

COURSE OBJECTIVES:**To make the students**

1. To know the DML and its application
2. To be familiar with Queries using Transaction commands
3. To be aware of constraints function and perform clause.
4. To utilize the date and character function and its application.
5. To Write the PL/SQL Program.

COURSE OUTCOMES:**Learners should be able to**

1. Create Table DML command and perform logical operations using the Oracle as IT platform
2. Design Query using Transaction command
3. Develop programme using various operations
4. Design query using character, date and joins function.
5. To critically analyse the requirement and Write a PL/SQL Programme.
6. Prepare the written report.
 1. Create an employee table using DML commands and perform logical operations.
 2. Create a student table using DML commands and perform logical operations.
 3. Perform the queries using transaction commands.
 4. Write a program for inventory with constraints and perform the following clause.
 - i) Select Clause
 - ii) Where Clause
 - iii) Order by clause
 5. Write a program for student database and perform the following operations
 - i) Arithmetic Operation
 - ii) Group Function
 6. Perform queries by using character and date functions.
 7. Perform a query for Joins.
 8. Write a PL/SQL program for Employee Payroll.
 9. Write a PL/SQL program for Student Database and Calculate Total, Average, Result.

SUGGESTED READINGS:

1. Rajiv Chopra (2016), "Database Management Systems (DBMS)", 5TH Edition, S.Chand, New Delhi.
2. Nilesh Shah, (2015), "Database Systems Using Oracle", 2nd Edition, Pearson Education, New Delhi.
3. Raghu Ramakrishnan, Johannes Gehrke, (2014), "Database Management Systems", 3rd Edition, McGraw Hill Education (India) Edition, New Delhi.
4. Abraham Silberschatz, Henry F. Korth and S. Sudharshan (2011), "Database System Concepts", Sixth Edition, Tata Mc Graw Hill.
5. G.K.Gupta, (2011), "Database Management Systems", Tata Mc Graw Hill, New Delhi.

Ex. No: 1	LOGICAL OPERATIONS IN EMPLOYEE TABLE
Date:	

AIM:

To create a table employee using DML commands and perform the logical operations.

ALGORITHM:

STEP1: start the process

STEP2: start-all programs-oracles-run set command lines

STEP3: connect to SQL.

STEP4: create a table named employee and insert the values.

STEP5: update the table using “update”command

STEP6: delete a record from a table using “delete” command

STEP7: perform the logical operation using AND,OR,NOT in the employee table

STEP8: stop the process.

QUERIES :

```
SQL> create table employee (empid number(7),empname varchar22(15),department  
varchar22(15),designation varchar22(20),salary number(10) );
```

Table created.

```
SQL> insert into employee values('E001','Askar', 'Sales','Manager',25000);
```

1 row created.

```
SQL> insert into employee values('E002','Abdul', 'Production','Manager',35000);
```

1 row created.

```
SQL> insert into employee values('E003','Bala', 'Marketing', 'Staff ',20000);
```

1 row created.

```
SQL> insert into employee values('E004',' Dinesh', 'Quality Analysis', 'Staff ',10000);
```

1 row created.

```
SQL> insert into employee values('E005','Narmadha', 'Sales', 'Staff ',18000);
```

1 row created.

```
SQL> select * from employee;
```

EMPID	EMPNAME	DEPARTMENT	DESIGNATION	SALARY
E001	Askar	Sales	Manager	25000
E002	Abdul	Production	Manager	35000
E003	Bala	Marketing	Staff	20000
E004	Dinesh	Quality Analysis	Staff	10000
E005	Narmadha	Sales	Staff	18000

SQL> select * from employee where department = 'Sales' and designation = 'Manager' ;

EMPID	EMPNAME	DEPARTMENT	DESIGNATION	SALARY
E001	Askar	Sales	Manager	25000

SQL> select * from employee where department = 'Sales' or designation = 'Manager' ;

EMPID	EMPNAME	DEPARTMENT	DESIGNATION	SALARY
E001	Askar	Sales	Manager	25000
E002	Abdul	Production	Manager	35000
E005	Narmadha	Sales	Staff	18000

SQL> select * from employee where not department = 'Sales';

EMPID	EMPNAME	DEPARTMENT	DESIGNATION	SALARY
E002	Abdul	Production	Manager	35000
E003	Bala	Marketing	Staff	20000
E004	Dinesh	Quality Analysis	Staff	10000

RESULT :

The above queries has been executed successfully.

Ex. No: 2	LOGICAL OPERATIONS IN STUDENT TABLE
Date:	

AIM:

To create a table student using DML command and perform logical operation.

ALGORITHM:

STEP1: start the process

STEP2: start-all program-oracles-run SQL

STEP3: connect to SQL

STEP4: create a table named student and insert the values

STEP5: update the table using “update” command

STEP6: perform the logical operations using AND,OR,NOT in the employee table

STEP7: stop the process.

QUERIES :

```
SQL> create table student32 ( regno varchar22(10),sname varchar22(15),mark1 number (3),mark2 number(3),total number(3));
```

Table created.

```
SQL> insert into student32(regno,sname,mark1,mark2)values('18CCU004','Aishwarya',80,90);
```

1 row created.

```
SQL> insert into student32(regno,sname,mark1,mark2)values('18CCU002','Anushya',50,70);
```

1 row created.

```
SQL> insert into student32(regno,sname,mark1,mark2)values('18CCU003','Bobby',70,70);
```

1 row created.

```
SQL> insert into student32(regno,sname,mark1,mark2)values('18CCU004','Devi',67,90);
```

1 row created.

```
SQL> insert into student32(regno,sname,mark1,mark2)values('18CCU005','Manish',98,90);
```

1 row created.

```
SQL> select * from student32;
```

REGNO	SNAME	MARK1	MARK2	TOTAL
-----	-----	-----	-----	-----
18CCU004	Aishwarya	80	90	
18CCU002	Anushya	50	70	
18CCU003	Bobby	70	70	
18CCU004	Devi	67	90	
18CCU005	Manish	98	90	

```
SQL> update student32 set total= mark1+mark2;
```

5 rows updated.

```
SQL> select * from student32;
```

REGNO	SNAME	MARK1	MARK2	TOTAL
-----	-----	-----	-----	-----
18CCU004	Aishwarya	80	90	170
18CCU002	Anushya	50	70	120
18CCU003	Bobby	70	70	140
18CCU004	Devi	67	90	157
18CCU005	Manish	98	90	188

SQL> select * from student32 where mark1 >70 and mark2 >70;

REGNO	SNAME	MARK1	MARK2	TOTAL
18CCU004	Aishwarya	80	90	170
18CCU005	Manish	98	90	188

SQL> select * from student32 where mark1 >70 or mark2 >70;

REGNO	SNAME	MARK1	MARK2	TOTAL
18CCU004	Aishwarya	80	90	170
18CCU004	Devi	67	90	157
18CCU005	Manish	98	90	188

SQL> select * from student32 where not mark1 >70;

REGNO	SNAME	MARK1	MARK2	TOTAL
18CCU002	Anushya	50	70	120
18CCU003	Bobby	70	70	140
18CCU004	Devi	67	90	157

RESULT :

The above queries has been executed successfully.

Ex. No: 3	TRANSACTION COMMANDS
Date:	

AIM:

To create the queries using transactions

ALGORITHM:

STEP1: start the process

STEP2: start - all programs- oracles-run SQL

STEP3: connect to SQL

STEP4: create a table named customer and insert the values

STEP5: perform the transaction command: ROLLBACK, COMMIT,SAVEPOINT in the table

STEP6: stop the process.

QUERIES :

SQL> create table employee(eid varchar2 (8),ename varchar2 (10),age number (9),doj date, salary number (9));

Table created.

SQL> insert into employee values ('E001','aji','23','21/march/2019','23000');

1 row created.

SQL> insert into employee values ('E002','dharshu','43','1/june/2009','24000');

1 row created.

SQL> insert into employee values ('E003','yamu','53','8/august/2005','24000');

1 row created.

SQL> insert into employee values ('E004','bobby','56','25/jan/2006','25000');

1 row created.

SQL> insert into employee values ('E005','santhiya','45','5/may/2016','27000');

1 row created.

SQL> insert into employee values ('E006','raji','52','4/april/2019','37000');

1 row created.

SQL> select * from employee;

EID	ENAME	AGE	DOJ	SALARY
1001	aji	23	21-MAR-19	23000
1002	dharshu	43	01-JUN-09	24000
1003	yamu	53	08-AUG-05	24000
1004	bobby	56	25-JAN-06	25000
1005	santhiya	45	05-MAY-16	27000
1006	raji	52	04-APR-19	37000

6 rows selected.

SQL> commit;

Commit complete.

SQL> delete from employee where eid ='E005';

1 row deleted.

SQL> select * from employee;

EID	ENAME	AGE	DOJ	SALARY
1001	aji	23	21-MAR-19	23000
1002	dharshu	43	01-JUN-09	24000
1003	yamu	53	08-AUG-05	24000
1004	bobby	56	25-JAN-06	25000
1006	raji	52	04-APR-19	37000

SQL> rollback ;

Rollback complete.

SQL> select * from employee;

EID	ENAME	AGE	DOJ	SALARY
1001	aji	23	21-MAR-19	23000
1002	dharshu	43	01-JUN-09	24000
1003	yamu	53	08-AUG-05	24000
1004	bobby	56	25-JAN-06	25000
1005	santhiya	45	05-MAY-16	27000
1006	raji	52	04-APR-19	37000

6 rows selected.

SQL> insert into employee values ('1007','sandya','46','5/march/2006','2000');

1 row created.

SQL> savepoint sp1;

Savepoint created.

SQL> insert into employee values ('1008','sam','45','5/may/2014','22000');

1 row created.

SQL> savepoint sp2;

Savepoint created.

SQL> select * from employee;

EID	ENAME	AGE	DOJ	SALARY
-----	-----	-----	-----	-----
1001	aji	23	21-MAR-19	23000
1002	dharshu	43	01-JUN-09	24000
1003	yamu	53	08-AUG-05	24000
1004	bobby	56	25-JAN-06	25000
1005	santhiya	45	05-MAY-16	27000
1006	raji	52	04-APR-19	37000
1007	sandya	46	05-MAR-06	2000
1008	sam	45	05-MAY-14	22000

8 rows selected.

SQL> rollback to sp1;

Rollback complete.

SQL> select * from employee;

EID	ENAME	AGE	DOJ	SALARY
-----	-----	-----	-----	-----
1001	aji	23	21-MAR-19	23000
1002	dharshu	43	01-JUN-09	24000
1003	yamu	53	08-AUG-05	24000
1004	bobby	56	25-JAN-06	25000
1005	santhiya	45	05-MAY-16	27000
1006	raji	52	04-APR-19	37000
1007	sandya	46	05-MAR-06	2000

7 rows selected.

RESULT :

The above queries has been executed successfully.

Ex. No: 4	INVENTORY TABLE
Date:	

AIM:

To create an inventory with constraints and the following select clause, where clause and order clause.

ALGORITHM:

STEP1: start the process

STEP2: start all programs-oracles-run SQL

STEP3: connect to SQL

STEP4: create a table named inventory and insert the values

STEP5: update the table using with constraints and the select clause , where clause and order clause

STEP6: stop the process.

QUERIES :

SQL> create table inventory(pid varchar2(9) not null primary key, pname varchar2 (7), amount number (8));

Table created.

SQL> insert into inventory values('P001','pencil','5');

1 row created.

SQL> insert into inventory values('P002','pen','10');

1 row created.

SQL> insert into inventory values('P003','eraser','3');

1 row created.

SQL> insert into inventory values('P004','dettol','20');

1 row created.

SQL> insert into inventory values('P005','paper','10',);

1 row created.

SQL> insert into inventory values('P006','hamam','20');

1 row created.

SQL> select * from inventory;

PID	PNAME	AMOUNT
1	pencil	5
2	pen	10
3	eraser	3
4	dettol	20
5	paper	10
6	hamam	20

6 rows selected.

SQL> select pname from inventory where amount > 3 ;

PNAME

pencil
pen
dettol
paper
hamam

5 rows selected.

SQL> select * from inventory order by amount ;

PID PNAME AMOUNT

3 eraser 3
1 pencil 5
5 paper 10
2 pen 10
4 dettol 20
6 hamam 20

6 rows selected.

SQL> select * from inventory order by amount desc ;

PID PNAME AMOUNT

6 hamam 20
4 dettol 20
2 pen 10
5 paper 10
1 pencil 5
3 eraser 3

RESULT :

The above queries has been executed successfully.

Ex. No: 5	ARITHMETIC AND GROUP FUNCTIONS
Date:	

AIM:

To create an student database and to perform arithmetic operations and group function.

ALGORITHM:

STEP1: start the process

STEP2: start –all programs- oracles-run SQL

STEP3: connect to SQL

STEP4: create a table named student and insert the values

STEP5: to perform the following operations: arithmetic operation and group function

STEP6: stop the process.

QUERIES :

```
SQL> create table student(regno varchar2(10),sname varchar2(20),internal varchar2(4));
```

Table created.

```
SQL> insert into student values('1001','ram','33');
```

1 row created.

```
SQL> insert into student values('1002','sam','34');
```

1 row created.

```
SQL> insert into student values('1003','adith','39');
```

1 row created.

```
SQL> insert into student values('1004','malar','39');
```

1 row created.

```
SQL> insert into student values('1005','nivin','35');
```

1 row created.

```
SQL> insert into student values('1006','koya','31');
```

1 row created.

ARITHMETIC FUNCTIONS :

```
SQL> select regno,sname,internal,internal+50 as total from student;
```

REGNO	SNAME	INTERNAL	TOTAL
1001	ram	33	83
1002	sam	34	84
1003	adith	39	89
1004	malar	39	89
1005	nivin	35	85
1006	koya	31	81

6 rows selected.

SQL> select regno,sname,internal,internal-20 as total from student;

REGNO	SNAME	INTERNAL	TOTAL
-------	-------	----------	-------

1001	ram	33	13
1002	sam	34	14
1003	adith	39	19
1004	malar	39	19
1005	nivin	35	15
1006	koya	31	11

6 rows selected.

SQL> select regno,sname,internal,internal*2 as total from student;

REGNO	SNAME	INTERNAL	TOTAL
-------	-------	----------	-------

1001	ram	33	66
1002	sam	34	68
1003	adith	39	78
1004	malar	39	78
1005	nivin	35	70
1006	koya	31	62

6 rows selected.

SQL> select regno,sname,internal,internal/2 as total from student;

REGNO	SNAME	INTERNAL	TOTAL
-------	-------	----------	-------

1001	ram	33	16.5
1002	sam	34	17
1003	adith	39	19.5
1004	malar	39	19.5
1005	nivin	35	17.5
1006	koya	31	15.5

6 rows selected.

GROUP FUNCTIONS :

SQL> select count(*) from student;

COUNT(*)

10

```
SQL> select sum(internal) from student;
```

```
SUM(INTERNAL)
```

```
-----
```

```
342
```

```
SQL> select min(internal) from student;
```

```
MIN(INTERNAL)
```

```
----
```

```
30
```

```
SQL> select max(internal) from student;
```

```
MAX(INTERNAL)
```

```
----
```

```
39
```

```
SQL> select avg(internal) from student;
```

```
AVG(INTERNAL)
```

```
-----
```

```
34.2
```

```
SQL> select distinct(internal) from student;
```

```
INTERNAL
```

```
-----
```

```
33
```

```
31
```

```
39
```

```
32
```

```
34
```

```
35
```

```
30
```

```
7 rows selected.
```

RESULT :

The above queries has been executed successfully.

Ex. No: 6	CHARACTER AND DATE FUNCTIONS
Date:	

AIM:

To create a student database and perform character and date function.

ALGORITHM:

STEP1: start the process

STEP2: start –all programs-oracles-run SQL

STEP3: connect the SQL

STEP4: create a table named student and insert the values

STEP5: to use character function like lower,upper,incat,upcat,replace,length,trim,substr and instr in table

STEP6: use date function in the table

STEP7: stop the process.

QUERIES:

```
SQL> create table student (rno varchar2(15),cname varchar2(20),ccode varchar2 (15));
```

Table created.

```
SQL> insert into student values('18ccu071','corporate accounting','18ccu301');
```

1 row created.

```
SQL> insert into student values('18ccu072','database management system','18ccu302');
```

1 row created.

```
SQL> insert into student values('18ccu073','computerised accounting system','18ccu303');
```

1 row created.

```
SQL> insert into student values('18ccu073','computerised accounting system','18ccu311');
```

1 row created.

```
SQL> insert into student values('18ccu073','database management system pratical','18ccu304');
```

1 row created.

```
SQL> insert into student values('18ccu074','computerised accounting system pratical','18ccu305');
```

1 row created.

```
SQL> insert into student values('18ccu075','auditing','18ccu311');
```

1 row created.

```
SQL> insert into student values('18ccu076','business law','18ccu315');
```

1 row created.

SQL> select * from student;

RNO	CNAME	CCODE
18ccu071	corporate accounting	18ccu301
18ccu072	database management system	18ccu302
18ccu073	computerised accounting system	18ccu303
18ccu073	computerised accounting system	18ccu311
18ccu073	database management system pratical	18ccu304
18ccu074	computerised accounting system pratical	18ccu305
18ccu075	auditing	18ccu311
18ccu076	business law	18ccu315

8 rows selected.

CHARACTER FUNCTIONS :

SQL> select lower(ccode) from student;

LOWER(CCODE)

18ccu301
18ccu302
18ccu303
18ccu311
18ccu304
18ccu305
18ccu311
18ccu315

8 rows selected.

SQL> select upper(ccode) from student;

UPPER(CCODE)

18CCU301
18CCU302
18CCU303
18CCU311
18CCU304
18CCU305
18CCU311
18CCU315

8 rows selected.

```
SQL> select initcap(cname) from student;
```

```
INITCAP(CNAME)
```

```
-----  
Corporate Accounting  
Database Management System  
Computerised Accounting System  
Computerised Accounting System  
Database Management System Pratical  
Computerised Accounting System Pratical  
Auditing  
Business Law
```

8 rows selected.

```
SQL> select concat(ccode,cname) from student;
```

```
CONCAT(CCODE,CNAME)
```

```
-----  
18ccu301corporate accounting  
18ccu302database management system  
18ccu303computerised accounting system  
18ccu311computerised accounting system  
18ccu304database management system pratical  
18ccu305computerised accounting system pratical  
18ccu311auditing  
18ccu315business law
```

8 rows selected.

```
SQL> select length(cname) from student;
```

```
LENGTH(CNAME)
```

```
-----  
20  
26  
30  
30  
35  
39  
8  
12
```

8 rows selected.

DATE FUNCTIONS :

SQL> select current_date from dual;

CURRENT_DATE

05-SEP-19

SQL> select sysdate from dual;

SYSDATE

05-SEP-19

SQL> select current_timestamp from dual;

CURRENT_TIMESTAMP

05-SEP-19 11.19.28.052000 AM +05:30

SQL> select extract(year from sysdate) from dual;

EXTRACT(YEARFROMSYSDATE)

2019

SQL> select extract(day from sysdate) from dual;

EXTRACT(DAYFROMSYSDATE)

5

SQL> select extract(month from sysdate) from dual;

EXTRACT(MONTHFROMSYSDATE)

9

SQL> select ADD_MONTHS(Date '2019-09-05',2) from dual;

ADD_MONTH

05-NOV-19

RESULT :

The above queries has been executed successfully.

Ex. No: 7	JOINS
Date:	

AIM:

To create a student database and perform the queire for joins

ALGORITHM:

STEP1: start the process

STEP2: start-all program-oracles-run SQL

STEP3: connect the SQL

STEP4: create a table named student and insert the values

STEP5: create another table named student course and insert the values

STEP6: use inner join, left join, full join and right join queries

STEP7: stop the process.

QUERIES:

```
SQL> create table student(rno varchar2(8),name varchar2(10),address varchar2 (10),phone number(10),age int);
```

Table created.

```
SQL> create table studentcourse (courseid varchar2(10),rno varchar2(15));
```

Table created.

```
SQL> desc student
```

Name	Null?	Type
RNO		VARCHAR2(8)
NAME		VARCHAR2(10)
ADDRESS		VARCHAR2(10)
PHONE		NUMBER(10)
AGE		INT

```
SQL> desc studentcourse
```

Name	Null?	Type
COURSEID		VARCHAR2(7)
RNO		VARCHAR2(8)

```
SQL> select * from student;
```

RNO	NAME	ADDRESS	PHONE	AGE
18CCU071	AJI	CBE	987654313	19
18CCU070	ABZUL	CBE	987654313	19
18CCU068	ABDUL	CBE	987654313	19
18CCU069	ABI	CBE	987654313	19

```
SQL> select * from studentcourse;
```

COURSEID	RNO
CCU101	18CCU070
CCU101	18CCU071
CCU102	18CCU073
CCU103	18CCU074
CCU101	18CCU072

```
SQL> select student.name,studentcourse.courseid,studentcourse.rno from student inner join studentcourse on student.rno = studentcourse.rno;
```

NAME	COURSEID	RNO
-----	-----	-----
AJI	CCU101	18CCU071
ABZUL	CCU101	18CCU070

```
SQL> select student.name,studentcourse.courseid,studentcourse.rno from student35 left join studentcourse on student.rno = studentcourse.rno;
```

NAME	COURSEID	RNO
-----	-----	-----
AJI	CCU101	18CCU071
ABZUL	CCU101	18CCU070
ABDUL		
ABI		

```
SQL> select student.name,studentcourse.courseid,studentcourse.rno from student35 right join studentcourse on student.rno = studentcourse.rno;
```

NAME	COURSEID	RNO
-----	-----	-----
ABZUL	CCU101	18CCU070
AJI	CCU101	18CCU071
	CCU102	18CCU073
	CCU103	18CCU074
	CCU101	18CCU072

```
SQL> select student.name,studentcourse.courseid,studentcourse.rno from student35 full join studentcourse on student.rno = studentcourse.rno;
```

NAME	COURSEID	RNO
-----	-----	-----
AJI	CCU101	18CCU071
ABZUL	CCU101	18CCU070
ABDUL		
ABI		
	CCU102	18CCU073
	CCU103	18CCU074
	CCU101	18CCU072

7 rows selected.

RESULT :

The above queries has been executed successfully.

Ex. No: 8	PL/SQL PROGRAM FOR EMPLOYEE PAY SLIP
Date:	

AIM:

To write a PL/SQL program for employee payroll

ALGORITHM:

STEP1: stop the process

STEP2: start- all programs-oracles-run SQL

STEP3: connect to SQL

STEP4: create a table named employee and insert the values like eno, name, basicpay

STEP5: calculate the da,hra,netpay with the given basicpay

STEP6: get the output

STEP7: stop the process

PROGRAM:

```
SQL> declare
  2 eno varchar(25);
  3 ename varchar(25);
  4 esalary number(5);
  5 edearnessallowance number(5);
  6 begin
  7 eno:='1';
  8 ename:='Sanjay';
  9 ebasicpay:='10000';
 10 eda:=ebasicpay*10/100;
 11 ehra:=ebasicpay*5/100;
 12 enetpay:=ebasicpay+eda+ehra;
 13 dbms_output.put_line('eno');
 14 dbms_output.put_line(enno);
 15 dbms_output.put_line('ename');
 16 dbms_output.put_line(ename);
 17 dbms_output.put_line('eda');
 18 dbms_output.put_line(eda);
 19 dbms_output.put_line('ehra');
 20 dbms_output.put_line(ehra);
 21 dbms_output.put_line('enetpay');
 22 dbms_output.put_line(enetpay);
 23 end;
 24 /
```

OUTPUT :

```
eno
1
ename
Sanjay
eda
1000
ehra
500
enetpay
11500
```

PL/SQL procedure successfully completed.

RESULT :

The above PL/SQL Program has been completed and executed successfully.

Ex. No: 9	PL/SQL PROGRAM FOR STUDENT DATABASE
Date:	

AIM:

To create program for student database and calculate total, average, result

ALGORITHM:

STEP1: start the process

STEP2: start –all programs-oracles-run SQL

STEP3: connect to SQL

STEP4: create a student table and declare the variables to be used in the student mark list

STEP5: enter the formula to calculate the average

STEP6: inner the marks

STEP7: total and average is calculated and printed

PROGRAM:

```
SQL> declare
  2  student varchar(15);
  3  dbms number(15);
  4  elective number(15);
  5  english number(15);
  6  ca number(15);
  7  total number(15);
  8  average number(15);
  9  begin
 10  dbms:=&dbms;
 11  elective:=&elective;
 12  english:=&english;
 13  ca:=&ca;
 14  total:=dbms+elective+english+ca;
 15  average:=total/4;
 16  dbms_output.put_line(total);
 17  dbms_output.put_line(' ');
 18  dbms_output.put_line(average);
 19  end;
 20 /
```

OUTPUT :

```
Enter value for dbms: 12
old 10: dbms:=&dbms;
new 10: dbms:=12;
Enter value for elective: 23
old 11: elective:=&elective;
new 11: elective:=23;
Enter value for english: 70
old 12: english:=&english;
new 12: english:=70;
Enter value for ca: 50
old 13: ca:=&ca;
new 13: ca:=50;

155
39
PL/SQL procedure successfully completed.
```

RESULT :

The above PL/SQL Program has been completed and executed successfully.