DEWAN VS GROUP OF INSTITUTIONS INDIA

Department of Artificial Intelligence
Academic Year 2025
(Session 2025-26)

LAB MANUAL

DATA BASE MANAGEMENT SYSTEM

BCS-551

B. Tech. 3rd Year CSE / AI

Dewan VS Institute of Engineering Technology, Meerut

Department of Artificial Intelligence

Academic Year 2025

(Session 2025-26)

DBMS LAB BCS-551 (5th Semester CSE)

COURSE OUTCOME (CO)	
AFTER STUDYING THIS COURSE, THE STUDENT WILL BE ABLE TO	
BCS-551.1	design and implement a database schema for a given problem-domain. Student could understand and remember certain features of Oracle, Entity relationship and creating Entity-Relationship diagrams.
BCS-551.2	analyze normalization and SQL/PL-SQL queries, sub queries for relational data base management system. To implement and execute cursor, procedure, functions, packages and triggers in oracle.
BCS-551.3	understand the concepts and techniques related to ODBC and its implementations. Students will design and implement a database for a given problems according to well-known design principles that balance data retrieval performance with data consistency.

LIST OF PROGRAMS

S. No.	*Name of Program*	*Date*	*Sign*
1.	SQL CONSISTING OF DDL, DML, DCL, TCL COMMANDS.		
2.	*CREATE, INSERT, UPDATE, DELETE, RENAME, TRUNCATE, ON TABLE.*		
3.	CREATING TABLES WITH CONSTRAINTS NOT NULL, CHECK, PRIMARY KEY, UNIQUE, FOREIGN KEY.		
4.	CREATING AND DROPING OF VIEWS		
5.	DRL-DATA RETRIEVAL LANGUAGE		
6.	SUBQUERIES (APPLYING IN, ALL, ANY, EXISTS, NOT EXISTS, UNION, INTERSECT, MINUS)		
7.	FUNCTIONS AGGREGATE, DATE, CONVERSION.		
8.	INTRODUCTION TO PL/SQL.		
9.	WRITE PL/SQL PROGRAMS • WRITE A PL/SQL PROGRAM TO SWAP TWO NUMBERS BY TAKING THIRD VARIABLE. • WRITE A PL/SQL PROGRAM TO SWAP TWO NUMBERS WITH OUT TAKING THIRD VARIABLE. • WRITE A PL/SQL PROGRAM TO FIND THE TOTAL AND AVERAGE OF 6 SUBJECTS AND DISPLAY THE GRAD		
10.	WRITE PL/SQL PROGRAMS • WRITE A PL/SQL CODE BLOCK TO CALCULATE THE AREA OF A CIRCLE FOR A VALUE OF RADIUS VARYING FROM 3 TO 7. STORE THE RADIUS AND THE CORRESPONDING VALUES OF CALCULATED AREA IN AN EMPTY TABLE NAMED AREAS, CONSISTING OF TWO COLUMNS RADIUS & AREA. • WRITE A PL/SQL CODE BLOCK THAT WILL ACCEPT AN ACCOUNT NUMBER FROM THE USER, CHECK IF THE USERS BALANCE IS LESS THAN MINIMUM BALANCE, ONLY THEN DEDUCT RS.100/- FROM THE BALANCE.THIS PROCESS IS FIRED ON THE ACCT TABLE.		

Program No. 1

Object: SQL Consisting of DDL,DML,DCL,TCL COMMANDS.

SYNTAX'S OF COMMANDS

CREATE TABLE

```
CREATE TABLE table_name (
column_name1   data_type,
column_name2   data_type,
column_name3 data_type,
....
);
```

ALTER A TABLE

To add a column in a table

```
ALTER TABLE table_name ADD
column_name datatype;
```

To delete a column in a table

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

DROP TABLE

```
DROP TABLE table_name;
```

TRUNCATE TABLE

```
TRUNCATE TABLE table_name;
```

INSERT

```
INSERT INTO table_name  VALUES
(value1, value2, value3,...);
```

(O R)

```
INSERT INTO table_name (column1, column2, column3,...) VALUES
(value1, value2, value3,...);
```

UPDATE

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value;
```

DELETE

```
DELETE FROM table_name WHERE
some_column=some_value;
```

SELECT

```
SELECT column_name(s)
FROM table_name;
```

Program No. 2

Object: **CREATE, INSERT, UPDATE, DELETE, RENAME, TRUNCATE, ON TABLES**

SQL>CREATE TABLE STUDENT (SNO NUMBER (5), SNAME VARCHAR2 (15), DOJ DATE);

OUTPUT:-TABLE CREATED

SQL> INSERT INTO STUDENT VALUES (&SNO,'&SNAME','&DOJ');

OUTPUT:-

Enter value for sno: 501
Enter value for sname: ABI
Enter value for doj: 12-OCT-07
old 1: INSERT INTO STUDENT100 VALUES(&SNO,'&SNAME','&DOJ')
new 1: INSERT INTO STUDENT100 VALUES(501,'ABI','12-OCT-07')

1 row created.

SQL> /

Enter value for sno: 502
Enter value for sname: ASHOK
Enter value for doj: 03-OCT-07
old 1: INSERT INTO STUDENT100 VALUES(&SNO,'&SNAME','&DOJ')
new 1: INSERT INTO STUDENT100 VALUES(502,'ASHOK','03-OCT-07')

1 row created.

SQL> /

Enter value for sno: 503
Enter value for sname: BHAVYA
Enter value for doj: 10-OCT-07
old 1: INSERT INTO STUDENT100 VALUES(&SNO,'&SNAME','&DOJ')
new 1: INSERT INTO STUDENT100 VALUES(503,'BHAVYA','10-OCT-07')

1 row created.
SQL> /

```
Enter value for sno: 504
Enter value for sname: AKASH
Enter value for doj: 05-OCT-07
old   1: INSERT INTO STUDENT100 VALUES(&SNO,'&SNAME','&DOJ')
new  1: INSERT INTO STUDENT100 VALUES(504,'AKASH','05-OCT-07')

1 row created.
SQL> /

Enter value for sno: 505
Enter value for sname: NIKHIL
Enter value for doj: 08-OCT-07
old   1: INSERT INTO STUDENT100 VALUES(&SNO,'&SNAME','&DOJ')
new  1: INSERT INTO STUDENT100 VALUES(505,'NIKHIL','08-OCT-07')

1 row created.
```

```
SQL> RENAME STUDENT TO CSESTUDENT;
```

OUTPUT:-Table renamed.

```
SQL>SELECT * FROM CSESTUDENT;
```

OUTPUT:-

```
     SNO SNAME          DOJ
- - - - - - ---------- - - - - -
     501 ABI            12-OCT-07
     502 ASHOK          03-OCT-07
     503 BHAVYA         10-OCT-07
     504 AKASH          05-OCT-07
     505 NIKHIL         08-OCT-07
```

```
SQL> UPDATE CSESTUDENT SET SNAME='ABILASH' WHERE
SNAME='ABI';
```

OUTPUT:-1 row updated.

```
SQL> ALTER TABLE CSESTUDENT ADD BRANCH VARCHAR2(6);
```

OUTPUT:-Table altered.

```
SQL> UPDATE CSESTUDENT SET BRANCH='CSE' WHERE SNO=501;
```

OUTPUT:-1 row updated.

SQL> UPDATE CSESTUDENT SET BRANCH='CSE' WHERE SNO=502;

OUTPUT:-1 row updated.

SQL> UPDATE CSESTUDENT SET BRANCH='CSE' WHERE SNO=503;

OUTPUT:-1 row updated.

SQL> UPDATE CSESTUDENT SET BRANCH='CSE' WHERE SNO=504;

OUTPUT:-1 row updated.

SQL> UPDATE CSESTUDENT SET BRANCH='CSE' WHERE SNO=505;

OUTPUT:-1 row updated.

SQL> SELECT * FROM CSESTUDENT;

SNO	SNAME	DOJ	BRANCH
501	ABILASH	12-OCT-07	CSE
502	ASHOK	03-OCT-07	CSE
503	BHAVYA	10-OCT-07	CSE
504	AKASH	05-OCT-07	CSE
505	NIKHIL	08-OCT-07	CSE

SQL> DELETE FROM CSESTUDENT WHERE SNO=501;

OUTPUT:-1 row deleted.

SQL> SELECT * FROM CSESTUDENT;

OUTPUT:-

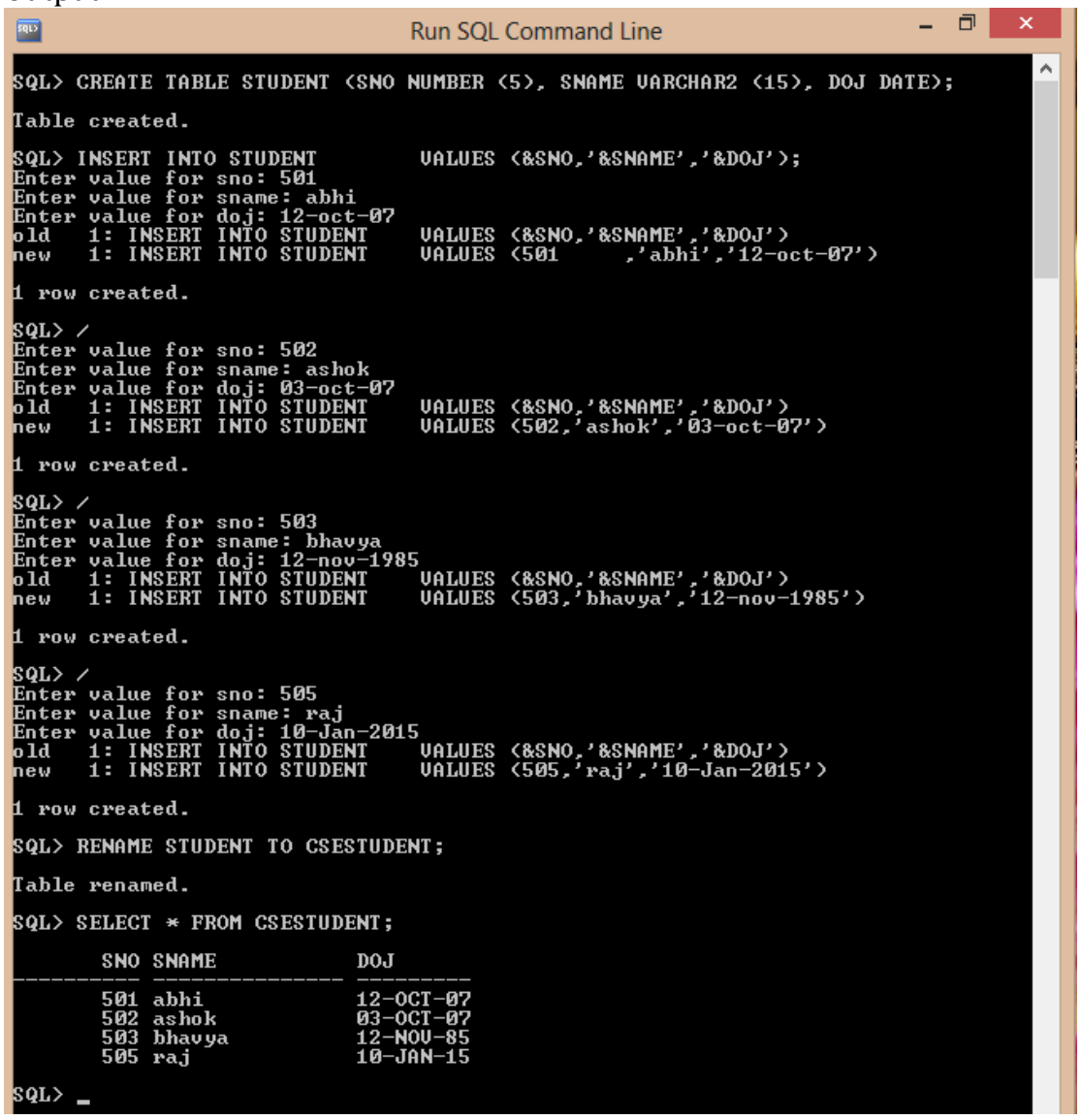
SNO	SNAME	DOJ	BRANCH
502	ASHOK	03-OCT-07	CSE
503	BHAVYA	10-OCT-07	CSE

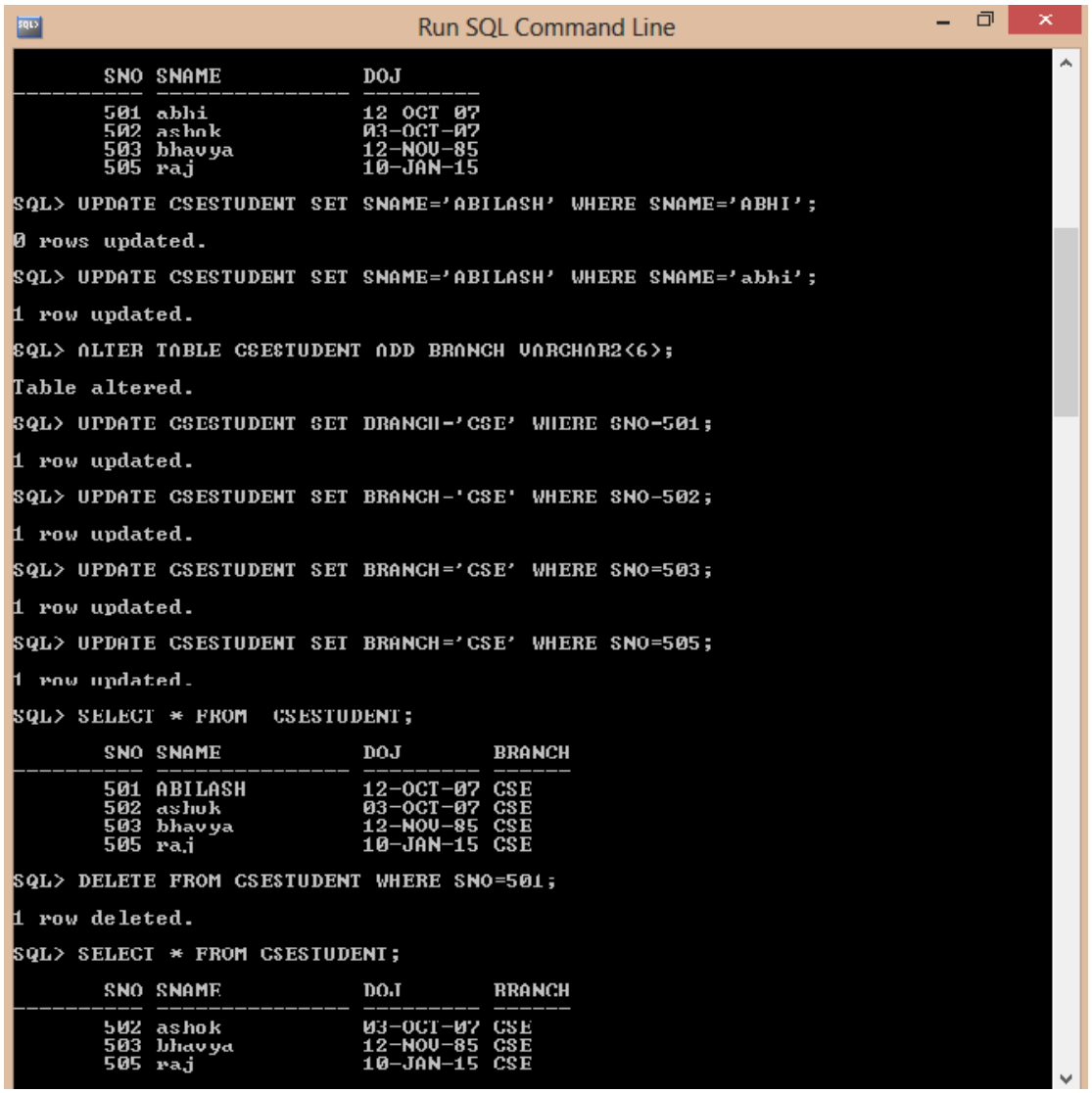
504 AKASH 05-OCT-07 CSE
505 NIKHIL 08-OCT-07 CSE
502 ASHOK 03-OCT-07 CSE

SQL> DROP TABLE CSESTUDENT;

OUTPUT:-Table dropped.

Output-




Run SQL Command Line
SNO SNAME DOJ
501 abhi 12 OCT 07
502 ashok 03-OCT-07
503 bhavya 12-NOV-85
505 raj 10-JAN-15
SQL> UPDATE CSESTUDENT SET SNAME='ABILASH' WHERE SNAME='ABHI';
0 rows updated.
SQL> UPDATE CSESTUDENT SET SNAME='ABILASH' WHERE SNAME='abhi';
1 row updated.
SQL> ALTER TABLE CSESTUDENT ADD BRANCH VARCHAR2(6);
Table altered.
SQL> UPDATE CSESTUDENT SET BRANCH='CSE' WHERE SNO=501;
1 row updated.
SQL> UPDATE CSESTUDENT SET BRANCH='CSE' WHERE SNO=502;
1 row updated.
SQL> UPDATE CSESTUDENT SET BRANCH='CSE' WHERE SNO=503;
1 row updated.
SQL> UPDATE CSESTUDENT SET BRANCH='CSE' WHERE SNO=505;
1 row updated.
SQL> SELECT * FROM CSESTUDENT;
SNO SNAME DOJ BRANCH
501 ABILASH 12-OCT-07 CSE
502 ashok 03-OCT-07 CSE
503 bhavya 12-NOV-85 CSE
505 raj 10-JAN-15 CSE
SQL> DELETE FROM CSESTUDENT WHERE SNO=501;
1 row deleted.
SQL> SELECT * FROM CSESTUDENT;
SNO SNAME DOJ BRANCH
502 ashok 03-OCT-07 CSE
503 bhavya 12-NOV-85 CSE
505 raj 10-JAN-15 CSE

Program No. 3

Object: **CREATING TABLES WITH CONSTRAINTS (NOT NULL)**

```
SQL> CREATE TABLE STUD(ROLLNO NUMBER(6) NOT NULL,NAME VARCHAR2(10),BRANCH VARCHAR2(6));

Table created.

SQL> DESC STUD;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROLLNO                                    NOT NULL NUMBER(6)
 NAME                                               VARCHAR2(10)
 BRANCH                                             VARCHAR2(6)


SQL> INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH');
Enter value for rollno: 501
Enter value for name: ABHILASH
Enter value for branch: CSE
old   1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new   1: INSERT INTO STUD VALUES(501,'ABHILASH','CSE')

1 row created.

SQL> /
Enter value for rollno: 502
Enter value for name: ABI
Enter value for branch: CSE
old   1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new   1: INSERT INTO STUD VALUES(502,'ABI','CSE')

1 row created.

SQL> SELECT * FROM STUD;

    ROLLNO NAME       BRANCH
---------- ---------- -------
       501 ABHILASH   CSE
       502 ABI        CSE


SQL> INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH');
```

```
Enter value for rollno:
Enter value for name: BHAVYA
Enter value for branch: CSE
old  1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new  1: INSERT INTO STUD VALUES(,'BHAVYA','CSE')
INSERT INTO STUD VALUES(,'BHAVYA','CSE')
                        *
ERROR:CANNOT INSERT NULL INTO("SCOTT",'STUD',ROLLNO)
```

(UNIQUE)

```
SQL> CREATE TABLE STUD(ROLLNO NUMBER(6) UNIQUE ,NAME
VARCHAR2(10),BRANCH VARCHAR2(6));

Table created.

SQL> INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH');
Enter value for rollno: 501
Enter value for name: abhilash
Enter value for branch: cse
old    1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new  1: INSERT INTO STUD VALUES(501,'abhilash','cse')

1 row created.

SQL> /
Enter value for rollno: 502
Enter value for name: ABI
Enter value for branch: CSE
old    1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new  1: INSERT INTO STUD VALUES(502,'ABI','CSE')

1 row created.



SQL> /
Enter value for rollno: 502
Enter value for name: BHAVYA
Enter value for branch: CSE
old  1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new  1: INSERT INTO STUD VALUES(502,'BHAVYA','CSE')
INSERT INTO STUD VALUES(502,'BHAVYA','CSE')
*
ERROR at line 1:
ORA-00001: unique constraint (SCOTT.SYS_C001290) violated
```

(PRIMARY KEY)

```
SQL> CREATE TABLE STUD(ROLLNO NUMBER(6) PRIMARY KEY ,NAME
VARCHAR2(10),BRANCH VARCHAR2(6));

Table created.

SQL> INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH');
Enter value for rollno: 501
Enter value for name: abhilash
Enter value for branch: cse
old   1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new   1: INSERT INTO STUD VALUES(501,'abhilash','cse')

1 row created.

SQL> /
Enter value for rollno: 502
Enter value for name: ABI
Enter value for branch: CSE
old   1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new   1: INSERT INTO STUD VALUES(502,'ABI','CSE')

1 row created.



SQL> /
Enter value for rollno: 502
Enter value for name: BHAVYA
Enter value for branch: CSE
old   1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new   1: INSERT INTO STUD VALUES(502,'BHAVYA','CSE')
INSERT INTO STUD VALUES(502,'BHAVYA','CSE')
*
ERROR at line 1:
ORA-00001: unique constraint (SCOTT.SYS_C001290) violated



SQL> INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH');
Enter value for rollno:
Enter value for name: BHAVYA
Enter value for branch: CSE
old   1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
```

```
new  1: INSERT INTO STUD VALUES(,'BHAVYA','CSE')
INSERT INTO STUD VALUES(,'BHAVYA','CSE')
                *
ERROR:CANNOT  INSERT  NULL  INTO("SCOTT",'STUD',ROLLNO)



SQL> SELECT * FROM STUD;

   ROLLNO NAME        BRANCH
- - - - - - ---------- -------
      501 ABHILASH   CSE
      502 ABI        CSE
```

(CHECK)

```
SQL>  create table stud1(rno number(5),name varchar2(10),sal number(10) constraint no_ck check(sal between 10000 and 30000));

Table created.

SQL> insert into stud1 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)

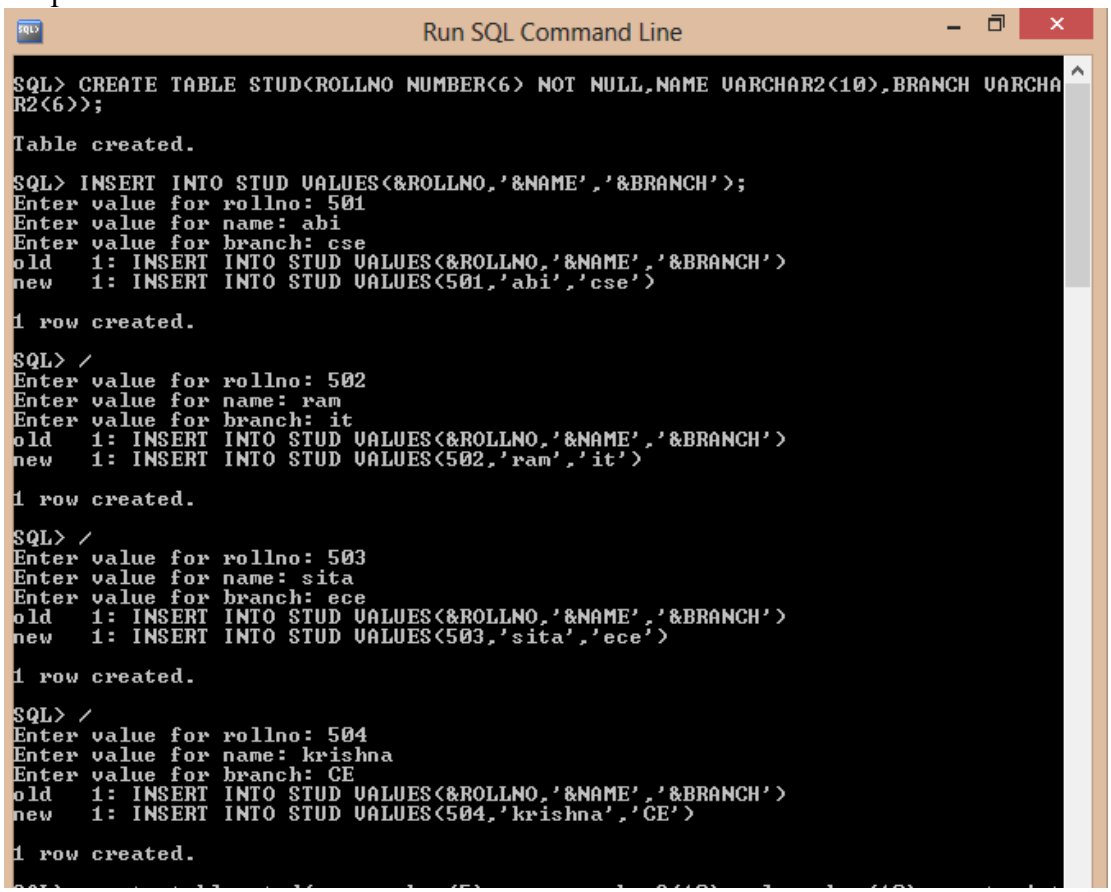
```
SOL>create table adm(stuid number(6) constraint stuid_pk primary
key,sname varchar2(15),per number(5));
```

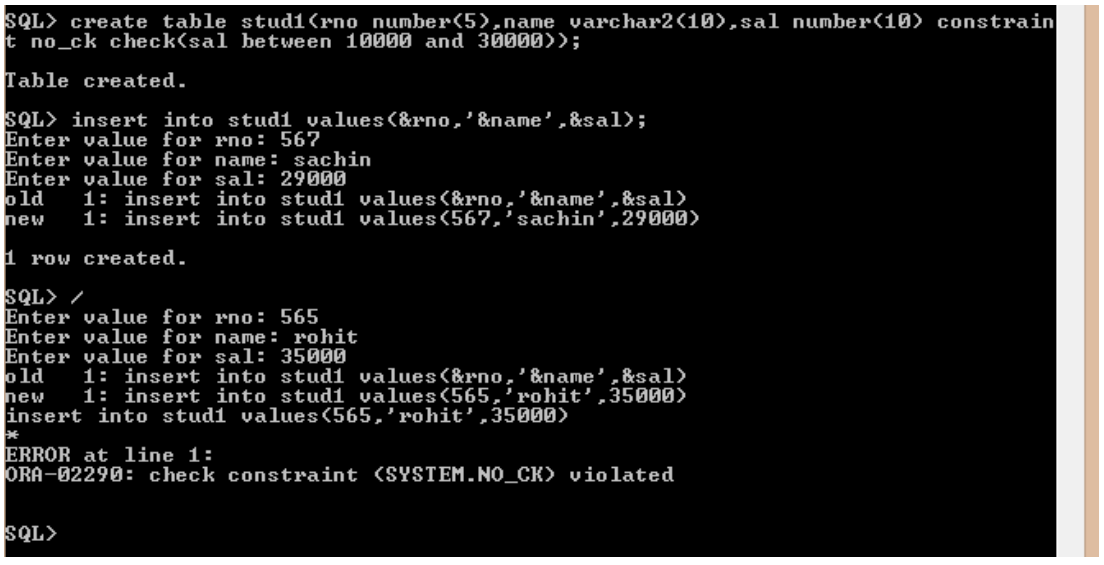
Table created.

```
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
```

Output-


Run SQL Command Line
SQL> CREATE TABLE STUD(ROLLNO NUMBER(6) NOT NULL,NAME VARCHAR2(10),BRANCH VARCHAR2(6));
Table created.
SQL> INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH');
Enter value for rollno: 501
Enter value for name: abi
Enter value for branch: cse
old 1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new 1: INSERT INTO STUD VALUES(501,'abi','cse')
1 row created.
SQL> /
Enter value for rollno: 502
Enter value for name: ram
Enter value for branch: it
old 1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new 1: INSERT INTO STUD VALUES(502,'ram','it')
1 row created.
SQL> /
Enter value for rollno: 503
Enter value for name: sita
Enter value for branch: ece
old 1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new 1: INSERT INTO STUD VALUES(503,'sita','ece')
1 row created.
SQL> /
Enter value for rollno: 504
Enter value for name: krishna
Enter value for branch: CE
old 1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&BRANCH')
new 1: INSERT INTO STUD VALUES(504,'krishna','CE')
1 row created.


SQL> create table stud1(rno number(5),name varchar2(10),sal number(10) constraint no_ck check(sal between 10000 and 30000));
Table created.
SQL> insert into stud1 values(&rno,'&name',&sal);
Enter value for rno: 567
Enter value for name: sachin
Enter value for sal: 29000
old 1: insert into stud1 values(&rno,'&name',&sal)
new 1: insert into stud1 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 stud1 values(&rno,'&name',&sal)
new 1: insert into stud1 values(565,'rohit',35000)
insert into stud1 values(565,'rohit',35000)
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.NO_CK) violated
SQL>

Program No. 4

Object: CREATING AND DROPING OF VIEWS

SQL> select * from emp2;

OUTPUT:-

ENAME	STREET	CITY
coyote	toon	hollywood
rabbit	tunnel	carrot ville
smith	revolver	death valley
williams	sea view	sea attle

SQL> create view emp22 as select * from emp2;

OUTPUT:-

View created.

SQL> select * from emp22;

OUTPUT:-

ENAME	STREET	CITY
coyote	toon	hollywood
rabbit	tunnel	carrot ville
smith	revolver	death valley
williams	sea view	sea attle

SQL> update emp22 set city='hyd' where ename='coyote';

OUTPUT:-

1 row updated.

SQL> select * from emp2;

OUTPUT:-

ENAME	STREET	CITY
coyote	toon	hyd
rabbit	tunnel	carrot ville

smith revolver death valley
williams sea view sea attle

SQL> select * from emp22;

OUTPUT:-

ENAME	STREET	CITY
coyote	toon	hyd
rabbit	tunnel	carrot ville
smith	revolver	death valley
williams	sea view	sea attle

SQL> drop table emp2;

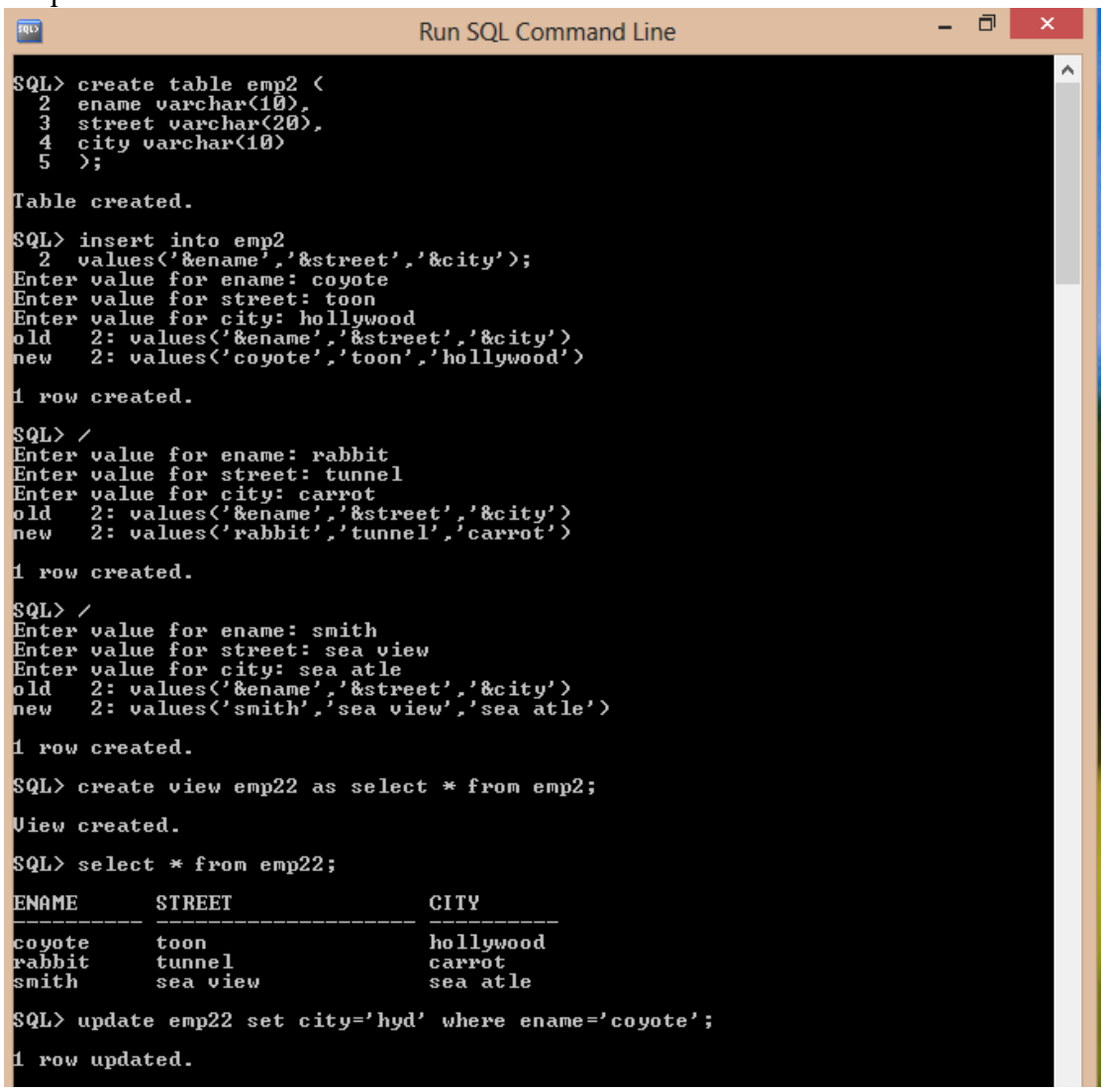
OUTPUT:-
Table dropped.

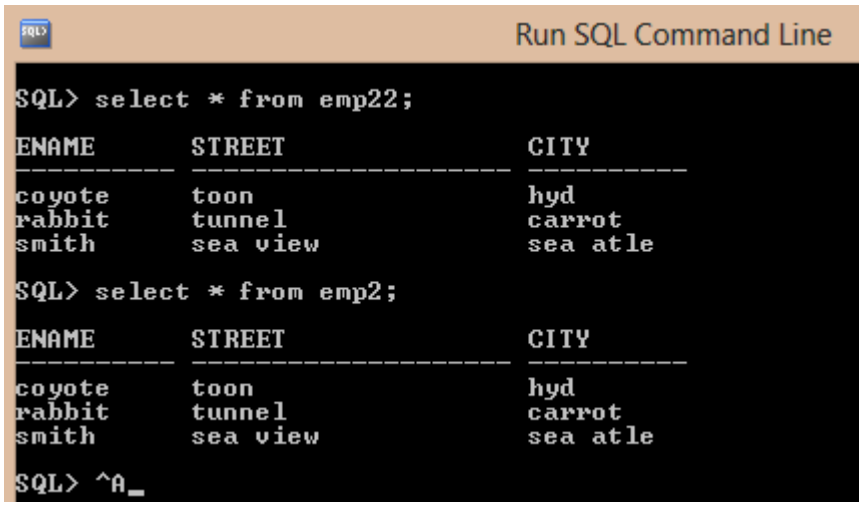
SQL> select * from emp22;

OUTPUT:-

```
select * from emp22
          *
ERROR at line 1:
ORA-04063: view "SCOTT.EMP22" has errors
```

Output-


Run SQL Command Line
SQL> create table emp2 (
2 ename varchar(10),
3 street varchar(20),
4 city varchar(10)
5);
Table created.
SQL> insert into emp2
2 values('&ename','&street','&city');
Enter value for ename: coyote
Enter value for street: toon
Enter value for city: hollywood
old 2: values('&ename','&street','&city')
new 2: values('coyote','toon','hollywood')
1 row created.
SQL> /
Enter value for ename: rabbit
Enter value for street: tunnel
Enter value for city: carrot
old 2: values('&ename','&street','&city')
new 2: values('rabbit','tunnel','carrot')
1 row created.
SQL> /
Enter value for ename: smith
Enter value for street: sea view
Enter value for city: sea atle
old 2: values('&ename','&street','&city')
new 2: values('smith','sea view','sea atle')
1 row created.
SQL> create view emp22 as select * from emp2;
View created.
SQL> select * from emp22;
ENAME STREET CITY
coyote toon hollywood
rabbit tunnel carrot
smith sea view sea atle
SQL> update emp22 set city='hyd' where ename='coyote';
1 row updated.


Run SQL Command Line
SQL> select * from emp22;
ENAME STREET CITY
coyote toon hyd
rabbit tunnel carrot
smith sea view sea atle
SQL> select * from emp2;
ENAME STREET CITY
coyote toon hyd
rabbit tunnel carrot
smith sea view sea atle
SQL> ^A

Program No. 5

Object: DRL-DATA RETRIEVAL LANGUAGE

IMPLEMENTING SELECT COMMANDS

```
CREATE TABLE Emp
(
EmpNo number (5),
EName VarChar (15),
Job Char (10),
Mgr number (5),
Hiredate Date,
DeptNo number (5)
);
```

```
SQL>     insert into emp Values(&empno,'&ename','&job',&mgr,'&date',&deptno,&SAL);
```

SQL> select * from emp;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10

```
7844 TURNER      SALESMAN        7698 08-SEP-81      1500        0    30

7876 ADAMS       CLERK           7788 23-MAY-87      1100             20

7900 JAMES       CLERK           7698 03-DEC-81      950              30

7902 FORD        ANALYST         7566 03-DEC-81      3000             20

7934 MILLER      CLERK           7782 23-JAN-82      1300             10


14 rows selected.


SQL> select empno,ename,sal from emp;

     EMPNO ENAME           SAL
---------- ---------- ----------
      7369 SMITH            800
      7499 ALLEN           1600
      7521 WARD            1250
      7566 JONES           2975
      7654 MARTIN          1250
      7698 BLAKE           2850
      7782 CLARK           2450
      7788 SCOTT           3000
      7839 KING            5000
      7844 TURNER          1500
      7876 ADAMS           1100

     EMPNO ENAME           SAL
----------
      7900 JAMES            950
      7902 FORD            3000
      7934 MILLER          1300

14 rows selected.

SQL> select empno,ename,sal from emp where sal between 2500 and 5000;

     EMPNO ENAME           SAL
```

```
------ ---------- ----------
  7566 JONES           2975
  7698 BLAKE           2850
  7788 SCOTT           3000
  7839 KING            5000
  7902 FORD            3000
```

SQL>select ename,job,sal,deptno from emp where sal not between 1500 and 5000;

```
ENAME      JOB           SAL     DEPTNO
------ --------- ------- -------
SMITH      CLERK          800       20
WARD       SALESMAN       1250      30
MARTIN     SALESMAN       1250      30
ADAMS      CLERK          1100      20
JAMES      CLERK          950       30
MILLER     CLERK          1300      10
```

6 rows selected.

SQL> select empno,ename,sal from emp where sal in (800,5000);

```
   EMPNO ENAME           SAL
------ ---------- ----------
  7369 SMITH            800
  7839 KING            5000
```

SQL> select empno,ename,sal from emp where sal not in(800,1250,3000,5000);

```
   EMPNO ENAME           SAL
------ ---------- ----------
  7499 ALLEN            1600
  7566 JONES           2975
  7698 BLAKE           2850
  7782 CLARK           2450
  7844 TURNER          1500
  7876 ADAMS           1100
  7900 JAMES           950
  7934 MILLER          1300
```

8 rows selected.

SQL> select empno,ename,sal from emp where comm is null;

EMPNO	ENAME	SAL
7369	SMITH	800
7566	JONES	2975
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7876	ADAMS	1100
7900	JAMES	950
7902	FORD	3000
7934	MILLER	1300

10 rows selected.

SQL> select empno,ename,sal from emp where comm is not null;

EMPNO	ENAME	SAL
7499	ALLEN	1600
7521	WARD	1250
7654	MARTIN	1250
7844	TURNER	1500

SQL> select empno,ename,job,sal from emp where ename like'S%';

EMPNO	ENAME	JOB	SAL
7369	SMITH	CLERK	800
7788	SCOTT	ANALYST	3000

SQL> select empno,ename,job,sal from emp where job not like'S%';

EMPNO	ENAME	JOB	SAL
7369	SMITH	CLERK	800
7566	JONES	MANAGER	2975
7698	BLAKE	MANAGER	2850
7782	CLARK	MANAGER	2450
7788	SCOTT	ANALYST	3000
7839	KING	PRESIDENT	5000

```
7876 ADAMS       CLERK         1100
7900 JAMES      CLERK          950
7902 FORD       ANALYST        3000
7934 MILLER      CLERK         1300
```

10 rows selected.

SQL> select ename,job,sal from emp where sal>2500;

```
ENAME      JOB            SAL
---------------------------------
JONES     MANAGER        2975
BLAKE      MANAGER        2850
SCOTT     ANALYST        3000
KING      PRESIDENT      5000
FORD      ANALYST        3000
```

SQL> select ename,job,sal from emp where sal<2500;

```
ENAME      JOB            SAL
--------------
SMITH      CLERK          800
ALLEN      SALESMAN       1600
WARD       SALESMAN       1250
MARTIN     SALESMAN       1250
CLARK      MANAGER        2450
TURNER     SALESMAN       1500
ADAMS      CLERK          1100
JAMES      CLERK          950
MILLER     CLERK          1300
```

9 rows selected.

SQL> select empno,ename,job,sal from emp order by sal;

```
    EMPNO ENAME      JOB            SAL
- - - - - - ---------- ---------- - - - - - -
     7369 SMITH       CLERK          800
     7900 JAMES      CLERK          950
     7876 ADAMS       CLERK         1100
     7521 WARD       SALESMAN       1250
     7654 MARTIN      SALESMAN       1250
     7934 MILLER      CLERK         1300
     7844 TURNER      SALESMAN       1500
```

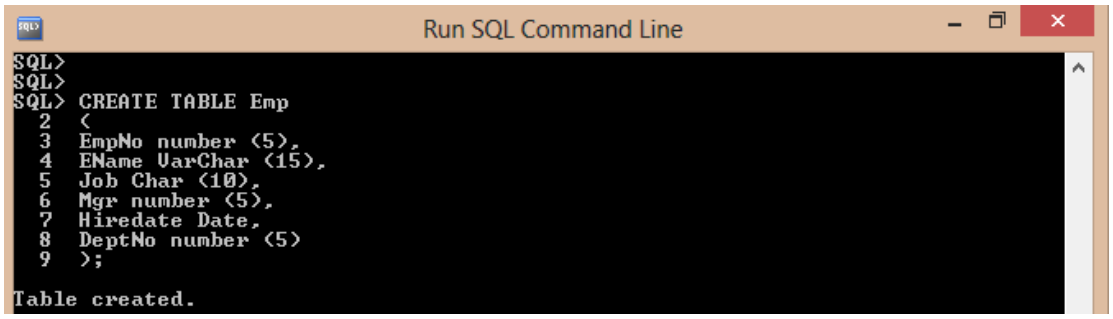
```
7499 ALLEN      SALESMAN      1600
7782 CLARK      MANAGER       2450
7698 BLAKE      MANAGER       2850
7566 JONES      MANAGER       2975

   EMPNO ENAME      JOB              SAL
- - - - - - ---------- ---------- - - - - - -
     7788 SCOTT      ANALYST       3000
     7902 FORD       ANALYST       3000
     7839 KING       PRESIDENT     5000
```

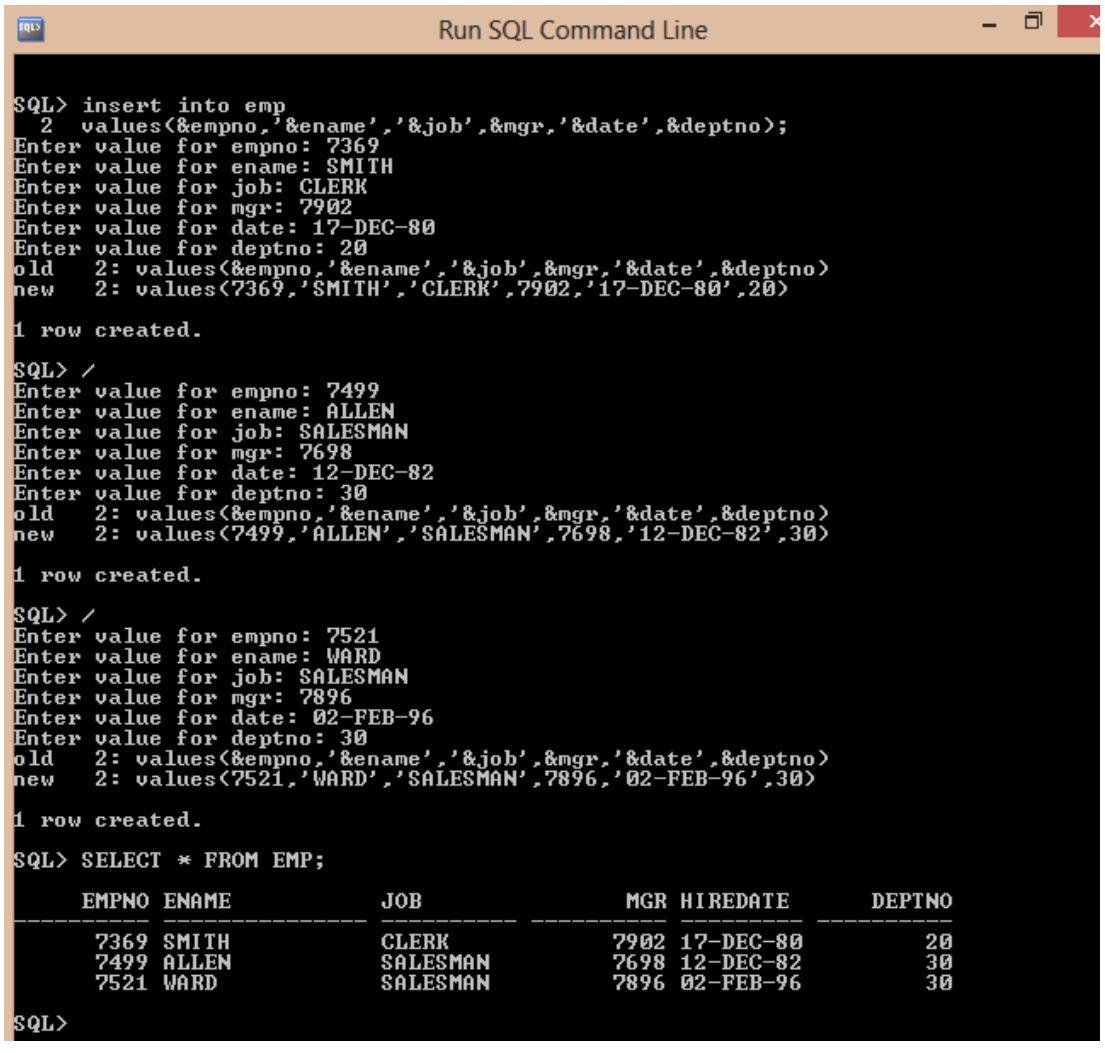
14 rows selected.

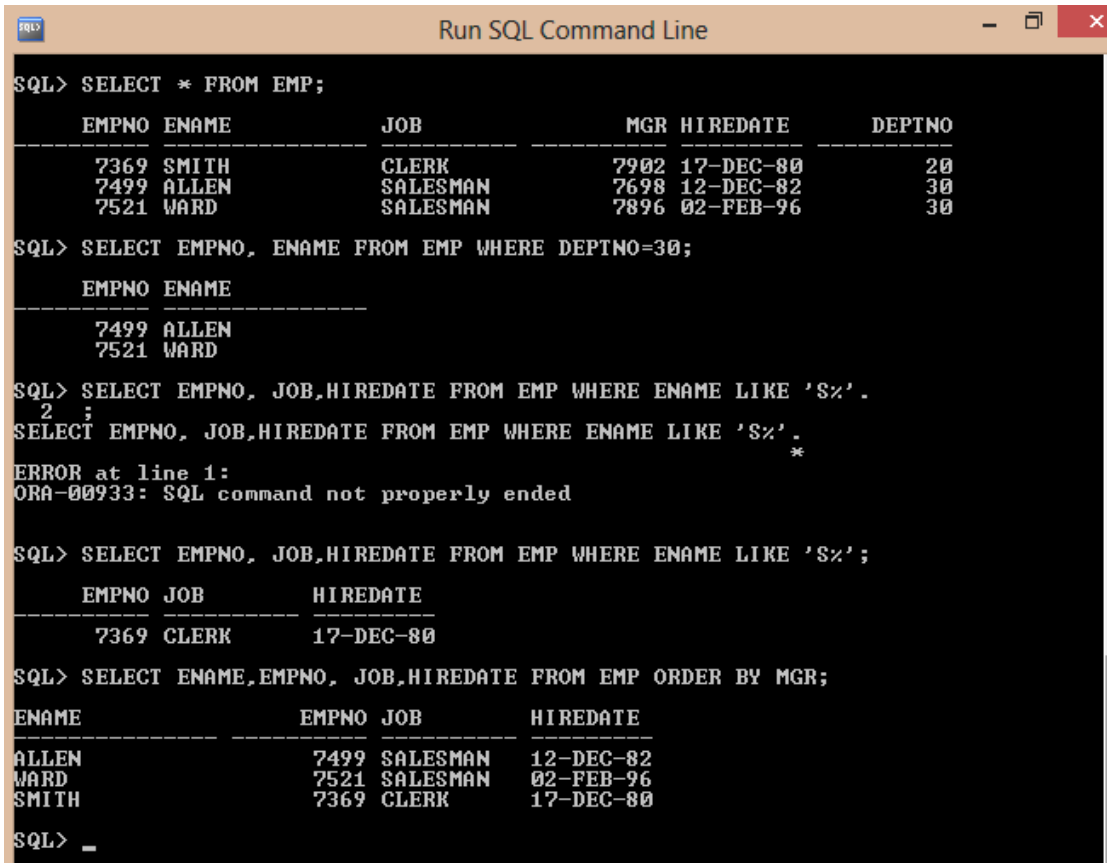
SQL> select empno,ename,job,sal from emp order by sal desc;

```
   EMPNO ENAME      JOB              SAL
- - - - - - ---------- ---------- - - - - - -
     7839 KING       PRESIDENT     5000
     7788 SCOTT      ANALYST       3000
     7902 FORD       ANALYST       3000
     7566 JONES      MANAGER       2975
     7698 BLAKE      MANAGER       2850
     7782 CLARK      MANAGER       2450
     7499 ALLEN      SALESMAN      1600
     7844 TURNER     SALESMAN      1500
     7934 MILLER     CLERK         1300
     7521 WARD       SALESMAN      1250
     7654 MARTIN     SALESMAN      1250

   EMPNO ENAME      JOB              SAL
- - - - - - ---------- ---------- - - - - - -
     7876 ADAMS      CLERK         1100
     7900 JAMES      CLERK         950
     7369 SMITH      CLERK         800
```

14 rows selected.


Run SQL Command Line
SQL>
SQL>
SQL> CREATE TABLE Emp
  2 (
  3 EmpNo number (5),
  4 EName VarChar (15),
  5 Job Char (10),
  6 Mgr number (5),
  7 Hiredate Date,
  8 DeptNo number (5)
  9);
Table created.


Run SQL Command Line
SQL> insert into emp
  2 values(&empno,'&ename','&job',&mgr,'&date',&deptno);
Enter value for empno: 7369
Enter value for ename: SMITH
Enter value for job: CLERK
Enter value for mgr: 7902
Enter value for date: 17-DEC-80
Enter value for deptno: 20
old   2: values(&empno,'&ename','&job',&mgr,'&date',&deptno)
new   2: values(7369,'SMITH','CLERK',7902,'17-DEC-80',20)
1 row created.
SQL> /
Enter value for empno: 7499
Enter value for ename: ALLEN
Enter value for job: SALESMAN
Enter value for mgr: 7698
Enter value for date: 12-DEC-82
Enter value for deptno: 30
old   2: values(&empno,'&ename','&job',&mgr,'&date',&deptno)
new   2: values(7499,'ALLEN','SALESMAN',7698,'12-DEC-82',30)
1 row created.
SQL> /
Enter value for empno: 7521
Enter value for ename: WARD
Enter value for job: SALESMAN
Enter value for mgr: 7896
Enter value for date: 02-FEB-96
Enter value for deptno: 30
old   2: values(&empno,'&ename','&job',&mgr,'&date',&deptno)
new   2: values(7521,'WARD','SALESMAN',7896,'02-FEB-96',30)
1 row created.
SQL> SELECT * FROM EMP;
EMPNO ENAME JOB MGR HIREDATE DEPTNO
7369 SMITH CLERK 7902 17-DEC-80 20
7499 ALLEN SALESMAN 7698 12-DEC-82 30
7521 WARD SALESMAN 7896 02-FEB-96 30
SQL>


Run SQL Command Line
SQL> SELECT * FROM EMP;
EMPNO ENAME JOB MGR HIREDATE DEPTNO
7369 SMITH CLERK 7902 17-DEC-80 20
7499 ALLEN SALESMAN 7698 12-DEC-82 30
7521 WARD SALESMAN 7896 02-FEB-96 30
SQL> SELECT EMPNO, ENAME FROM EMP WHERE DEPTNO=30;
EMPNO ENAME
7499 ALLEN
7521 WARD
SQL> SELECT EMPNO, JOB,HIREDATE FROM EMP WHERE ENAME LIKE 'S%'.
2 ;
SELECT EMPNO, JOB,HIREDATE FROM EMP WHERE ENAME LIKE 'S%'.
*
ERROR at line 1:
ORA-00933: SQL command not properly ended
SQL> SELECT EMPNO, JOB,HIREDATE FROM EMP WHERE ENAME LIKE 'S%';
EMPNO JOB HIREDATE
7369 CLERK 17-DEC-80
SQL> SELECT ENAME,EMPNO, JOB,HIREDATE FROM EMP ORDER BY MGR;
ENAME EMPNO JOB HIREDATE
ALLEN 7499 SALESMAN 12-DEC-82
WARD 7521 SALESMAN 02-FEB-96
SMITH 7369 CLERK 17-DEC-80
SQL> _

Program No. 6

Object: **SUBQUERIES (APPLYING IN, ALL, ANY, EXISTS, NOT EXISTS, UNION, INTERSECT, MINUS)**

SQL>select * from emp;

OUTPUT:-

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20

```
    7900 JAMES      CLERK           7698      03-DEC-81          950
30

    7902 FORD       ANALYST         7566      03-DEC-81         3000
              20

    7934 MILLER     CLERK           7782      23-JAN-82         1300
              10
```

SQL>select * from dept;

OUTPUT:-

```
 DEPTNO DNAME          LOC
------- -------------- --------
     10 ACCOUNTING     NEW YORK
     20 RESEARCH       DALLAS
     30 SALES          CHICAGO
     40 OPERATIONS     BOSTON
```

SQL> select * from emp where sal in(3000,5000);

OUTPUT:-

```
  EMPNO ENAME      JOB         MGR             HIREDATE          SAL
COMM        DEPTNO
------- ---------- ------- ------ ------- ------
------- ---------- --------

    7788 SCOTT      ANALYST         7566 19-APR-87          3000
20
    7839 KING       PRESIDENT                 17-NOV-81
5000                    10
    7902 FORD       ANALYST         7566 03-DEC-81          3000
20
```

SQL> select empno,ename from emp where sal in(select max(sal) from emp);

OUTPUT:-

```
 EMPNO ENAME
------- ----------
    7839 KING
```

SQL> select empno,ename from emp where sal in(select max(sal) from emp group by deptno);

OUTPUT:-

```
 EMPNO ENAME
---------- ----------
    7698 BLAKE
    7788 SCOTT
    7902 FORD
    7839 KING
```

SQL> select empno,ename,job,sal from emp where sal>all(select avg(sal) from emp group by deptno);

OUTPUT:-

```
EMPNO ENAME      JOB              SAL
------ ---------- ------ ------
    7566 JONES      MANAGER       2975
    7788 SCOTT      ANALYST       3000
    7839 KING       PRESIDENT     5000
    7902 FORD       ANALYST       3000
```

SQL> select empno,ename,job,sal from emp where sal<all(select avg(sal) from emp group by deptno);

OUTPUT:-

```
 EMPNO ENAME      JOB              SAL
------ ---------- ------ ------
    7369 SMITH      CLERK             800
    7521 WARD       SALESMAN         1250
     7654 MARTIN    SALESMAN         1250
     7844 TURNER    SALESMAN         1500
    7876 ADAMS      CLERK            1100
    7900 JAMES      CLERK             950
    7934 MILLER     CLERK            1300
```

7 rows selected.

SQL> select ename,job,sal from emp where sal>any(select sal from emp where job='CLERK');

OUTPUT:-

```
ENAME      JOB              SAL
```

```
          ---------------------------
ALLEN      SALESMAN       1600
WARD       SALESMAN       1250
JONES      MANAGER        2975
MARTIN     SALESMAN       1250
BLAKE      MANAGER        2850
CLARK      MANAGER        2450
SCOTT      ANALYST        3000
KING       PRESIDENT      5000
TURNER     SALESMAN       1500
ADAMS      CLERK          1100
JAMES      CLERK           950

ENAME      JOB             SAL
---------- ---------------------------
FORD       ANALYST         3000
MILLER     CLERK           1300
```

13 rows selected.

SQL> select ename,job,sal from emp where sal<any(select sal from emp where job='CLERK');

OUTPUT:-

```
ENAME      JOB               SAL
---------- ---------------------------
SMITH      CLERK              800
WARD       SALESMAN          1250
MARTIN     SALESMAN          1250
ADAMS      CLERK             1100
JAMES      CLERK              950
```

SQL> select ename,job,sal from emp where sal=any(select sal from emp where job='CLERK');

OUTPUT:-

```
ENAME      JOB             SAL
---------- ---------- ----------
SMITH      CLERK           800
JAMES      CLERK           950
ADAMS      CLERK          1100
MILLER     CLERK          1300
```

SQL> select deptno,dname from dept d where exists(select * from emp e where d.deptno=e.deptno);

OUTPUT:-

```
   DEPTNO DNAME
- - - - - -      ---------------
       10       ACCOUNTING
       20       RESEARCH
       30       SALES
```

SQL> select deptno,dname from dept d where not exists(select * from emp e where d.deptno=e.deptno);

OUTPUT:-

```
 DEPTNO    DNAME
- - - - - -      ---------------
       40       OPERATIONS
```

SQL> create table s1(sid number(5),sname varchar2(10),rating number(5),age number(10));

Table created.

SQL> insert into s1 values(&sid,'&sname',&rating,&age);
Enter value for sid: 22
Enter value for sname: dustin
Enter value for rating: 7
Enter value for age: 45
old 1: insert into s1 values(&sid,'&sname',&rating,&age)
new 1: insert into s1 values(22,'dustin',7,45)

1 row created.

SQL> /
Enter value for sid: 31
Enter value for sname: lubber
Enter value for rating: 8
Enter value for age: 56
old 1: insert into s1 values(&sid,'&sname',&rating,&age)
new 1: insert into s1 values(31,'lubber',8,56)

1 row created.

SQL> /

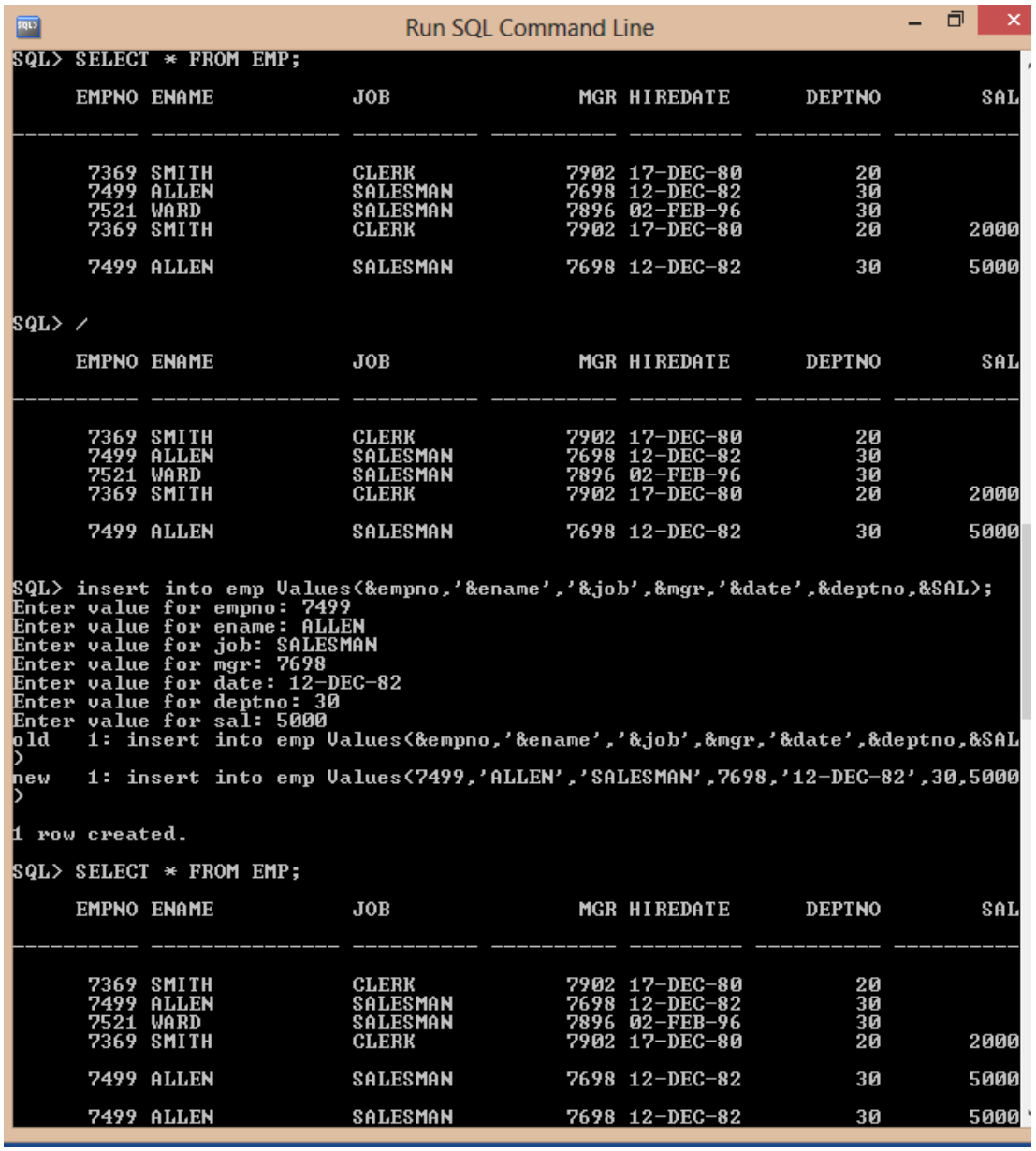
```
Enter value for sid: 58
Enter value for sname: rusty
Enter value for rating: 10
Enter value for age: 35
old  1: insert into s1 values(&sid,'&sname',&rating,&age)
new  1: insert into s1 values(58,'rusty',10,35)

1 row created.
SQL>  create table s2(sid number(5),sname varchar2(10),rating number(5),age
number(10));

Table created.

SQL> insert into s2 values(&sid,'&sname',&rating,&age);
Enter value for sid: 28
Enter value for sname: yuppy
Enter value for rating: 9
Enter value for age: 25
old  1: insert into s2 values(&sid,'&sname',&rating,&age)
new  1: insert into s2 values(28,'yuppy',9,25)

1 row created.

SQL> /
Enter value for sid: 31
Enter value for sname: lubber
Enter value for rating: 8
Enter value for age: 55
old  1: insert into s2 values(&sid,'&sname',&rating,&age)
new  1: insert into s2 values(31,'lubber',8,55)

1 row created.

SQL> /
Enter value for sid: 44
Enter value for sname: guppy
Enter value for rating: 5
Enter value for age: 35
old  1: insert into s2 values(&sid,'&sname',&rating,&age)
new  1: insert into s2 values(44,'guppy',5,35)

1 row created.

SQL> /
Enter value for sid: 58
Enter value for sname: rusty
```

```
Enter value for rating: 10
Enter value for age: 35
old   1: insert into s2 values(&sid,'&sname',&rating,&age)  new
1: insert into s2 values(58,'rusty',10,35)

1 row created.

SQL> select * from s2;

       SID SNAME           RATING        AGE
---------- ---------- ---------- ----------
        28 yuppy               9         25
        31 lubber              8         55
        44 guppy               5         35
        58 rusty              10         35
SQL> select sname from s1 union select sname from s2; SNAME
------
dustin
guppy
lubber
rusty
yuppy
SQL> select sname from s1 union select sname from s2; SNAME
------
dustin
lubber
rusty
yuppy
lubber
guppy
rusty

7 rows selected.
SQL> select sid from s1 intersect select sid from s2; SID
------
        31
        58
```

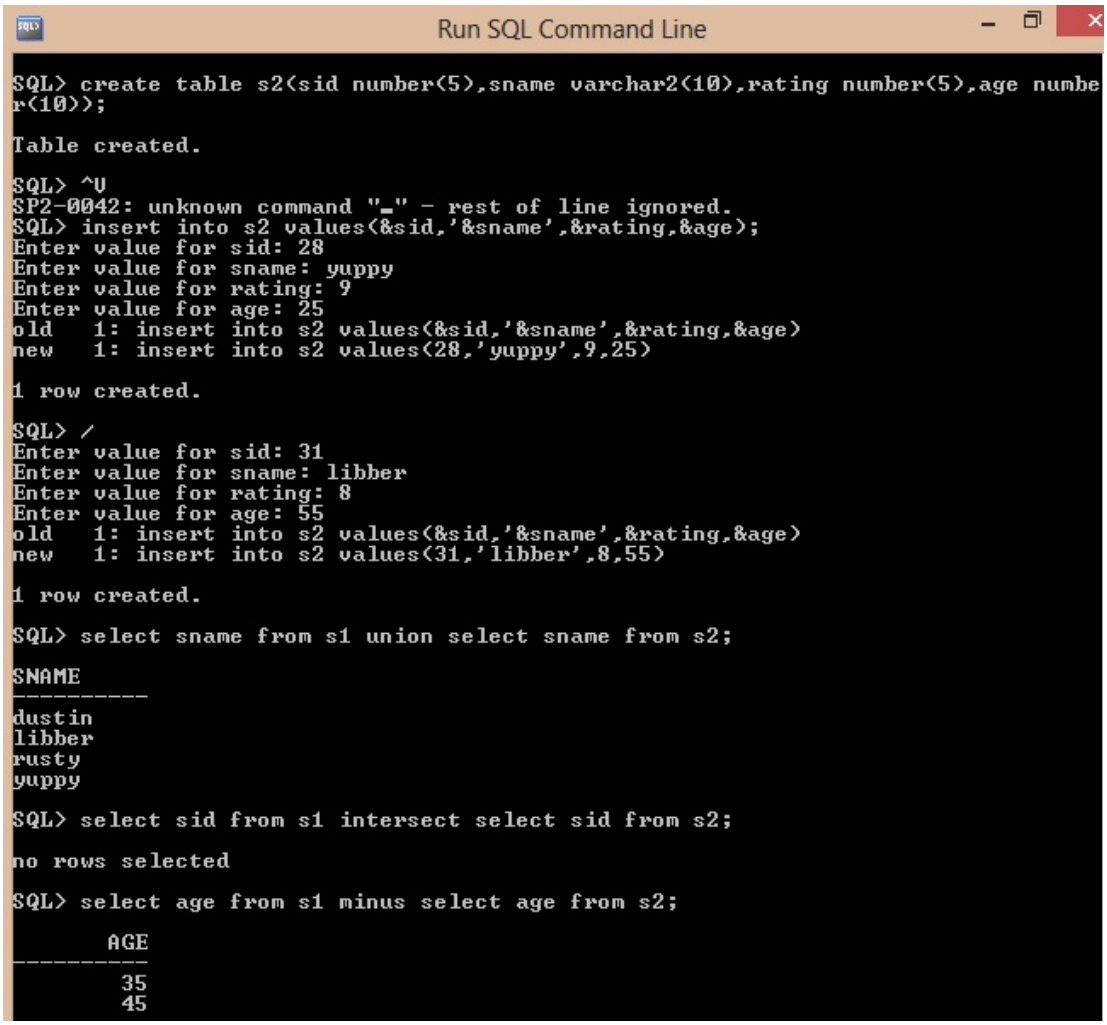
SQL> select age from s1 minus select age from s2;

```
       AGE
------
        45
        56
```


Run SQL Command Line
SQL> CREATE TABLE DEPT
2 (
3 DEPTNO NUMBER(3),
4 DNAME VARCHAR(10),
5 LOC VARCHAR(20)
6);
Table created.
SQL> INSERT INTO DEPT
2 VALUES(&DEPTNO,'&DNAME','&LOC');
Enter value for deptno: 10
Enter value for dname: ACC
Enter value for loc: NEWYORK
old 2: VALUES(&DEPTNO,'&DNAME','&LOC')
new 2: VALUES(10,'ACC','NEWYORK')
1 row created.
SQL> /
Enter value for deptno: 20
Enter value for dname: RESEARCH
Enter value for loc: DALLAS
old 2: VALUES(&DEPTNO,'&DNAME','&LOC')
new 2: VALUES(20,'RESEARCH','DALLAS')
1 row created.
SQL> /
Enter value for deptno: 30
Enter value for dname: SALES
Enter value for loc: CHICAGO
old 2: VALUES(&DEPTNO,'&DNAME','&LOC')
new 2: VALUES(30,'SALES','CHICAGO')
1 row created.
SQL> SELECT * FROM DEPT;
DEPTNO DNAME LOC
10 ACC NEWYORK
20 RESEARCH DALLAS
30 SALES CHICAGO
SQL> DESC EMP;
Name Null? Type
EMPNO NUMBER(5)
ENAME VARCHAR2(15)
JOB CHAR(10)
MGR NUMBER(5)
HIREDATE DATE
DEPTNO NUMBER(5)


Run SQL Command Line
SQL> SELECT * FROM EMP;
EMPNO ENAME JOB MGR HIREDATE DEPTNO SAL
7369 SMITH CLERK 7902 17-DEC-80 20
7499 ALLEN SALESMAN 7698 12-DEC-82 30
7521 WARD SALESMAN 7896 02-FEB-96 30
7369 SMITH CLERK 7902 17-DEC-80 20 2000
7499 ALLEN SALESMAN 7698 12-DEC-82 30 5000
SQL> /
EMPNO ENAME JOB MGR HIREDATE DEPTNO SAL
7369 SMITH CLERK 7902 17-DEC-80 20
7499 ALLEN SALESMAN 7698 12-DEC-82 30
7521 WARD SALESMAN 7896 02-FEB-96 30
7369 SMITH CLERK 7902 17-DEC-80 20 2000
7499 ALLEN SALESMAN 7698 12-DEC-82 30 5000
SQL> insert into emp Values(&empno,'&ename','&job',&mgr,'&date',&deptno,&SAL);
Enter value for empno: 7499
Enter value for ename: ALLEN
Enter value for job: SALESMAN
Enter value for mgr: 7698
Enter value for date: 12-DEC-82
Enter value for deptno: 30
Enter value for sal: 5000
old 1: insert into emp Values(&empno,'&ename','&job',&mgr,'&date',&deptno,&SAL
)
new 1: insert into emp Values(7499,'ALLEN','SALESMAN',7698,'12-DEC-82',30,5000
)
1 row created.
SQL> SELECT * FROM EMP;
EMPNO ENAME JOB MGR HIREDATE DEPTNO SAL
7369 SMITH CLERK 7902 17-DEC-80 20
7499 ALLEN SALESMAN 7698 12-DEC-82 30
7521 WARD SALESMAN 7896 02-FEB-96 30
7369 SMITH CLERK 7902 17-DEC-80 20 2000
7499 ALLEN SALESMAN 7698 12-DEC-82 30 5000
7499 ALLEN SALESMAN 7698 12-DEC-82 30 5000


Run SQL Command Line
6 rows selected.
SQL> select empno,ename from emp where sal in(select max(sal) from emp);
EMPNO ENAME
7499 ALLEN
7499 ALLEN
SQL> select empno,ename from emp where sal in(select max(sal) from emp group by deptno);
EMPNO ENAME
7369 SMITH
7499 ALLEN
7499 ALLEN
SQL> select empno,ename,job,sal from emp where sal>all(select avg(sal) from emp group by deptno);
no rows selected
SQL> select deptno,dname from dept d where exists(select * from emp e where d.deptno=e.deptno);
DEPTNO DNAME
20 RESEARCH
30 SALES
SQL> create table s1(sid number(5),sname varchar2(10),rating number(5),age number(10));
Table created.
SQL> insert into s1 values(&sid,'&sname',&rating,&age);
Enter value for sid: 22
Enter value for sname: dustin
Enter value for rating: 7
Enter value for age: 45
old 1: insert into s1 values(&sid,'&sname',&rating,&age)
new 1: insert into s1 values(22,'dustin',7,45)
1 row created.
SQL> /
Enter value for sid: 58
Enter value for sname: rusty
Enter value for rating: 10
Enter value for age: 35
old 1: insert into s1 values(&sid,'&sname',&rating,&age)
new 1: insert into s1 values(58,'rusty',10,35)
1 row created.
SQL>


Run SQL Command Line
SQL> create table s2(sid number(5),sname varchar2(10),rating number(5),age numbe
r(10));
Table created.
SQL> ^U
SP2-0042: unknown command "_" - rest of line ignored.
SQL> insert into s2 values(&sid,'&sname',&rating,&age);
Enter value for sid: 28
Enter value for sname: yuppy
Enter value for rating: 9
Enter value for age: 25
old 1: insert into s2 values(&sid,'&sname',&rating,&age)
new 1: insert into s2 values(28,'yuppy',9,25)
1 row created.
SQL> /
Enter value for sid: 31
Enter value for sname: libber
Enter value for rating: 8
Enter value for age: 55
old 1: insert into s2 values(&sid,'&sname',&rating,&age)
new 1: insert into s2 values(31,'libber',8,55)
1 row created.
SQL> select sname from s1 union select sname from s2;
SNAME
dustin
libber
rusty
yuppy
SQL> select sid from s1 intersect select sid from s2;
no rows selected
SQL> select age from s1 minus select age from s2;
AGE
35
45

Program No. 7

Object: Aggregate Function, Conversion, Count, Date, String

SQL> SELECT * FROM EMP;

output:-

ENO	ENAME	JOB	SAL
1001	STEVE	SALESMAN	1500
1002	ADAM	CLERK	1000
1003	EVE	MANAGER	5220
1004	JAMES	DIRECTOR	7000

SQL> SELECT COUNT(*)FROM EMP;

output:-

COUNT(*)
4

SQL> SELECT SUM(SAL) FROM EMP;

output:-

SUM(SAL)
14720

SQL> SELECT AVG(SAL) FROM EMP;

output:-

AVG(SAL)
3680

SQL> SELECT MAX(SAL) FROM EMP;

output:-

```
  MAX(SAL)
----------
      7000
```

SQL> SELECT MIN(SAL) FROM EMP;

output:-

```
  MIN(SAL)
----------
      1000
```

SQL>select * from emp;

OUTPUT:-

```
     EMPNO ENAME      JOB              SAL     DEPTNO
---------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK             800         20
      7499 ALLEN      SALESMAN         1600         30
      7521 WARD       SALESMAN         1250         30
      7566 JONES      MANAGER          2975         20
      7654 MARTIN     SALESMAN         1250         30
      7698 BLAKE      MANAGER          2850         30
      7782 CLARK      MANAGER          2450         10
      7788 SCOTT      ANALYST          3000         20
      7839 KING       PRESIDENT        5000         10
      7844 TURNER     SALESMAN         1500         30
      7876 ADAMS      CLERK            1100         20
      7900 JAMES      CLERK             950         30
      7902 FORD       ANALYST          3000         20
      7934 MILLER     CLERK            1300         10
```

14 rows selected.

APPLYING BY GROUP BY ---

SQL> select deptno,max(sal) from emp group by deptno;

OUTPUT:-

```
DEPTNO          MAX(SAL)
----------      ----------
      10            5000
      20            3000
      30            2850
```

SQL> select deptno,min(sal) from emp group by deptno;

OUTPUT:-

```
   DEPTNO       MIN(SAL)
   ----------   ----------
      10              1300
      20             800
      30             950
```

SQL> select deptno,max(sal) from emp group by deptno having max(sal)<3000;

OUTPUT:-

```
    DEPTNO     MAX(SAL)
   ------      ----------
      30            2850
```

SQL>select deptno,min(sal) from emp group by deptno having min(sal)>1000;

OUTPUT:-

```
   DEPTNO   MIN(SAL)
   ------   ----------
      10         1300
```

SQL> select ename,count(*) from emp group by deptno;

OUTPUT:-

```
   DEPTNO   COUNT(*)
    ----------    ---------
      10         3
      20         5
      30         6
3 rows selected.
```

CONVERSION FUNCTIONS(TO_CHAR)

SQL>select to_char(65,'RN')from dual;
OUTPUT:LXV

SQL>select to_char(65,'rn')from dual;
OUTPUT:lxv

SQL>select to_char(58,'s9999')from dual;
OUTPUT:+58

SQL>select to_char(-100,'s9999')from dual;
OUTPUT:-100

SQL> select to_char(41,'XXXX')from dual;
OUTPUT:29

SQL> select to_char(10,'XXXX')from dual;
OUTPUT:A

SQL> select to_char(sysdate,'day')from dual;
OUTPUT:MONDAY

SQL> SELECT TO_CHAR(SYSDATE,'MONTH')FROM DUAL;
OUTPUT:JANUARY

SQL> SELECT TO_CHAR(SYSDATE,'YEAR')FROM DUAL;
OUTPUT:TWO THOUSAND NINE

SQL> select to_char(123456,'9g99g999')from dual;
OUTPUT:1,23,456

SQL> select to_char(1234,'l9999')from dual;
OUTPUT:$1234

SQL> select to_char(123456,'9g99g999d999')from dual;
OUTPUT:1,23,456.000

SELECT TO_CHAR(2234,'L9999','NLS_CURRENCY=RS')FROM DUAL;
OUTPUT:RS2234

STRING FUNCTIONS

SQL>SELECT CONCAT('ORACLE','CORPORATION')FROM DUAL;

OUTPUT:-ORACLECORPORATION

SQL>SELECT LPAD('ORACLE',15,'*')FROM DUAL;

OUTPUT:-*********ORACLE

SQL>SELECT RPAD('ORACLE',15,'*')FROM DUAL;

OUTPUT:-ORACLE*********

SQL>SELECT LTRIM('SSMITHSS','S')FROM DUAL;

OUTPUT:-MITHSS

SQL>SELECT RTRIM('SSMITHSS','S')FROM DUAL;

OUTPUT:-SSMITH

SQL>SELECT LOWER('DBMS')FROM DUAL;

OUTPUT:-dbms

SQL>SELECT UPPER('dbms')FROM DUAL;

OUTPUT:-DBMS

SQL>SELECT INITCAP('ORACLE','CORPORATION')FROM DUAL;

OUTPUT:-Oracle Corporation

SQL>SELECT LENGTH('DATABASE')FROM DUAL;

OUTPUT:-8

SQL>SELECT SUBSTR('ABCDEFGHIJ'3,4)FROM DUAL;

OUTPUT:-CDEF

SQL>SELECT INSTR('CORPORATE FLOOR','OR',3,2)FROM DUAL;

OUTPUT:-14

DATE FUNCTIONS

SQL>SELECT SYSDATE FROM DUAL;

OUTPUT:-29-DEC-08

SQL>SELECT NEXT_DAY(SYSDATE,’WED’)FROM DUAL;

OUTPUT:-05-JAN-09

SQL>SELECT ADD_MONTHS(SYSDATE,2)FROM DUAL;

OUTPUT:-28-FEB-09

SQL>SELECT LAST_DAY(SYSDATE)FROM DUAL;

OUTPUT:-31-DEC-08

SQL>SELECT MONTHS_BETWEEN(SYSDATE,HIREDATE)FROM EMP;

OUTPUT:-

SQL>SELECT LEAST('10-JAN-07','12-OCT-07')FROM DUAL;

OUTPUT:-10-JAN-07

SQL>SELECT GREATEST('10-JAN-07','12-OCT-07')FROM DUAL;

OUTPUT:-10-JAN-07

SQL>SELECT TRUNC(SYSDATE,'DAY')FROM DUAL;

OUTPUT:-28-DEC-08

SQL>SELECT TRUNC(SYSDATE,'MONTH')FROM DUAL;

OUTPUT:-01-DEC-08
SQL>SELECT TRUNC(SYSDATE,'YEAR')FROM DUAL;

OUTPUT:-01-JAN-08

SQL>SELECT ROUND(SYSDATE,'DAY')FROM DUAL;

OUTPUT:-28-DEC-08

SQL>SELECT ROUND(SYSDATE,'MONTH')FROM DUAL;

OUTPUT:-01-JAN-09

SQL>SELECT ROUND(SYSDATE,'YEAR')FROM DUAL;

OUTPUT:-01-JAN-09

NUMBER FUNCTIONS

SQL> select round(12.36), round(14.63) from dual;

OUTPUT:-

```
ROUND(12.36)       ROUND(14.63)
- - - - - - - -          - - - - - - - -
       12                  15
```

SQL> select floor(12.87), floor(11.23) from dual;

OUTPUT:-

```
FLOOR(12.87)         FLOOR(11.23)
  - - - - - - - -          ----------
       12                  11
```

SQL> select ceil(16.23), ceil(12.78) from dual;

OUTPUT:-

```
CEIL(16.23)       CEIL(12.78)
- - - - - - -          - - - - - - -
      17              13
```

SQL> select trunc(56.63) from dual;

OUTPUT:-

```
TRUNC(56.63)
- - - - - - - -
       56
```

SQL> select mod(11,4) from dual;

OUTPUT:-

```
 MOD(11,4)
----------
         3
```

SQL> select power(2,3) from dual;

OUTPUT:-

```
POWER(2,3)
----------
         8
```

SQL> select sign(0),sign(34),sign(-56) from dual;

OUTPUT:-

```
  SIGN(0)   SIGN(34)      SIGN(-56)
 ---------- ----------    ----------
         0          1            -1
```

SQL> select abs(12),abs(-89) from dual;

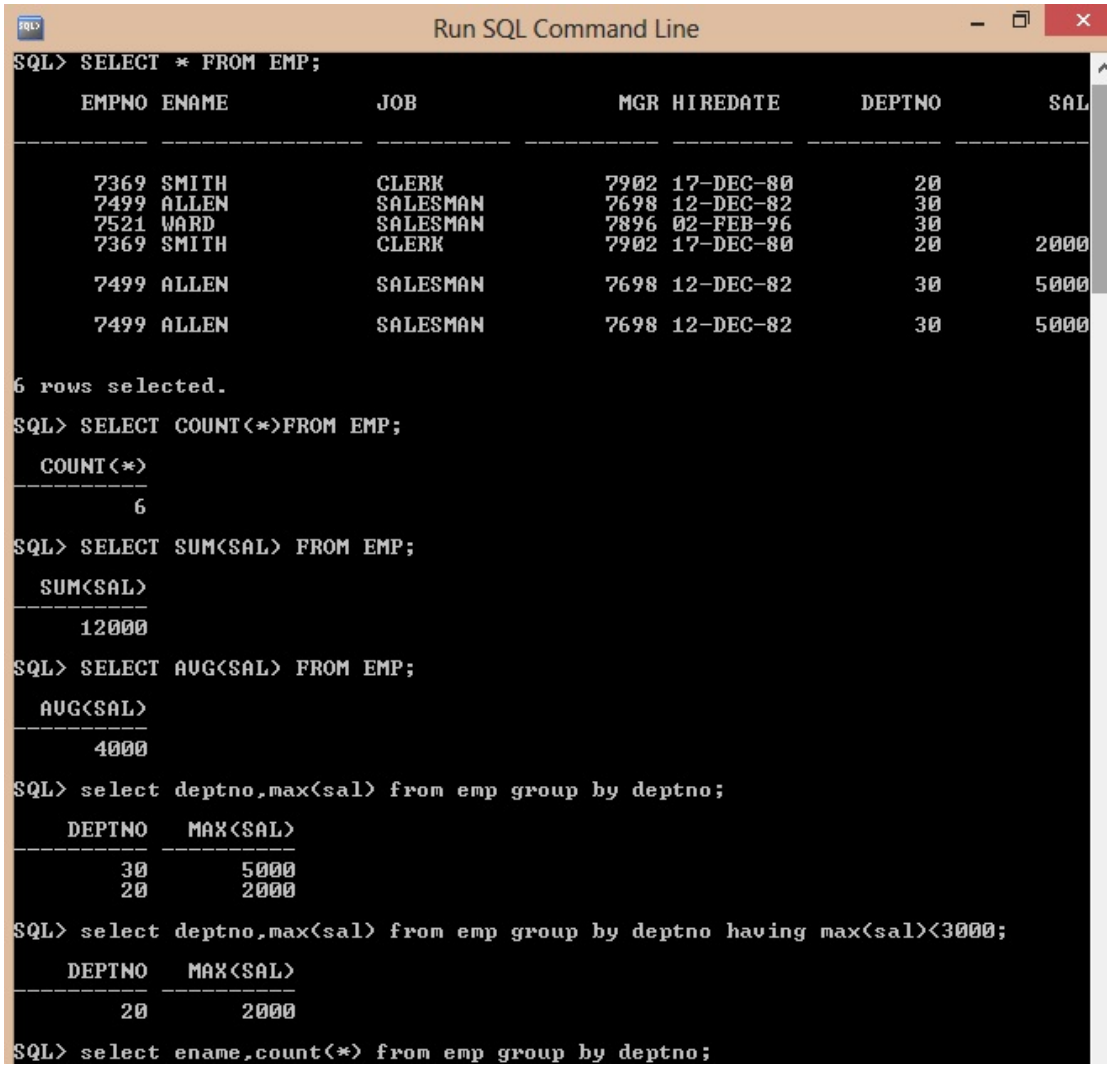
OUTPUT:-

```
  ABS(12)    ABS(-89)
 ---------- ----------
        12         89
```

SQL> select sqrt(25) from dual;

OUTPUT:-

```
 SQRT(25)
 ---------
        5
```


Run SQL Command Line
SQL> SELECT * FROM EMP;
EMPNO ENAME JOB MGR HIREDATE DEPTNO SAL
7369 SMITH CLERK 7902 17-DEC-80 20
7499 ALLEN SALESMAN 7698 12-DEC-82 30
7521 WARD SALESMAN 7896 02-FEB-96 30
7369 SMITH CLERK 7902 17-DEC-80 20 2000
7499 ALLEN SALESMAN 7698 12-DEC-82 30 5000
7499 ALLEN SALESMAN 7698 12-DEC-82 30 5000
6 rows selected.
SQL> SELECT COUNT(*)FROM EMP;
COUNT(*)
6
SQL> SELECT SUM(SAL) FROM EMP;
SUM(SAL)
12000
SQL> SELECT AVG(SAL) FROM EMP;
AVG(SAL)
4000
SQL> select deptno,max(sal) from emp group by deptno;
DEPTNO MAX(SAL)
30 5000
20 2000
SQL> select deptno,max(sal) from emp group by deptno having max(sal)<3000;
DEPTNO MAX(SAL)
20 2000
SQL> select ename,count(*) from emp group by deptno;

Program No. 8

Object: INTRODUCTION TO PL/SQL

PL/SQL stands for PROCEDURAL Language Extensions to SQL.
PL/SQL extends SQL by adding programming structures and subroutines available in any high level language.
PL/SQL can be used for both server-side and Client side Development.
PL/SQL has syntax and rules that determine how programming statements work together.
PL/SQL is not a stand alone Programming Language.
PL/SQL is a part of the ORACLE RDBMS and hence can reside in two environments,the CLIENT and the SERVER.
Any MODULE that is developed using PL/SQL can be moved easily between SERVER SIDE and CLIENT SIDE applications.
Either in CLIENT/SERVER environments any PL/SQL Block or the PL/SQL Engine processes Subroutine.
PL/SQL Engine is a special component that processes and executes any PL/SQL statements and sends any SQL statement to the SQL statement processor.
The SQL statement processes are always located on the ORACLE SERVER.

As per the necessity the PL/SQL Engine can be located either at

SERVER
CLIENT

When PL/SQL Engine is locted upon the SERVER,the whole PL/SQL block is passed to the PL/SQL Engine on the ORACLE SERVER.
When the PL/SQL Engine is located upon the CLIENT,the PL/SQL processing is done on the CLIENT SIDE.All SQL statenents that are embedded within the PL/SQL block,are sent to the ORACLE SERVER for further processing.

If the PL/SQL block does not contain any SQL statements,the entire block is executed on the CLIENT SIDE.

PL/SQL BLOCK

DECLARE
--Declarations of memory variables,constants,cursors etc.,in PL/SQL
BEGIN
--SQL executable statements
--PL/SQL executable statements
EXCEPTION

/*SQL or PL/SQL code to handle errors that may arise during the execution of the code block between BEGIN and EXCEPTION section
END;

SYNTAX's of CONTROL STATEMENTS in PL/SQL

1. BRANCHING
2. SELECTION
3. LOOPING

BRANCHING STATEMENTS

1.Simple IF
2.ELSIF
3.ELSE IF

SIMPLE IF

```
IF condition THEN
   statement1;
   statement2;
END IF;
```

IF-THEN-ELSE STATEMENT

```
IF condition THEN
   statement1;
ELSE
   statement2;
END IF;
```

ELSIF STATEMENTS

```
IF condition1 THEN
  statement1;
ELSIF condition2 THEN
  statement2;
ELSIF condition3 THEN
  statement3;
ELSE
  statementn;
END IF;
```

NESTED IF

```
IF condition THEN
   statement1;
ELSE
 IF condition THEN
    statement2;
 ELSE
    statement3;
 END IF;
END IF;
ELSE
 statement3;
END IF;
```

SELECTION IN PL/SQL

SIMPLE CASE

```
CASE SELECTOR
    WHEN Expr1 THEN statement1;
    WHEN  Expr2  THEN  statement2;
          :
          :
          :
ELSE
     statementn;

END CASE;
```

SEARCHED CASE

```
CASE
  WHEN  searchcondition1 THEN statement1;
  WHEN   searchcondition2  THEN  statement2;
          :
          :
          :
ELSE
          statementn;
END CASE;
```

ITERATIONS IN PL/SQL

SIMPLE LOOP

```
LOOP
  statement1;
EXIT [ WHEN Condition];
END LOOP;
```

WHILE LOOP

```
WHILE condition LOOP
        statement1;
        statement2;
END LOOP;
```

FOR LOOP

```
FOR counter IN [REVERSE]
   LowerBound..UpperBound
LOOP
        statement1;
        statement2;
END LOOP;
```

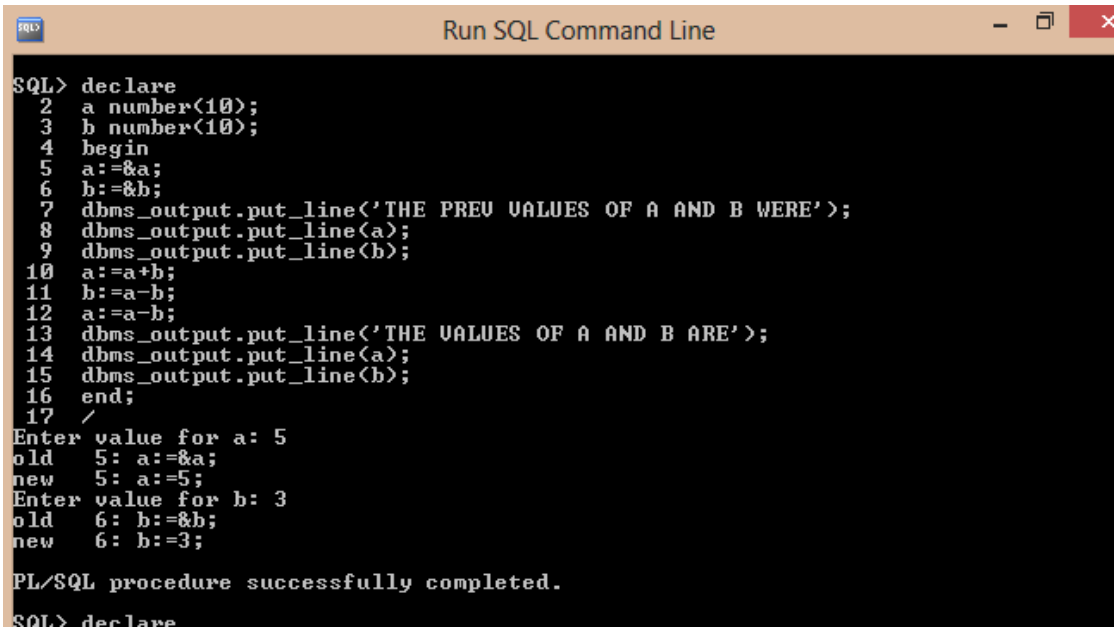
Program No. 9

Object: (1) **WRITE A PL/SQL PROGRAM TO SWAP TWO NUMBERS WITH OUT TAKING THIRD VARIABLE**

```
declare
a number(10);
b number(10);
begin
a:=&a;
b:=&b;
dbms_output.put_line('THE PREV VALUES OF A AND B WERE');
dbms_output.put_line(a);
dbms_output.put_line(b);
a:=a+b;
b:=a-b;
a:=a-b;
dbms_output.put_line('THE VALUES OF A AND B ARE');
dbms_output.put_line(a);
dbms_output.put_line(b);
end;
```

OUTPUT:

```
SQL> @ SWAPPING.SQL
 17 /
Enter value for a: 5
old  5: a:=&a;
new   5: a:=5;
Enter value for b: 3
old  6: b:=&b;
new   6: b:=3;
THE PREV VALUES OF A AND B WERE
5
3
THE VALUES OF A AND B ARE
3
5

PL/SQL procedure successfully completed.
```



(2) WRITE A PL/SQL PROGRAM TO SWAP TWO NUMBERS BY TAKING THIRD VARIABLE

```
declare
a number(10);
b number(10);
c number(10);
begin
dbms_output.put_line('THE PREV VALUES OF A AND B WERE');
dbms_output.put_line(a);
dbms_output.put_line(b);
a:=&a;
b:=&b;
c:=a;
a:=b;
b:=c;
dbms_output.put_line('THE VALUES OF A AND B ARE');
dbms_output.put_line(a);
dbms_output.put_line(b);
end;
```

OUTPUT:

```
SQL> @ SWAPPING2.SQL
 19 /
Enter value for a: 5
old  6: a:=&a;
new   6: a:=5;
Enter value for b: 3
```

```
old   7: b:=&b;
new   7: b:=3;
THE PREV VALUES OF A AND B WERE
5
3
THE VALUES OF A AND B ARE
3
5

PL/SQL procedure successfully completed.
```

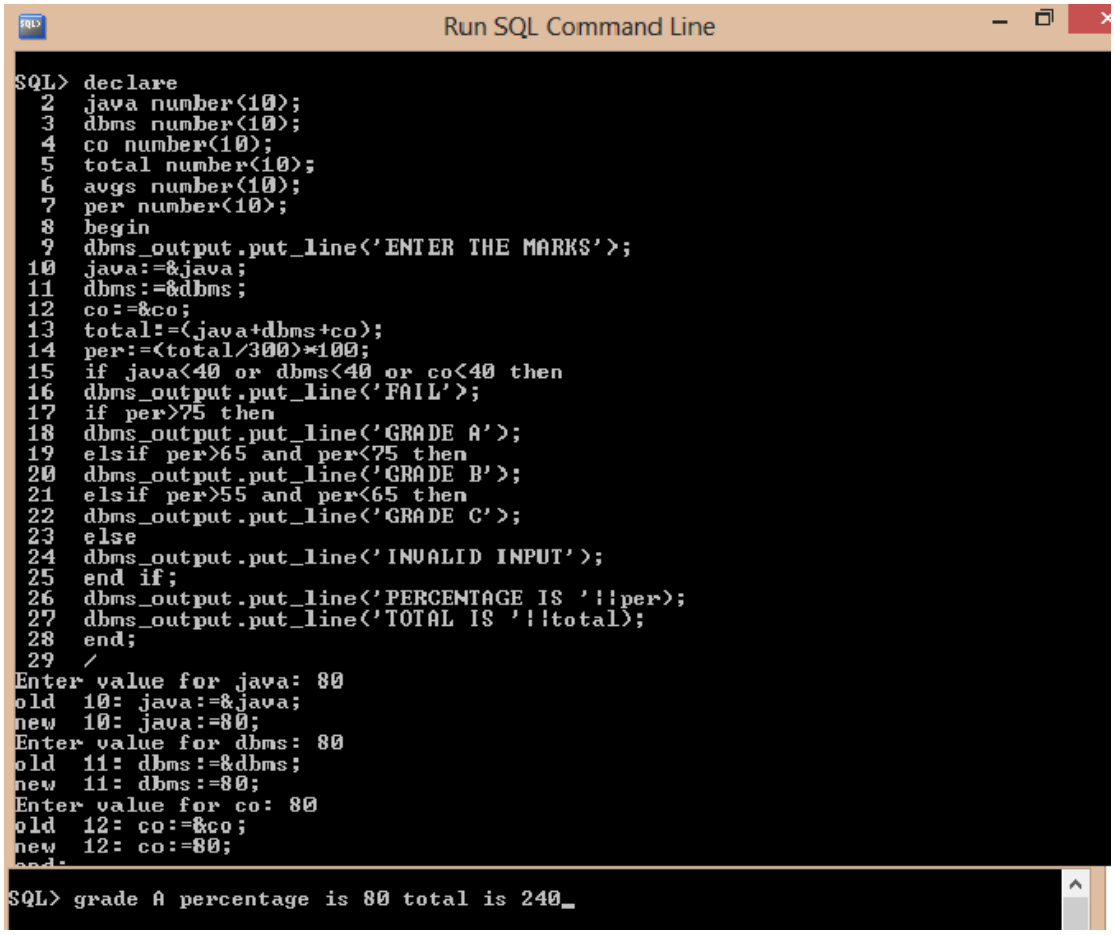
(3) WRITE A PL/SQL PROGRAM TO FIND THE TOTAL AND AVERAGE OF 6 SUBJECTS AND DISPLAY THE GRADE

```
declare
java number(10);
dbms number(10);
co number(10);
total number(10);
avgs number(10);
per number(10);
begin
dbms_output.put_line('ENTER THE MARKS');
java:=&java;
dbms:=&dbms;
co:=&co;
total:=(java+dbms+co);
per:=(total/300)*100;
if java<40 or dbms<40 or co<40 then
dbms_output.put_line('FAIL');
if per>75 then
dbms_output.put_line('GRADE A');
elsif per>65 and per<75 then
dbms_output.put_line('GRADE B');
elsif per>55 and per<65 then
dbms_output.put_line('GRADE C');
else
dbms_output.put_line('INVALID INPUT');
end if;
dbms_output.put_line('PERCENTAGE IS '||per);
dbms_output.put_line('TOTAL IS '||total);
```

```
end;
```

OUTPUT:

```
SQL> @ GRADE.sql
 31 /
Enter value for java: 80
old  12: java:=&java;
new  12: java:=80;
Enter value for dbms: 70
old  13: dbms:=&dbms;
new  13: dbms:=70;
Enter value for co: 89
old  14: co:=&co;
new  14: co:=89;
Enter value for se: 72
old  15: se:=&se;
new  15: se:=72;
Enter value for es: 76
old  16: es:=&es;
new  16: es:=76;
Enter value for ppl: 71
old  17: ppl:=&ppl;
new  17: ppl:=71;
GRADE A
PERCENTAGE IS 76
TOTAL  IS  458
PL/SQL procedure successfully completed.
```

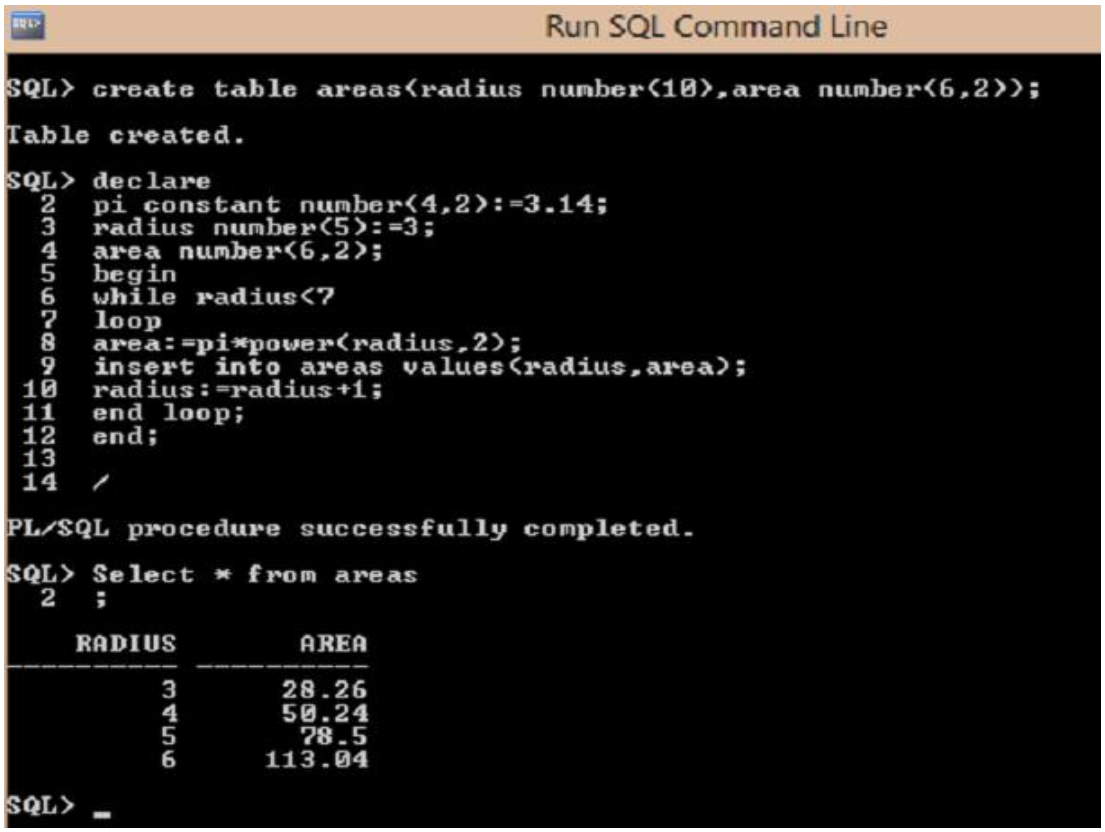

Run SQL Command Line
SQL> declare
2 java number(10);
3 dbms number(10);
4 co number(10);
5 total number(10);
6 avgs number(10);
7 per number(10);
8 begin
9 dbms_output.put_line('ENTER THE MARKS');
10 java:=&java;
11 dbms:=&dbms;
12 co:=&co;
13 total:=(java+dbms+co);
14 per:=(total/300)*100;
15 if java<40 or dbms<40 or co<40 then
16 dbms_output.put_line('FAIL');
17 if per>75 then
18 dbms_output.put_line('GRADE A');
19 elsif per>65 and per<75 then
20 dbms_output.put_line('GRADE B');
21 elsif per>55 and per<65 then
22 dbms_output.put_line('GRADE C');
23 else
24 dbms_output.put_line('INVALID INPUT');
25 end if;
26 dbms_output.put_line('PERCENTAGE IS '||per);
27 dbms_output.put_line('TOTAL IS '||total);
28 end;
29 /
Enter value for java: 80
old 10: java:=&java;
new 10: java:=80;
Enter value for dbms: 80
old 11: dbms:=&dbms;
new 11: dbms:=80;
Enter value for co: 80
old 12: co:=&co;
new 12: co:=80;
SQL> grade A percentage is 80 total is 240

Program No. 10

Object: (1) WRITE A PL/SQL CODE BLOCK TO CALCULATE THE AREA OF A CIRCLE FOR A VALUE OF RADIUS VARYING FROM 3 TO 7. STORE THE RADIUS AND THE CORRESPONDING VALUES OF CALCULATED AREA IN AN EMPTY TABLE NAMED AREAS ,CONSISTING OF TWO COLUMNS RADIUS & AREA
TABLE NAME: AREAS
RADIUS AREA

```
SQL> create table areas(radius number(10),area number(6,2));
Table created.
--PROGRAM
declare
pi constant number(4,2):=3.14;
radius number(5):=3;
area number(6,2);
begin
while radius<7
loop
area:=pi*power(radius,2);
insert into areas values(radius,area);
radius:=radius+1;
end loop;
end;
```

OUTPUT:



(2) WRITE A PL/SQL CODE BLOCK THAT WILL ACCEPT AN ACCOUNT NUMBER FROM THE USER,CHECK IF THE USERS BALANCE IS LESS THAN MINIMUM BALANCE,ONLY THEN DEDUCT RS.100/- FROM THE BALANCE.THIS PROCESS IS FIRED ON THE ACCT TABLE.

SQL> create table acct(name varchar2(10),cur_bal number(10),acctno number(6,2));

SQL> insert into stud values('&sname',&rollno,&marks); SQL>
select * from acct;

ACCTNO	NAME	CUR_BAL
777	sirius	10000
765	john	1000
855	sam	500
353	peter	800

--PROGRAM

```
declare
mano    number(5);
mcb number(6,2);
minibal   constant   number(7,2):=1000.00;
fine number(6,2):=100.00;
begin
mano:=&mano;
select cur_bal into mcb from acct where acctno=mano;   if
mcb<minibal then
update acct set cur_bal=cur_bal-fine where acctno=mano; end if;
end;
```

OUTPUT:

```
SQL> @ BANKACC.sql
 13 /
Enter value for mano: 855 old
7: mano:=&mano;
new   7: mano:=855;

PL/SQL procedure successfully completed.
```

SQL> select * from acct;

ACCTNO	NAME	CUR_BAL
777	sirius	10000
765	john	1000
855	sam	400
353	peter	800