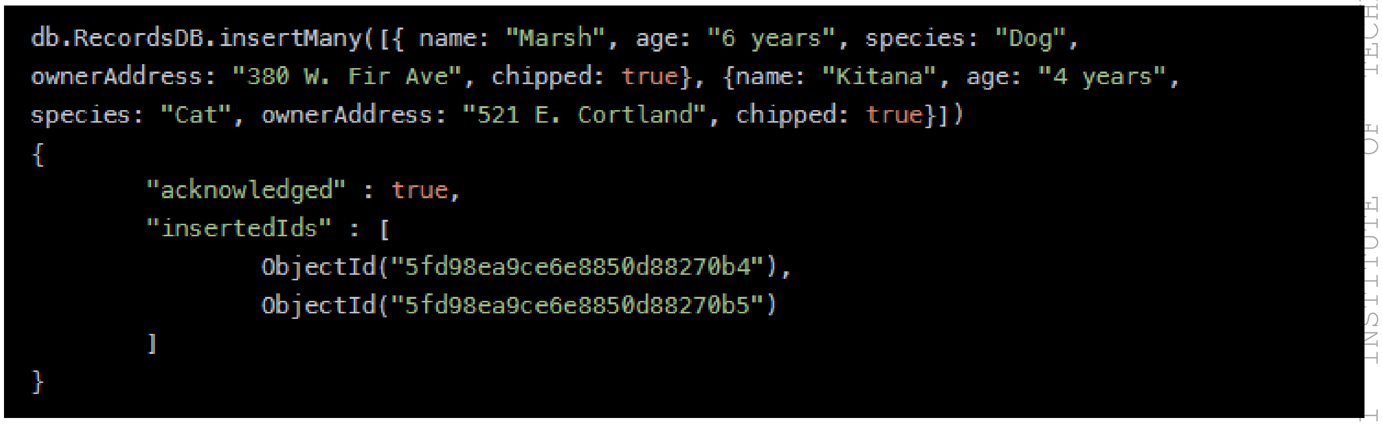
PROGRAM

Step 1: Create operations using insertOne() function



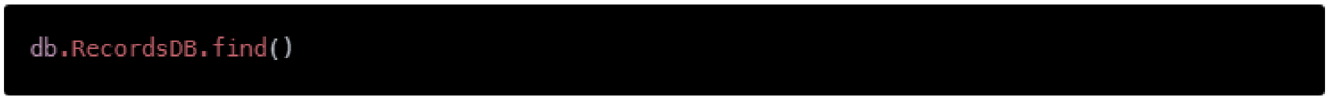
```
db.RecordsDB.insertOne({
    name: "Marsh",
    age: "6 years",
    species: "Dog",
    ownerAddress: "380 W. Fir Ave",
    chipped: true
})
```

Step 2: Create operations using insertMany() function

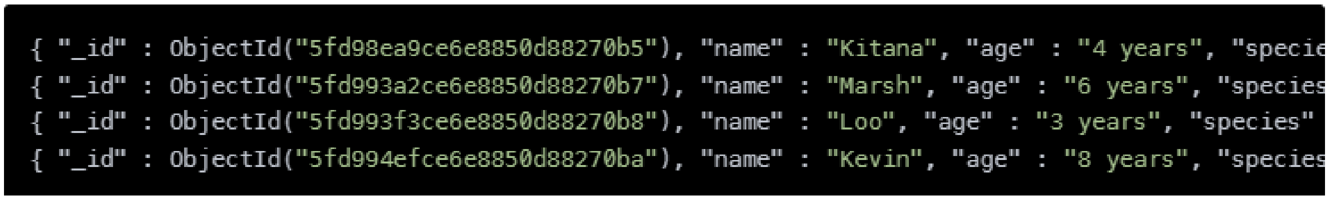


```
db.RecordsDB.insertMany([{ name: "Marsh", age: "6 years", species: "Dog",
ownerAddress: "380 W. Fir Ave", chipped: true}, {name: "Kitana", age: "4 years",
species: "Cat", ownerAddress: "521 E. Cortland", chipped: true}])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5fd98ea9ce6e8850d88270b4"),
                ObjectId("5fd98ea9ce6e8850d88270b5")
        ]
}
```

Step 3: Read Operation

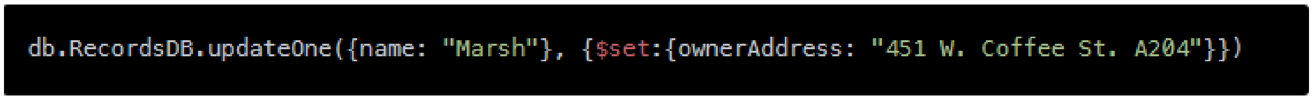


```
db.RecordsDB.find()
```

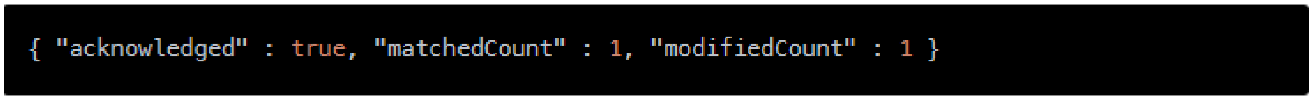


```
{ "_id" : ObjectId("5fd98ea9ce6e8850d88270b5"), "name" : "Kitana", "age" : "4 years", "specie
{ "_id" : ObjectId("5fd993a2ce6e8850d88270b7"), "name" : "Marsh", "age" : "6 years", "species
{ "_id" : ObjectId("5fd993f3ce6e8850d88270b8"), "name" : "Loo", "age" : "3 years", "species"
{ "_id" : ObjectId("5fd994efce6e8850d88270ba"), "name" : "Kevin", "age" : "8 years", "species
```

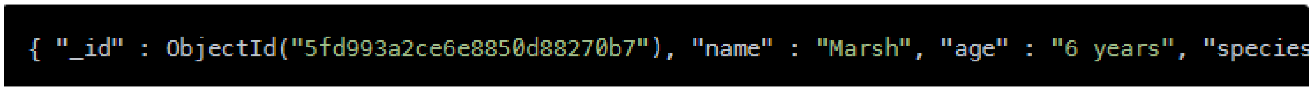
Step 4: update Operation



```
db.RecordsDB.updateOne({name: "Marsh"}, {$set:{ownerAddress: "451 W. Coffee St. A204"}})
```

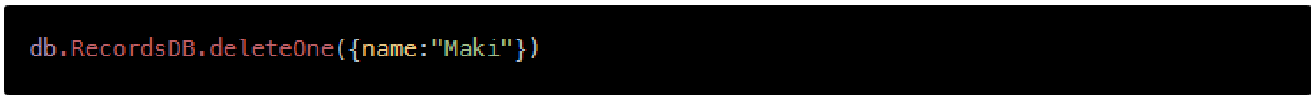


```
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

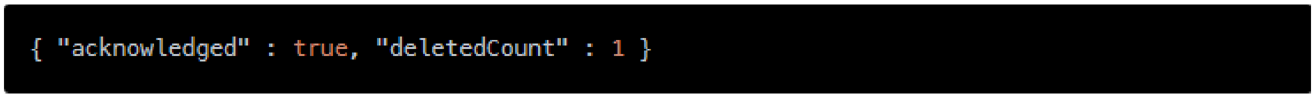


```
{ "_id" : ObjectId("5fd993a2ce6e8850d88270b7"), "name" : "Marsh", "age" : "6 years", "species
```

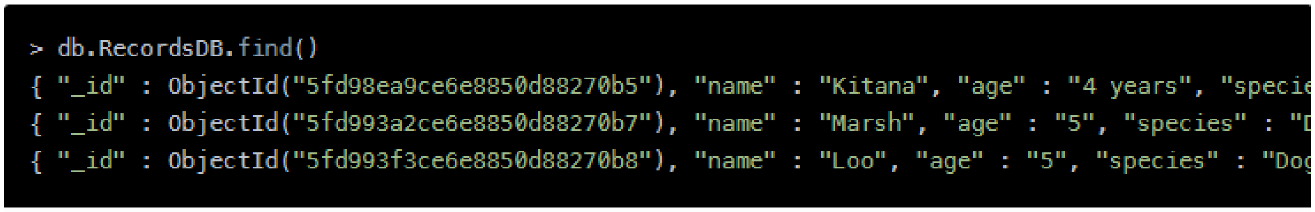
Step 5: Delete Operation



```
db.RecordsDB.deleteOne({name:"Maki"})
```



```
{ "acknowledged" : true, "deletedCount" : 1 }
```



```
> db.RecordsDB.find()
{ "_id" : ObjectId("5fd98ea9ce6e8850d88270b5"), "name" : "Kitana", "age" : "4 years", "specie
{ "_id" : ObjectId("5fd993a2ce6e8850d88270b7"), "name" : "Marsh", "age" : "5", "species" : "D
{ "_id" : ObjectId("5fd993f3ce6e8850d88270b8"), "name" : "Loo", "age" : "5", "species" : "Dog
```

RESULT:

Thus CURD operations using MongoDB successfully executed and verified.

Ex.No:7 **Date:**	**MONGODB INDEXING AND SHARDING**

AIM: To Perform Indexing and Sharding using MongoDB

PROGRAM:

Indexing

Step 1: MongoDB provides a method called createIndex() that allows user to create an index.

```
db.mycol.createIndex({“age”:1})
{
“createdCollectionAutomatically” : false,
“numIndexesBefore” : 1,
“numIndexesAfter” : 2,
“ok” : 1
}
```

Step 2: In order to drop an index, MongoDB provides the dropIndex() method.

```
db.NAME_OF_COLLECTION.dropIndex({KEY:1})
```

In order to delete (or drop) multiple indexes from the collection, MongoDB provides the dropIndexes() method that takes multiple indexes as its parameters.

```
db.NAME_OF_COLLECTION.dropIndexes({KEY1:1, KEY2: 1})
```

Step 3: The getIndexes() method in MongoDB gives a description of all the indexes that exists in the given collection.

```
db.NAME_OF_COLLECTION.getIndexes()
```

It will retrieve all the description of the indexes created within the collection.

Sharding

Step 1: Creating a Directory for Config Server

The first step to be performed in order to set up MongoDB Sharding would be to create a separate directory for Config Server. This can be done using the following command:

```
mkdir /data/configdb
```

Step 2: Starting MongoDB Instance in Configuration Mode

One Server has to be set up as the Configuration Server. Suppose you have a Server named "**ConfServer**" which would be used as the Configuration Server, the following command can be executed to perform that operation:

```
mongod –configdb ConfServer: 27019
```

Step 3: Starting Mongos Instance

Once the Configuration Server has been set up, the Mongos Instance can be started by executing the following command along with the **name of your Configuration Serve**r:

```
mongos –configdb ConfServer: 27019
```

Step 4: Connecting to Mongos Instance

A connection can be formed **to the Mongos Instance** by running the following command from the Mongo Shell:

```
mongo –host ConfServer –port 27017
```

Step 5: Adding Servers to Clusters

All Servers that have to be included in the Cluster can be added by the following command:

```
sh.addShard("SA:27017")
```

"SA" here has to be replaced with the name of your Server that has to be added to the Cluster. This command can be executed for all Servers that have to be added to the Cluster.

Step 6: Setting up Replica sets for Shard Servers

Convert the shard instances into **replicas**. To set up replica sets, run the following command.

```
sh.addShardToZone("shardInstance", "replicaSetName")
```

Step 7: Initialize mongos and add shards to cluster

Whatever shards you have created so far are running currently but not a part of the Sharded cluster. To include them into sharded cluster you will need **mongos query**. Follow the given command to add shards to cluster.

```
mongos --configdb <configdb_connection_string>
sh.addShard("<shard1_connection_string>")
sh.addShard("<shard2_connection_string>")
# Repeat for additional shards if needed
```

Step 8: Enabling Sharding for Database

Once the Sharded Cluster has been set up, Sharding for the required **database** has to be enabled. This can be done by the following command:

```
sh.enableSharding(db_test)
```

In the above command, "db_test" has to be replaced with the name of the database that you wish to Shard. This completes the MongoDB sharding tutorial to help set up MongoDB sharding.

Step 9: Evaluate the Shard Usage

Sharding is implemented to enhance the **scalability of a database system**, and its effectiveness is maximized when efficiently **supporting database queries**. If a significant portion of your queries requires **scanning** every shard in the cluster for execution, the advantages of sharding may be compromised by the increased **complexity** of the system. This step assesses whether a query is optimized and utilizes a single shard or if it spans multiple shards to fetch results.

RESULT

The Indexing and Sharding using MongoDB is Executed Successfully

Ex.No:8

Date:

XML DATABASE AND TABLE CREATIONS

AIM: To Perform SQL queries to create XML DB and Table

PROGRAM:

Step 1: Creating a table that can store XML data

```
CREATE SCHEMA POSAMPLE;

SET CURRENT SCHEMA POSAMPLE;

CREATE TABLE Customer (Cid BIGINT NOT NULL PRIMARY KEY, Info XML);
```

Step 2: Insert three XML documents into the Customer table

```
INSERT INTO Customer (Cid, Info) VALUES (1000,
'<customerinfo xmlns="http://posample.org" Cid="1000">
  <name>Kathy Smith</name>
  <addr country="Canada">
    <street>5 Rosewood</street>
    <city>Toronto</city>
    <prov-state>Ontario</prov-state>
    <pcode-zip>M6W 1E6</pcode-zip>
  </addr>
  <phone type="work">416-555-1358</phone>
</customerinfo>');
```

```
INSERT INTO Customer (Cid, Info) VALUES (1002,
'<customerinfo xmlns="http://posample.org" Cid="1002">
  <name>Jim Noodle</name>
  <addr country="Canada">
    <street>25 EastCreek</street>
    <city>Markham</city>
    <prov-state>Ontario</prov-state>
    <pcode-zip>N9C 3T6</pcode-zip>
  </addr>
  <phone type="work">905-555-7258</phone>
</customerinfo>');
```

```
INSERT INTO Customer (Cid, Info) VALUES (1003,
'<customerinfo xmlns="http://posample.org" Cid="1003">
  <name>Robert Shoemaker</name>
  <addr country="Canada">
    <street>1596 Baseline</street>
    <city>Aurora</city>
    <prov-state>Ontario</prov-state>
    <pcode-zip>N8X 7F8</pcode-zip>
  </addr>
  <phone type="work">905-555-2937</phone>
</customerinfo>');
```

Step 3: You can confirm that the records were successfully inserted as follows:

```
SELECT * from Customer;
```

Result:

Thus the above program executed successfully

Ex.No:9

Date:

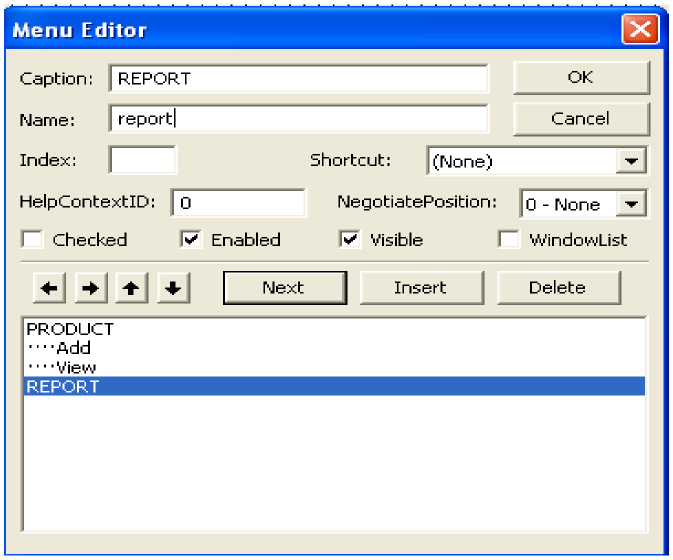
DESIGN AND IMPLEMENTATION OF PRODUCT MANAGEMENT

AIM:

To create a program to working with forms, menus and reports in product management exercise using Visual Basic 2000.

PROGRAM:

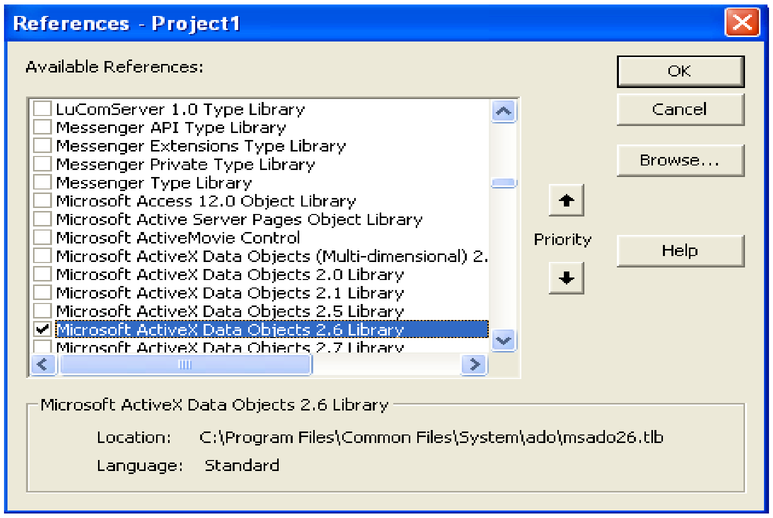
1. Create a Database product with the field's pname, pno, amt, quan.
2. Create a menu product with sub menu view and add, Create another menu report using menu editor.



Caption: REPORT
Name: report
Index:
Shortcut: (None)
HelpContextID: 0
NegotiatePosition: 0 - None
[ ] Checked [✓ Enabled] [✓ Visible] [ ] WindowList

```
PRODUCT
····Add
····View
REPORT
```

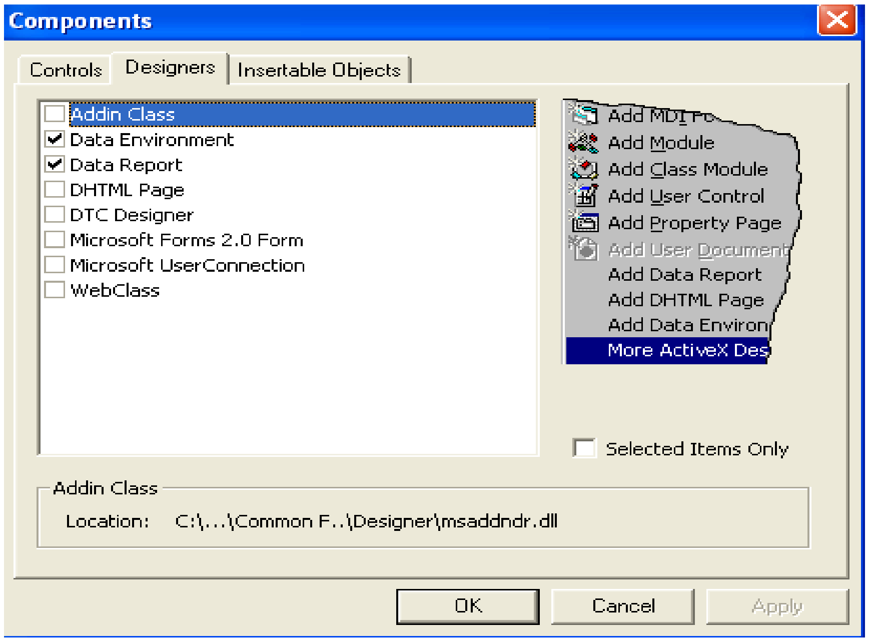
4. Set Reference->Microsoft activex data objects 2.6 library.



```
[ ] LuComServer 1.0 Type Library
[ ] Messenger API Type Library
[ ] Messenger Extensions Type Library
[ ] Messenger Private Type Library
[ ] Messenger Type Library
[ ] Microsoft Access 12.0 Object Library
[ ] Microsoft Active Server Pages Object Library
[ ] Microsoft ActiveMovie Control
[ ] Microsoft ActiveX Data Objects (Multi-dimensional) 2.
[ ] Microsoft ActiveX Data Objects 2.0 Library
[ ] Microsoft ActiveX Data Objects 2.1 Library
[ ] Microsoft ActiveX Data Objects 2.5 Library
[✓] Microsoft ActiveX Data Objects 2.6 Library
[ ] Microsoft ActiveX Data Objects 2.7 Library
```

Microsoft ActiveX Data Objects 2.6 Library
Location: C:\Program Files\Common Files\System\ado\msado26.tlb
Language: Standard

4. Set Components->Designer tab->Data environment, Data report.


Components

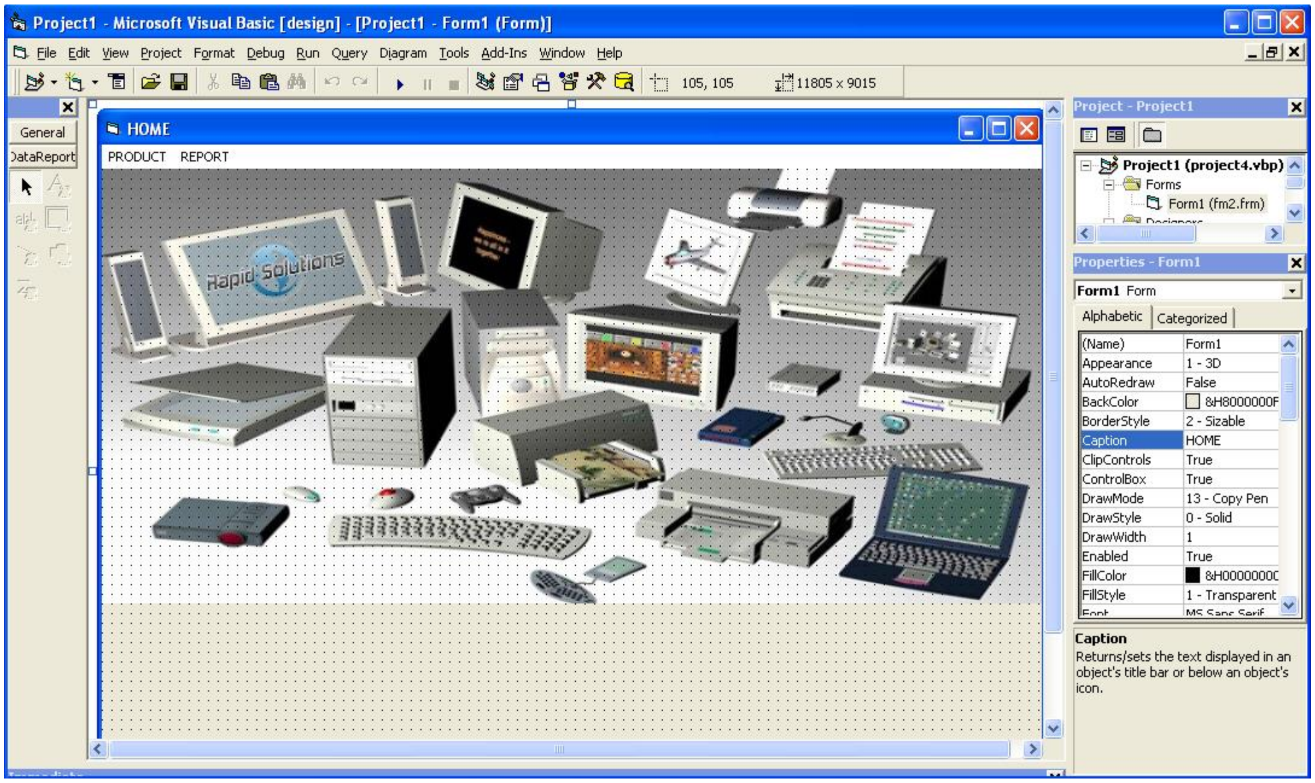
Controls | Designers | Insertable Objects

- [ ] Addin Class
- [✓] Data Environment
- [✓] Data Report
- [ ] DHTML Page
- [ ] DTC Designer
- [ ] Microsoft Forms 2.0 Form
- [ ] Microsoft UserConnection
- [ ] WebClass

[ ] Selected Items Only

Addin Class
Location: C:\...\Common F..\Designer\msaddndr.dll

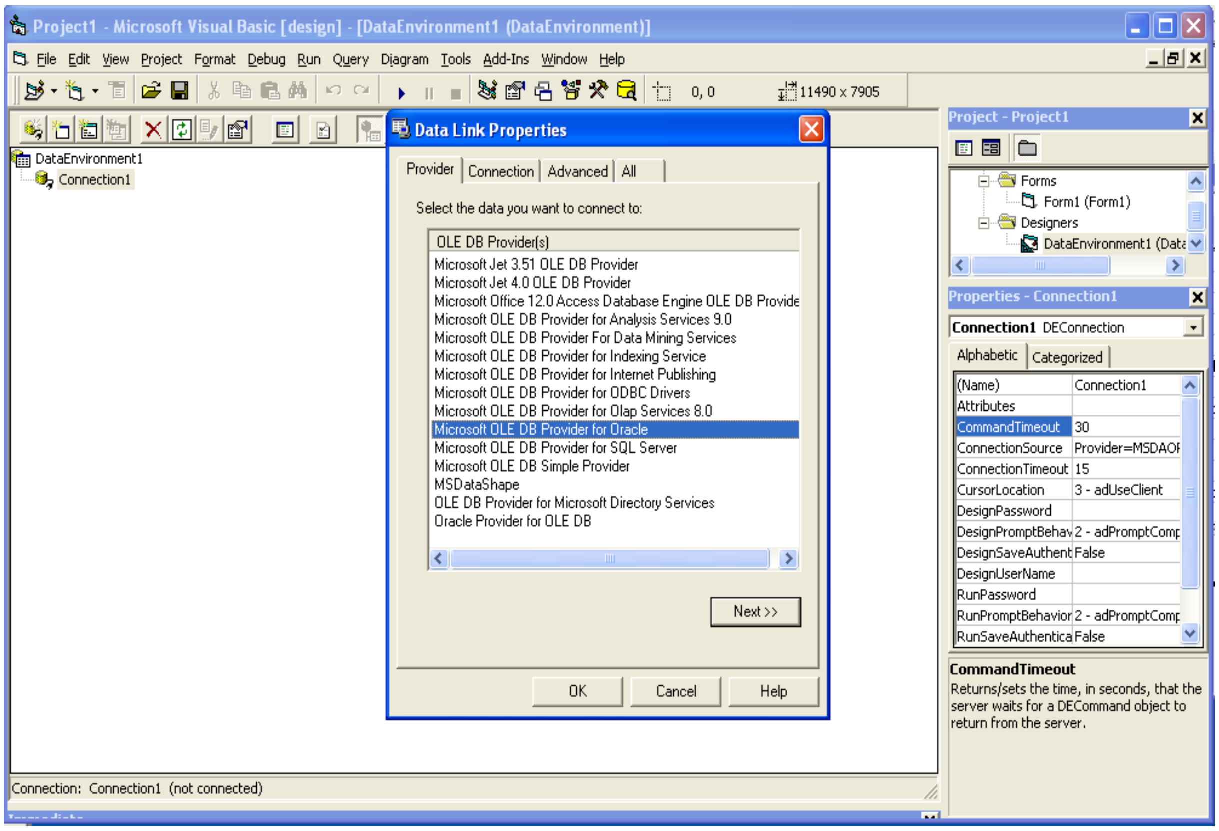
5. After creating menu the form will be:


Project1 - Microsoft Visual Basic [design] - [Project1 - Form1 (Form)]

(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	☐ &H8000000F
BorderStyle	2 - Sizable
Caption	HOME
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	■ &H0000000C
FillStyle	1 - Transparent

Caption
Returns/sets the text displayed in an object's title bar or below an object's icon.

6. Right click the data environment->properties->microsoft OLEDB provider for oracle->click next.



Select the data you want to connect to:

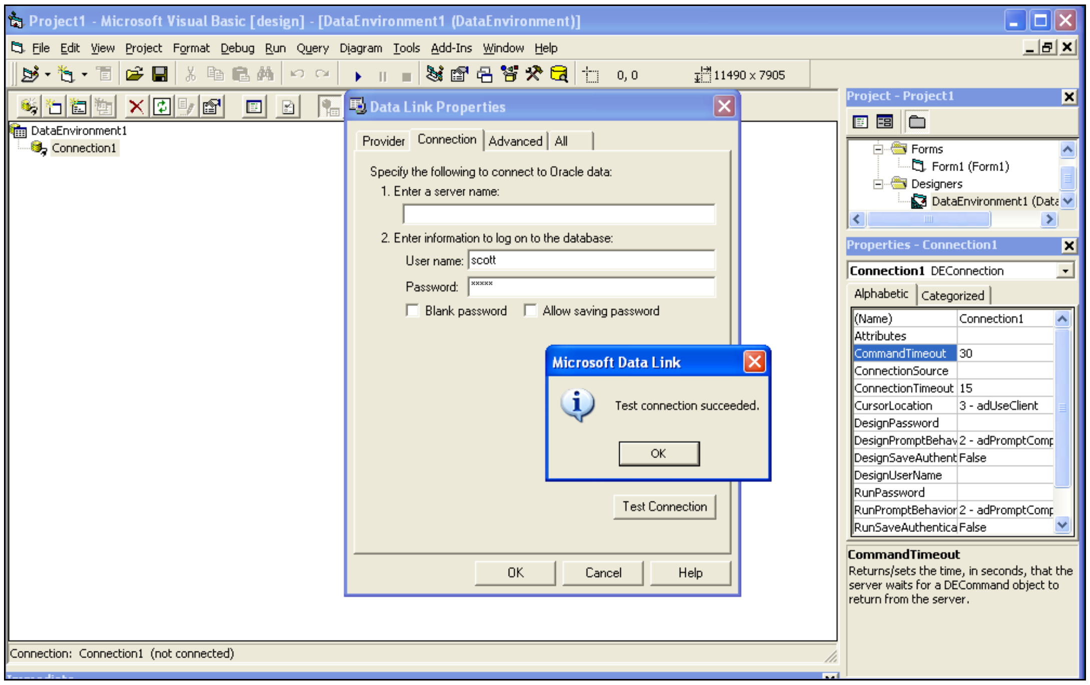
OLE DB Provider(s)
Microsoft Jet 3.51 OLE DB Provider
Microsoft Jet 4.0 OLE DB Provider
Microsoft Office 12.0 Access Database Engine OLE DB Provide
Microsoft OLE DB Provider for Analysis Services 9.0
Microsoft OLE DB Provider For Data Mining Services
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for Internet Publishing
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for Olap Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape
OLE DB Provider for Microsoft Directory Services
Oracle Provider for OLE DB

(Name)	Connection1
Attributes	
CommandTimeout	30
ConnectionSource	Provider=MSDAOR
ConnectionTimeout	15
CursorLocation	3 - adUseClient
DesignPassword	
DesignPromptBehav	2 - adPromptComp
DesignSaveAuthent	False
DesignUserName	
RunPassword	
RunPromptBehavior	2 - adPromptComp
RunSaveAuthentica	False

CommandTimeout
Returns/sets the time, in seconds, that the server waits for a DECommand object to return from the server.

Connection: Connection1 (not connected)

7.connection->oracle uname->pwd->click test connection->ok.


Project1 - Microsoft Visual Basic [design] - [DataEnvironment1 (DataEnvironment)]

DataEnvironment1
Connection1

Data Link Properties

Specify the following to connect to Oracle data:
1. Enter a server name:
2. Enter information to log on to the database:
User name: scott
Password: *****
Blank password   Allow saving password

Microsoft Data Link
Test connection succeeded.

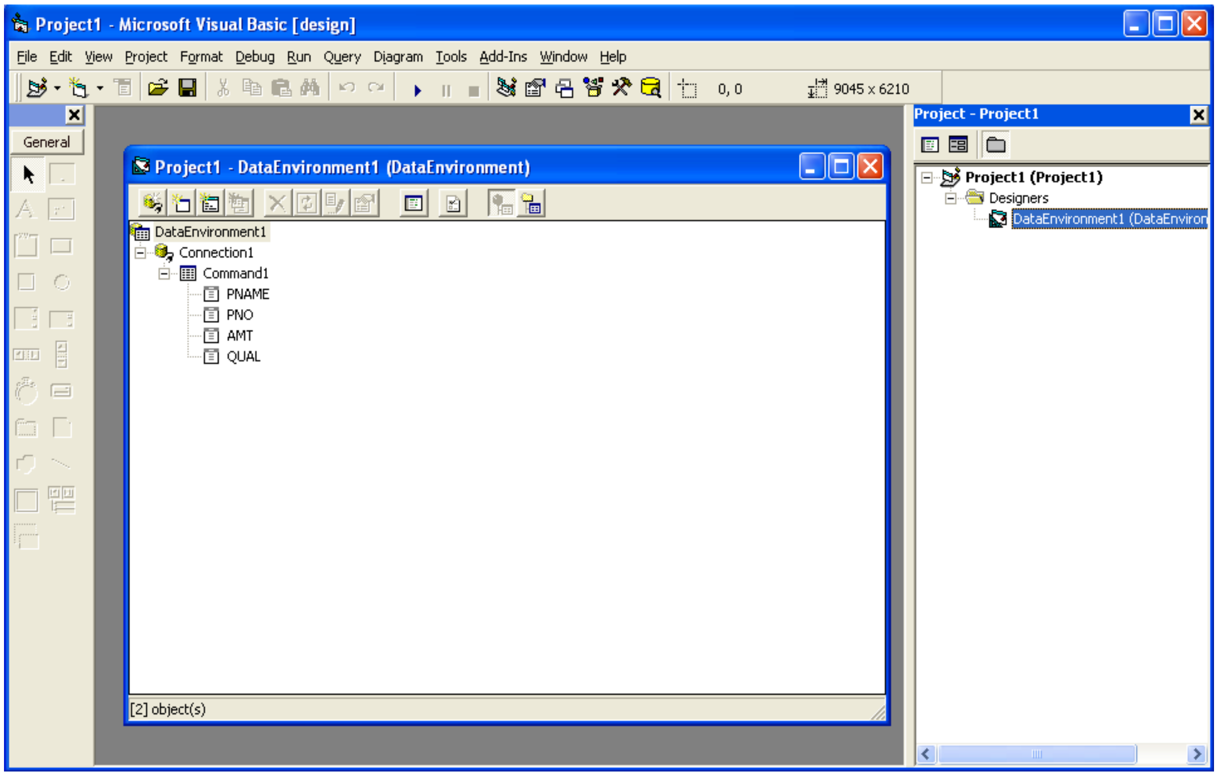
Properties - Connection1
Connection1 DEConnection

(Name)	Connection1
Attributes	
CommandTimeout	30
ConnectionSource	
ConnectionTimeout	15
CursorLocation	3 - adUseClient
DesignPassword	
DesignPromptBehav	2 - adPromptComp
DesignSaveAuthent	False
DesignUserName	
RunPassword	
RunPromptBehavior	2 - adPromptComp
RunSaveAuthentica	False

CommandTimeout
Returns/sets the time, in seconds, that the server waits for a DECommand object to return from the server.

Connection: Connection1 (not connected)

8. Right click the connection->add command.


Project1 - Microsoft Visual Basic [design]

Project1 - DataEnvironment1 (DataEnvironment)

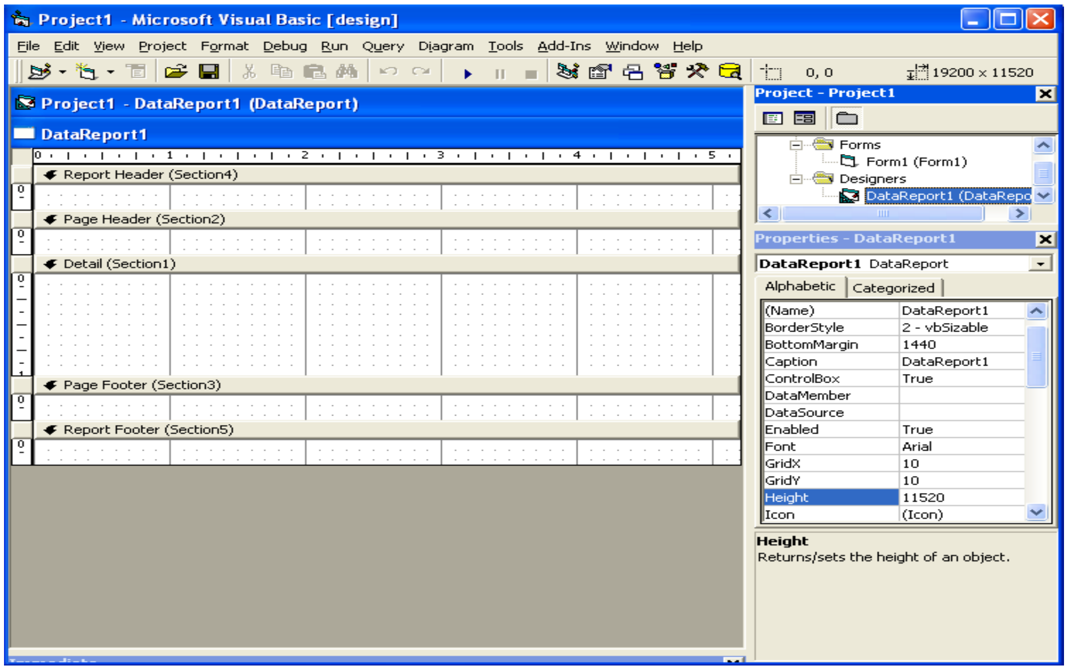
```
DataEnvironment1
  Connection1
    Command1
      PNAME
      PNO
      AMT
      QUAL
```

[2] object(s)

Project - Project1
Project1 (Project1)
Designers
DataEnvironment1 (DataEnviron

9.In project window right click ->add->Data report.Set the

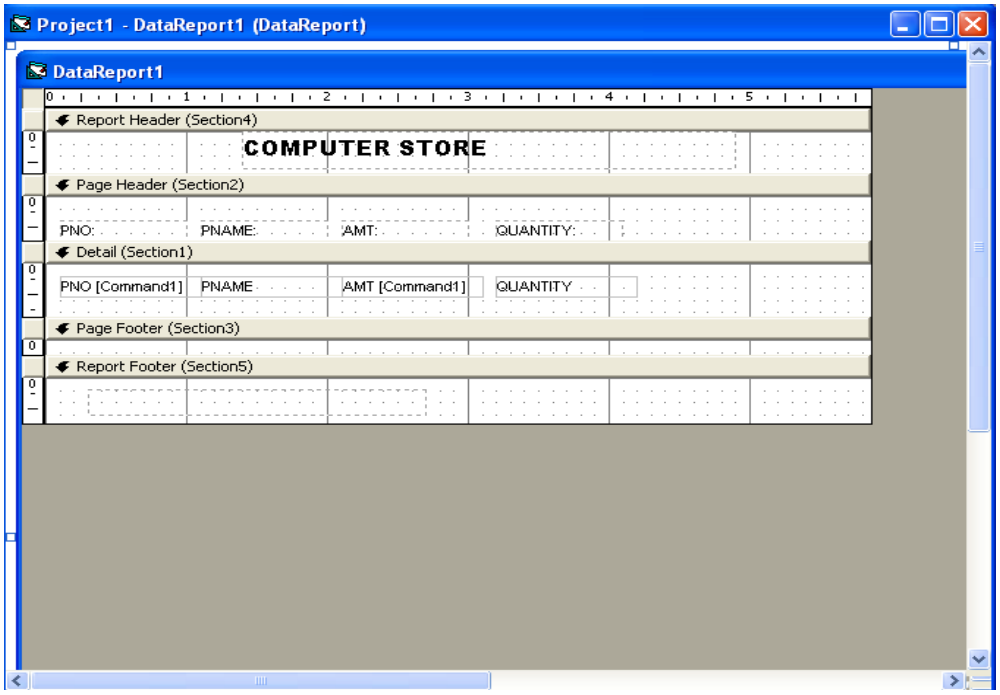
Datasource & Data member in the properties window of Report



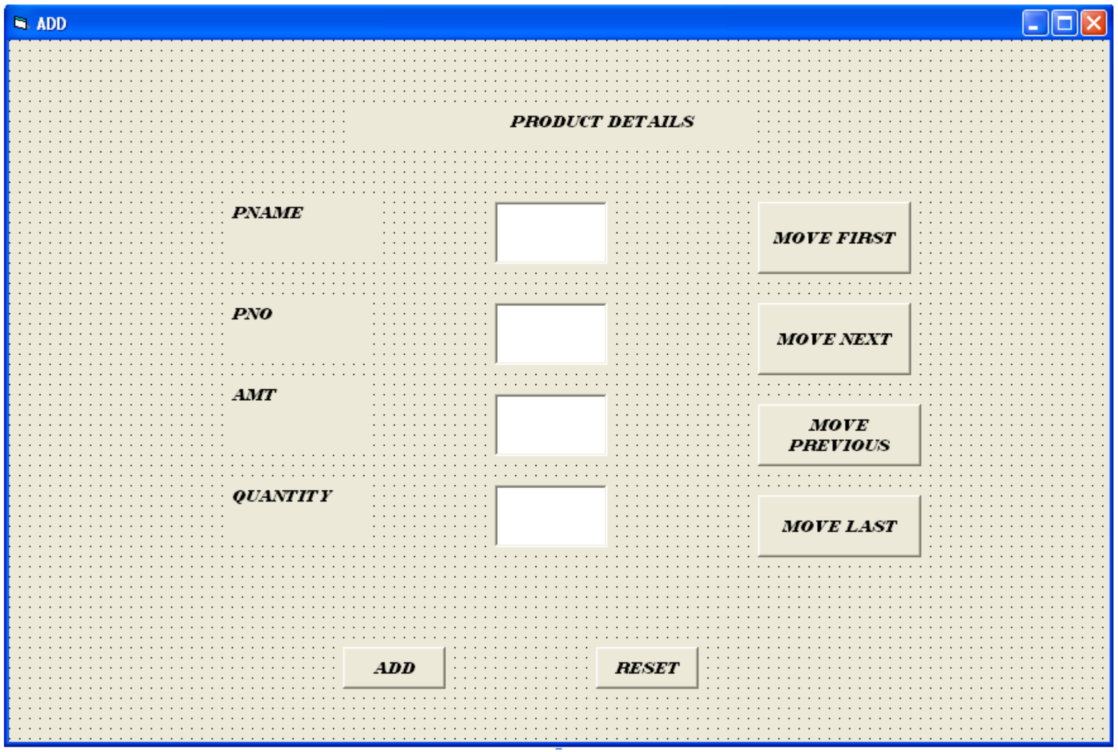
(Name)	DataReport1
BorderStyle	2 - vbSizable
BottomMargin	1440
Caption	DataReport1
ControlBox	True
DataMember	
DataSource	
Enabled	True
Font	Arial
GridX	10
GridY	10
Height	11520
Icon	(Icon)

Height
Returns/sets the height of an object.

10. In command drag the product fields and paste it in the data report window.



11.create a form design for product details:



CODING FOR ADD PRODUCT FORM:

```
Public cs As New ADODB.Connection

Public rs As New ADODB.Recordset
```

Coding for Add command:

```
Private Sub ADD_Click()

rs.AddNew

rs.Fields(0) = Text1.Text

rs.Fields(1) = Text2.Text

rs.Fields(2) = Text3.Text

rs.Fields(3) = Text4.Text

rs.Update
```

```
MsgBox "inserted successfully!"

End Sub
```

Coding for Movelast command:

```
Private Sub Command4_Click()

rs.MoveLast

If rs.EOF Then

    MsgBox "this is the last record"

Else

    Text1.Text = rs.Fields(0)

    Text2.Text = rs.Fields(1)

    Text3.Text = rs.Fields(2)

End If

End Sub
```

Coding for Form_Load:

```
Private Sub Form_Load()

cs.Open "Aarthi", "scott", "tiger"

rs.Open "select * from product", cs, adOpenDynamic,

adLockPessimistic

End Sub
```

Coding for Movefirst command:

```
Private Sub MOVEFIRST_Click()

rs.MOVEFIRST

Text1.Text = rs.Fields(0)

Text2.Text = rs.Fields(1)

Text3.Text = rs.Fields(2)

Text4.Text = rs.Fields(3)
```

```
End Sub
```

Coding for Movenext command:

```
Private Sub MOVENEXT_Click()

rs.MOVENEXT

If rs.EOF = True Then

MsgBox " First record"

Else

Text1.Text = rs.Fields(0)

Text2.Text = rs.Fields(1)

Text3.Text = rs.Fields(2)

End If

End Sub
```

Coding for Moveprevious command:

```
Private Sub MOVEPREVIOUS_Click()

rs.MOVEPREVIOUS

If rs.BOF = True Then

MsgBox " First record"

Else

Text1.Text = rs.Fields(0)

Text2.Text = rs.Fields(1)

Text3.Text = rs.Fields(2)

End If

End Sub

Private Sub RESET_Click()

Text1.Text = ""

Text2.Text = ""
```

```
Text3.Text = ""

Text4.Text = ""

End SubProduct:
```

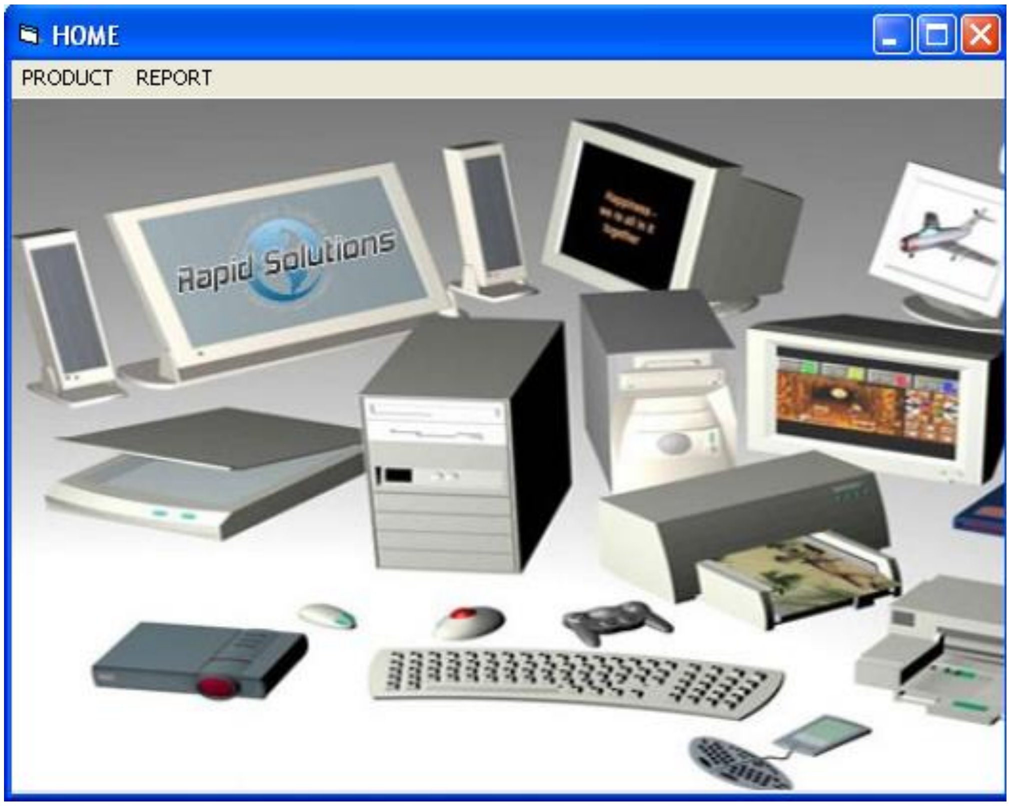
CODING FOR HOME PAGE:

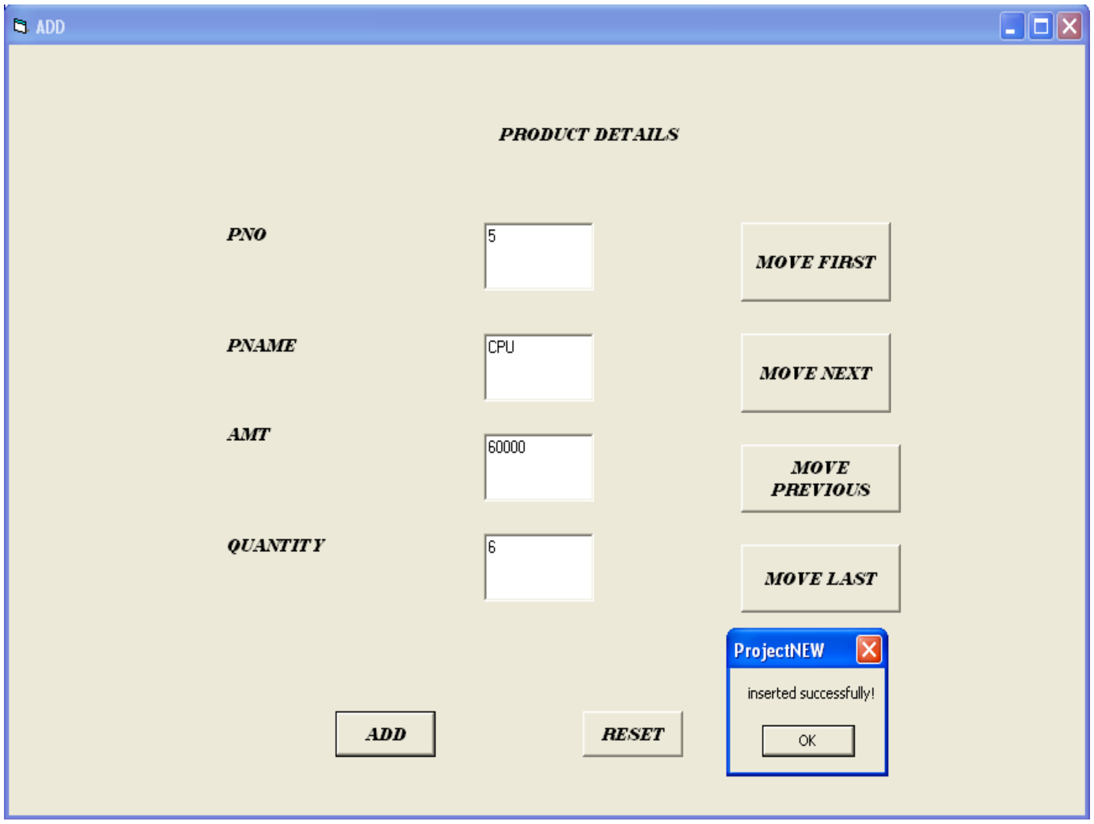
```
Private Sub ad_Click()

Form1.Hide

ADD.Show

End Sub
```

CODING FOR REPORT MENU:

```
Private Sub repo_Click()

DataReport1.Show

End Sub
```

OUTPUT:


HOME
PRODUCT
REPORT
Rapid Solutions


ADD

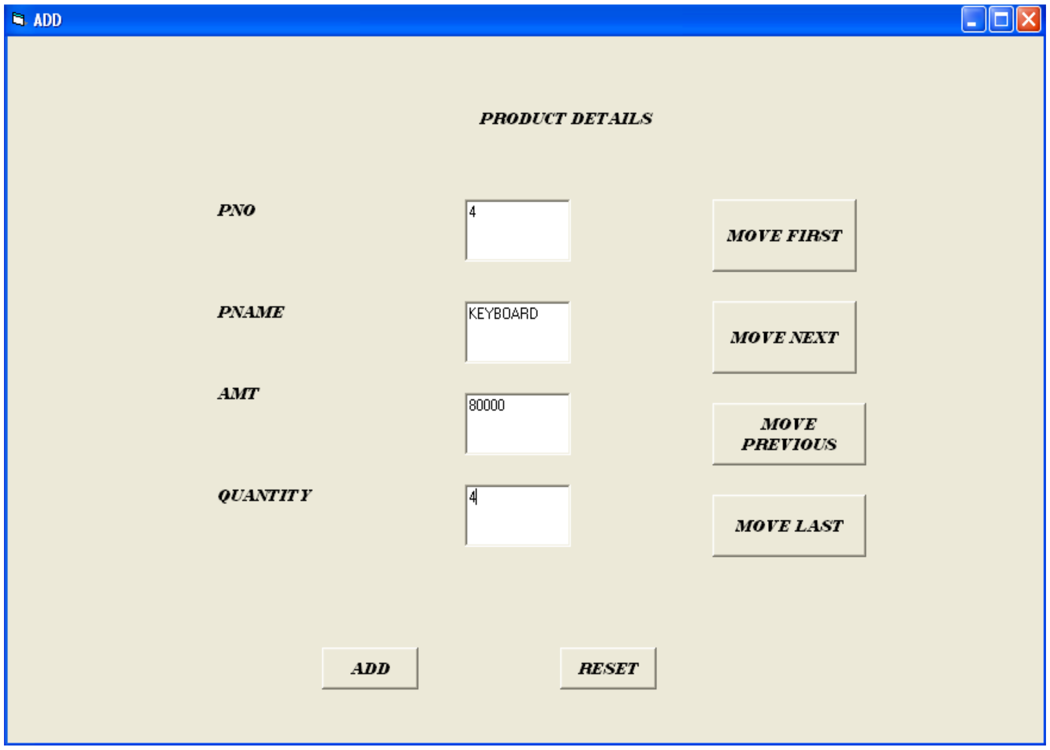
PRODUCT DETAILS

PNO	5
PNAME	CPU
AMT	60000
QUANTITY	6

ProjectNEW

inserted successfully!

View:



ADD

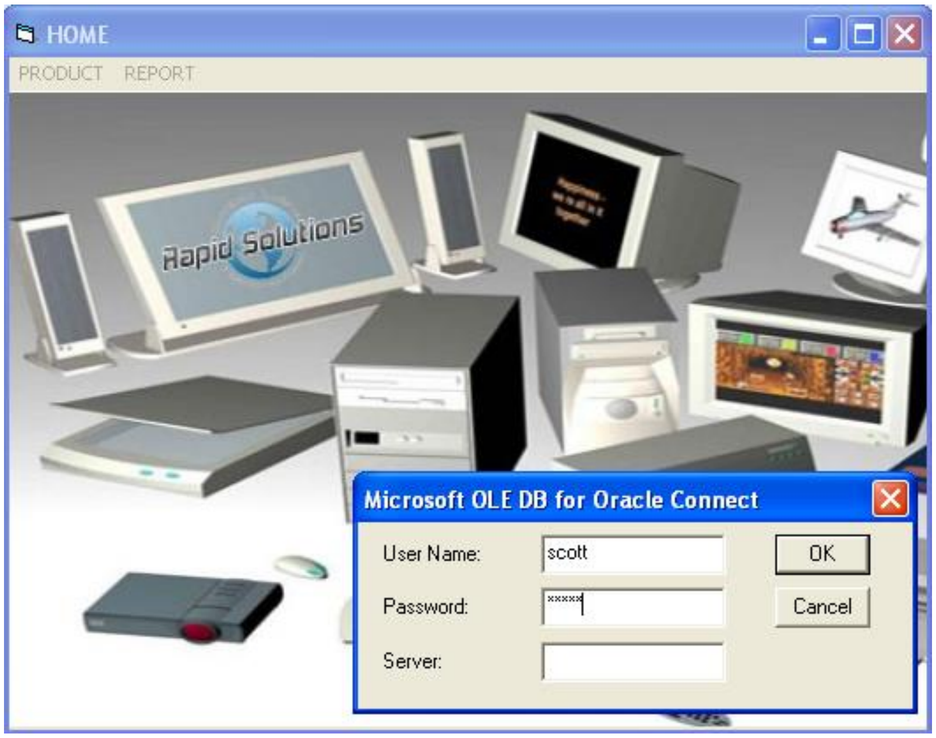
PRODUCT DETAILS

PNO 4

PNAME KEYBOARD

AMT 80000

QUANTITY 4



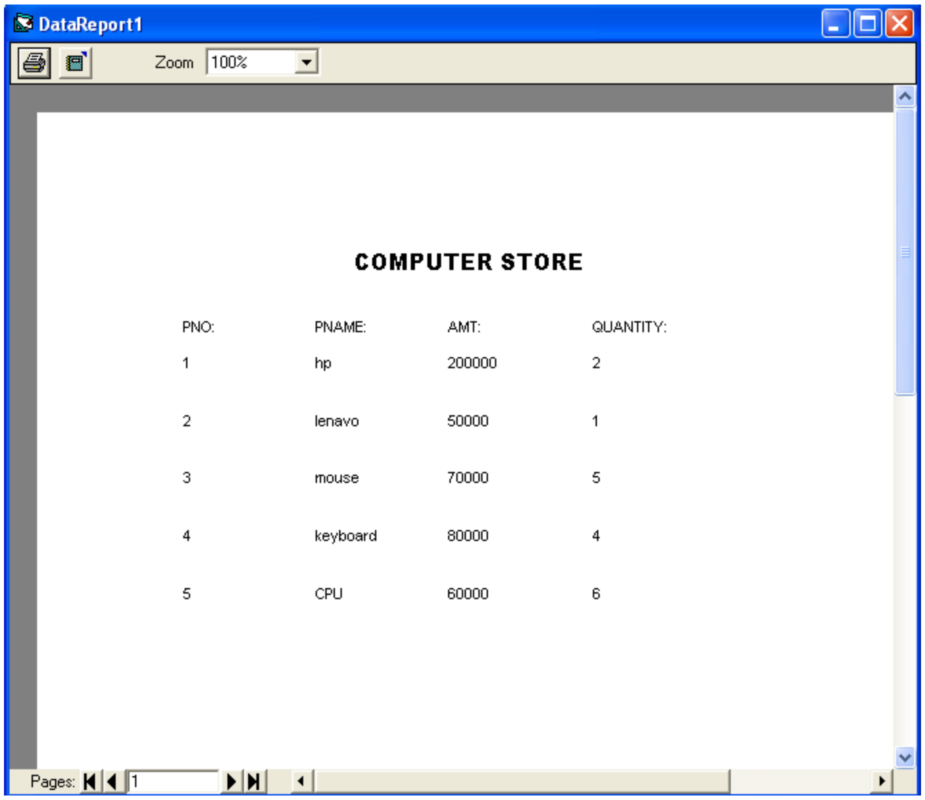
HOME

Microsoft OLE DB for Oracle Connect

User Name: scott

Password: *****

Server:



COMPUTER STORE

```
PNO:        PNAME:      AMT:        QUANTITY:
1           hp          200000      2

2           lenavo      50000       1

3           mouse       70000       5

4           keyboard    80000       4

5           CPU         60000       6
```

Result:

Thus the above program form, menu and reports using visual Basic was created and executed successfully.

Ex.No:10	DESIGN AND IMPLEMENTATION OF ACCOUNT MANAGEMENT

AIM

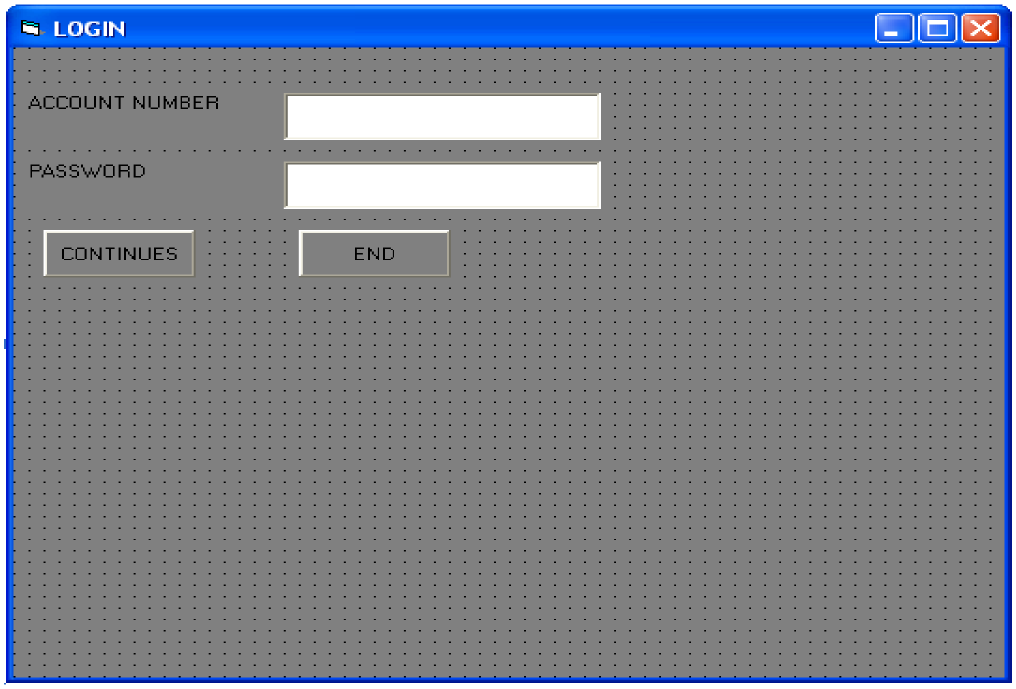
To create account management exercise for connecting oracle using visual basic 2000.

ALGORITHM

1. Create table account with the field's acc_no, pin_no, Name, Bal.

```
ACC_NO     PIN_NO        NAME           BAL
--------- ---------- ---------- ----------
123       456          priya          29900
345       678          sangi          45000
567       890          aarthi         46000
```

2. Design a login form using visual basic, add label box & text box for acc_no & pin_no, add command button for continue & end.



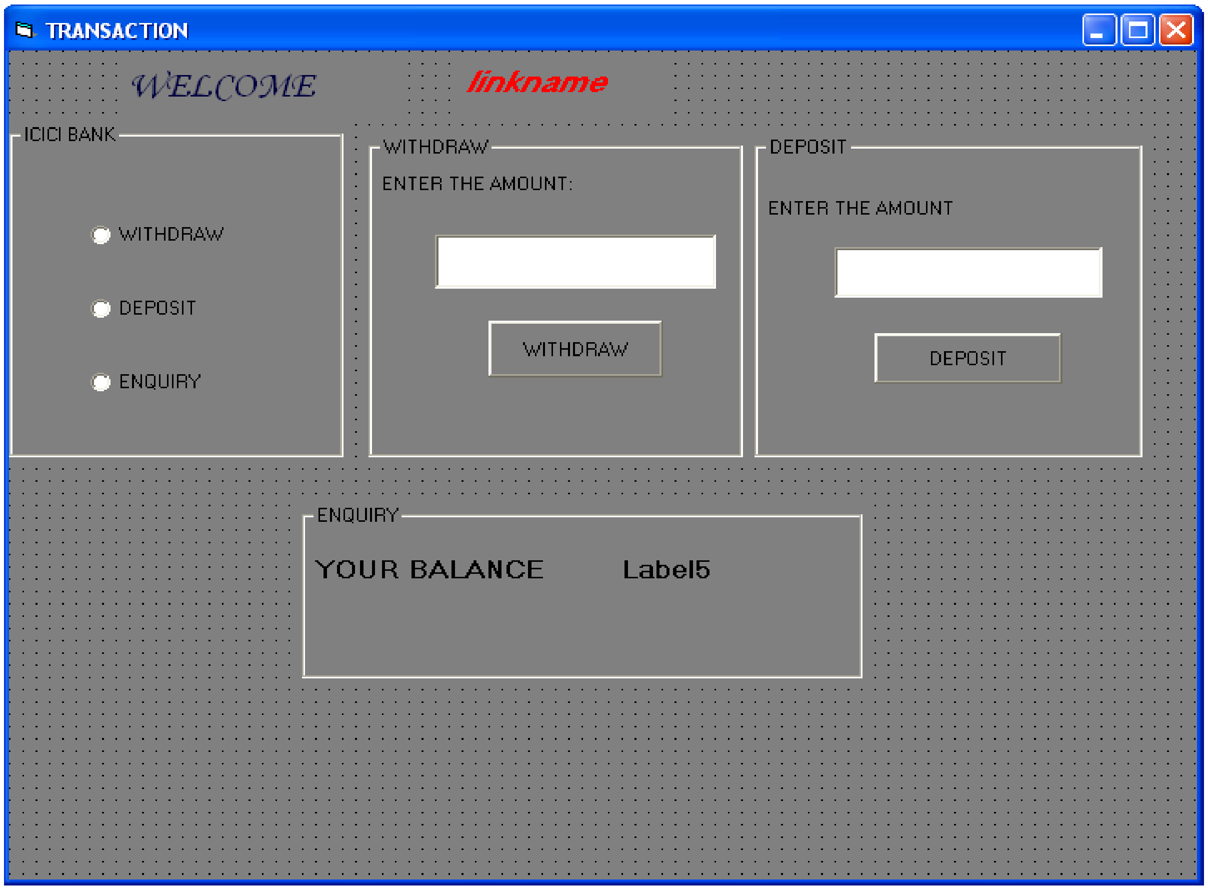
FORM1:

LOGIN CODING:

```
Public cs As New ADODB.Connection

Public rs As New ADODB.Recordset

Private Sub Command1_Click()

Dim flag As Integer

flag = 0

rs.MoveFirst

While (rs.EOF = False)

   If Val(Text1) = rs.Fields(0) And Val(Text2) = rs.Fields(1) Then

     MsgBox "WELCOME ICICI BANK", vbOKOnly, "ICICI"

     flag = 1

     Form2.Show

     Form1.Hide

     Form2.linkname.Caption = rs.Fields(2)

  End If

 rs.MoveNext

Wend

If flag = 0 Then

  MsgBox "invalid paid", vbCritical, "ICICI"

End If

End Sub

Private Sub Form_Load()

cs.Open "krish1", "scott", "tiger"
```

```
rs.Open "select * from bank", cs, adOpenDynamic, adLockPessimistic

End Sub
```

3. create another form transaction with frame name icici bank in that option buttons of withdraw, deposit & enquiry.
   - Create 3 frames for withdraw, deposit & enquiry in the same transaction form.



FORM2:

TRANSACTION:

```
Public cs As New ADODB.Connection

Public rs As New ADODB.Recordset
```

Coding for withdraw command:

```
Private Sub Command1_Click()

rs.MoveFirst
```

```
a = Val(Text1)

While (rs.EOF = False)

  If Form1.Text1.Text = rs.Fields(0) And Form1.Text2.Text =

    rs.Fields(1) Then

    If (rs.Fields(3) > Val(a)) Then

     rs.Fields(3) = rs.Fields(3) - a

     rs.Update

     MsgBox "withdraw is successful", vbInformation, "ICICI"

     MsgBox "your current balance is" & rs.Fields(3), vbInformation

    Else

MsgBox "your account balance is below" & rs.Fields(3), vbInformation

End If

End If

rs.MoveNext

Wend

End Sub
```

Coding for deposit command:

```
Private Sub Command2_Click()

rs.MoveFirst

a = Val(Text2)

While (rs.EOF = False)

If Form1.Text1.Text = rs.Fields(0) And Form1.Text2.Text =
rs.Fields(1) Then

     rs.Fields(3) = rs.Fields(3) + a

     rs.Update

     MsgBox "deposit is successful", vbInformation, "ICICI"
     MsgBox "your current balance is" & rs.Fields(3), vbInformation
End If
rs.MoveNext
Wend
End Sub
Private Sub Form_Load()
frame2.Visible = False
Frame3.Visible = False
Frame4.Visible = False
cs.Open "krish1", "scott", "tiger"
rs.Open "select * from bank", cs, adOpenDynamic, adLockPessimistic
End Sub
```

Coding for withdraw option:

```
Private Sub Option1_Click()
frame2.Visible = True
Frame3.Visible = False
Frame4.Visible = False
End Sub
```

Coding for deposit option:

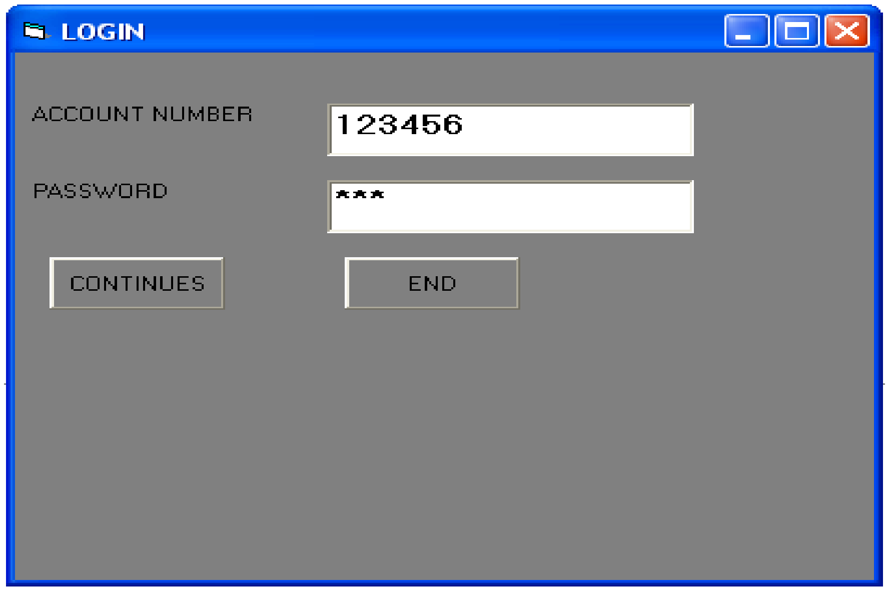
```
Private Sub Option2_Click()
Frame3.Visible = True
frame2.Visible = False
Frame4.Visible = False
End Sub
```

Coding for enquiry option:

```
Private Sub Option3_Click()
```

```
Frame4.Visible = True

frame2.Visible = False

Frame3.Visible = False

rs.MoveFirst

While (rs.EOF = False)

If Form1.Text1.Text = rs.Fields(0) And Form1.Text2.Text =
rs.Fields(1) Then

Label5.Caption = rs.Fields(3)

End If

rs.MoveNext

Wend

End Sub
```

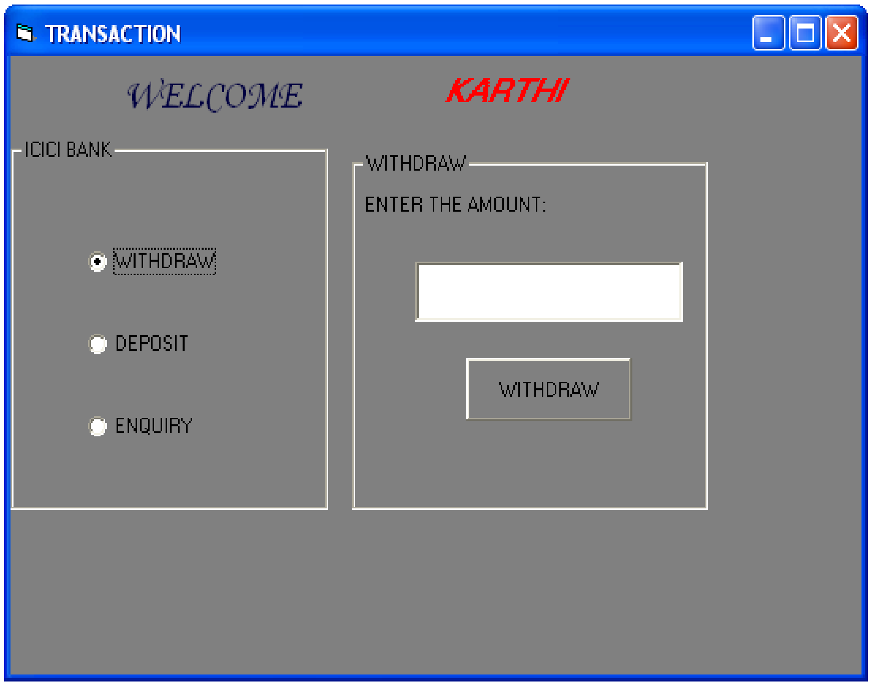
OUTPUT

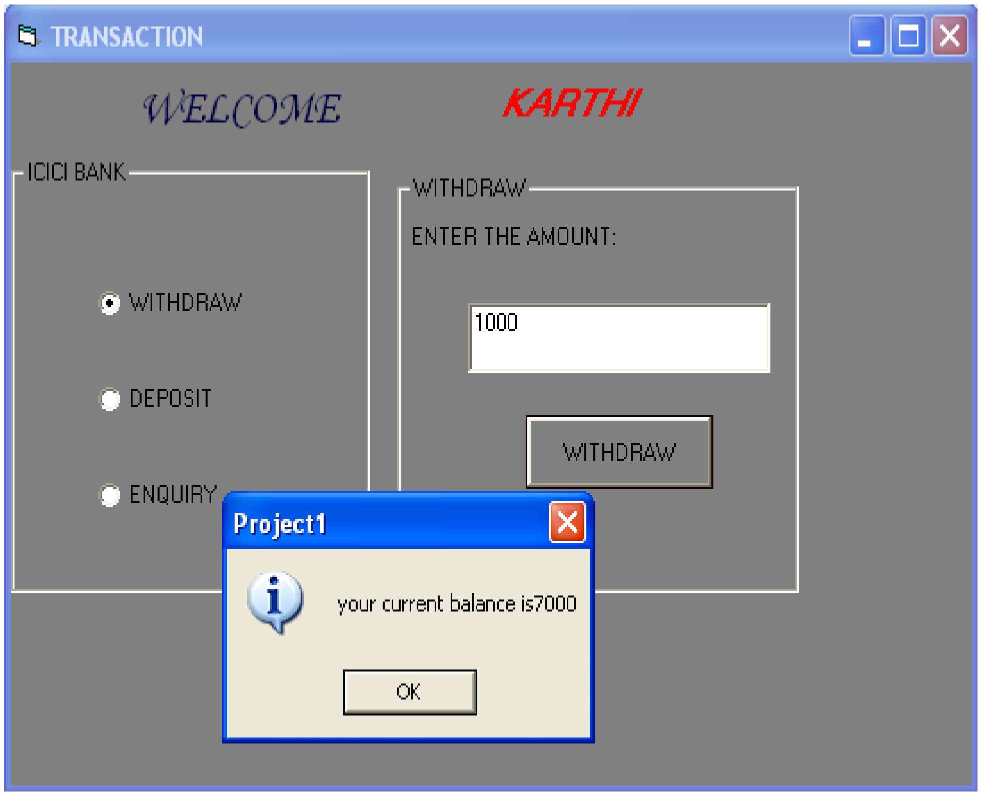


LOGIN

ACCOUNT NUMBER 123456

PASSWORD ***

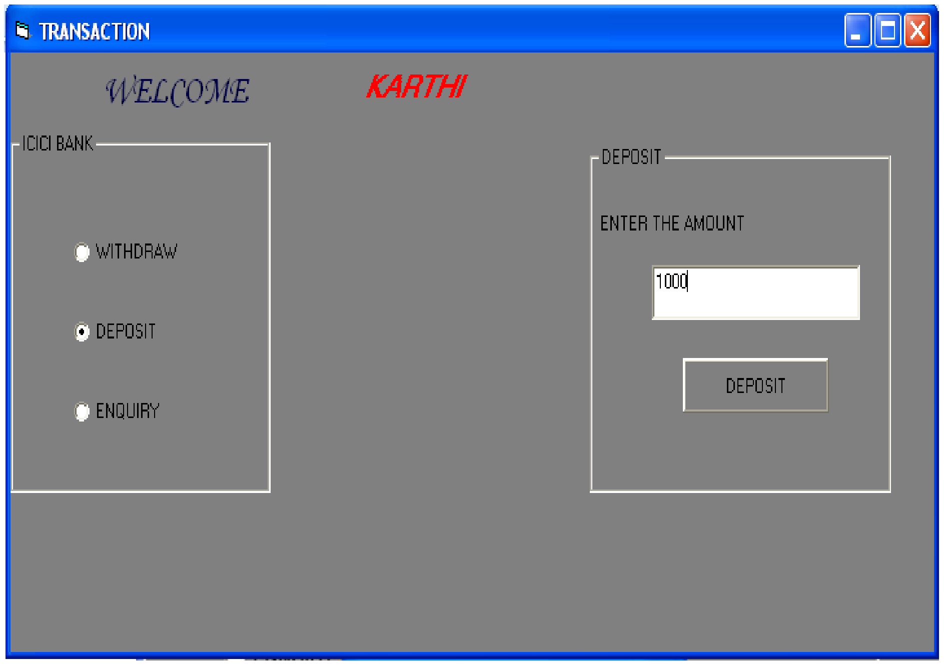

TRANSACTION
WELCOME
KARTHI
ICICI BANK
WITHDRAW
DEPOSIT
ENQUIRY
WITHDRAW
ENTER THE AMOUNT:
WITHDRAW



● WITHDRAW

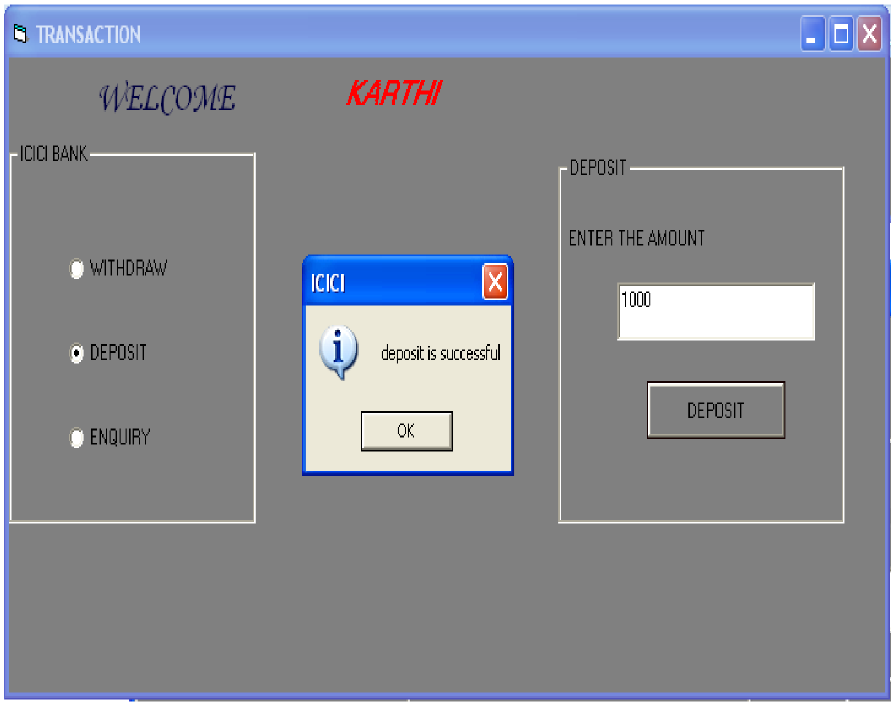
ENTER THE AMOUNT: 1000

your current balance is7000



● DEPOSIT

ENTER THE AMOUNT 1000



TRANSACTION

WELCOME KARTHI

ICICI BANK

○ WITHDRAW

● DEPOSIT

○ ENQUIRY

DEPOSIT

ENTER THE AMOUNT

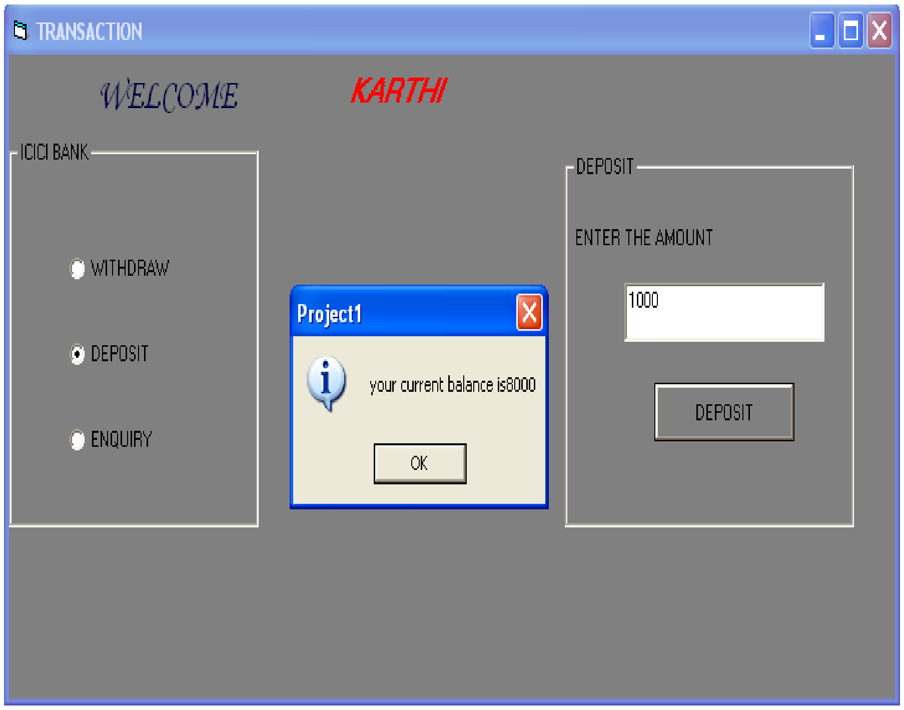
1000

DEPOSIT

ICICI

deposit is successful

OK



TRANSACTION

WELCOME KARTHI

ICICI BANK

○ WITHDRAW

● DEPOSIT

○ ENQUIRY

DEPOSIT

ENTER THE AMOUNT

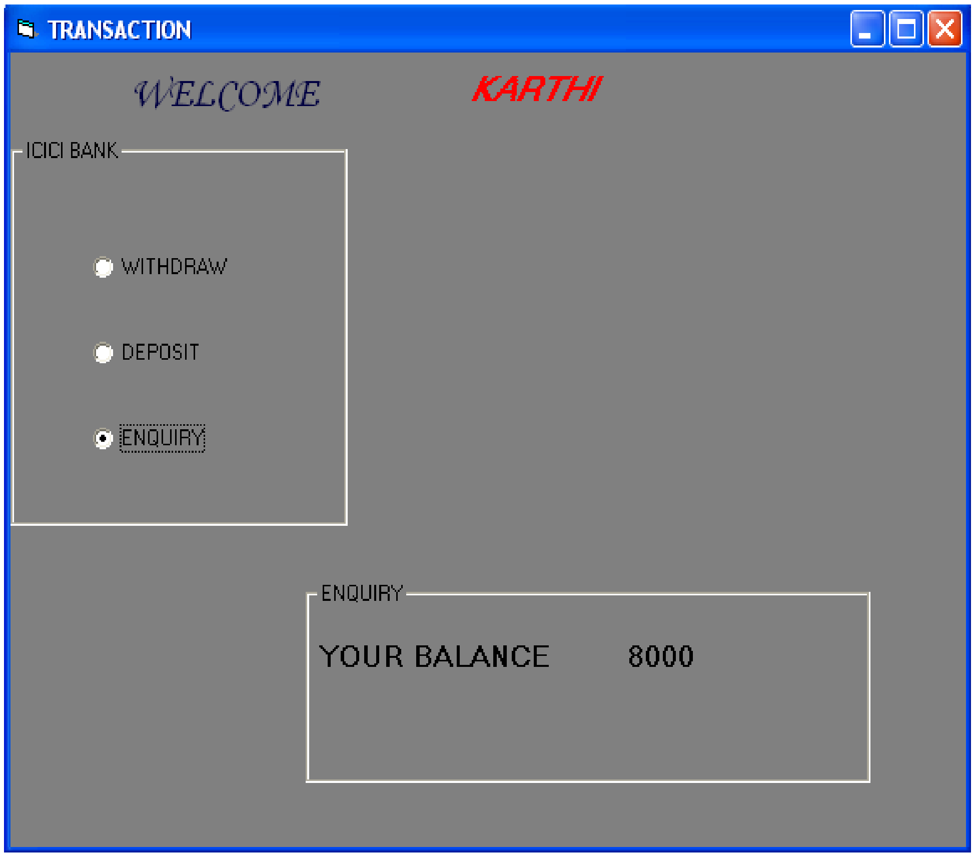
1000

DEPOSIT

Project1

your current balance is8000

OK



WELCOME KARTHI

ICICI BANK

- ○ WITHDRAW
- ○ DEPOSIT
- ● ENQUIRY

ENQUIRY

YOUR BALANCE 8000

RESULT

Thus the program using front end tool using visual basic was executed successfully.

Vision and Mission of the Institution

Vision

To shape the institution into a globally renowned centre for

Mission

education and research in engineering and technology. We aim to foster pragmatic, ingenious ideas that would help in the advancementof the individual and the society.

Mission

To provide wisdom through educational and job oriented training specially targeting young minds pursuing engineering and technology programmes thereby making them intellectually bright, critically intense and creatively powerful.