

Ex-No: 1

STUDENT MARKSHEET PROCESSING

Aim

Algorithm

Program code:

Query 01: Create a student marksheet table

```
SQL> create table student_marksheet(  
2  sno number(3),  
3  regno varchar2(7),  
4  sname varchar2(20),  
5  mark1 number(3),  
6  mark2 number(3),  
7  mark3 number(3),  
8  total number(3),  
9  average number (5,2));
```

Table created.

Query 02: Insert minimum 5 records in the student marksheet table

```
SQL> insert into student_marksheet  
values(&sno, '&regno', '&sname', &mark1, &mark2, &mark3, &total, &ave  
rage);
```

Enter value for sno: 1

Enter value for regno: U06CS093

Enter value for sname: Sathya pradeep.p

Enter value for mark1: 88

Enter value for mark2: 89

Enter value for mark3: 99

Enter value for total: 0

Enter value for average: 0

```
old 1: insert into student_marksheet  
values(&sno, '&regno', '&sname', &mark1, &mark2, &mark3, &total, &av
```

```
new 1: insert into student_marksheet  
values(1, 'U06CS093', 'Sathya pradeep.p', 88, 89, 99, 0, 0)
```

1 row created.

Query 03: Display all the records in the student marksheet table

```
SQL> select * from student_marksheet;
```

SNO	REGNO	SNAME	DBMS	DSP	OS	TOTAL	AVERAGE
1	U06CS093	Sathya pradeep.p	88	89	99	0	0
2	U06CS079	Ravindar	79	89	88	0	0
3	U06CS092	Sathish kumar.R	89	88	95	0	0
4	U06CS502	Vinod	98	89	95	0	0
5	U06CS078	Ram.R.V	85	78	88	0	0

Query 04: Calculate the total marks of each student

```
SQL> update student_marksheet set  
total=mark1+mark2+mark3;
```

5 rows updated.

Query 05: Calculate the average marks of each student

```
SQL> update student_marksheet set average=total/3;
```

5 rows updated.

Query 06: Calculate the sum of DBMS marks

```
SQL> select sum(DBMS) from student_marksheet;
```

SUM(DBMS)

439

Query 07: Find the total number of students

```
SQL> select count(sno) from student_marksheet;
```

COUNT(SNO)

5

Query 08: Find who has got maximum marks in DBMS

```
SQL> select sname,dbms from student_marksheet where dbms
in(select max(dbms) from student_marksheet);
```

```
SNAME          DBMS
-----
Vinod          98
```

Query 09: Find who has got minimum marks in OS

```
SQL> select sname,os from student_marksheet where os
in(select min(os) from student_marksheet);
```

```
SNAME          OS
-----
Ravindar       88
Ram.R.V        88
```

Query 10: Find who has got the highest average in the class

```
SQL> select sname,average from student_marksheet where average in(select
max(average) from student_marksheet);
```

```
SNAME          AVERAGE
-----
Vinod          94
```

Query 11: Find the overall percentage

```
SQL> select (4/5)*100 as "Percentage" from dual;
```

```
Percentage
-----
80
```

Query 12: Display all the updated records

```
SQL> select * from student_marksheet;
```

SNO	REGNO	SNAME	DBMS	DSP	OS	TOTAL	AVERAGE
1	U06CS093	Sathya pradeep.p	88	89	99	276	92
2	U06CS079	Ravindar	79	89	88	256	85.33
3	U06CS092	Satish kumar.R	89	88	95	272	90.67

4	U06CS502	Vinod	98	89	95	282	94
5	U06CS082	Ram.R.V	85	78	88	251	83.67

Front-end Of the Application

```
Dim cn As New ADODB.Connection
```

```
Dim r1 As New ADODB.Recordset
```

```
-----  
-----
```

```
Private Sub Form_Load()
```

```
    cn.Open "vinodh"."pradeep","venki"
```

```
    r1.Open "select * from student_marksheet", cn,  
    adOpenDynamic, adLockOptimistic
```

```
End Sub
```

```
-----  
-----
```

```
Private Sub cmdPrev_Click()
```

```
    r1.MovePrevious
```

```
    If r1.BOF = True Then
```

```
        MsgBox "First Record", vbInformation
```

```
        r1.MoveLast
```

```
    End If
```

```
End Sub
```

```
-----  
-----  
-----
```

```
Private Sub cmdNext_Click()
```

```
    r1.MoveNext
```

```
    If r1.EOF = True Then
```

```
        MsgBox "Last Record", vbInformation
```

```
        r1.MoveFirst
```

```
        End If
    End Sub

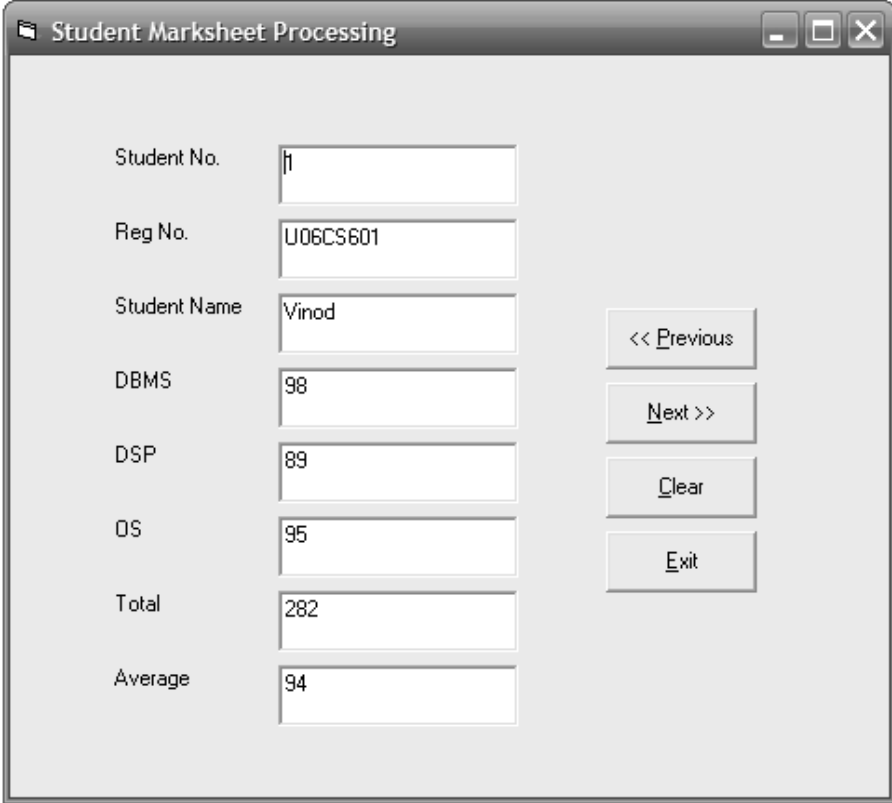
-----
-----

Private Sub cmdClear_Click()
    txtStudno.Text = ""
    txtRegno.Text = ""
    txtStudname.Text = ""
    txtDbms.Text = ""
    txtDsp.Text = ""
    txtOs.Text = ""
    txtTotal.Text = ""
    txtAvg.Text = ""
End Sub

-----
-----

Private Sub cmdExit_Click()
    End
End Sub
```

Output



The screenshot shows a window titled "Student Marksheet Processing" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a form with the following fields and values:

Student No.	11
Reg No.	U06CS601
Student Name	Vinod
DBMS	98
DSP	89
OS	95
Total	282
Average	94

To the right of the form are four buttons: "<< Previous", "Next >>", "Clear", and "Exit".

Result

Ex-No: 2

PAYROLL PROCESSING

Aim

Algorithm

Program code:

Query 01: Create a employee table

```
SQL> create table employee(  
2 empno number(5),  
3 empname varchar2(30),  
4 dept varchar2(20));
```

Table created.

Query 02: Insert minimum 5 records in the employee table

```
SQL> insert into employee  
values (&empno, '&empname', '&dept');  
  
Enter value for empno: 101  
  
Enter value for empname: Salil Dixit  
  
Enter value for dept: DataBase  
  
old 1: insert into employee  
values (&empno, '&empname', '&dept')  
  
new 1: insert into employee values (101, 'Sathya  
pradeep.p', 'DataBase')  
  
1 row created.
```

Query 03: Display all the records in employee table

EMPNO	EMPNAME	DEPT
101	Sathya pradeep.P	DataBase
102	Ravindar	System Analysis
103	Vinoth	Software Development
104	Siva	Software Development
105	Ram.r.v	Human Resources
106	Raj kumar	Hardware

Query 04: Create a payment table

```
SQL> create table payment(  
  2 empno number(5),  
  3 empname varchar2(20),  
  4 dept varchar2(20),  
  5 basic number(8,2),  
  6 hra number(7,2),  
  7 cca number(7,2),  
  8 da number(7,2),  
  9 gross_sal number(8,2),  
10 pf number(7,2),  
10 annual_income number(15,2),  
11 pf number(7,2),  
12 annual_income number(15,2),  
13 net_sal number(9,2));
```

Table created.

Query 05: Insert minimum 5 records in the payment table

```
SQL> insert into payment  
values(&empno, '&design', &basic, &hra, &cca, &da, &gross_sal,  
&pf, &annual_income, &tax, &net_sal);
```

Enter value for empno: 101

Enter value for design: DBA

Enter value for basic: 27580

Enter value for hra: 0

Enter value for cca: 0

Enter value for da: 0

Enter value for gross_sal: 0

Enter value for pf: 0

Enter value for annual_income: 0

Enter value for tax: 0

Enter value for net_sal: 0

```
old 1: insert into payment
values (&empno, '&design', &basic, &cca, &da, &gross_sal, &pf, &annual
```

```
new 1: insert into payment
values (101, 'DBA', 27580, 0, 0, 0, 0, 0, 0, 0, 0)
```

1 row created.

Query 06: Display all records

EMPNO	DESIGN	BASIC	HRA	CCA	DA	GROSS	PF	ANNUAL_INCOME	TAX	NET_SAL
101	DBA	27580	0	0	0	0	0	0	0	0
102	Sen. Analyst	25900	0	0	0	0	0	0	0	0
103	Developer	20985	0	0	0	0	0	0	0	0
104	Developer	23450	0	0	0	0	0	0	0	0
105	Manager	25689	0	0	0	0	0	0	0	0
106	Developer	22568	0	0	0	0	0	0	0	0

Query 07: Calculate HRA amount (15% of BASIC)

```
SQL> update payment set hra=basic*(15/100);
```

6 rows updated.

Query 08: Calculate CCA amount (15% of BASIC)

```
SQL> update payment set cca=basic*(15/100);
```

6 rows updated.

Query 09: Calculate DA amount (30% of BASIC)

```
SQL> update payment set da=basic*(30/100);
```

6 rows updated.

Query 10: Calculate gross salary

```
SQL> update payment set gross=basic+hra+cca+da;
```

6 rows updated.

Query 11: Calculate PF amount (12% of BASIC)

```
SQL> update payment set pf=basic*(12/100);
```

6 rows updated.

Query 12: Calculate ANNUAL INCOME

```
SQL> update payment set annual_income=gross*12;
```

6 rows updated.

Query 13: Calculate TAX with following condition

More than: 1 Lakh 10% of BASIC

2 Lakh 20% of BASIC

3 Lakh 30% of BASIC

```
SQL> update payment set tax=annual_income*(10/100) where  
annual_income>=100000 and annual_income<200000;
```

0 rows updated.

```
SQL> update payment set tax=annual_income*(20/100) where  
annual_income>=200000 and annual_income<300000;
```

0 rows updated.

```
SQL> update payment set tax=annual_income*(30/100) where  
annual_income>300000;
```

6 rows updated.

Query 14: Calculate NET SAL

```
SQL> update payment set net_sal=gross_sal-pf-tax;
```

6 rows updated.

Query 15: Display empno,dept and net sal

```
SQL> select employee.empno,employee.dept,payment.net_sal  
from employee,payment where employee.empno=payment.empno;
```

EMPNO DEPT	NET_SAL
-----	-----
101 DataBase	27580
102 System Analysis	25900
103 Software Development	20985
104 Software Development	23450
105 Human Resources	25689
106 Hardware	22568

6 rows selected.

Front-end Of the Application

```
Dim cn As New ADODB.Connection
```

```
Dim r1 As New ADODB.Recordset
```

```
-----  
-----
```

```
Private Sub Form_Load()
```

```
    cn.Open "Pradeep"."Pradeep","venki"
```

```
    r1.Open "select * from payment", cn, adOpenDynamic,  
    adLockOptimistic
```

```
End Sub
```

```
-----  
-----
```

```
Private Sub cmdPrev_Click()  
    r1.MovePrevious  
    If r1.BOF = True Then  
        MsgBox "First Record", vbInformation  
        r1.MoveLast  
    End If  
End Sub
```

```
-----  
-----  
Private Sub cmdNext_Click()  
    r1.MoveNext  
    If r1.EOF = True Then  
        MsgBox "Last Record", vbInformation  
        r1.MoveFirst  
    End If  
End Sub
```

```
-----  
-----  
-----  
Private Sub cmdClear_Click()  
    txtStudno.Text = ""  
    txtRegno.Text = ""  
    txtStudname.Text = ""  
    txtDbms.Text = ""  
    txtDsp.Text = ""  
    txtOs.Text = ""  
    txtTotal.Text = ""  
    txtAvg.Text = ""  
End Sub
```



```
Private Sub cmdExit_Click()
```

```
    End
```

```
End Sub
```


Output

The screenshot shows a window titled "Payroll Processing System" with the following data:

Emp. No.	101	Annual Income	529536
Designation	DBA	Tax	13238.40
Basic Salary	27580	Net Salary	27580
HRA	4137		
CCA	4137	<< Previous	
DA	8274	Next >>	
Gross Salary	44128	Clear	
PF	3309.60	Exit	

Result

Ex-No: 3

BANK TRANSACTION

Aim

Algorithm

Program code:

Query 01: Create a bank table

```
SQL> create table bank(  
2  accno number(6),  
3  cname varchar2(20),  
4  address varchar2(25),  
5  mob number(10),  
6  deposit number(8,2),  
7  withdraw number(8,2),  
8  balance number(8,2),  
9  tran_date date);
```

Table created.

Query 02: Insert minimum 5 records in the bank table

```
SQL> insert into bank  
values(&accno, '&cname', '&address', &mob, &deposit, &withdraw  
, &balance, '&tran_date');
```

Enter value for accno: 14232

Enter value for cname: Sathya pradeep.p

Enter value for address: chennai

Enter value for mob: 9884697779

Enter value for deposit: 500000

Enter value for withdraw: 150000

Enter value for balance: 350000

Enter value for tran_date: 26-sep-2008

```
old 1: insert into bank  
values(&accno, '&cname', '&address', &mob, &deposit, &withdraw  
, &balance, '&trans
```

```
new 1: insert into bank values(14232, 'Sathya  
pradeep.p', 'chennai', 9884697779, 500000, 150000, 350000, '26-  
sep-
```

1 row created.

Query 03: Display all records from bank table

```
SQL> select * from bank;
```

ACCNO	CNAME	ADDRESS	MOB	DEPOSITE	WITHDRAW	BALANCE	TRAN_DATE
14232	Vinodh	Chennai	9962420365	500000	150000	350000	26-sep-2008
14233	Sanat Soumyakant	Orissa	9789800794	250000	150000	100000	28-sep-2008
14234	Rahul Sharmah	Gauhati	9884860030	300000	100000	200000	25-sep-2008
14235	Pushker Pandey	Gujarat	9840646790	150000	50000	100000	20-aug-2008
14236	Tasaduq Aziz	Kashmir	9941090598	400000	340000	60000	29-sep-2008

Query 04: Add Rs. 50000 to account of 'Vinodh'

```
SQL> update bank set balance=balance+50000 where  
cname='Vinodh';
```

1 row updated.

Query 05: Display the updated balance

```
SQL> select cname,balance from bank where  
cname='Vinodh';
```

```
CNAME          BALANCE  
-----
```

```
Vinodh 400000
```

Query 06: Display the balance of 'Sanat Soumyakant'

```
SQL> select cname,balance from bank where cname='Sanat  
Soumyakant';
```

```
CNAME          BALANCE  
-----
```

```
Sanat Soumyakant 100000
```

Query 07: Withdraw Rs. 30000 from account of 'Sanat Soumyakant'

```
SQL> update bank set balance=balance-30000 where  
cname='Sanat Soumyakant';
```

1 row updated.

Query 08: Display the customer details who did transaction on 29-sep-08

```
SQL> select * from bank where trabsac_date='29-sep-08';
```

ACCN O	CNAME	ADDRES S	MOB	DEPOSIT E	WITHDRA W	BALANC E	TRAN_DAT E
14236	Tasadu q Aziz	Kashmir	994109059 8	400000	340000	60000	29-sep- 2008

Query 09: Display the customer name who lives in Gauhati

```
SQL> select cname from bank where address='Gauhati';
```

CNAME

Rahul Sharmah

Query 10: Display the customer name who is having maximum balance

```
SQL> select cname from bank where balance in(select  
max(balance) from bank);
```

CNAME

Vinodh

Front-end Of the Application

```
Dim cn As New ADODB.Connection
```

```
Dim r1 As New ADODB.Recordset
```

```
-----  
-----  
Private Sub Form_Load()  
    cn.Open "pradeep"."pradeep","venki"  
    r1.Open "select * from bank", cn, adOpenDynamic,  
    adLockOptimistic  
End Sub
```

```
-----  
-----  
Private Sub cmdPrev_Click()  
    r1.MovePrevious  
    If r1.BOF = True Then  
        MsgBox "First Record", vbInformation  
        r1.MoveLast  
    End If  
End Sub
```

```
Private Sub cmdNext_Click()  
    r1.MoveNext  
    If r1.EOF = True Then  
        MsgBox "Last Record", vbInformation  
        r1.MoveFirst  
    End If  
End Sub
```

```
-----  
-----  
Private Sub cmdClear_Click()  
    txtAccno.Text = ""  
    txtCust.Text = ""  
    txtAdd.Text = ""
```

```
txtMob.Text = ""  
txtDep.Text = ""  
txtWith.Text = ""  
txtBal.Text = ""  
txtTran.Text = ""
```

```
End Sub
```

```
-----  
-----
```

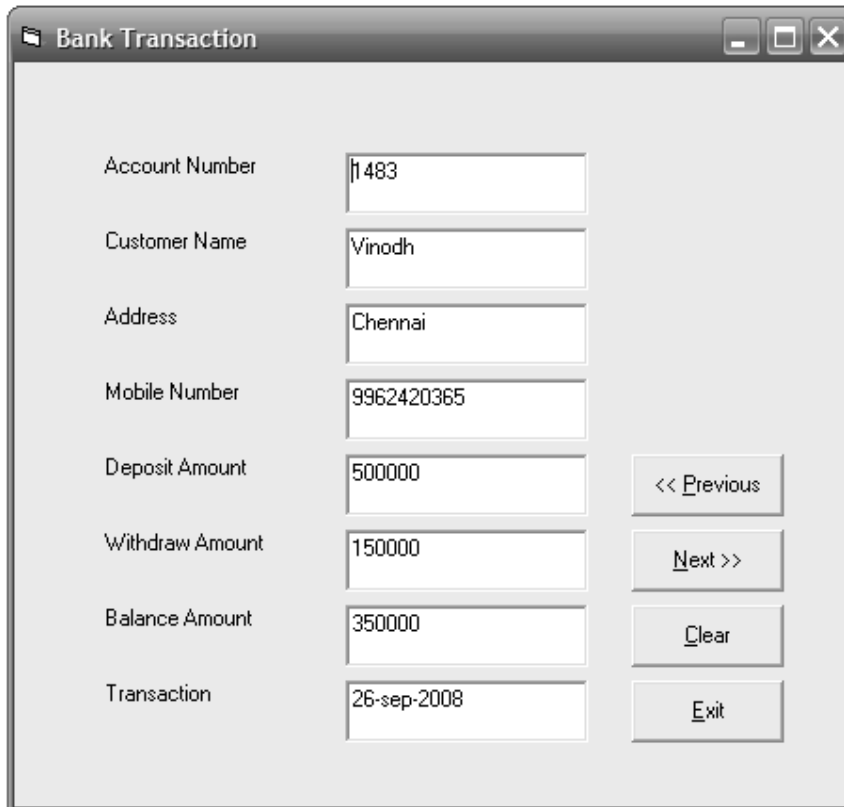
```
Private Sub cmdExit_Click()
```

```
End
```

```
End Sub
```

```
-----  
-----
```


Output



A screenshot of a Java Swing window titled "Bank Transaction". The window contains a form with the following fields and values:

Account Number	1483
Customer Name	Vinodh
Address	Chennai
Mobile Number	9962420365
Deposit Amount	500000
Withdraw Amount	150000
Balance Amount	350000
Transaction	26-sep-2008

On the right side of the form, there are four buttons: "<< Previous", "Next >>", "Clear", and "Exit".

Result

Ex-No: 4

LIBRARY INFORMATION SYSTEM

Aim

Algorithm

Program code:

Query 01: Create a library table

```
SQL> create table library(  
2  book_no number(3),  
3  book_title varchar2(15),  
4  book_auth varchar2(15),  
5  quan_books number(3),  
6  avail_books number(3),  
7  stud_name varchar2(15),  
8  dept varchar2(5),  
9  issue_date date,  
10 ret_date date,  
11 extra_days number(3),  
12 ren_date date,  
13 fine_amt number(5,2));
```

Table created.

Query 02: Insert minimum 5 records in the library table

```
SQL> insert into library  
values(&book_no, '&book_title', '&book_auth',  
&quan_books, &avail_books,  
'&stud_name', '&dept', '&issue_date', '&ret_date', &extra_days, '&re  
n_date', &fine_amt);
```

Enter value for book_no: 101

Enter value for book_title: OOPS Programing

Enter value for book_auth: Balaguruswami

Enter value for quan_books: 85

Enter value for avail_books: 84

Enter value for stud_name: Salil

Enter value for dept: CSE

Enter value for issue_date: 03-aug-2008

Enter value for ret_date: 02-sep-2008

Enter value for extra_days: 0

Enter value for ren_date: 18-sep-2008

Enter value for fine_amt: 0

```
old 1: insert into library
values(&book_no, '&book_title', '&book_auth', &quan_books, &avail_
books, '
```

```
new 1: insert into library values(101, 'OOPS
Programing', 'Balaguruswami', 85, 84, 'Salil Dixit', 'CSE',
```

1 row created.

Query 03: Display all records

```
SQL> select * from library;
```

B_NO	Book_TITL E	BOOK_ AUTH	QUAN_ BOOKS	AVAIL_ BOOKS	STUD_ NAME	DE PT	ISSUE_D ATE	RET_ DATE	EXTA _DAY S	REN_ DATE	FINE_ AMT	DAYS TAKE N
10 1	OOPS Prog	Bala	85	84	Salil	CSE	03-aug- 2008	02-sep- 2008	0	18-sep- 2008	0	0
10 2	DBMS	Ramez Elmasri	70	69	Priya	CSE	04-sep- 2008	05-oct- 2008	0	21-oct- 2008	0	0
10 3	DSP	Nagoor Koni	45	44	Smruti	CSE	06-aug- 08	07-sep- 08	0	10-sep- 08	0	0
10 4	C Prog	Yashwa nt K	80	79	Suraj	CSE	25-aug- 08	26-sep- 08	0	27-sep- 08	0	0
10 5	Data Struct	Sahni S	16	15	Sunil	CSE	29-sep- 08	30-oct- 08	0	05-nov- 08	0	0

Query 04: Find out total books in library

```
SQL> select sum(quan_books) as "Total books in library"
from library;
```

Total books in library

296

Query 05: Find out total quantity of books on DBMS

```
SQL> select sum(quant_books) as "Total books on DBMS" from
library where book_title='rdbms';
```

Total books on DBMS

70

Query 06: Calculate number of days taken

```
SQL> update library set days_taken=ret_date-issue_date;
```

5 rows updated.

Query 07: Calculate the return date

```
SQL> update library set ret_date=issue_date+30;
```

5 rows updated.

Query 08: Calculate the renewal date

```
SQL> update library set ren_date=ret_date+15;
```

5 rows updated.

Query 09: Find out who has taken C Prog book

```
SQL> select stud_name from library where book_title='C
Prog';
```

STUD_NAME

Suraj

Query 10: Find out whether author "Balaguruswami" book is available

```
SQL> select book_title,book_auth from library where
book_auth='Balaguruswami' and avail_books>0;
```

BOOK_TITLE BOOK_AUTHOR

OOPS Programing Balaguruswami

Query 11: Calculate the fine amount

```
SQL> update library set extra_days=days_taken-15;
```

5 rows updated.

```
SQL> update library set fine_amt=extra_days*0.50;
```

5 rows updated.

Query 12: Display the updated table

B_NO	Book_TITL E	BOOK_ AUTH	QUAN_ BOOKS	AVAIL_ BOOKS	STUD_ NAME	DE PT	ISSUE_D ATE	RET_ DATE	EXTA_ _DAY S	REN_ DATE	FINE_ AMT	DAYS TAKE N
101	OOPS Prog	Bala	85	84	Salil	CSE	03-aug- 2008	02-sep- 2008	15	18-sep- 2008	7.50	30
102	DBMS	Ramez Elmasri	70	69	Priya	CSE	04-sep- 2008	05-oct- 2008	16	21-oct- 2008	8	31
103	DSP	Nagoor Koni	45	44	Smruti	CSE	06-aug- 08	07-sep- 08	17	10-sep- 08	8.50	32
104	C Prog	Yashwa nt K	80	79	Suraj	CSE	25-aug- 08	26-sep- 08	17	27-sep- 08	8.5	32
105	Data Struct	Sahni S	16	15	Sunil	CSE	29-sep- 08	30-oct- 08	16	05-nov- 08	8	31

Front End Of The Application

```
Dim cn As New ADODB.Connection
```

```
Dim r1 As New ADODB.Recordset
```

```
Private Sub Form_Load()
```

```
    cn.Open "pradeep"."pradeep","venki"
```

```
    r1.Open "select * from library", cn, adOpenDynamic, adLockOptimistic
```

```
End Sub
```

```
Private Sub cmdPrev_Click()
```

```
    r1.MovePrevious
```

```
    If r1.BOF = True Then
```

```
        MsgBox "First Record", vbInformation
```

```
    r1.MoveLast
```

End If

End Sub

Private Sub cmdNext_Click()

 r1.MoveNext

 If r1.EOF = True Then

 MsgBox "Last Record", vbInformation

 r1.MoveFirst

 End If

End Sub

Private Sub cmdClear_Click()

 txtBookno.Text = ""

 txtBooktitle.Text = ""

 txtAuth.Text = ""

 txtQuan.Text = ""

 txtAvail.Text = ""

 txtStudname.Text = ""

 txtDept.Text = ""

 txtIssue.Text = ""

 txtRet.Text = ""

 txtRen.Text = ""

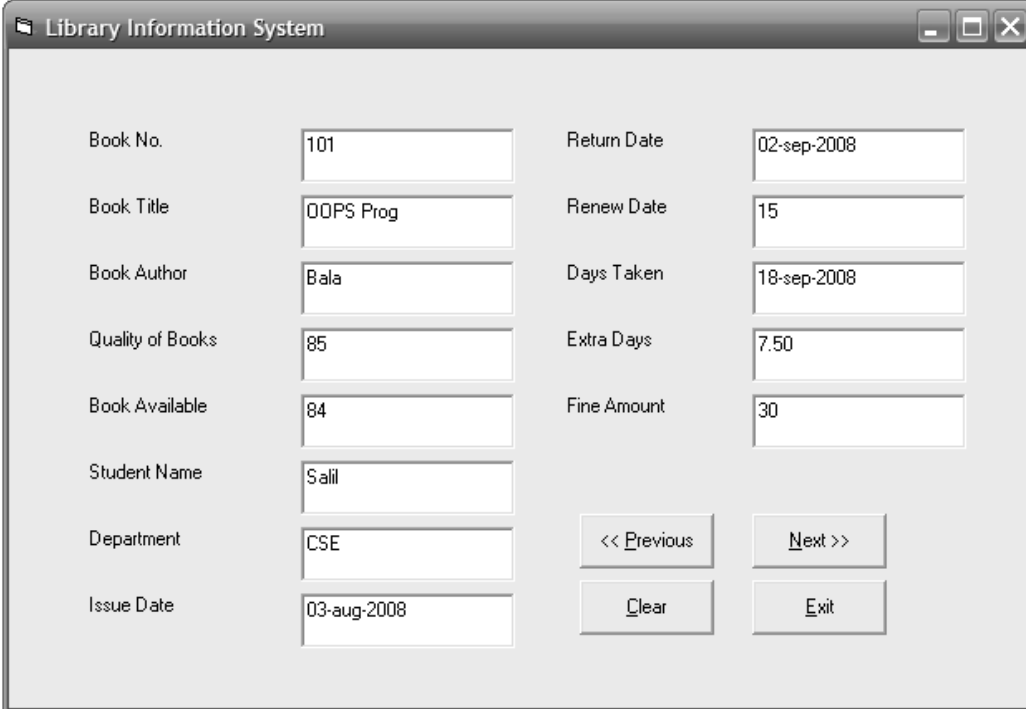
 txtDaystaken.Text = ""

 txtExtradays.Text = ""

 txtFine.Text = ""

End Sub

Output



The screenshot shows a window titled "Library Information System" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a form with the following fields and buttons:

Book No.	101	Return Date	02-sep-2008
Book Title	OOPS Prog	Renew Date	15
Book Author	Bala	Days Taken	18-sep-2008
Quality of Books	85	Extra Days	7.50
Book Available	84	Fine Amount	30
Student Name	Salil		
Department	CSE	<< Previous	Next >>
Issue Date	03-aug-2008	Clear	Exit

Result

Ex-No: 5

ELECTRICITY BILL PROCESSING

Aim

Algorithm

Program code:

Query 01: Create a elec bill table

```
SQL> create table elec_bill(  
2   cno number(5),  
3   cname varchar2(20),  
4   address varchar2(25),  
5   city varchar2(15),  
6   prev_read number(5),  
7   curr_read number(5),  
8   units number(5),  
9   rate number(5),  
10  amount number(7,2));
```

Table created.

Query 02: Insert minimum 5 records in the elec bill table

```
SQL> insert into elec_bill  
values(&cno, '&cname', '&address', '&city', &prev_read, &curr_r  
ead, &units, &rate, &amount);
```

Enter value for cno: 101

Enter value for cname: Pradeep

Enter value for address: perambur

Enter value for city: chennai

Enter value for prev_read: 355

Enter value for curr_read: 865

Enter value for units: 0

Enter value for rate: 0

Enter value for amount: 0

```
old 1: insert into elec_bill  
values(&cno, '&cname', '&address', '&city', &prev_read, &curr_  
read
```

```
new 1: insert into elec_bill
values (101, 'Pradeep', 'perambur', 'chennai', 355, 865, 0, 0, 0)
```

1 row created.

Query 03: Display all the records

```
SQL> select * from elec_bill;
```

CN O	CNAME	ADDRESS	CITY	PREV_ READ	CURR_RE AD	UNIT S	RAT E	AMOUN T
101	Vinod	Velachery	Chennai	355	865	0	0	0
102	Soroujit	Selaiyur	Chennai	211	355	0	0	0
103	Rishi	R K Puram	Delhi	123	234	0	0	0
104	Sunil	Rampur	Puri	321	489	0	0	0
105	Sujoy	Hawrah	Kolkata	289	456	0	0	0

Query 04: Calculate the units

```
SQL> update elec_bill set units=curr_read-prev_read;
```

5 rows updated.

Query 05: Find out the rate for each customer

<100 = Rs.0.80

100-200 = Rs.1.00

200-300 = Rs.1.50

300-400 = Rs.2.00

>400 = Rs.3.00

```
SQL> update elec_bill set rate=0.80 where units<=100;
```

0 rows updated.

```
SQL> update elec_bill set rate=1.00 where units>100 and
units<=200;
```

4 rows updated.

```
SQL> update elec_bill set rate=1.50 where units>200 and
units<=300;
```

0 rows updated.

```
SQL> update elec_bill set rate=2.00 where units>300 and
units<=400;
```

0 rows updated.

```
SQL> update elec_bill set rate=3.00 where units>400;
```

1 row updated.

Query 06: Find out the amount payable for each customer

```
SQL> update elec_bill set amount=units*rate;
```

5 rows updated.

Query 07: Display cname, units, amount

```
SQL> select cname,units,amount from elec_bill;
```

CNAME	UNITS	AMOUNT
-----	-----	-----
Vinod	510	1530
Soroujit	144	144
Rishi	111	111
Sunil	168	168
Sujoy	167	167

Query 08: Display cname, units, amount who are living in Chennai

```
SQL> select cname,units,amount from elec_bill where
city='Chennai';
```

CNAME	UNITS	AMOUNT
-------	-------	--------

```

-----
Vinod          510      1530
Soroujit      144       144

```

Query 09: Display cname, city who used the maximum units

```
SQL>select cname,city from elec_bill where units
in(select max(units) from elec_bill);
```

```

CNAME      CITY
-----
Vinod      Chennai

```

Query 10: Display cname, who paid the lowest amount

```
SQL>select cname from elec_bill where amount in(select
min(amount) from elec_bill);
```

```

CNAME
-----
Rishi

```

Front End Of The Application

```
Dim cn As New ADODB.Connection
```

```
Dim r1 As New ADODB.Recordset
```

```
-----
Private Sub Form_Load()
```

```
    cn.Open "pradeep"."pradeep","venki"
```

```
    r1.Open "select * from elec_bill", cn, adOpenDynamic, adLockOptimistic
```

```
End Sub
```

```
-----
Private Sub cmdPrev_Click()
```

```
    r1.MovePrevious
```

```
    If r1.BOF = True Then
```

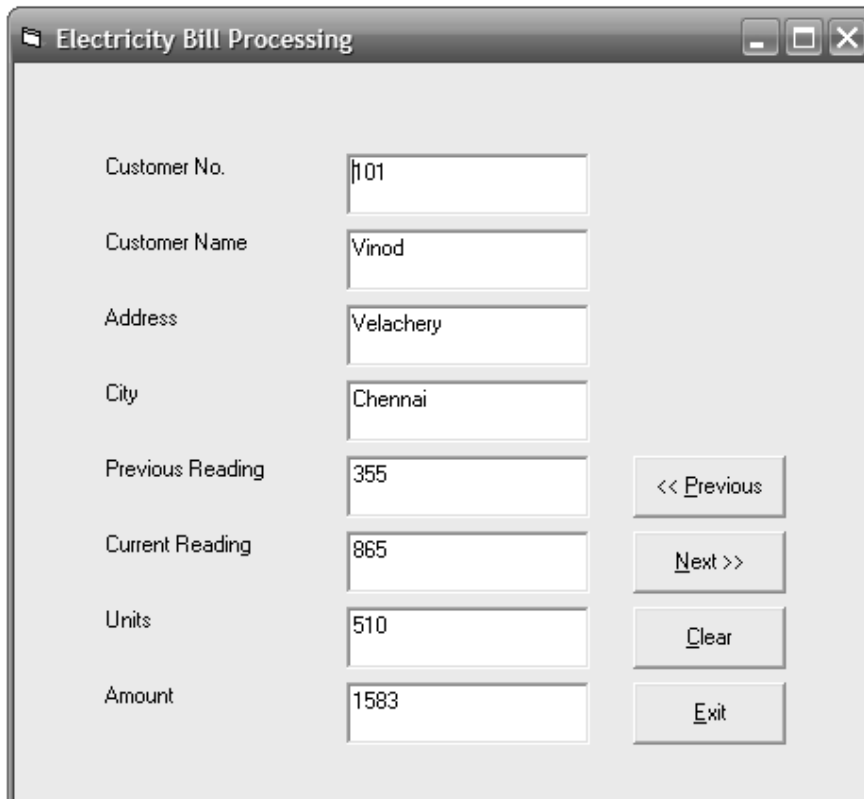
```
        MsgBox "First Record", vbInformation
        r1.MoveLast
    End If
End Sub
```

```
Private Sub cmdNext_Click()
    r1.MoveNext
    If r1.EOF = True Then
        MsgBox "Last Record", vbInformation
        r1.MoveFirst
    End If
End Sub
```

```
Private Sub cmdClear_Click()
    txtCustno.Text = ""
    txtCustname.Text = ""
    txtAdd.Text = ""
    txtCity.Text = ""
    txtPrev.Text = ""
    txtCurr.Text = ""
    txtUnit.Text = ""
    txtAmt.Text = ""
End Sub
```

```
Private Sub cmdExit_Click()
    End
End Sub
```


Output



The screenshot shows a window titled "Electricity Bill Processing" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a form with the following fields and values:

Field	Value
Customer No.	101
Customer Name	Vinod
Address	Velachery
City	Chennai
Previous Reading	355
Current Reading	865
Units	510
Amount	1583

Navigation buttons are located on the right side of the form:

- << Previous
- Next >>
- Clear
- Exit

Result

Ex-No: 6

ADDITION OF THREE NUMBERS

Program code:

```
SQL> set serveroutput on
```

```
SQL> declare
```

```
2 a number;
```

```
3 b number;
```

```
4 c number;
```

```
5 d number;
```

```
6 begin
```

```
7 a:=&a;
```

```
8 b:=&b;
```

```
9 c:=&c;
```

```
10 d:=a+b+c;
```

```
11 dbms_output.put_line('Sum of three numbers is' ||d);
```

```
12 end;
```

```
13 /
```

Output

Enter value for a: 5

old 7: a:=&a;

new 7: a:=5;

Enter value for b: 5

old 8: b:=&b;

new 8: b:=5;

Enter value for c: 5

old 9: c:=&c;

new 9: c:=5;

Sum of three numbers is 15

PL/SQL procedure successfully completed.

Result

Ex-No: 7

FINDING THE MAXIMUM NUMBER

Aim

Algorithm

Program code:

```
SQL> set serveroutput on

SQL> declare
  2  a number;
  3  b number;
  4  c number;
  5  d number;
  6  begin
  7  dbms_output.put_line('enter a:');
  8  a:=&a;
  9  dbms_output.put_line('enter b:');
10  b:=&b;
11  dbms_output.put_line('enter c:');
12  c:=&b;
13  if(a>b)and(a>c) then
14  dbms_output.put_line('A is maximum');
15  elsif(b>a)and(b>c) then
16  dbms_output.put_line('B is maximum');
17  else
18  dbms_output.put_line('C is maximum');
19  end if;
20  end;
21  /
```

Output

Enter value for a: 9

old 8: a:=&a;

new 8: a:=9;

Enter value for b: 6

old 10: b:=&b;

new 10: b:=6;

Enter value for c: 10

old 12: c:=&b;

new 12: c:=10;

enter a:

enter b:

enter c:

C is maximum

PL/SQL procedure successfully completed.

Result

Ex-No: 8

CALCULATE SIMPLE INTEREST

Aim

Algorithm

Program code:

```
SQL> set serveroutput on

SQL> declare

  2  simple_interest number;
  3  p number;
  4  n number;
  5  r number;
  6  begin
  7  dbms_output.put_line('Enter the Principal amount(
p): ');
  8  p:=&p;
  9  dbms_output.put_line('Enter the No of Years( n): ');
 10  n:=&n;
 11  dbms_output.put_line('Enter the rate of interest
(r): ');
 12  r:=&r;
 13  simple_interest:=p*n*(r/100);
 14  dbms_output.put_line(' The simple interest
is' ||simple_interest);
 15  end;
 16  /
```

Output

Enter the Principal amount(p): 10000

old 8: p:=&p;

new 8: p:=10000;

Enter the No of Years(n): 2

old 10: n:=&n;

new 10: n:=2;

Enter the rate of interest(r) :8

old 12: r:=&r;

new 12: c:=8;

The simple interest is 1600

PL/SQL procedure successfully completed.

Result

Ex-No: 9

SUM OF 100 NUMBERS

Aim

Algorithm

Program code:

```
SQL> set serveroutput on
```

```
SQL> declare
```

```
2 a number;
```

```
3 s1 number default 0;
```

```
4 begin
```

```
5 a:=1;
```

```
6 loop
```

```
7 s1:=s1+a;
```

```
8 exit when (a=100);
```

```
9 a:=a+1;
```

```
10 end loop;
```

```
11 dbms_output.put_line('sum between 1 to 100 is' ||  
s1);
```

```
12 end;
```

```
13 /
```

Output

sum between 1 to 100 is 5050

Result

Ex-No: 10

REVERSING A STRING

Aim

Algorithm

Program code:

```
SQL> set serveroutput on
SQL> declare
    2  given_string  varchar2(5);
    3  str_length number(2);
    4  inverted_string  varchar2(5);
    5  begin
    6  given_string:='&given_string';
    7  str_length:=length(given_string);
    8  for cnt in reverse 1..str_length
    9  loop
10  inverted_string:=inverted_string||substr(given_string,
        cnt,1);
11  end loop;
12  dbms_output.put_line('the given string
is'||given_string);
13  dbms_output.put_line('the inverted string is'||
inverted_string);
14  end;
15 /
```

Output

```
old 6: given_string:='&given_string';
```

```
new 6: given_string:='ashoka';
```

```
the given string is ashok
```

```
the inverted string is kohsa
```

```
PL/SQL procedure successfully completed.
```

Result

Ex-No: 11

PL/SQL CURSOR

Aim

Algorithm

Program code:

Query 01: Create a std_cse table

```
SQL> create table std_cse(  
2 regno number(5),  
3 name varchar2(10),  
4 mark1 number(3),  
5 mark2 number(3),  
6 mark3 number(3),  
7 total number(3),  
8 average number(10),  
9 remarks varchar2(10));
```

Table created.

Query 02: Insert minimum 5 records in the std_cse table

```
SQL> insert into std_cse  
values(&regno, '&name', &mark1, &mark2, &mark3, &total, &average,  
&remarks');
```

Enter value for regno: 1

Enter value for name: Vinodh

Enter value for mark1: 98

Enter value for mark2: 89

Enter value for mark3: 86

Enter value for total: 0

Enter value for average: 0

Enter value for remarks: abc

```
old 1: insert into std_cse  
values(&regno, '&name', &mark1, &mark2, &mark3, &total, &average, '&remarks')
```

```
new 1: insert into std_cse values(1,'Vinodh',98,89,86,0,0,'abc')
```

1 row created.

Query 03: Display all the records

```
SQL> select * from std_cse;
```

REGNO	NAME	MARK1	MARK2	MARK3	TOTAL	AVERAGE	REMARKS
1	Pradeep	98	89	86	0	0	abc
2	Anand	89	95	85	0	0	abc
3	Soroujit	85	86	99	0	0	abc
4	Swagat	85	87	89	0	0	abc
5	Suraj	85	94	89	0	0	abc

CURSOR:

```
SQL> declare
```

```
    2  cursor c_student is select
    3  regno,mark1,mark2,mark3,total,average,remarks from
std_cse;
    4  sno std_cse.regno%type;
    5  m1 std_cse.mark1%type;
    6  m2 std_cse.mark2%type;
    7  m3 std_cse.mark3%type;
    8  tot std_cse.total%type;
    9  avr std_cse.average%type;
   10  rmk std_cse.remarks%type;
   11  begin
   12  open c_student;
   13  fetch c_student into sno,m1,m2,m3,tot,avr,rmk;
   14  while c_student%found
   15  loop
```



```
16 tot:=m1+m2+m3;
17 update std_cse set total=tot where regno=sno;
18 avr:=(tot)/3;
19 update std_cse set average=avr where regno=sno;
20 if(avr>=80) then rmk:='V Good';
21 elsif(avr>=60) then rmk:='Good';
22 elsif(avr>=40) then rmk:='Better';
23 elsif(avr<40) then rmk:='Poor';
24 end if;
25 update std_cse set remarks=rmk where regno=sno;
26 fetch c_student into sno,m1,m2,m3,tot,avr,rmk;
27 end loop;
28 end;
29 /
```

PL/SQL procedure successfully completed.

Output

Query 04: Display all the records

REGNO	NAME	MARK1	MARK2	MARK3	TOTAL	AVERAGE	REMARKS
1	Vinodh	98	89	86	273	91	V Good
2	Anand	89	95	85	269	90	V Good
3	Soroujit	85	86	99	270	90	V Good
4	Swagat	85	87	89	261	87	V Good
5	Suraj	85	94	89	268	89	V Good

Result

Ex-No: 12

**EMPLOYEE SALARY PROCESSING USING
EXCEPTION HANDLING**

Aim

Algorithm

Program code:

Query 01: Create a emp table

```
SQL> create table emp(  
2   eno number(4),  
3   salary number(6,2));
```

Table created.

Query 02: Insert records in the emp table

Enter value for eno: 123

Enter value for salary: 5000

```
old 1: insert into emp values(&eno,&salary)
```

```
new 1: insert into emp values(123,5000)
```

1 row created.

Old values

ENO	SALARY
123	5400
124	6000
125	4700

Procedure:

```
SQL> declare  
2   empno number(4) := '&empno';  
3   sal number(6,2);  
4   sal_missing exception;  
5   begin  
6   select salary into sal from emp where eno=empno;
```

```
7  if sal is NULL then
8  raise sal_missing;
9  end if;
10 if sal<2000 then
11 sal:=sal+sal*(5/100);
12 elsif sal<5000 then
13 sal:=sal+sal*(6/100);
14 elsif sal<1000 then
15 sal:=sal+sal*(7/100);
16 elsif sal>1000 then
17 sal:=sal+sal*(8/100);
18 end if;
19 update emp set salary=sal where empno=eno;
20 exception
21 when sal_missing then
22 dbms_output.put_line('Salary is null');
23 when no_data_found then
24 dbms_output.put_line('Invalidempno');
25 end;
26 /
```

Enter value for empno: 125

```
old 2: empno number(4):='&empno';
```

```
new 2: empno number(4):='125';
```

PL/SQL procedure successfully completed.

Output

New values

ENO	SALARY
123	5400
124	6000
125	4982

Result