Exp.No: 01 Date :	Data Definition Language

AIM: To study and work with DDL Commands in Oracle

DESCRIPTION:

CREATE: To create a table, the basic structure to hold user data, specifying this information:

- column definitions
- integrity constraints
- the table's tablespace
- storage characteristics
- an optional cluster
- data from an arbitrary query

Syntax

CREATE TABLE tablename (columnname datatype [constraint [constraint name]], [columnname .....] [, constraints]);

Constraints:

- NOT NULL every record must have a value for the column
- UNIQUE value for each record must be distinct
- CHECK checks to make sure the condition is satisfied
- PRIMARY KEY defines the primary key of the table
- FOREIGN KEY (columname) REFERENCES tablename (columnname) defines the foreign key of the table

ALTER: To alter the definition of a table in one of these ways:

- to add a column
- to add an integrity constraint
- to redefine a column (data type, size, default value)

- to explicitly allocate an extent

Syntax

ALTER TABLE tablename MODIFY (columnname [datatype] [constraint]);

ALTER TABLE tablename ADD (columnname datatype [constraint]);

DROP: This command is used todrop the table.

Syntax

DROP TABLE tablename;

RENAME : RENAME tablename to new-tablename;

TRUNCATE: Delete all the records in the table retaining its structure.
Syntax: Truncate table <table name>;

Example SQL> truncate table employee;

PROCEDURE

1.Create a table with the given fields

2.Display the structure of the table

3.Alter the size and datatype of the fields in the table

4.Rename the table names

5.Drop the tables

1. **Write a query to create a table student with the following list of attributes Sno, Name, Register number, Marks.**

SQL> create table stud(sno number,name varchar2(15),regno number,marks number);

Table created.

SQL> desc st;

Name	Null?	Type
SNO		NUMBER
NAME		VARCHAR2 (15)
REGNO		NUMBER
MARKS		NUMBER

2. Write a query to rename the table name.

SQL> alter table stud rename to student;

Table altered.

3. Write a query to describe the table.

SQL> desc student;

Name	Null?	Type
SNO		NUMBER
NAME		VARCHAR2(15)
REGNO		NUMBER
MARKS		NUMBER

4. Write a query to change the type of an attribute Name as Number.

SQL> alter table student modify name number;

Table altered.

To Verify: SQL> desc student;

Name	Null?	Type
SNO		NUMBER
NAME		NUMBER
REGNO		NUMBER
MARKS		NUMBER

Table Creation and adding Constraints

5. Write a query to create a table BankAccount and assign Primary key Constraint for the attribute Accno and assign NOTNULL Constraint for the field CustomerName.

SQL> create table BankAccount(sno number,customername varchar2(10) not null,acno number Primary key,bal number);

Table created.

To Verify: SQL> desc BankAccount;

Name	Null?	Type
SNO		NUMBER
CUSTOMERNAME	NOT NULL	VARCHAR2 (10)
ACCNO	NOT NULL	NUMBER
BAL		NUMBER

6. Write a query to Add a Primary key for the attribute Regno of the table Student.

SQL> alter table student add(primary key(regno));

Table altered.

To Verify: SQL> desc student;

Name	Null?	Type
SNO		NUMBER
NAME		VARCHAR2(15)
REGNO	NOT NULL	NUMBER
MARKS		NUMBER

7. **Write a query to Alter "Student" table to add "Gender" and "City" columns and verify it.**

SQL> alter table student add(Gender varchar2(5), City varchar2(15));

Table altered.

To Verify: SQL> desc student;

8. Write a query to Alter "student" table to modify the column "Name" as "Sname" and add Unique Constraint on it and verify it.

SQL> alter table student modify(Name varchar2(20) constraint Sname unique);

Table altered.

To Verify: SQL> desc student;

9. Write a query to Alter "student" table to drop the "City" column and verify it.

SQL> alter table student drop column City;

Table altered.

To Verify: SQL> desc student;

10. Write a query to alter "student" table to set the "Gender" column as unused and verify it.

SQL> alter table student set unused(Gender);

Table altered.

To Verify: SQL> desc student;

11. Write a query to view the constraint names and its types of the "student" table from the User_Constraints table.

SQL> select constraint_name,constraint_type from user_constraints where table_name='student';

12. Write a query to view the column associated with the constraints names of the BankAccount table from the user_cons_columns view.

SQL> select constraint_name,column_name from user_cons_columns where table_name='BankAccount';

13. Write a query to create a FOREIGN KEY constraint on the "Accno" column when the "Loan" table which is already created.

SQL> alter table Loan add foreign key(Accno) references BankAccount(Accno);

(CHECK)

14. Write a query to create a table stud with attributes rno,name,sal and add check constraint to attribute sal.

SQL> create table stud(rno number(5),name varchar2(10),sal number(10) constraint no_ck check(sal between 10000 and 30000));

Table created.
SQL> insert into stud values(&rno,'&name',&sal);
Enter value for rno: 567
Enter value for name: sachin
Enter value for sal: 29000
old 1: insert into stud values(&rno,'&name',&sal)
new 1: insert into stud values(567,'sachin',29000)

1 row created.
SQL> /
Enter value for rno: 565
Enter value for name: rohit
Enter value for sal: 35000
old 1: insert into stud values(&rno,'&name',&sal)
new 1: insert into stud values(565,'rohit',35000)
insert into stud values(565,'rohit',35000)
*
ERROR at line 1:
ORA-02290: check constraint (SCOTT.NO_CK) violated

(FOREIGN KEY)

15. Write a query to create a table adm with attributes stuid,sname,percentage(per) and add primary key constraint to stuid. Create another table course with attributes stuid,branch and sec and add foreign key constraint to stuid.

SOL>create table adm(stuid number(6) constraint stuid_pk primary key,sname varchar2(15),per number(5));
Table created.
SQL> insert into adm values(&stuid,'&sname',&per);
Enter value for stuid: 1
Enter value for sname: abi
Enter value for per: 80
old 1: insert into adm values(&stuid,'&sname',&per)
new 1: insert into adm values(1,'abi',80)

1 row created.
SQL> /
Enter value for stuid: 2
Enter value for sname: rohit
Enter value for per: 89
old 1: insert into adm values(&stuid,'&sname',&per)
new 1: insert into adm values(2,'rohit',89)
1 row created.
SQL> /
Enter value for stuid: 3

```
Enter value for sname: sachin
Enter value for per: 99
old   1: insert into adm values(&stuid,'&sname',&per)
new   1: insert into adm values(3,'sachin',99)

1 row created.
SQL> /
Enter value for stuid: 4
Enter value for sname: naveen
Enter value for per: 70
old   1: insert into adm values(&stuid,'&sname',&per)
new   1: insert into adm values(4,'naveen',70)
1 row created.
SQL> select * from adm;
    STUID SNAME       PER
---------- ---------- ----------
         1 abi         80
         2 rohit       89
         3 sachin      99
         4 naveen      70

SQL> create table course(stuid number(6) constraint sid_fk references adm(stuid),branch
varchar2(5),sec varchar2(10));

Table created.

SQL> insert into course values(&stuid,'&branch','&sec');
Enter value for stuid: 1
Enter value for branch: cse
Enter value for sec: a
old   1: insert into course values(&stuid,'&branch','&sec')
new   1: insert into course values(1,'cse','a')

1 row created.

SQL> /
Enter value for stuid: 5
Enter value for branch: cse
Enter value for sec: b
old   1: insert into course values(&stuid,'&branch','&sec')
new   1: insert into course values(5,'cse','b')
insert into course values(5,'cse','b')
*
ERROR at line 1:
ORA-02291: integrity constraint (SCOTT.SID_FK) violated - parent key not found
```

```
SQL> delete from adm where stuid=1;
*
ERROR at line 1:
ORA-02292: integrity constraint (SCOTT.SID_FK) violated - child record found

SQL> delete from course where stuid=1;

1 row deleted.

SQL> delete from adm where stuid=1;

1 row deleted.

SQL>select * from adm;

STUID SNAME     PER
----- --------------- - ----------
    2       rohit      89
    3       sachin           99
    4       naveen           70
```

Disabling a Constraint:

17. Write a query to disable the Unique key (Sname) Constraint on the student table without Dropping it or Recreating it and verify it.

SQL> alter table student disable constraint Sname;

Table altered.

To Verify: SQL> desc student;

18. Write a query to enable or activate the Unique key Constraint(Sname) on "student" table and verify it.

SQL> alter table student enable constraint Sname;

Table altered.

To Verify: SQL> desc student;

Dropping Constraints:

19. Write a query to remove the Unique key(Sname) Constraint from the student table and

verify it.

SQL> alter table student drop constraint Sname;

Table altered.

To Verify: SQL> desc student;

20. Write a query to remove the Primary key Constraint on the "BankAccount" table and Drop the associated Foreign key Constraint on the Loan.Accno column and verify it.

SQL> alter table student drop primary key cascade;

Table altered.

To Verify: SQL> desc student;

21. Write a query to Rename the "student" table to "Student" and verify it.

SQL> rename student to Student;

To Verify: SQL> desc Student;

Truncating and Deleting the table:

22. Write a query to Truncate the table.

SQL> truncate table student;

Table truncated.

To Verify: SQL> select *from student;

No rows selected

23. Write a query to delete the table permanently.

SQL> drop table student;

Table dropped.

To Verify: SQL> desc student;

ERROR:
ORA-04043: object student does not exist.

Exp.No: 2 Date :	DATA MANIPULATION LANGUAGE (DML)

1.INSERT:
Insert command insert one or more rows into a table.
Syntax:
INSERT INTO<table name> values (value1, value2, value3….);
Example: SQL>insert into emp1 values (7954,'SMITH','CLERK', 7902,'17-DEC-1980' ,800,NULL,20); SQL> insert into emp1 values (&empno,'&ename','&job',&mgr,'&hiredate',&sal, comm);

2. SIMPLE SELECT STATEMENT:
To view records from a table
Syntax: SELECT * from <tablename> SELECT column1, column2 from <tablename>
Example
SQL>select * from emp1
SQL>select eno,ename from emp1

3.Update
The UPDATE command can be used to modify information contained within a table, either in bulk or individually.

Syntax

1.UPDATE tablename SET fieldname=new value;
2. UPDATE table name SET fieldname=new value where condition;

4. DELETE: To delete particular record from a table.
Syntax: Delete from <tablename> where <condition>
Example: SQL> Delete from emp1 where ename='john';

To Insert the values into the Table:

1. Write a query to insert a new row containing values for each column into the "Student" table and verify it.

```
SQL>  insert into student values(&sno,'&name',&regno,&contact_no);
Enter value for sno: 1
Enter value for name: shiva
Enter value for regno: 1017191
Enter value for contact_no: 9941416491
old   1:  insert into student values(&sno,'&name',&regno,&contact_no)
new   1:  insert into student values(1,'shiva',1017191,9941416491)
```

1 row created.

```
SQL> /
Enter value for sno: 2
Enter value for name: vignesh kumar
Enter value for regno: 1017208
Enter value for contact_no: 9710352789
old   1:  insert into student values(&sno,'&name',&regno,&contact_no)
new   1:  insert into student values(2,'vignesh kumar',1017208,9710352789)
```

1 row created.

```
SQL> /
Enter value for sno: 3
Enter value for name: uva
Enter value for regno: 1017207
Enter value for contact_no: 9952974163
old   1:  insert into student values(&sno,'&name',&regno,&contact_no)
new   1:  insert into student values(3,'uva',1017207,9952974163)
```

1 row created.

```
SQL> /
Enter value for sno: 4
Enter value for name: vignesh
Enter value for regno: 1017209
Enter value for contact_no: 9444715807
old   1:  insert into student values(&sno,'&name',&regno,&contact_no)
new   1:  insert into student values(4,'vignesh p',1017209,9444715807)
1 row created.
```

[Note: Backslash(/) is used to get the values consecutively]

2. Write a query to display the table content.

SQL> select * from student;

SNO	NAME	REGNO	CONTACT_NO
1	shiva	1017191	9941416491
2	vignesh kumar	1017208	9710352789
3	uva	1017207	9952974163
4	vignesh p	1017209	9444715807

3. Write a query to display the name and contact_no Columns from "student" table.

SQL> select name,contact_no from student;

NAME	CONTACT_NO
shiva	9941416491
vignesh kumar	9710352789
uva	9952974163
vignesh p	9444715807

4. Write a query to display the distinct values of "regno" from "student" table.

SQL> select distinct regno from student;

REGNO
1017191
1017207
1017208
1017209

5. Write a query to insert new row containing values for each column into the "Student" table and verify it.

SQL>insert into student values (5, 'Raja', 1017215, 984111555);

1 row created

6. Create a table "Employee" with sno, name,empid and salary as its attributes.

SQL> create table empdata(sno number,name varchar2(20),empid number primary key,salary number);

Table created.

7. Write a query to insert values for each column into the "Employee" table and verify it.

```
SQL> insert into Employee values(&sno,'&name',&empid,&salary);
Enter value for sno: 1
Enter value for name: monica
Enter value for empid: 10232
Enter value for salary: 25000
old   1: insert into Employee values(&sno,'&name',&empid,&salary)
new   1: insert into Employee values(1,'monica',10232,25000)

1 row created.

SQL> /
Enter value for sno: 2
Enter value for name: vidhya
Enter value for empid: 10923
Enter value for salary: 29000
old   1: insert into Employee values(&sno,'&name',&empid,&salary)
new   1: insert into Employee values(2,'vidhya',10923,29000)

1 row created.

SQL> /
Enter value for sno: 3
Enter value for name: monisha
Enter value for empid: 10330
Enter value for salary: 50000
old   1: insert into Employee values(&sno,'&name',&empid,&salary)
new   1: insert into Employee values(3,'monisha',10330,50000)

1 row created.

SQL> /
Enter value for sno: 4
Enter value for name: cindhya
Enter value for empid: 10113
Enter value for salary: 45000
old   1: insert into Employee values(&sno,'&name',&empid,&salary)
new   1: insert into Employee values(4,'cindhya',10113,45000)

1 row created.

SQL> /
Enter value for sno: 5
Enter value for name: priya
```

Enter value for empid: 14234
Enter value for salary: 60000
old 1: insert into Employee values(&sno,'&name',&empid,&salary)
new 1: insert into Employee values(5,'priya',14234,60000)

1 row created.

To verify: SQL> select *from Employee;

SNO	NAME	EMPID	SALARY
1	monica	10232	25000
2	vidhya	10923	29000
3	monisha	10330	50000
4	cindhya	10113	45000
5	priya	14234	60000

8. Write a query to insert new row containing values for each column into the Employee table and verify it.

SQL> insert into Employee values(10,'monisha',12345,60000);

1 row created.

To verify: SQL> select *from Employee;

SNO	NAME	EMPID	SALARY
1	monica	10232	25000
2	vidhya	10923	29000
3	monisha	10330	50000
4	cindhya	10113	45000
5	priya	14234	60000
10	monisha	12345	60000

Updating Rows in a table:

9. Write a query to update sno column to 6 whose empid is 12345 and verify it.

SQL> update empdata set sno=6 where sno=12345;

1 row updated.

SQL> select *from Employee;

SNO	NAME	EMPID	SALARY
1	monica	10232	25000
2	vidhya	10923	29000
3	monisha	10330	50000
4	cindhya	10113	45000
5	priya	14234	60000
6	monisha	12345	60000

6 rows selected.

10. Write a query to update the "salary" column to 20000 whose empid is 14234 and verify it.

SQL> update Employee set salary=20000 where empid=14234;

To verify: SQL> select *from Employee;

SNO	NAME	EMPID	SALARY
1	monica	10232	25000
2	vidhya	10923	29000
3	monisha	10330	50000
4	cindhya	10113	45000
5	priya	14234	20000
6	monisha	12345	60000

Deleting rows from a table:

11. Write a query to delete the record from the "Employee" table whose empid is "12345" and verify it.

SQL> delete from Employee where empid=12345;

To verify: SQL> select *from Employee;

SNO	NAME	EMPID	SALARY
1	monica	10232	25000
2	vidhya	10923	29000
3	monisha	10330	50000
4	cindhya	10113	45000
5	priya	14234	20000

Exp.No: 3 **DATA CONTROL LANGUAGE (DCL)**

Date :

Aim: To write Data control language commands and verify the same

1. Write a query to end your current transaction and make permanent all changes performed in the transaction.

```
SQL> commit;
Commit complete.
```

2. Write a query to create a table goods with sno,itemcode,itemname,costnumber as its attributes and assign primary key constraint for the column "itemcode".

```
SQL> create table goods(sno number,itemcode number primary key,itemname varchar2(10),cost
number);

Table created.

SQL> insert into goods values(&sno,&itemcode,'&itemname',&cost);
Enter value for sno: 1
Enter value for itemcode: 1025
Enter value for itemname: dell moniters
Enter value for cost: 5000
old   1: insert into goods values(&sno,&itemcode,'&itemname',&cost)
new   1: insert into goods values(1,1025,'dell moniters',5000)
insert into goods values(1,1025,'dell moniters',5000)
            *
ERROR at line 1:
ORA-01401: inserted value too large for column

SQL> insert into goods values(&sno,&itemcode,'&itemname',&cost);
Enter value for sno: 1
Enter value for itemcode: 1025
Enter value for itemname: moniter
Enter value for cost: 5000
old   1: insert into goods values(&sno,&itemcode,'&itemname',&cost)
new   1: insert into goods values(1,1025,'moniter',5000)

1 row created.

SQL> /
Enter value for sno: 2
Enter value for itemcode: 1026
```

```
Enter value for itemname: mouse
Enter value for cost: 250
old   1: insert into goods values(&sno,&itemcode,'&itemname',&cost)
new   1: insert into goods values(1026,1026,'mouse',250)

1 row created.

SQL> /
Enter value for sno: 3
Enter value for itemcode: 1027
Enter value for itemname: RAM
Enter value for cost: 1500
old   1: insert into goods values(&sno,&itemcode,'&itemname',&cost)
new   1: insert into goods values(3,1027,'RAM',1500)

1 row created.

SQL> /
Enter value for sno: 4
Enter value for itemcode: 1028
Enter value for itemname: webcam
Enter value for cost: 350
old   1: insert into goods values(&sno,&itemcode,'&itemname',&cost)
new   1: insert into goods values(4,1028,'webcam',350)

1 row created.

SQL> /
Enter value for sno: 5
Enter value for itemcode: 1029
Enter value for itemname: pendrive
Enter value for cost: 500
old   1: insert into goods values(&sno,&itemcode,'&itemname',&cost)
new   1: insert into goods values(5,1029,'pendrive',500)

1 row created.
SQL> select *from goods;

       SNO     ITEMCODE      ITEMNAME        COST
----------     ---------- ----------     ----------
         1          1025           moniter          5000
         2          1026           mouse             250
         3          1027           RAM              1500
         4          1028           webcam            350
         5          1029            pendrive          500
```

SQL> commit;

Commit complete.

3. Write a query to add the record into the table "goods" and set the Savepoint S1, S2 and S3 and verify it.

SQL> insert into goods values(6,1030,'keyboard',500);
1 row created.

SQL> savepoint s1;

Savepoint created.

SQL> insert into goods values(7,1031,'DVD drive',2500);

1 row created.

SQL> savepoint s2;

Savepoint created.

SQL> insert into goods values(8,1032,'UPS',3000);

1 row created.

SQL> insert into goods values(9,1033,'CPU',5000);

1 row created.

SQL> savepoint s3;

Savepoint created.

4. Write a query to Rollback to Savepoint S3 and verify it.

SQL> rollback to savepoint s3;

Rollback complete.

To Verify: SQL> select *from goods;

SNO	ITEMCODE	ITEMNAME	COST

1	1025	moniter	5000
2	1026	mouse	250
3	1027	RAM	1500
4	1028	webcam	350
5	1029	pendrive	500
6	1030	keyboard	500
7	1031	DVD drive	2500
8	1032	UPS	3000
9	1033	CPU	5000

9 rows selected.

4. Write a query to Rollback to Savepoint S2 and verify it.

SQL> rollback to savepoint s2;

Rollback complete.

To Verify: SQL> select *from goods;

SNO	ITEMCODE	ITEMNAME	COST
1	1025	moniter	5000
2	1026	mouse	250
3	1027	RAM	1500
4	1028	webcam	350
5	1029	pendrive	500
6	1030	keyboard	500
7	1031	DVD drive	2500

7 rows selected.

5. Write a query to Rollback completely and verify it.

SQL> rollback;

Rollback complete.

To Verify: SQL> select *from goods;

SNO	ITEMCODE	ITEMNAME	COST
1	1025	moniter	5000
2	1026	mouse	250
3	1027	RAM	1500
4	1028	webcam	350
5	1029	pendrive	500

Exp.No: 4 (a)

Date :

DATA QUERY LANGUAGE (DQL)

Aim:

To write Data Query Language Commands and verify the same.

Selecting Rows and Columns:

1.Write a query to create a table Employee with the following list of attributes Sno, EName, EmpID, Salary and Designation.

```
SQL>  create table Employee(sno number,Ename varchar2(15),Empid number primary key,Salary
      Number,Designation varchar2(20));

Table created.

SQL> desc Employee;
 Name                      Null?      Type
 ------------------------  ---------- -------------------------
 SNO                                  NUMBER
 NAME                                 VARCHAR2(15)
 EMPID                     NOT NULL   NUMBER
 SALARY                               NUMBER
DESIGNATION                           VARCHAR2(15)


SQL> insert into Employee values(&Sno,'&Ename',&empid,&salary,'&designation');
Enter value for sno: 1
Enter value for name: Raja
Enter value for empid: 101
Enter value for salary: 12000
Enter value for desig: Manager
old   1: insert into employee values(&sno,'&name',&empid,&salary,'&desig')
new 1: insert into employee values(1,'Raja',101,12000,'Manager')

1 row created.

SQL> /
Enter value for sno: 2
Enter value for name: Guru
Enter value for empid: 102
Enter value for salary: 13000
```

```
Enter value for desig: Asst
old   1: insert into empl values(&sno,'&name',&empid,&salary,'&desig')
new   1: insert into empl values(2,'vp',102,13000,'asst')

1 row created.

SQL> /
Enter value for sno: 3
Enter value for name: yuva
Enter value for empid: 103
Enter value for salary: 14000
Enter value for desig: sen asst
old   1: insert into empl values(&sno,'&name',&empid,&salary,'&desig')
new   1: insert into empl values(3,'yuva',103,14000,'sen asst')

1 row created.

SQL> /
Enter value for sno: 4
Enter value for name: siva
Enter value for empid: 105
Enter value for salary: asst manager
Enter value for desig: asst mngr
old   1: insert into empl values(&sno,'&name',&empid,&salary,'&desig')
new   1: insert into empl values(4,'siva',105,asst manager,'asst mngr')
insert into empl values(4,'siva',105,asst manager,'asst mngr')
                        *
ERROR at line 1:
ORA-00917: missing comma


SQL> /
Enter value for sno: 4
Enter value for name: siva
Enter value for empid: 105
Enter value for salary: 25000
Enter value for desig: ass mngr
old   1: insert into empl values(&sno,'&name',&empid,&salary,'&desig')
new   1: insert into empl values(4,'siva',105,25000,'ass mngr')

1 row created.

SQL> /
Enter value for sno: 5
Enter value for name: sriram
Enter value for empid: 106
```

```
Enter value for salary: 23000
Enter value for desig: mngr
old   1: insert into empl values(&sno,'&name',&empid,&salary,'&desig')
new   1: insert into empl values(5,'sriram',106,23000,'mngr')

1 row created.


SQL> insert into employee values(&sno,'&name',&empid,&salary);
Enter value for sno: 1
Enter value for name: vidhya
Enter value for empid: 10125
Enter value for salary: 25000
old   1: insert into employee values(&sno,'&name',&empid,&salary)
new   1: insert into employee values(1,'vidhya',10125,25000)

1 row created.

SQL> /
Enter value for sno: 2
Enter value for name: deepthi
Enter value for empid: 10254
Enter value for salary: 12000
old   1: insert into employee values(&sno,'&name',&empid,&salary)
new   1: insert into employee values(2,'deepthi',10254,12000)

1 row created.

SQL> /
Enter value for sno: 3
Enter value for name: monisha
Enter value for empid: 10265
Enter value for salary: 50000
old   1: insert into employee values(&sno,'&name',&empid,&salary)
new   1: insert into employee values(3,'monisha',10265,50000)

1 row created.

SQL> /
Enter value for sno: 4
Enter value for name: ileana
Enter value for empid: 10101
Enter value for salary: 29000
old   1: insert into employee values(&sno,'&name',&empid,&salary)
new   1: insert into employee values(4,'ileana',10101,29000)
```

1 row created.

SQL> /
Enter value for sno: 5
Enter value for name: ria
Enter value for empid: 10252
Enter value for salary: 33000
old 1: insert into employee values(&sno,'&name',&empid,&salary)
new 1: insert into employee values(5,'ria',10252,33000)

1 row created.
SQL> select *from Employee;

SNO	ENAME	EMPID	SALARY
1	vidhya	10125	25000
2	deepthi	10254	12000
3	monisha	10265	50000
4	ileana	10101	29000
5	ria	10252	33000

1. Write a query to display the entire Employee ID and Name of the Employee from Employee table.

SQL> select Empid,EName from Employee;

ENAME	EMPID
vidhya	10125
deepthi	10254
monisha	10265
ileana	10101
ria	10252

Rows using Arithmetic Operators:

1. Write a query to calculate the salary increase of 1000 for all the employees and display a new salary + 1000 column in the output.

SQL> select sno,name,salary+1000,empid from empdata;

SNO	NAME	SALARY+1000	EMPID
1	monica	26000	10232
2	vidhya	30000	10923
3	monisha	51000	10330

4	cindhya	46000	10113
5	priya	61000	14234
6	monisha	61000	12345

6 rows selected.

Selecting Rows with Where Clause:

2. Write a query to retrieve Name,Empid and Salary for all employees whose esignation is "Manager".

SQL> select name,empid,salary from employee where designation='vp';

NAME	EMPID	SALARY
ram	a125	30000

Selecting Rows using Comparison Condition:

3.Write a query to retrieve the Name and Salary for all employees whose Salary is less than or equal to 15000.

SQL> select name,empid,salary from employee where salary<=1500000;

NAME	EMPID	SALARY
ragu	a123	100000
rama	a124	200000
ram	a125	30000

Selecting Rows using "BETWEEN" and "AND"

4.Write a query to retrieve Name and Salary of all employees whose salary is between 10000 and 150000.

SQL> select name,salary from employee where salary between 10000 and 150000;

NAME	SALARY
ragu	100000
ram	30000

Selecting Rows using "IN" condition

5. Write a query to retrieve the Empid, Name, and salary of all employees whose empid is a123 and a125.

SQL> select name,empid,salary from employee where empid in('a123','a125');

NAME	EMPID	SALARY
ragu	a123	100000
ram	a125	30000

Selecting Rows using "LIKE" keyword with % and _ symbols.

6. Write a query to retrieve name of all employees whose name begins with "r".

SQL> select name,salary,designation from employee where name like'r%';

NAME	SALARY	DESIGNATION
ragu	100000	chairman
rama	200000	ceo
ram	30000	vp

7. Write a query to retrieve name of all employees whose second letter of name is "a".

SQL> select name from employee where name like '_a%';

NAME
ragu
rama
ram

8.**Write a query to retrieve name of all employees who have "a" and "u" letters in their name.**

SQL> select name from employee where name like '%a%u%';

NAME
Ragu

Selecting Rows with Logical Conditions "AND,OR and NOT" operator:

9.Write a query to retrieve the Name and Empid , Salary and Designation of all employees who have a Designation that contains the string "cha" and earns Rs.100000 or more.

SQL> select name,empid,salary,designation from employee where salary >=100000 and designation like 'cha%';

NAME	EMPID	SALARY	DESIGNATION
ragu	a123	100000	chairman

10. Write a query to retrieve the Name and Empid , Salary and Designation of all employees whose Designation is "NOT IN" "Chairman and VP".

SQL> select name,empid,salary,designation from employee where designation not in 'chairman';

NAME	EMPID	SALARY	DESIGNATION
rama	a124	200000	ceo
ram	a125	30000	vp

11. Write a query to retrieve the Name and Empid , Salary and Designation of all employees whose salary is "NOT BETWEEN" "Rs.200000 and Rs.300000".

SQL> select name,empid,salary,designation from employee where salary not between 200000 and 300000;

NAME	EMPID	SALARY	DESIGNATION
ragu	a123	100000	chairman
ram	a125	30000	vp

12.Write a query to retrieve Name,Empid, Salary and Designation of all employees whose Name doesn't contain "m".

SQL> select name,empid,salary,designation from employee where name not like '%m%';

NAME	EMPID	SALARY	DESIGNATION
ragu	a123	100000	chairman

13.**Write a query to retrieve Name,Empid, Salary and Designation of all employees if an employee is 'VP' and earns more than 10000 OR if the employee is 'CEO'.**

SQL> select name,empid,salary,designation from employee where designation='vp' or designation='ceo' and salary>10000;

NAME	EMPID	SALARY	DESIGNATION

rama a124 200000 ceo
ram a125 30000 vp

Selecting Rows using "ORDER BY" Clause with "ASC" and "DESC" keyword:

14.Write a query to retrieve Name,Empid, Salary and Designation of all employees and sort the result by "Name".

SQL> select name,empid,salary,designation from employee order by name;

NAME	EMPID	SALARY	DESIGNATION
dinesh	c128	10000	manager
ragu	a123	100000	chairman
ram	a125	30000	vp
rama	a124	200000	ceo
sag	b128	10000	manager

15.Write a query to retrieve Name,Empid, Salary and Annual Salary of all employees and sort the result by Annual Salary.

SQL> select name,empid,salary,salary*12 as "AnnualSalary" from employee order by salary*12;

NAME	EMPID	SALARY	AnnualSalary
sag	b128	10000	120000
dinesh	c128	10000	120000
ram	a125	30000	360000
ragu	a123	100000	1200000
rama	a124	200000	2400000

16. Write a query to add an attribute hiredate to the employee table.

SQL> alter table employee add(hiredate date);

Table altered.
SQL> select * from employee;

SLNO	NAME	EMPID	SALARY	DESIGNATION	HIREDATE
1	ragu	a123	100000	chairman	
2	rama	a124	200000	ceo	
3	ram	a125	30000	vp	
4	sag	b128	10000	manager	
5	dinesh	c128	10000	manager	

17.Write a query to update the value for hiredate attribute of the employee table;

SQL> update employee set hiredate='10-May-10' where slno=1;

1 row updated.

SQL> update employee set hiredate='10-May-11' where slno=2;

1 row updated.

SQL> update employee set hiredate='10-June-11' where slno=3;

1 row updated.

SQL> update employee set hiredate='12-June-12' where slno=4;

1 row updated.

SQL> update employee set hiredate='14-June-10' where slno=5;

1 row updated.

SQL> select * from employee;

SLNO	NAME	EMPID	SALARY	DESIGNATION	HIREDATE
6	abc	d124	200000	admin	15-MAR-08
1	ragu	a123	100000	chairman	10-MAY-10
2	rama	a124	200000	ceo	10-MAY-11
3	ram	a125	30000	vp	10-JUN-11
4	sag	b128	10000	manager	12-JUN-12
5	dinesh	c128	10000	manager	14-JUN-10

6 rows selected.

18. Write a query to retrieve the name,designation,and hiredate of an employee hired between 10 May 2010 and 12 May 2012.

SQL> select ename,designation,hiredate from employee where hiredate between '10-May-10' and '12-Jun-12';

NAME	DESIGNATION	HIREDATE
ragu	chairman	10-MAY-10
rama	ceo	10-MAY-11
ram	vp	10-JUN-11
sag	manager	12-JUN-12
dinesh	manager	14-JUN-10

19 .Write a query to retrieve the name and empid of all employees with empid a125, b128, c128 in alphabetical order by name.

SQL> select name,empid from employee where empid in('a125','b128','c128') order by name asc;

NAME	EMPID
dinesh	c128
ram	a125
sag	b128

20. Write a query to retrieve the name and Hire date of every employee who was hired in 2010.

SQL> select name,hiredate from employee where hiredate like'%10%';

NAME	HIREDATE
ragu	10-MAY-10
rama	10-MAY-11
ram	10-JUN-11
dinesh	14-JUN-10

Ex.No:4(b) **DATA QUERY LANGUAGE(SINGLE ROW FUNCTIONS)**

Date:

Aim:

To write DQL Commands for single row functions and verify the same.

1. **Write a query to display the Name and Designation of all employees in the following format.**

EMPLOYEE DETAILS
DESIGNATION OF "KUMAR" IS MANAGER

SQL> select 'The designation for '||upper(name)||' is '||designation as "Employee Details" from employee;

```
Employee Details
----------------------------------------------------------------
The designation for ABC is admin
The designation for RAGU is chairman
The designation for RAMA is ceo
The designation for RAM is vp
The designation for SAG is Manager
The designation for DINESH is manager

6 rows selected.
```

2. **Write a query to display Name,Empid and Designation of all employees whose Designation starts with Capital letter following small letters.**

SQL> select name,empid,designation from employee where designation=initcap(designation);

```
NAME                 EMPID                DESIGNATION
-------------------- -------------------- --------------------
sag                   b128                 Manager
```

Selecting Rows using Character Manipulation Functions:

1. **Write a query to display Name,Empid and Designation of all employees who have the string "ager" contained in the designation starting at the 4th position of designation.**

SQL> select name,empid,salary from employee where substr(designation,4)='ager';

NAME	EMPID	SALARY
sag	b128	10000
dinesh	c128	10000

2. **Write a query to display the empid,name and the numeric position of the letter ‘a’ in the name for all the employees whose last names end with an ‘h’.**

SQL> select id,name,instr(name,'a') as "Contains 'a'" from employee where substr(name,-1)=’u';

EMPID	NAME	Contains 'a'
a123	ragu	1

3. **Write a query to display the salary value as right justified for a length 10 of all employees.**

SQL> select RPAD(salary,10,'*') from employee;

RPAD(SALARY)

10000*****
20000*****

4. Write a query to display the salary value as left justified from a length 10 of all employees.

SQL> select lpad(salary,10,'*') from employee;

LPAD(SALARY)

*****10000
*****20000

5.Write a query to trim and display the letter ‘H’ from the ‘Hello World’.

SQL> select trim('h' from 'hello world') from dual;

TRIM('H'FR
---

ello world

Selecting Rows using Number Functions from Dual table:

(ROUND, TRUNCATE and MOD)

1.Write a query to round the value 45.923 into 45.92,46 and 50.

```
SQL> select round(45.923,2),round(45.923,0),round(45.923,-1) from dual;

ROUND(45.923,2) ROUND(45.923,0) ROUND(45.923,-1)
--------------- --------------- ----------------
          45.92              46               50
```

2. Write a query to truncate the value 45.923 into 45.92,45 and 0.

```
SQL> select name,salary,mod(salary,5000) from employee where designation='manager'

NAME                 SALARY     MOD(SALARY,5000)
-------------------- ---------- ----------------
sag                       10000                0
dinesh                    10000                0
```

Selecting Rows with Date Functions;

MONTHS_BETWEEN,ADD_MONTHS,NEXT_DAY,LAST_DAY,ROUND,TRUNCATE

1. Write a query to display system date, label the column "System Date".

```
SQL> select sysdate as "system date" from dual;

system date
-----------
10-JAN-13
```

2.Write a query to display the name and number of weeks employed for all employees who are managers. Label the column "Weeks".

```
SQL> select id,hiredate,months_between(sysdate,hiredate) as "tenure" from employee;

        ID HIREDATE      tenure
---------- ---------  ----------
```

```
1   01-OCT-11     15.3108875
2   10-SEP-10       28
3    12-JUN-09    42.9560488
4   20-MAR-10    33.6979842
```

3.Write a query to display the employee id,hiredate and first Friday after hire date,of all employees fewer than 20 months.

```
SQL> select id,hiredate,next_day(hiredate,'friday') from employee where
months_between(sysdate,hiredate)<20;

        ID  HIREDATE    NEXT_DAY
---------- --------- ---------------
         1  01-OCT-11    07-OCT-11
```

4. Write a query to display the employee id,hiredate and last day of the hire month for all employees fewer than 36 months.

```
SQL> select empid,hiredate,last_day(hiredate) from employee where months_between
(sysdate,hiredate)<36;

EMPID          HIREDATE      LAST_DAY(10-MAY-10)
---------------- ---------------      -----------------
a123             10-MAY-10      31-MAY-10
a124             10-MAY-1       30-JUN-11
a128             12-JUN-12      30-JUN-12
c128             14-JUN-10      30-JUN-10
d124             15-MAR-18      31-MAR-18

6 rows selected.
```

Selecting Rows using Data Conversion Functions:
TO_CHAR,TO_NUMBER,TO_DATE

1.Write a query to display the system date and date in the following format: 'DD/MON/YYYY'. Label the column DATE.

```
SQL> select sysdate,to_char(sysdate,'DD/MON/YYYY') as "DATE" from dual;

SYSDATE       DATE
---------------- -----------
17-JAN-13   17/JAN/2013
```

2.Write a query to display the system date and year to be spelled out. Label the column Year.

```
SQL> select sysdate,to_char(sysdate,'year') as "YEAR" from dual;
```

```
SYSDATE        YEAR
---------      ------------
17-JAN-13      twenty thirteen
```

3.Write a query to display the system date,full name of the month,three-letter abbreviation of the day of week.Label the columns Month and Day.

```
SQL> select sysdate,to_char(sysdate,'month') as "Month",to_char(sysdate,'DY') from dual;

SYSDATE   Month     TO_
---------- ---------    -----
17-JAN-13   january    THU
```

4.Write a query to display the employee id,hiredate,and month on which the employee started. Label the column month_hired of all employees whose name is 'ram'.

```
SQL> select empid,hiredate,to_char(hiredate,'DaY') as "Hired month" from employee where
        name='ram';

EMPID          HIREDATE   Hired mon
-------------      --------        ---------
a125             10-JUN-11      Friday
```

5.Write a query to display the name,hiredate,and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week starting with Monday.

```
SQL> select name,hiredate,to_char(hiredate,'DAY') as "DAY" from employee order by DAY;

NAME                HIREDATE       DAY
-------------------- ---------------       ---------
ram                  10-JUN-11          FRIDAY
dinesh               14-JUN-10          MONDAY
ragu                 10-MAY-10          MONDAY
abc                 15-MAR-18          THURSDAY
rama                 10-MAY-11          TUESDAY
sag                  12-JUN-12          TUESDAY

6 rows selected.
```

6.Write a query to display the salary of all employees in the following format: Rs.6,000.00

```
SQL> select to_char(salary,'$9,99,999.00') salary from employee;

SALARY
```

```
-------------
 $1,00,000.00
 $2,00,000.00
   $30,000.00
   $10,000.00
   $10,000.00
   $2,00,000.00

6 rows selected.
```

7.Write a query to convert the character string '01/jan/2008' to a date format.

```
SQL> select to_date ('01/jan/2008','dd-mon-yy') from dual;

TO_DATE('
---------
01-JAN-08
```

8. Write a query to display name and employees hired prior to 1999.
Hint: Use the RR format in TO_DATE function.

```
SQL> select name,to_char(hiredate,'DD-MON-YYYY') from employee where hiredate<(t
o_date('01-Jan-11','DD-MON-YY'));

NAME                 TO_CHAR(HIR
-------------------- ------------------
ragu                 10-MAY-2010
dinesh               14-JUN-20106 rows
```

Ex.No: 5 **GROUP FUNCTIONS**

Date:

Aim:

To write queries for group functions using 'group by' and 'having' clause and verify the same.

1.Write a query to create a table employee with attributes sno,ename,deptid,eid,salary and designation. Assign the constraint primary key for the attribute eid and assign not null constraints for the remaining attributes.

SQL> create table employee(sno number,ename varchar2(20) NOT NULL,deptid varchar2(30) NOT NULL,eid varchar2(10) PRIMARY KEY,salary number NOT NULL,designation varchar2(20) NOT NULL);

To verify: desc employee;

Name	Null?	Type
SNO		NUMBER
ENAME	NOT NULL	VARCHAR2(20)
DEPTID	NOT NULL	VARCHAR2(30)
EID	NOT NULL	VARCHAR2(10)
SALARY	NOT NULL	NUMBER
DESIGNATION	NOT NULL	VARCHAR2(20)

Selecting rows using Group Functions:

2.Write a query to display the sum and average of salary for all employees and verify it. Label the column as "Sum" and "Average".

SQL> select sum(salary),avg(salary) from employee;

SUM(SALARY)	AVG(SALARY)
20000	10000

3. Write a query to display the sum and average of salary of all "managers" and verify it. Label the column as "Sum" and "Average".

SQL> select sum(salary),avg(salary) from employee where designation like '%man%';

```
SUM(SALARY)     AVG(SALARY)
-----------     -----------
     20000           10000
```

4. **Write a query to display highest and lowest salary for all managers and verify it.**

```
SQL> select MAX (salary),min(salary) from employee where designation='manager';


MAX(SALARY)         MIN(SALARY)
-----------         -----------
     50000           10000
```

5. Write a query to display the name of the employee in an alphabetized list of all employees and verify it.

```
SQL> select min(ename),max(ename) from employee;

MIN(ENAME)          MAX(ENAME)
--------------------  --------------------
A                     C
```

6. Write a query to display the number of employees in the department CSE and verify it.

```
SQL> select count(*) from employee where deptid like '%CS%';

  COUNT(*)
----------
         3
```

<u>Selecting Rows using Group By Clause:</u>

7. Write a query to display the department number and the average salary for each department and verify it.

```
SQL> select deptid,avg(salary) from employee group by deptid;

DEPTID                         AVG(SALARY)
------------------------------ -----------
CS001                            30000
CS002                            10000
```

8. Write a query to display the number of employees working for each department and verify it.

```
SQL> select deptid,count(eid)from employee group by deptid;
```

DEPTID COUNT(EID)
---------- ----------
CS001 2
CS002 1

9. Write a query to display deptid,designation,and sum of salary for all employees in each department, designation and verify it.

SQL> select deptid,designation,sum(salary) from employee group by deptid,designation;

DEPTID DESIGNATION SUM(SALARY)
---------- ---------- ----------
CS001 manager 60000
CS002 manager 13000
CS002 manufacuter 10000

10. Write a query to display the average salaries of those departments that have an average salary greater than Rs.8000/- and verify it.

SQL> select deptid,avg(salary)from employee having avg(salary)>8000 group by deptid;

DEPTID AVG(SALARY)
---------- ----------
CS001 30000

11. Write a query to display the designation and total monthy salary for each designation with a total payroll exceeding Rs 10000. Sort the list by the total monthly salary and verify it. Label the column as "Designation" and "Payroll".

SQL> select designation as "Designation",sum(salary) as "payroll" from employee having sum(salary)>1
0000 group by designation order by sum(salary);

Designation payroll
---------- ----------
manager 73000

<u>Selecting Rows using Having Clause: [Excluding the Group Results]</u>

12. Write a query to display the department numbers and maximum salaries for those departments whose maximum salary is greater than Rs.10000 and verify it.

SQL> select deptid,max(salary)from employee group by deptid having max(salary)>10000;

DEPTID MAX(SALARY)
---------- ----------

CS001 50000
CS002 60000

13. Write a query to display the designation and total monthly salary for each designation with a total payroll exceeding Rs 13000. The result should exclude manufactures and sort the list by the total monthly salary and verify it. Label the column as "Designation" and "Payroll".

SQL> select designation as "Designation",sum(salary) as "Payroll" from employee where designation NO
T IN('manufacturer') group by designation having sum(salary)>13000 order by sum(salary);

Designation	Payroll
programmer	60000
manager	66000

<u>Selecting Rows using Group Functions:</u>[Nesting Group Functions]

14. Write a query to display the maximum average salary and verify it. Label the column as "Maximum Average Salary" in each department.

SQL> select max(avg(salary)) as "Maximum Average Salary" from employee group by deptid;

Maximum Average Salary
35000

Ex.No: 6

Date :

NESTED QUERIES (SUB QUERIES)

Aim:

To write DQL Commands for nested queries and verify the same.

1. Write a query to display the name of all employees who earn more than C's Salary and verify it.

```
SQL> select ename from employee where salary>(select salary from employee where
ename='C');

ENAME
--------------------
D
```

2. Write a query to display the name and salary of all employees who are in the same department as 'C' and verify it.

```
SQL> select ename,salary from employee where deptid=(select deptid from employee
        where ename='C');

ENAME                SALARY
-------------------- ----------
A                     10000
C                     50000
E                     6000
```

3. Write a query to display the name and designation of all employees whose designation is the same as that of employee 002

```
SQL> select ename,designation from employee where designation=(select designation
        from employee where eid=002);

ENAME                DESIGNATION
-------------------- --------------------
B                    manufacuter
```

4. Write a query to display the name and designation of all the employees whose designation is the same as that of employee a124 and whose salary is greater than that of employee a125.

```
SQL> select name,designation from employee where designation=(select designation
 from employee where empid='a124') and salary>(select salary from employee where
 empid='a125');

NAME                 DESIGNATION
-------------------- --------------------
rama                 ceo
```

Selecting Rows using Group Functions in Subquery

4. **Write a query to display the name,empid,and salary of all employees whose salary is equal to the minimum salary.**

```
SQL> select name,empid,salary from employee where salary=(select min(salary) from
        employee);

NAME                 EMPID                SALARY
-------------------- -------------------- ----------
sag                  a128                  10000
dinesh               c128                  10000
```

5. **Write a query to display the name,empid,and salary of all employees who earns more than average salary. Sort the result in ascending order of salary.**

```
SQL> select name,empid,salary from employee where salary>(select avg(salary) fro
m employee) order by salary;

NAME                 EMPID                SALARY
-------------------- -------------------- ----------
ragu                 a123                 100000
abc                  d124                 200000
rama                 a124                 200000
```

Selecting Rows using Having Clause in Subquery.

6. **Write a query to display all the designation which have a lowest average salary and verify it.**

```
SQL> select designation,avg(salary) from employee group by designation having av
g(salary)=(select min(avg(salary)) from employee group by designation);

DESIGNATION          AVG(SALARY)
-------------------- -----------
manager               10000
```

Selecting Rows using Multiple Row Subqueries:

IN,<ANY(less than the minimum),>ANY(more than the minimum),=ANY(equivalent to IN),<ALL(less than the minimum),>ALL(more than the minimum),=ALL(equivalent to IN)

7. **Write a query to display the name,salary,empid of all the employees who earns the same salary as the minimum salary for each designation and verify it.**

   SQL> select name,salary,empid from employee where salary IN(select MIN(salary) from employee group by designation);

NAME	SALARY	EMPID
ragu	100000	a123
abc	200000	d124
rama	200000	a124
ram	30000	a125
dinesh	10000	c128
sag	10000	a128

   6 rows selected.

8. **Write a query to display name,empid,salary of all employees who earns the same salary as the average salary for each department and verify it.**

   SQL> select name,empid,salary from employee where salary IN(select avg(salary) from employee group by depid);

NAME	EMPID	SALARY
abc	d124	200000
rama	a124	200000
ragu	a123	200000

9. **Write a query to display the empid,name,designation and salary of all employees who are not managers and whose salary is less than that of any manager's maximum salary and verify it. (use 'any' keyword).**

   SQL> select empid,name,designation,salary from employee where salary<any(select salary from employee where designation='manager') and designation<>'manager';

EMPID	NAME	DESIGNATION	SALARY
d124	abc	admin	5000

10. **Write a query to display the empid,name,designation and salary of all employees who are not managers and whose salary is less than that of the manager's maximum salary and verify it. (use 'all' keyword).**

SQL> select empid,name,designation,salary from employee where salary<all(select salary from employee where designation='manager') and designation<>'manager';

EMPID	NAME	DESIGNATION	SALARY
d124	abc	admin	5000

Ex.No: 7 **JOIN QUERIES**

Date :

Aim:

To write DQL Commands to join, Restrict and Retrieve information from one or more tables execute it and verify the same.

Selecting Rows with Equijoin using table Aliases

1. Write a query to display empid,name,deptid,deptname and location for all employees and verify it.

```
SQL> select employee.empid,employee.name,employee.depid,department.deptname,depa
rtment.location from employee,department where employee.depid=department.depid;

EMPID       NAME        DEPID           DEPTNAME                 LOCATION
----------- ----------- ------------------ -------------------- --------------------
a123          ragu        CS000          COMPUTER_SCIENCE         CHENNAI
a124          rama        CS000          COMPUTER_SCIENCE         CHENNAI
a125          ram         EE000          ELECT_ELECTRO            MUMBAI
a128          sag         EE000          ELECT_ELECTRO            MUMBAI
c128          dinesh      EC000          ELECT_COMM               DELHI
d124          abc         EC000          ELECT_COMM               DELHI

6 rows selected.
```

2. Write a query to display the “dinesh” depid and deptname and verify it.

```
SQL> select e.empid,e.name,e.depid,d.deptname from employee e,department d where
 e.depid=d.depid and e.name='dinesh';

EMPID            NAME             DEPID         DEPTNAME
---------------- ---------------- ------------  ----------------
c128             dinesh           EC000         ELECT_COMM
```

Selecting Rows with Non-Equijoin using table Aliases:
[Other Conditions such as >=, <= and BETWEEN, AND]

1. Write a query to display the name,salary and deptname of all employees whose salary is greater than 10000 and verify it.

SQL> select e.name,e.salary,d.deptname from employee e,department d where e.salary>10000 and e.depid=d.depid;

NAME	SALARY	DEPTNAME
ragu	200000	COMPUTER_SCIENCE
rama	200000	COMPUTER_SCIENCE
ram	30000	ELECT_ELECTRO

Selecting Rows using Outer Joins:[Left Outerjoin,Right Outerjoin using "+" symbol]

1. Write a query to display the name, depid and deptname of all employees.Make sure that employees without department are included as well and verify it.

SQL> select e.name,e.depid,d.deptname from employee e,department d where e.depid =d.depid(+);

NAME	DEPID	DEPTNAME
rama	CS000	COMPUTER_SCIENCE
ragu	CS000	COMPUTER_SCIENCE
sag	EE000	ELECT_ELECTRO
ram	EE000	ELECT_ELECTRO
abc	EC000	ELECT_COMM
dinesh	EC000	ELECT_COMM
www		

7 rows selected.

2. Write a query to display the name, salary,depid and deptname of all employees. Make sure that departments without employees are included as well and verify.

SQL> select e.name,e.salary,e.depid,d.deptname from employee e,department d wher e e.depid(+)=d.depid;

NAME	SALARY	DEPID	DEPTNAME
ragu	200000	CS000	COMPUTER_SCIENCE
rama	200000	CS000	COMPUTER_SCIENCE
ram	30000	EE000	ELECT_ELECTRO
sag	10000	EE000	ELECT_ELECTRO
dinesh	10000	EC000	ELECT_COMM
abc	5000	EC000	ELECT_COMM
			MECHANICAL
			CHEMICAL

8 rows selected.

Selecting Rows using Self Joins:

1. **Write a query to find and display the name of each employee's deptname and verify it.**

Employee Details
works in ELECT_ELCTRO Dept

SQL> select e.name||' Works In '||d.deptname as "Employee Details" from employee e,department d where e.depid=d.depid;

Employee Details
---
ragu Works In COMPUTER_SCIENCE
rama Works In COMPUTER_SCIENCE
ram Works In ELECT_ELECTRO
sag Works In ELECT_ELECTRO
dinesh Works In ELECT_COMM
abc Works In ELECT_COMM

6 rows selected.

Selecting Rows using Cross Join: [like Cartesian Product of 2 Tables]

1. **Write a query to display the name and department name of all employees and verify it.[NOTE: Use CROSS JOIN]**

SQL> select name,deptname from employee cross join department;

NAME	DEPTNAME
www	COMPUTER_SCIENCE
ragu	COMPUTER_SCIENCE
rama	COMPUTER_SCIENCE
ram	COMPUTER_SCIENCE
sag	COMPUTER_SCIENCE
dinesh	COMPUTER_SCIENCE
abc	COMPUTER_SCIENCE
www	ELECT_ELECTRO
ragu	ELECT_ELECTRO
rama	ELECT_ELECTRO
ram	ELECT_ELECTRO

NAME DEPTNAME
---------- ----------
sag ELECT_ELECTRO
dinesh ELECT_ELECTRO
abc ELECT_ELECTRO
www ELECT_COMM
ragu ELECT_COMM
rama ELECT_COMM
ram ELECT_COMM
sag ELECT_COMM
dinesh ELECT_COMM
abc ELECT_COMM
www CHEMICAL

NAME DEPTNAME
---------- ----------
ragu CHEMICAL
rama CHEMICAL
ram CHEMICAL
sag CHEMICAL
dinesh CHEMICAL
abc CHEMICAL
www MECHANICAL
ragu MECHANICAL
rama MECHANICAL
ram MECHANICAL
sag MECHANICAL

NAME DEPTNAME
---------- ----------
dinesh MECHANICAL
abc MECHANICAL

35 rows selected.

Selecting Rows Using Natural or Inner Join:[like equi-join]

1. Write a query to display the empid,name,depid,deptname and location and verify it.[NOTE: Use Natural join]

SQL> select empid,name,depid,deptname from employee natural join department;

EMPID NAME DEPID DEPTNAME

a123	ragu	CS000	COMPUTER_SCIENCE
a124	rama	CS000	COMPUTER_SCIENCE
a125	ram	EE000	ELECT_ELECTRO
a128	sag	EE000	ELECT_ELECTRO
c128	dinesh	EC000	ELECT_COMM
d124	abc	EC000	ELECT_COMM

6 rows selected.

2. Write a query to display the empid,name,depid,deptname and location. Limit the row output to those with a empid equal to a124 or a125 and verify it.

```
SQL> select empid,name,depid,deptname,location from employee natural join depart
ment where empid IN('a124','a125');
```

EMPID	NAME	DEPID	DEPTNAME	LOCATION
a124	rama	CS000	COMPUTER_SCIENCE	CHENNAI
a125	ram	EE000	ELECT_ELECTRO	MUMBAI

Selecting Rows using Join with USING clause: [like Non-equi-join]

1. Write a query to display the empid,name and location of all employees and verify it. [NOTE: Use JOIN with USING clause]

```
SQL> select e.empid,e.name,d.location from employee e join department d using (d
epid);
```

EMPID	NAME	LOCATION
a123	ragu	CHENNAI
a124	rama	CHENNAI
a125	ram	MUMBAI
a128	sag	MUMBAI
c128	dinesh	DELHI
d124	abc	DELHI

6 rows selected.

Selecting Rows using Join with ON clause: [like self-join]

1. Write a query to display the empid,name,depid,deptname and location of all employees and verify it. [NOTE: Use JOIN with ON clause]

SQL> select e.empid,e.name,e.depid,d.deptname,d.location from employee e join department d on (e.depid=d.depid);

EMPID	NAME	DEPID	DEPTNAME	LOCATION
a123	ragu	CS000	COMPUTER_SCIENCE	CHENNAI
a124	rama	CS000	COMPUTER_SCIENCE	CHENNAI
a125	ram	EE000	ELECT_ELECTRO	MUMBAI
a128	sag	EE000	ELECT_ELECTRO	MUMBAI
c128	dinesh	EC000	ELECT_COMM	DELHI
d124	abc	EC000	ELECT_COMM	DELHI

6 rows selected.

2. **Write a query to display the employee id,name,depid,deptname and location of all customers where deptid is CS000 and verify it. [NOTE: Use JOIN with ON clause]**

SQL> select e.empid,e.name,d.depid,d.deptname,d.location from employee e join de partment d on(e.depid=d.depid) where e.depid='CS000';

EMPID	NAME	DEPID	DEPTNAME	LOCATION
a123	ragu	CS000	COMPUTER_SCIENCE	CHENNAI
a124	rama	CS000	COMPUTER_SCIENCE	CHENNAI

Selecting Rows using Outer Join with ON Clause: [LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN]

1. Write a query to display the name,depid and department name of all employees. Make sure that employees without department are included as well and justify it.[NOTE: Use LEFT OUTER JOIN with ON clause].

SQL> select e.name,e.depid,d.deptname from employee e left outer join department d on(e.depid=d.depid);

NAME	DEPID	DEPTNAME
rama	CS000	COMPUTER_SCIENCE

```
ragu            CS000           COMPUTER_SCIENCE
sag             EE000           ELECT_ELECTRO
ram             EE000           ELECT_ELECTRO
abc             EC000           ELECT_COMM
dinesh          EC000           ELECT_COMM
www

7       rows selected.
```

2. Write a query to display the employee name,depid, deptname of all employees. Make sure that departments without employees are included as well and verify it. [NOTE: Use RIGHT OUTER JOIN with ON Clause]

```
SQL> select e.name,d.depid,d.deptname from employee e right outer join departmen
t d ON(e.depid=d.depid);

NAME                DEPID           DEPTNAME
------------------- --------------- -------------------
ragu                CS000           COMPUTER_SCIENCE
rama                CS000           COMPUTER_SCIENCE
ram                 EE000           ELECT_ELECTRO
sag                 EE000           ELECT_ELECTRO
dinesh              EC000           ELECT_COMM
abc                 EC000           ELECT_COMM
                    MECH000         MECHANICAL
                    MA001           CHEMICAL

8 rows selected.
```

Ex.No: 8 **VIEWS**

Date :

AIM:

To write a DDL command to create views to restrict data access from one or more tables, execute it and verify the same.

Creating views:

1. **Create a view that contains employee id as "ID_NUMBER", employee name as "NAME" and salary as "SALARY" for each employee in department 90.**

SQL> create view vw_emp80(id,name,salary) as select empid,name,salary from employee where deptid=90;

OUTPUT:

View created.

SQL> select * from vw_emp80;

ID	NAME	SALARY
s203	vidhya	29000

2. **Create a view vw_sal that contains employee id as "ID" , employee name as "NAME" and annual salary as "ANNUAL_SAL" for each employee in the department 100.**

SQL> create view vw_sal(id,name,"annual_sal") as select empid,name,salary*12
from employee where deptid=100;

View created.

To Verify: SQL> select * from vw_emp80;

ID	NAME	SALARY
s203	vidhya	29000

Selecting views with group functions:

3. **To create a view vw_dept_sal that contains department name as "DEPNAME" , minimum salary as "MIN_SAL" , maximum salary as "MAX_SAL"**

```
SQL> create or replace view vw_dept_sal(depname,min_sal,max_sal) as select
       deptname,min(employee.salary),max(employee.salary) from employee, dep
        where employee.deptid=dep.deptid group by deptname;

        View created
```

To Verify: SQL> select * from vw_dept_sal;

DEPNA	MIN_SAL	MAX_SAL
cse	25000	25000
ece	29000	29000
eee	45000	45000
it	50000	50000
mech	49000	49000

Selecting views with check option:

4. **To create a view vw_emp12 that contains all the details of employees whose department no is 102 and does not allow the department no for those employees to be changed through views.**

SYNTAX:

```
SQL> create or replace view vw_emp12 as select * from employee where deptid=102 with
       check option constraint vw_emp12_ck;

View created.

SQL> select * from vw_emp12;
```

SNO	NAME	SALARY	DESIG	LAST	EMPID	DEPTID	HIRE_DATE
1	monica	25000	rep	a	a102	102	13-MAR-08

5. **To create a view to set the salary Rs.15000/- to the employee id 'a102' in the vw_dept_salary and verify the result in vw_emp12_sal and employees table.**

SQL> update vw_emp12 set salary=15000 where empid='a102';

1 row updated.

To Verify: SQL> select * from vw_emp12;

SNO	NAME	SALARY	DESIG	LAST	EMPID	DEPTID	HIRE_DATE
1	monica	15000	rep	a	a102	102	13-MAR-08

6. **To create a view vw_emp10 that contains employee id as "EMPLOYEE ID", employee name as "EMPLOYEE NAME" and deptno as "DEPTNO" of all employees whose deptno is 10 and does not allow inserting, deleting or modifying a row through view .**

STNTAX:

SQL> create or replace view vw_emp10(id,name,deptid) as select empno, ename,deptno from emp where deptno=10 with read only;

OUTPUT:

View created.

SQL> select * from vw_emp10;

ID	NAME	DEPTID
7782	CLARK	10
7839	KING	10
7934	MILLER	10

7. **To display the top three carner names and salaries of employees. Label the rownum as "RANK".**

SQL> select rownum as rank,emp.ename,emp.sal from(select ename,sal from emp order by sal desc)emp where rownum<=3;

RANK	ENAME	SAL
1	KING	5000

	2	SCOTT	3000
	3	FORD	3000

Ex.No: 9 (a) **PL / SQL PROCEDURES**

Date :

1. **Write a PL/SQL block to assign the salary of an employee whose employee id is "CS01" to bind the variable "gsal" and display the value of the bind variable using print command.**

PROCEDURE:

```
declare
        gsal number(6);
begin
        select salary into gsal from employee where eid='cs06';
        dbms_output.put_line(gsal);
        dbms_output.put_line('The salary is assigned to gsal');
end;
/
```

OUTPUT:

```
SQL> select *from employee;

EID     NAME       SALARY      DESIGNATIO
-------  ----------  ----------  ----------
cs06     aaa         2666        clerk
cs02     bbb         3597        Technician
cs03     ccc         3597        manager

SQL>  /
2666
The salary is assigned to gsal

PL/SQL procedure successfully completed.
```

2. **Write a PL/SQL block to calculate the monthly salary by getting the annual salary at run time.**

PROCEDURE:

```
declare
annual_sal number(6);
```

```
        monthly_sal number(6);
begin
        monthly_sal:=&annual_sal/12;
        dbms_output.put_line('The monthly salary is:'||monthly_sal);
        end;
        /
```

OUTPUT:

```
 Enter value for annual_sal: 10000
old   5: monthly_sal:=&annual_sal/12;
new   5: monthly_sal:=10000/12;

PL/SQL procedure successfully completed.
```

3. Write a PL/SQL block to update the salary for the employee "c201" and display it.

PROCEDURE:

```
declare
        e_sal employee.salary%type:=500;
        sal number(6);
begin
        update employee set salary=salary+e_sal where eid='cs06';
        select salary into sal from employee where eid='cs06';
        dbms_output.put_line('the salary is updated and the updated value is'||sal);
          end;
           /
```

OUTPUT:

```
SQL> /
PL/SQL procedure successfully completed.
The salary is updated and the updated value is 3166
```

4. Write a PL/SQL block to retrieve the hired date and employee id for the employee "cs06" and verify it.

PROCEDURE:

```
declare
        e_start_date emp1.start_date%TYPE;
        e_id emp1.emp_id%type;
 begin
        select start_date,emp_id into e_start_date,e_id from emp1 where
```

```
                emp_id='cs06';
                 dbms_output.put_line('the hire date is'||e_start_date);
                 dbms_output.put_line('the empid is'||e_id);
end;
/
```

5.Write a PL/SQL block to update the employee id of the employee whose name is "aaa" using control structure.

PROCEDURE:

```
declare
        e_name employee.name%type;
begin
        select name into e_name from employee where name='aaa';
           if e_name='aaa' then
             update employee set eid='c205' where name='aaa';
             dbms_output.put_line('Record has been updated!!!');
           end if;
         end;
          /
```

OUTPUT:

Record has been updated

To Verify: SQL> select * from employee;

EID	NAME	SALARY	DESIGNATIO
c205	aaa	4433	clerk
cs02	bbb	4957	Technician
cs03	ccc	4957	manager

6. Write a PL/SQL block to update salary of all employees who are managers.

PROCEDURE:

```
declare
        sal employee.salary%type :=1000;
begin
        for i in 1..2
        loop
        update employee set salary=salary+sal where designation='manager';
    end loop;
end;
  /
```

OUTPUT:

SQL> /

PL/SQL procedure successfully completed.

To Verify: SQL> select * from employee;

EID	NAME	SALARY	DESIGNATIO
c205	aaa	4433	clerk
cs02	bbb	4957	Technician
cs03	ccc	6957	manager

7. Write a PL/SQL block to assign the value 100 to the variable "num" and increase the "num" variable by 25 till num value gets 250.

PROCEDURE:

```
declare
 num number:=100;
begin
   while num<=200 loop
 dbms_output.put_line('The value of variable num is');
 dbms_output.put_line(TO_CHAR(num));
                    num:=num+25;
   end loop;
end;
/
```

OUTPUT:

SQL>/

The value of variable num is 200
PL/SQL procedure successfully completed.

8. Write a PL/SQL block to delete the rows that belong to the designation "Manager".

PROCEDURE:

```
begin
          delete from employee where designation='Manager';
dbms_output.put_line('The row with the designation Manager is deleted');
```

```
 end;
 /
```

OUTPUT:

```
SQL>/

The row with the designation Manager is deleted
PL/SQL procedure successfully completed.
```

To Verify: SQL> select * from employee;

EID	NAME	SALARY	DESIGNATION
c205	aaa	4433	clerk
cs02	bbb	4957	Technician

9. Write a PL/SQL block to check if the name is “aaa” then set the employee id is “cs06” and verify it.

PROCEDURE:

```
declare
  v_name employee.name%type;
  v_eid employee.eid%type:='cs06';
begin
  select name into v_name from employee where name='aaa';
  if v_name='aaa' then
        goto updation;
  end if;
  <<updation>>
update employee set eid=v_eid where name=v_name;


dbms_output.put_line('update the record!!!');
end; /
```

OUTPUT:

```
SQL> /
The record has been updated!!!
PL/SQL procedure successfully completed.

        SQL> select * from employee;

        EID       NAME      SALARY DESIGNATIO
        ---- -------------------- ---------- ------------------
```

cs06	aaa	4433	clerk
cs02	bbb	4957	Technician

EXCEPTION HANDLING:

10. Write a PL/SQL block to check whether the employee whose employee id is "cs01" is not present then display the error message "The Employee id cs01 is not available" using predefined exception "NO_DATA_FOUND" and verify it.

PROCEDURE:

```
declare
        v_eid employee.eid%type;
begin
        select eid into v_eid from employee where eid='cs10';
        exception
         when NO_DATA_FOUND then
 dbms_output.put_line('The employee id cs10 is not available');
end;
/
```

OUTPUT:

```
SQL> /
        The employee id cs10 is not available
        PL/SQL procedure successfully completed.
```

Ex.No: 9 (b)	STORED PROCEDURES
Date :	

[To execute use – exec procedure name]

1. **Create a procedure to display empid, name and salary of all employees record and verify it.**

PROCEDURE:

```
CREATE OR REPLACE PROCEDURE employee_detail
    IS
CURSOR emp_cur IS
    SELECT empid,name,salary FROM employee;
BEGIN
   FOR emp_rec IN emp_cur
 LOOP
   dbms_output.put_line(emp_rec.empid || ' ' ||emp_rec.name || ' ' ||emp_rec.salary);
END LOOP;
END;
/
```

Procedure created.

OUTPUT:

```
SQL> exec employee_detail;
cs001 www 10000
a123 ragu 204000
a124 rama 200000
a125 ram 30000
a128 sag 10000
c128 dinesh 10000
d124 abc 5000

PL/SQL procedure successfully completed.
```

2. **Create a procedure update_salary to update the salary column with an increase of 1000 and verify it.**

PROCEDURE:

```
create or replace procedure up_sal is
   begin
 update employee set salary=salary+1000;
   end;
/
```

OUTPUT:

```
SQL> /
Procedure created.

SQL> exec up_sal;

PL/SQL procedure successfully completed.

SQL> select *from employee;

EID    NAME      SALARY    DESIGNATIO
-------  ----------   ------------   --------------------
cs06      aaa          2933        clerk
cs02      bbb          4957        Technician
cs03      ccc          4957        manager
```

Ex.No: 10

Date :

TRIGGERS

1. **Write a PL/SQL block to create a trigger "T" when inserts,deletes and updates into the employee table with reference the value of a column before and after the data change by prefixing it with old or new qualifier.**

```
SQL> select *from employee1;

EID     NAME              SALARY        DESIGNATIO
----    ----------        ----------    ----------
cs06    ann               2666          clerk
cs02    becky             3597          technician
cs03    carl              3597          manager

SQL> desc temp1;
 Name                          Null?    Type
 ----------------------------- -------- --------------------
 E_EID                                  VARCHAR2(4)
 E_NAME                                 VARCHAR2(20)
 E_SALARY                               NUMBER(6)
 E_DESIGNATION                          VARCHAR2(10)

SQL>
    create or replace trigger T after update or delete on employee1 for each row
     declare
     e_eid varchar2(4);
     e_name varchar2(20);
     e_salary number(6);
     e_designation varchar2(10);
     op varchar2(8);
     begin
     if updating then
    op:='update';
    end if;
    if deleting then
    op:='delete';
```

```
    end if;
    e_eid:=:new.eid;
    e_name:=:new.name;
    e_salary:=:new.salary;
    e_designation:=:new.designation;
    insert into  temp1 values(e_eid,e_name,e_salary,e_designation);
   end;
   /
```

Trigger created.

Update:

```
SQL> update employee1 set name='ddd' where name='aaa';
```

1 row updated.

After Updating:

```
SQL> select *from temp1;
```

E_EI	E_NAME	E_SALARY	E_DESIGNAT
cs06	dave	2666	clerk

Ex.No: 11 **SIMPLE FORMS IN VISUAL BASIC**

Date :

CODING:

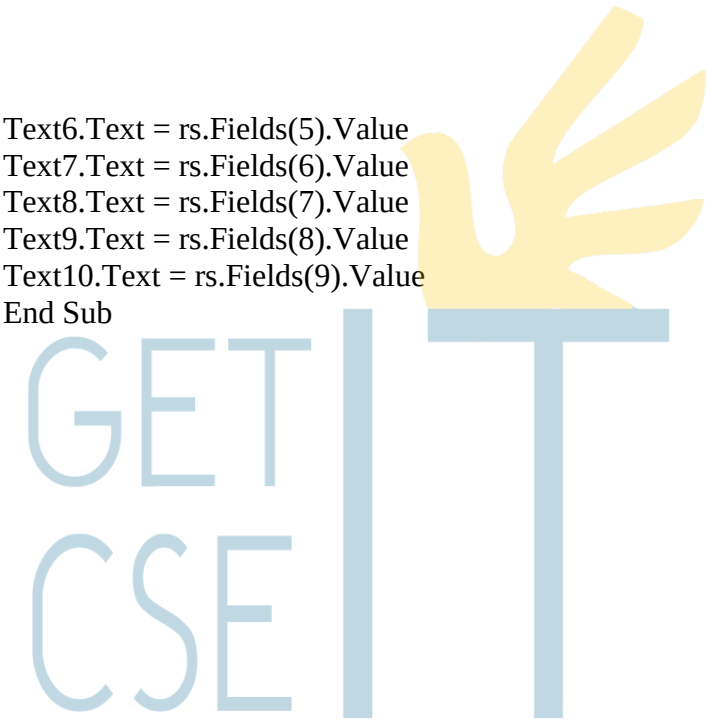
```
Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset

Private Sub Command1_Click()
rs.MoveFirst
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
MsgBox ("First record")
End Sub

Private Sub Command10_Click()
rs.AddNew
rs.Fields(0).Value = Text1.Text
rs.Fields(1).Value = Text2.Text
rs.Fields(2).Value = Text3.Text
rs.Fields(3).Value = Text4.Text
rs.Fields(4).Value = Text5.Text
rs.Fields(5).Value = Text6.Text
rs.Fields(6).Value = Text7.Text
rs.Fields(7).Value = Text8.Text
rs.Fields(8).Value = Text9.Text
rs.Fields(9).Value = Val(Text4.Text) + Val(Text5.Text) + Val(Text6.Text) +
Val(Text7.Text) + Val(Text8.Text) + Val(Text9.Text)
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
MsgBox "new record added"
rs.Save
```

```
End Sub


Private Sub Command2_Click()
rs.MoveLast
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
MsgBox "last record"
End Sub

Private Sub Command3_Click()
rs.MovePrevious
While rs.BOF = True
MsgBox "End of record-Previous"
rs.MoveFirst
Wend
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
End Sub

Private Sub Command4_Click()
rs.MoveNext
While rs.EOF = True
MsgBox "End of Record-Last"
rs.MoveFirst
Wend
```

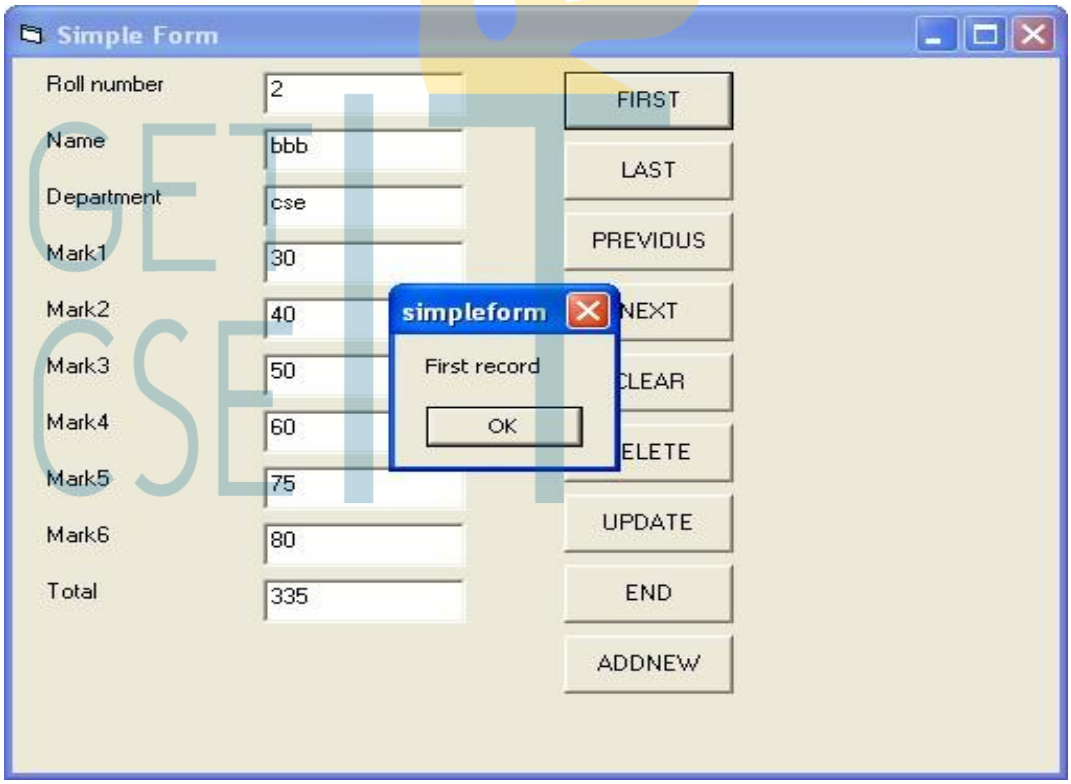
```
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
End Sub

Private Sub Command5_Click()
Text1.Text = " "
Text2.Text = " "
Text3.Text = " "
Text4.Text = " "
Text5.Text = " "
Text6.Text = " "
Text7.Text = " "
Text8.Text = " "
Text9.Text = " "
Text10.Text = " "
MsgBox "Cleared"
End Sub

Private Sub Command6_Click()
rs.Fields(0).Value = Text1.Text
rs.Fields(1).Value = Text2.Text
rs.Fields(2).Value = Text3.Text
rs.Fields(3).Value = Text4.Text
rs.Fields(4).Value = Text5.Text
rs.Fields(5).Value = Text6.Text
rs.Fields(6).Value = Text7.Text
rs.Fields(7).Value = Text8.Text
rs.Fields(8).Value = Text9.Text
rs.Fields(9).Value = Val(Text4.Text) + Val(Text5.Text) + Val(Text6.Text) +
Val(Text7.Text) + Val(Text8.Text) + Val(Text9.Text)
rs.Save
MsgBox "Saved"
End Sub

Private Sub Command7_Click()
Text1.Text = rs.Fields(0).Value
```

```
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
rs.Delete
MsgBox "record deleted"
End Sub

Private Sub Command8_Click()
rs.Fields(0).Value = Text1.Text
rs.Fields(1).Value = Text2.Text
rs.Fields(2).Value = Text3.Text
rs.Fields(3).Value = Text4.Text
rs.Fields(4).Value = Text5.Text
rs.Fields(5).Value = Text6.Text
rs.Fields(6).Value = Text7.Text
rs.Fields(7).Value = Text8.Text
rs.Fields(8).Value = Text9.Text
rs.Fields(9).Value = Val(Text4.Text) + Val(Text5.Text) + Val(Text6.Text) +
Val(Text7.Text) + Val(Text8.Text) + Val(Text9.Text)
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
rs.Update
MsgBox "record updated"
End Sub
Private Sub Command9_Click()
End
End Sub

Private Sub Form_Load()
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Documents and
Settings\venkat\Desktop\dbms lab\simpleform.mdb;Persist Security Info=False"
rs.Open "form", cn, adOpenDynamic, adLockOptimistic
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
```

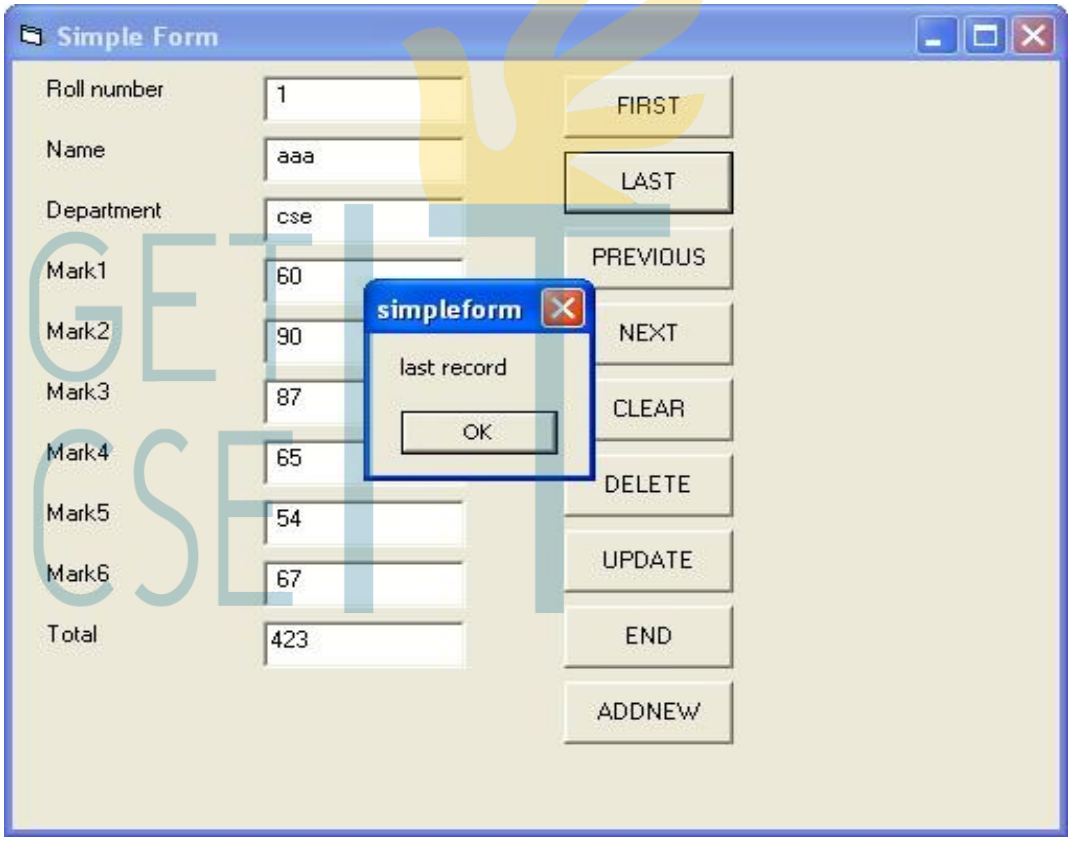
```
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = rs.Fields(9).Value
End Sub
```



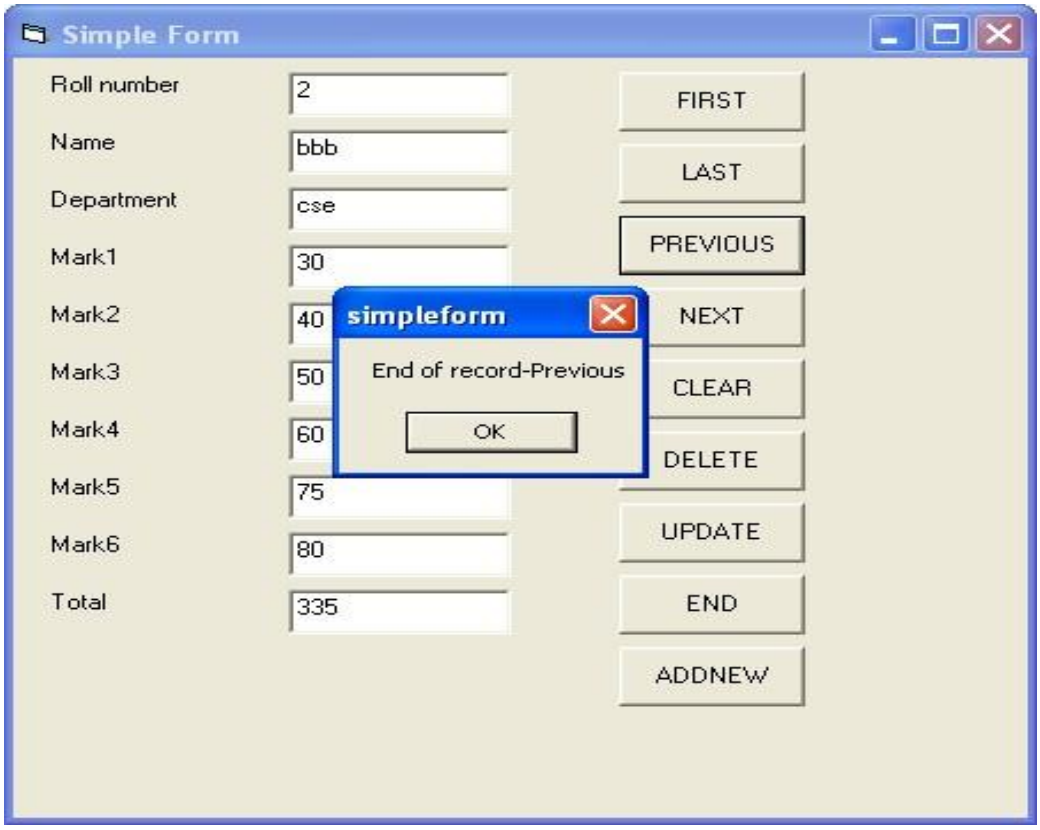
SCREENSHOT:



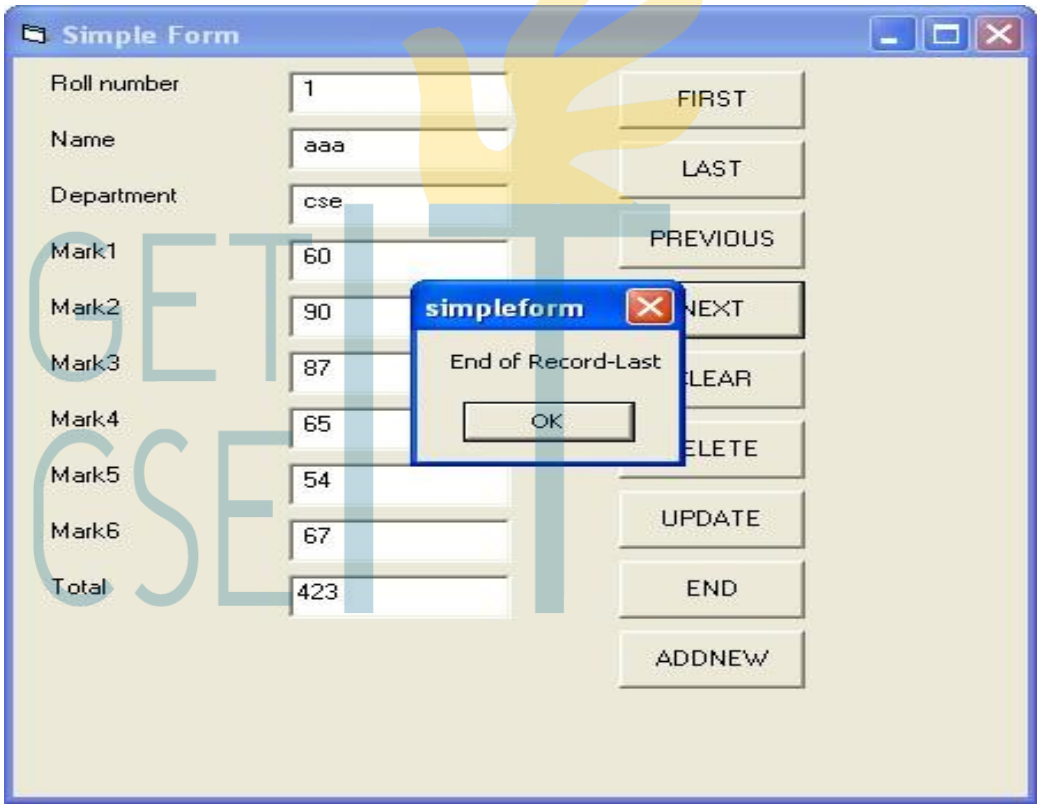
Last Record:



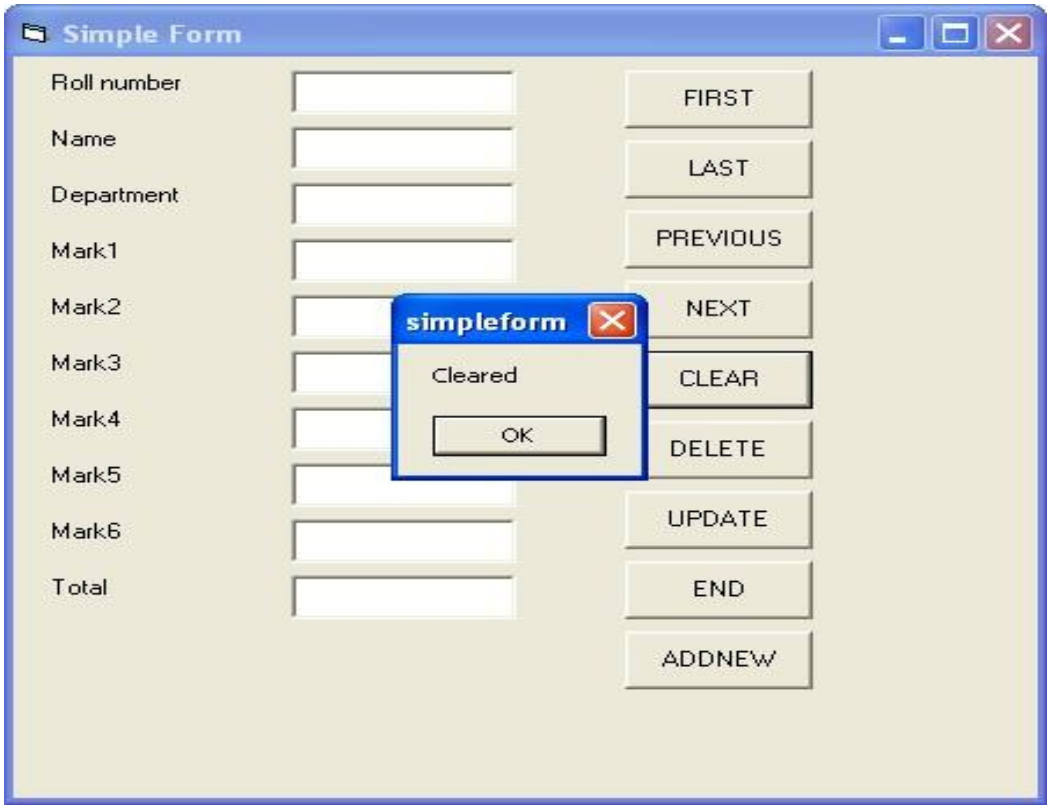
Previous:



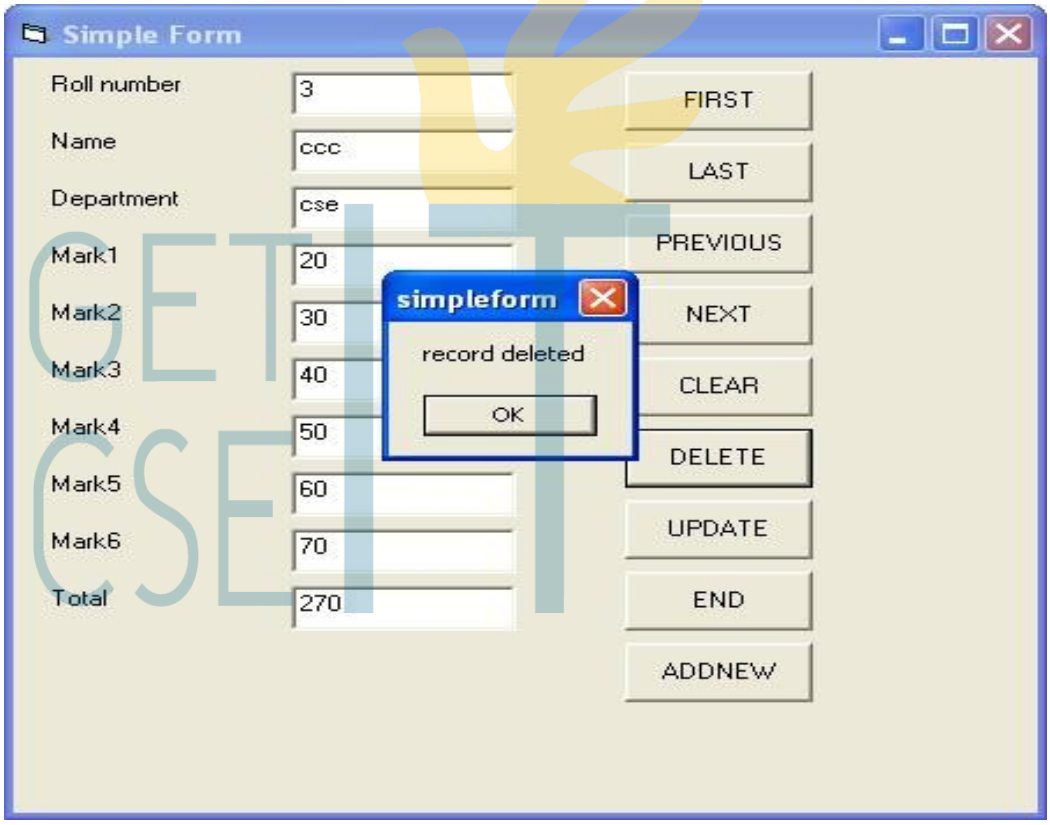
Next:



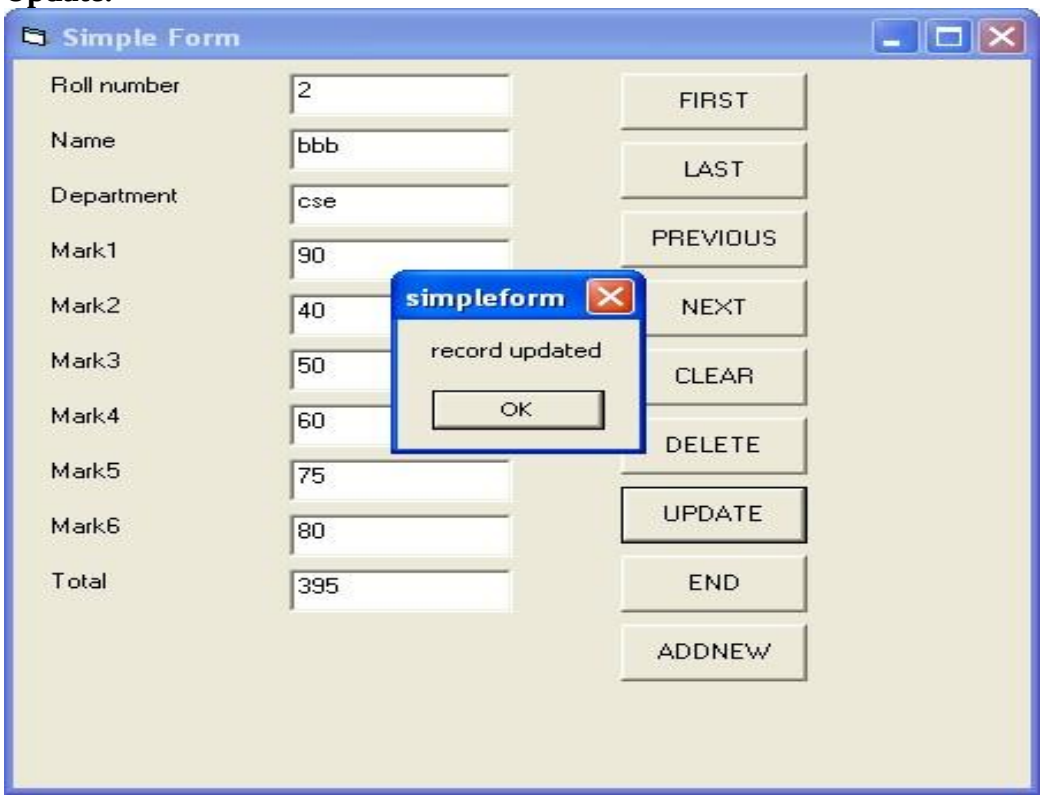
Clear:



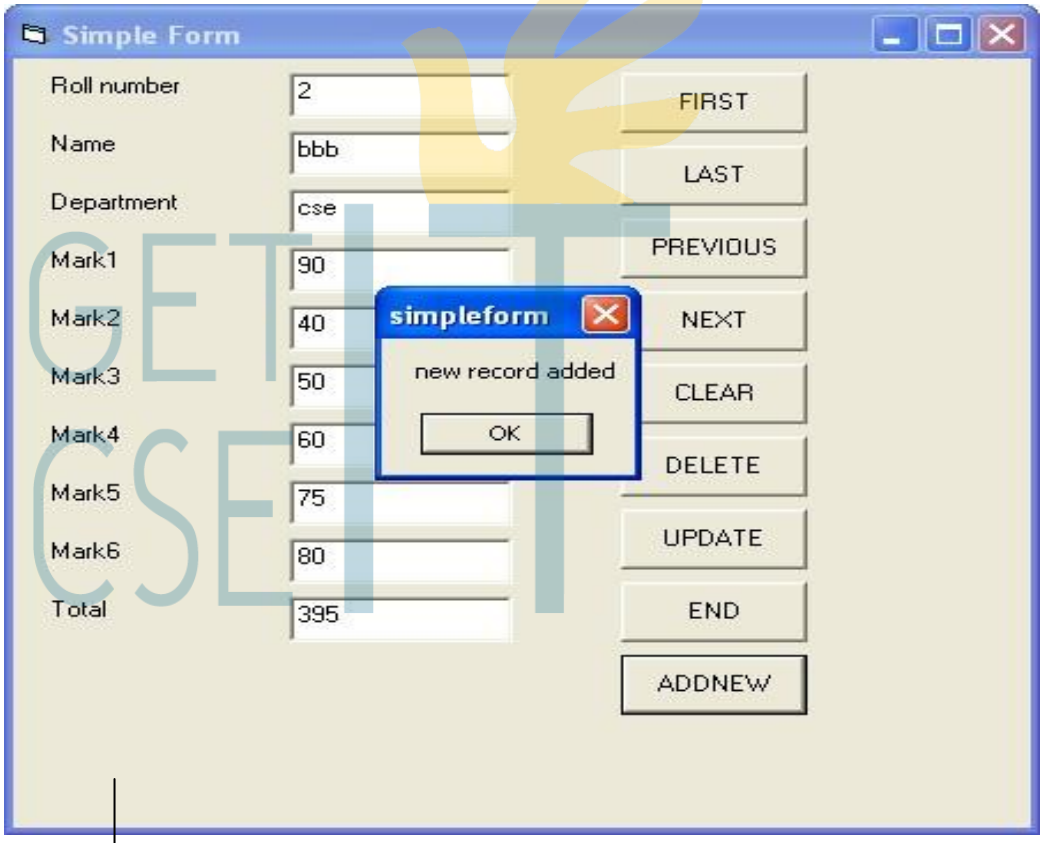
Delete:



Update:



AddNew:



Ex.No: 12 **EMPLOYEE PAYROLLS**

Date :

CODING:

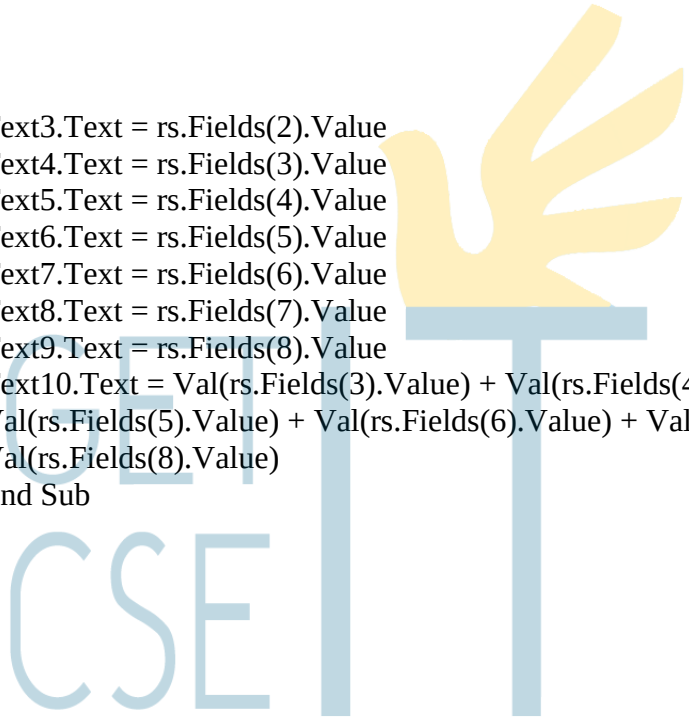
```
Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset
Private Sub Command1_Click()
rs.MoveFirst
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
```

```
MsgBox "First Record"
End Sub

Private Sub Command2_Click()
rs.MoveLast
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
MsgBox "Last Record"
End Sub

Private Sub Command3_Click()
rs.MovePrevious
While rs.BOF = True
MsgBox "End of record-First"
rs.MoveFirst
Wend
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)End Sub

Private Sub Command4_Click()
rs.MoveNext
While rs.EOF = True
MsgBox "End of Record-last"
rs.MoveFirst
Wend
Text1.Text = rs.Fields(0).Value
```

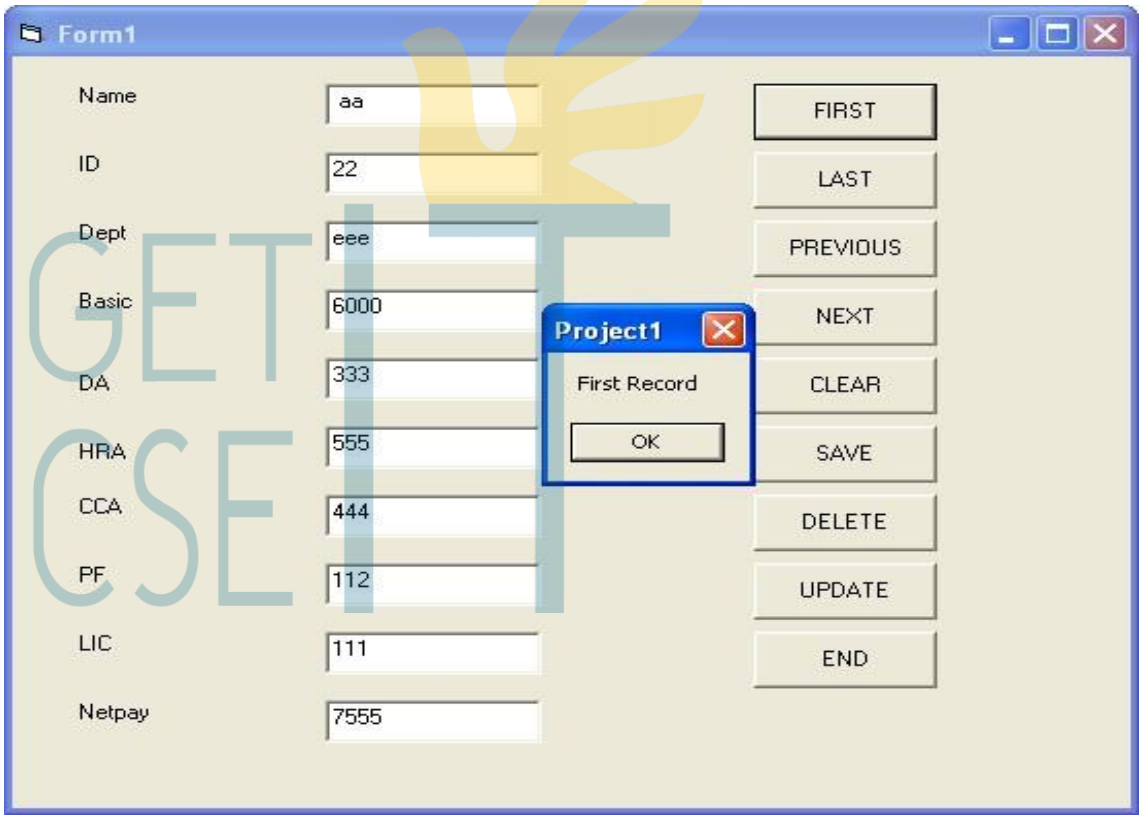
```
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)End Sub

Private Sub Command5_Click()
Text1.Text = " "
Text2.Text = " "
Text3.Text = " "
Text4.Text = " "
Text5.Text = " "
Text6.Text = " "
Text7.Text = " "
Text8.Text = " "
Text9.Text = " "
Text10.Text = " "
MsgBox "Text Cleared"
End Sub

Private Sub Command6_Click()
rs.Fields(0).Value = Text1.Text
rs.Fields(1).Value = Text2.Text
rs.Fields(2).Value = Text3.Text
rs.Fields(3).Value = Text4.Text
rs.Fields(4).Value = Text5.Text
rs.Fields(5).Value = Text6.Text
rs.Fields(6).Value = Text7.Text
rs.Fields(7).Value = Text8.Text
rs.Fields(8).Value = Text9.Text
rs.Fields(9).Value = Val(Text4.Text) + Val(Text5.Text) + Val(Text6.Text) +
Val(Text7.Text) + Val(Text8.Text) + Val(Text9.Text)
rs.Save
MsgBox "Record Saved"
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
End Sub

Private Sub Command7_Click()
```

```
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
rs.Delete
MsgBox "record deleted"
End Sub

Private Sub Command8_Click()
rs.Fields(0).Value = Text1.Text
rs.Fields(1).Value = Text2.Text
rs.Fields(2).Value = Text3.Text
rs.Fields(3).Value = Text4.Text
rs.Fields(4).Value = Text5.Text
rs.Fields(5).Value = Text6.Text
rs.Fields(6).Value = Text7.Text
rs.Fields(7).Value = Text8.Text
rs.Fields(8).Value = Text9.Text
rs.Fields(9).Value = Val(Text4.Text) + Val(Text5.Text) + Val(Text6.Text) +
Val(Text7.Text) + Val(Text8.Text) + Val(Text9.Text)
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
rs.Update
MsgBox "record updated"
End Sub

Private Sub Command9_Click()
End
End Sub

Private Sub Form_Load()
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Documents and
Settings\venkat\Desktop\pay\pay.mdb;Persist Security Info=False"
rs.Open "VBpay", cn, adOpenDynamic, adLockOptimistic
Text1.Text = rs.Fields(0).Value
Text2.Text = rs.Fields(1).Value
```

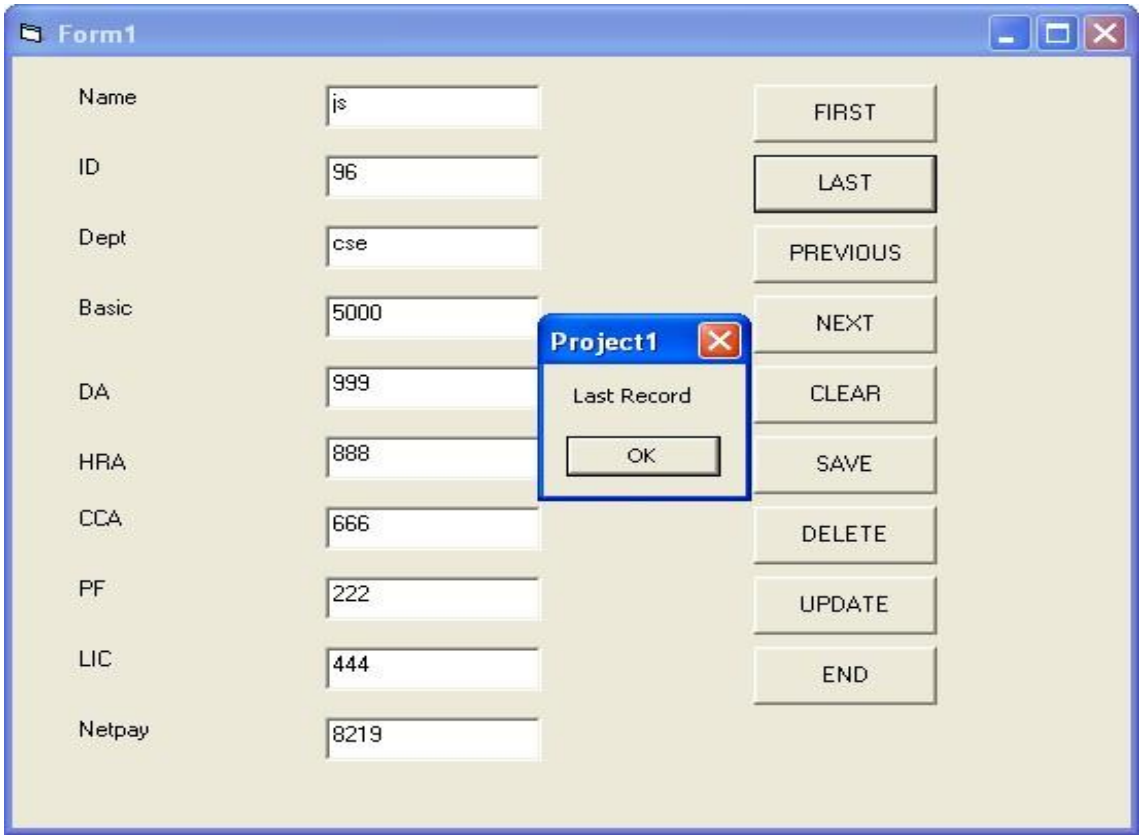
```
Text3.Text = rs.Fields(2).Value
Text4.Text = rs.Fields(3).Value
Text5.Text = rs.Fields(4).Value
Text6.Text = rs.Fields(5).Value
Text7.Text = rs.Fields(6).Value
Text8.Text = rs.Fields(7).Value
Text9.Text = rs.Fields(8).Value
Text10.Text = Val(rs.Fields(3).Value) + Val(rs.Fields(4).Value) +
Val(rs.Fields(5).Value) + Val(rs.Fields(6).Value) + Val(rs.Fields(7).Value) +
Val(rs.Fields(8).Value)
End Sub
```



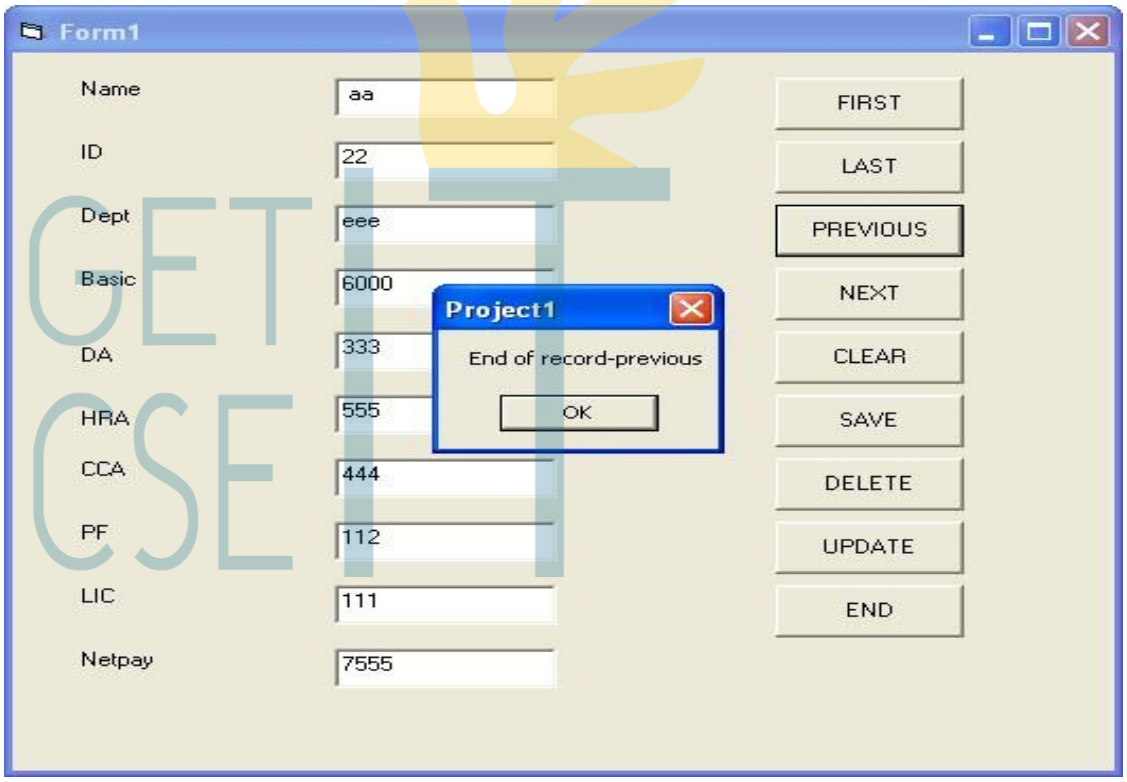
First Record:



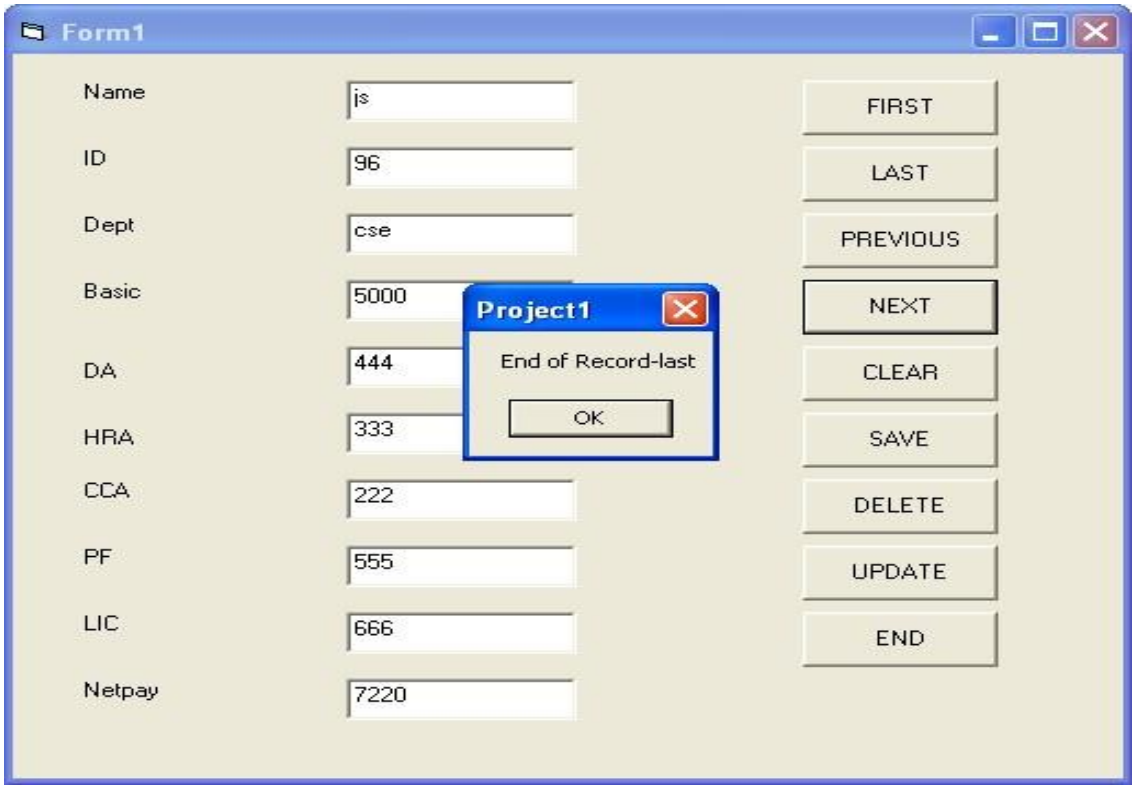
Last Record:



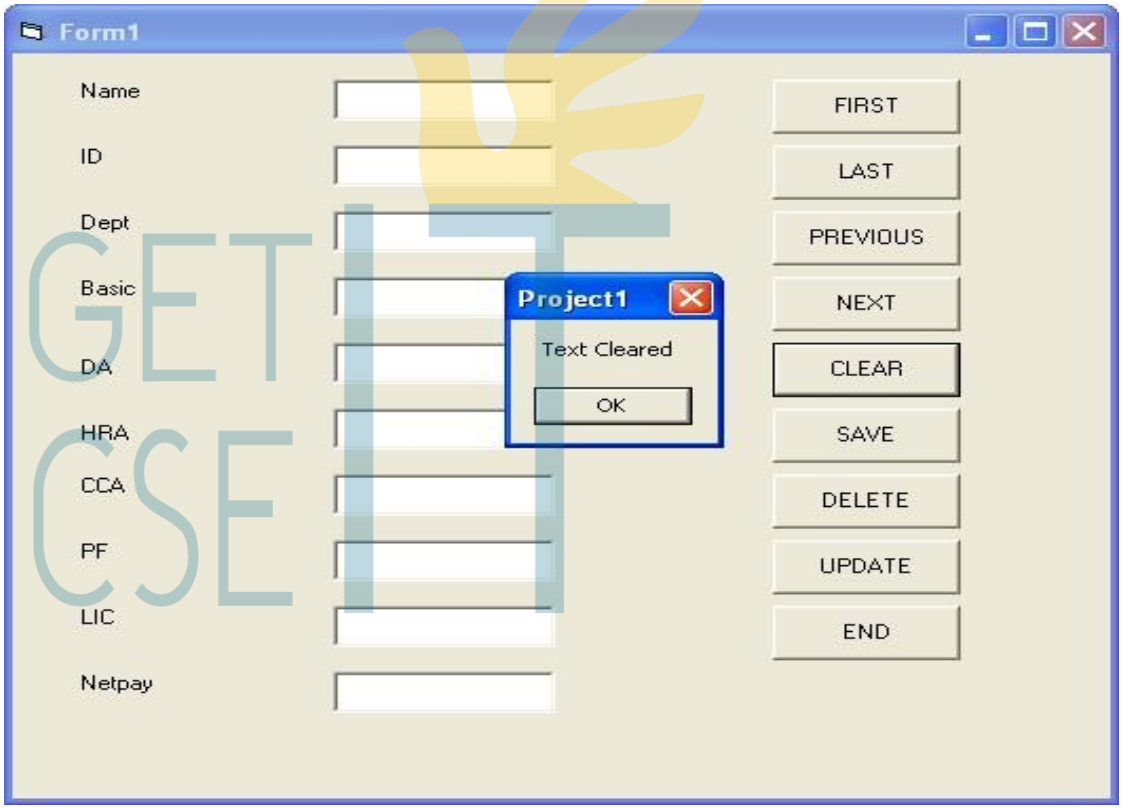
Previous:



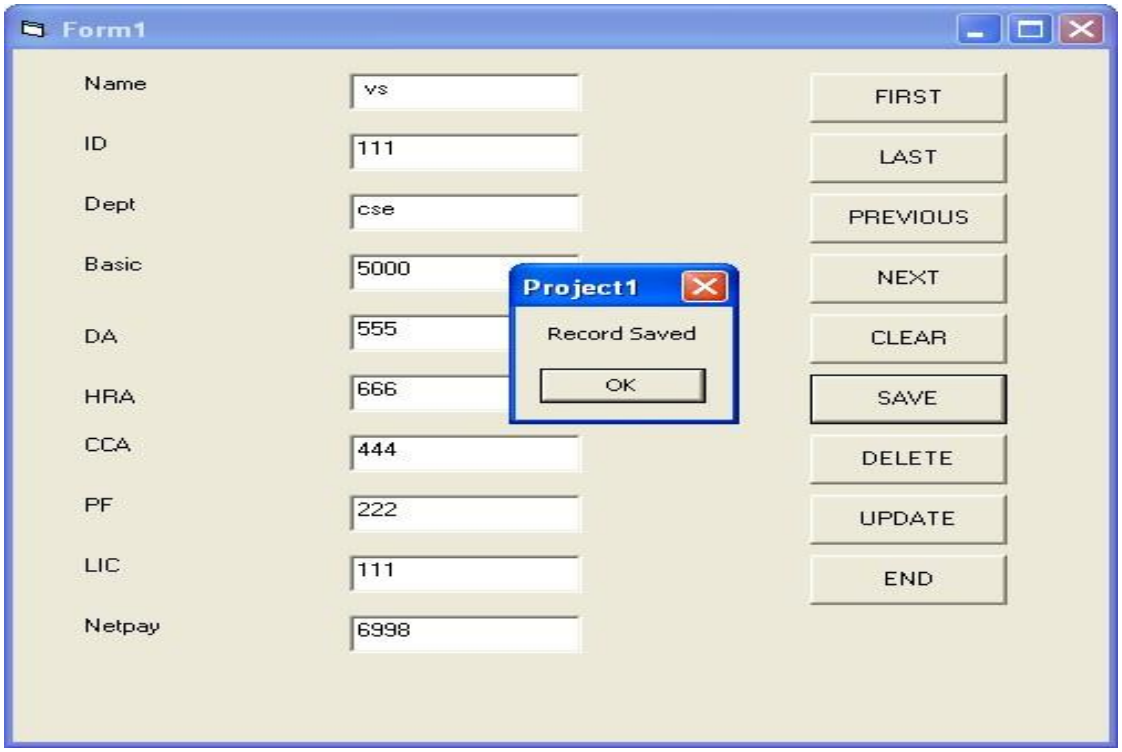
Next:



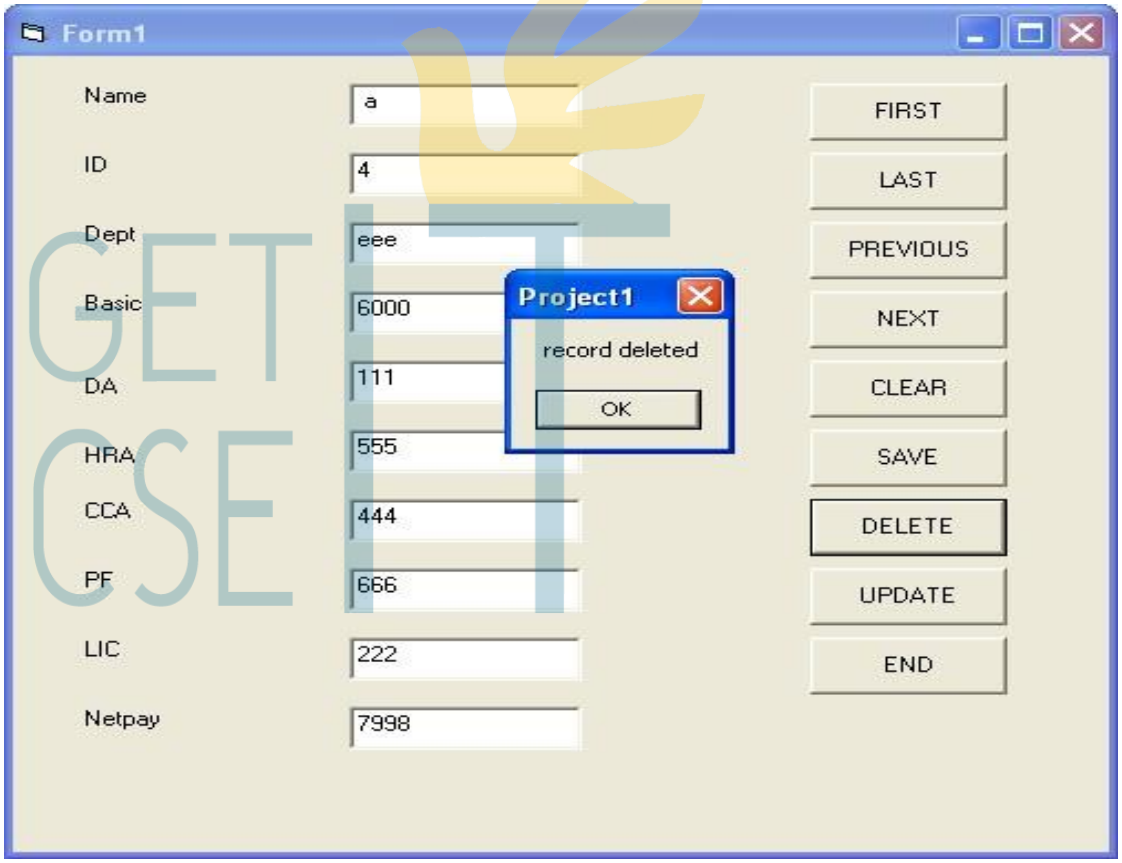
Clear:



Save:



Delete:



Update: