1) Queries for Creating, Altering and Dropping Tables, and Constraints.

**Creating a table**

SQL> create table students

2 (

3 rollno varchar2(30),

4 name varchar2(30),

5 branch varchar2(15)

6 );

Table created.

**To View Schema**

SQL> describe students

Name Null? Type

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

ROLLNO VARCHAR2(30)

NAME VARCHAR2(30)

BRANCH VARCHAR2(15)

**Altering the table**

SQL> alter table students

2 add age integer;

Table altered.

SQL> alter table students

2 drop column branch;

Table altered.

SQL> desc students

Name Null? Type

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

ROLLNO VARCHAR2(30)

NAME VARCHAR2(30)

AGE NUMBER(38)

**Truncate**

SQL> truncate table students;

Table truncated.

**Rename a table**

SQL> rename students to stu\_details2 ;

Table renamed.

**Dropping table**

SQL> drop table stu\_details;

Table dropped.

SQL> select \* from tab;

TNAME TABTYPE CLUSTERID

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

STUDENT1 TABLE

STUDENTS2 TABLE

STU TABLE

STU1 TABLE

STU2 TABLE

STUD TABLE

STU3 TABLE

STU4 TABLE

STU5 TABLE

BIN$QdFqkisyT1CzNapxV9Kqpw==$0 TABLE

BIN$OYWM4Kc6S3+WIMslmtnp/w==$0 TABLE

11 rows selected.

**Constraints**

**Not null**

SQL> create table stu

2 (

3 rno integer not null,

4 name varchar2(20)

5 );

Table created.

SQL> insert into stu values(501,'rani');

1 row created.

SQL> insert into stu values(501,'rani');

1 row created.

SQL> insert into stu values(null,'kamala');

insert into stu values(null,'kamala')

\*

ERROR at line 1:

ORA-01400: cannot insert NULL into ("CSE20561"."STU"."RNO")

**Unique**

SQL> create table stu1

2 (

3 rno integer,

4 name varchar2(20),

5 unique(rno)

6 );

Table created.

SQL> insert into stu1 values(501,'rani');

1 row created.

SQL> insert into stu1 values(501,'rani');

insert into stu1 values(501,'rani')

\*

ERROR at line 1:

ORA-00001: unique constraint (CSE20561.SYS\_C005728) violated

SQL> insert into stu1 values(null,'kamala');

1 row created.

**Primary key**

SQL> create table stu2

(

rno integer,

name varchar2(20),

primary key(rno)

);

Table created.

SQL> insert into stu2 values(501,'rani');

1 row created.

SQL> insert into stu2 values(501,'rani');

insert into stu2 values(501,'rani')

\*

ERROR at line 1:

ORA-00001: unique constraint (CSE20561.SYS\_C005730) violated

SQL> insert into stu2 values(null,'kamala');

insert into stu2 values(null,'kamala')

\*

ERROR at line 1:

ORA-01400: cannot insert NULL into ("CSE20561"."STU2"."RNO")

SQL> create table stu5

2 (

3 rno integer,

4 name varchar2(20)

5 );

Table created.

SQL> alter table stu5

2 add primary key(rno);

Table altered.

SQL> desc stu5

Name Null? Type

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

RNO NOT NULL NUMBER(38)

NAME VARCHAR2(20)

**Foreign key**

SQL> create table stud

2 (

3 rno integer,

4 fee integer,

5 foreign key(rno) references stu2(rno)

6 );

Table created.

SQL> insert into stud values(501,6000);

1 row created.

SQL> insert into stud values(502,8000);

insert into stud values(502,8000)

\*

ERROR at line 1:

ORA-02291: integrity constraint (CSE20561.SYS\_C005731) violated - parent key

not found

SQL> delete from stud where rno=501;

1 row deleted.

SQL> delete from stu2 where rno=501;

1 row deleted.

**Default**

SQL> create table stu3

2 (

3 rno integer,

4 name varchar2(20),

5 gender char(1) default 'F'

6 );

Table created.

SQL> insert into stu3 values(501,'ravi','M');

1 row created.

SQL> insert into stu3 values(501,'suma',default);

1 row created.

**Check**

SQL> create table stu4

2 (

3 rno integer,

4 name varchar2(20),

5 marks integer,

6 check (marks<101)

7 );

Table created.

SQL> insert into stu4 values(501,'ravi',101);

insert into stu4 values(501,'ravi',101)

\*

ERROR at line 1:

ORA-02290: check constraint (CSE20561.SYS\_C005733) violated

**2) Queries to Retrieve and Change Data: Select, Insert, Delete and Update.**

**Creating table**

SQL> create table students

2 (

3 rollno varchar2(30),

4 name varchar2(30)

5 );

Table created.

**Inserting Data into the table**

SQL> insert into students values('20A91A0501','Ravi');

1 row created.

SQL> insert into students values('20A91A0502','Suma');

1 row created.

**Displaying Data from the table**

SQL> select \* from students;

ROLLNO NAME

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

20A91A0501 Ravi

20A91A0502 Suma

SQL> select name from students;

NAME

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

Ravi

Suma

SQL> select \* from students where rollno='20A91A0501';

ROLLNO NAME

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

20A91A0501 Ravi

**Deleting a row from the table**

SQL> delete from students where rollno='20A91A0501';

1 row deleted.

**Updating a row in the table**

SQL> update students

2 set name='Rose'

3 where rollno='20A91A0502';

1 row updated.

SQL> insert into students values(&rollno,'&name');

Enter value for rollno: 501

Enter value for name: sai

old 1: insert into students values(&rollno,'&name')

new 1: insert into students values(501,'sai')

1 row created.

SQL> /

Enter value for rollno: 517

Enter value for name: sree

old 1: insert into students values(&rollno,'&name')

new 1: insert into students values(517,'sree')

1 row created.

SQL> insert into students(name,rollno) values('padma',503);

1 row created.

SQL> select \* from students;

ROLLNO NAME

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

20A91A0502 Rose

501 sai

517 sree

503 padma

**3.1) Queries to facilitate acquaintance of Built-in Functions: String Functions, Numeric Functions, Date Functions and Conversion Functions.**

**STRING FUNCTIONS:**

SQL> select concat('aditya','engg') from dual;

CONCAT('AD

----------

adityaengg

SQL> select concat(concat('aditya','engg'),'college') from dual;

CONCAT(CONCAT('AD

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

adityaenggcollege

SQL> select 'aditya'||'engg' from dual;

'ADITYA'||

----------

adityaengg

SQL> select lpad('aditya',15,'\*')as lpad from dual;

LPAD

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

61616161\*aditya

SQL> select rpad('aditya',15,'\*')as rpad from dual;

RPAD

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

aditya61616161\*

SQL> select ltrim('123123123rama123','123')from dual;

LTRIM('

-------

rama123

SQL> select rtrim('123123123rama123','123')from dual;

RTRIM('123123

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

123123123rama

SQL> select upper('aditya') from dual;

UPPER(

------

ADITYA

SQL> select lower('ADITYA') from dual;

LOWER(

------

aditya

SQL> select length('aditya') from dual;

LENGTH('ADITYA')

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

6

SQL> select substr('abcdefg',-3,2)from dual;

SU

--

ef

SQL> select instr('abab','b')from dual;

INSTR('ABAB','B')

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

2

SQL> select instr('abab','b',3)from dual;

INSTR('ABAB','B',3)

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

4

SQL> select ASCII('A') from dual;

ASCII('A')

----------

65

SQL> select chr(97) from dual;

C

-

a

SQL> select reverse('aditya') from dual;

REVERS

------

aytida

SQL> select initcap('aditya engg') from dual;

INITCAP('AD

-----------

Aditya Engg

SQL> CREATE TABLE student(rollno varchar2(30),name varchar2(30),branch varchar2(15));

Table created.

SQL> insert into student values('20A91A0501','Ravi','cse');

1 row created.

SQL> insert into student values('20A91A0502','Suma','cse');

1 row created.

SQL> select \* from student;

ROLLNO NAME BRANCH

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

20A91A0501 Ravi cse

20A91A0502 Suma cse

SQL> select reverse(name) from student;

REVERSE(NAME)

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

ivaR

amuS

SQL> select initcap(name) from student;

INITCAP(NAME)

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

Ravi

Suma

**NUMERIC FUNCTIONS:**

SQL> select abs(19) from dual;

ABS(19)

----------

19

SQL> select abs(-19) from dual;

ABS(-19)

----------

19

SQL> select sign(19) from dual;

SIGN(19)

----------

1

SQL> select sign(-19) from dual;

SIGN(-19)

----------

-1

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

POWER(3,2)

----------

9

SQL> select sqrt(9) from dual;

SQRT(9)

----------

3

SQL> select ceil(2.2) from dual;

CEIL(2.2)

----------

3

SQL> select ceil(-2.2) from dual;

CEIL(-2.2)

----------

-2

SQL> select floor(2.2) from dual;

FLOOR(2.2)

----------

2

SQL> select floor(-2.2) from dual;

FLOOR(-2.2)

-----------

-3

SQL> select mod(150,7) from dual;

MOD(150,7)

----------

3

SQL> select round(66.666,2) from dual;

ROUND(66.666,2)

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

66.67

SQL> select trunc(66.666,2) from dual;

TRUNC(66.666,2)

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

66.66

SQL> select exp(3) from dual;

EXP(3)

----------

20.0855369

SQL> select log(2,2) from dual;

LOG(2,2)

----------

1

**DATE FUNCTIONS:**

SQL> select sysdate from dual;

SYSDATE

---------

07-OCT-21

SQL> select sysdate+1 from dual;

SYSDATE+1

---------

08-OCT-21

SQL> select sysdate-1 from dual;

SYSDATE-1

---------

06-OCT-21

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

EXTRACT(YEARFROMSYSDATE)

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

2021

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

EXTRACT(MONTHFROMSYSDATE)

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

10

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

EXTRACT(DAYFROMSYSDATE)

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

7

SQL> select to\_char(sysdate,'yyyy/mm/dd') from dual;

TO\_CHAR(SY

----------

2021/10/07

SQL> select to\_char(sysdate,'HH:MM:SS') from dual;

TO\_CHAR(

--------

05:10:24

SQL> select add\_months(sysdate,2) from dual;

ADD\_MONTH

---------

07-DEC-21

SQL> select next\_day(sysdate,'Thursday') from dual;

NEXT\_DAY(

---------

14-OCT-21

SQL> select next\_day('10-dec-2019','Tuesday') from dual;

NEXT\_DAY(

---------

17-DEC-19

SQL> select last\_day(sysdate) from dual;

LAST\_DAY(

---------

31-OCT-21

SQL> select months\_between(to\_date('09-dec-2020','dd-mm-yyyy'),to\_date('09-dec-2019','dd-mm-yyyy'))

from dual;

MONTHS\_BETWEEN(TO\_DATE('09-DEC-2020','DD-MM-YYYY'),TO\_DATE('09-DEC-2019','DD-MM-

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

12

**On Delete Cascade:**

SQL> create table t1

2 (

3 t1\_id integer,

4 t1\_desc varchar2(30),

5 primary key(t1\_id));

Table created.

SQL> create table t2

2 (

3 t1\_t2\_id integer,

4 t2\_id integer,

5 t2\_desc varchar2(30),

6 primary key(t2\_id),

7 foreign key(t1\_t2\_id) references t1(t1\_id) on delete cascade

8 );

Table created.

SQL> create table t3

2 (

3 t2\_t3\_id integer,

4 t3\_id integer,

5 t3\_desc varchar2(30),

6 primary key(t3\_id),

7 foreign key(t2\_t3\_id) references t2(t2\_id) on delete cascade

8 );

Table created.

SQL> insert into t1 values(1,'t1',one);

insert into t1 values(1,'t1',one)

\*

ERROR at line 1:

ORA-009\*\*: too many values

SQL> insert into t1 values(1,'t1 one');

1 row created.

SQL> insert into t2 values(1,1,'t2 one');

1 row created.

SQL> insert into t3 values(1,1,'t3 one');

1 row created.

SQL> delete from t1 cascade;

1 row deleted.

SQL> select \* from t1;

no rows selected

SQL> select \* from t2;

no rows selected

SQL> select \* from t3;

no rows selected

SQL> insert into t1 values(1,'t1 one');

1 row created.

SQL> insert into t2 values(1,1,'t2 one');

1 row created.

SQL> insert into t3 values(1,1,'t3 one');

1 row created.

SQL> delete from t2 cascade;

1 row deleted.

SQL> select \* from t1;

T1\_ID T1\_DESC

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

1 t1 one

SQL> select \* from t2;

no rows selected

SQL> select \* from t3;

no rows selected

SQL> drop table t3;

Table dropped.

SQL> drop table t2;

Table dropped.

SQL> drop table t1;

Table dropped.

**On Delete SetNull:**

SQL> create table t1

2 (

3 t1\_id integer,

4 t1\_desc varchar2(30),

5 primary key(t1\_id)

6 );

Table created.

SQL> create table t2

2 (

3 t2\_id integer,

4 t1\_t2\_id integer,

5 t2\_desc varchar2(30),

6 primary key(t2\_id),

7 foreign key(t1\_t2\_id) references t1(t1\_id) on delete set NULL

8 );

Table created.

SQL> create table t3

2 (

3 t3\_id integer,

4 t2\_t3\_id integer,

5 t3\_desc varchar2(30),

6 primary key(t3\_id),

7 foreign key(t2\_t3\_id) references t2(t2\_id) on delete set NULL

8 );

Table created.

SQL> insert into t1 values(1,'t1 one');

1 row created.

SQL> insert into t2 values(1,1,'t1 one');

1 row created.

SQL> insert into t3 values(1,1,'t3 one');

1 row created.

SQL> delete from t1 where t1\_id = 1;

1 row deleted.

SQL> select \* from t1;

no rows selected

SQL> select \* from t2;

T2\_ID T1\_T2\_ID T2\_DESC

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

1 t1 one

SQL> select \* from t3;

T3\_ID T2\_T3\_ID T3\_DESC

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

1 1 t3 one

**3.2) Queries using operators in SQL.**

**Arithmetic Operators:**

SQL> select 4+5 as "add" from dual;

add

----------

9

SQL> select 9-5 from dual;

9-5

----------

4

SQL> select 2\*3 as mul from dual;

MUL

----------

6

SQL> select 8/4 from dual;

8/4

----------

2

SQL> select mod(8,4) from dual;

MOD(8,4)

----------

0

SQL> desc student

Name Null? Type

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

SID VARCHAR2(20)

SNAME VARCHAR2(30)

AGE NUMBER(38)

SQL> insert into student values(501,'akash',21);

1 row created.

SQL> insert into student values(502,'thanmayi',24);

1 row created.

SQL> select \* from student;

SID SNAME AGE

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

501 akash 21

502 thanmayi 24

**Comparison Operators:**

SQL> select sid,sname from student where age=21;

SID SNAME

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

501 akash

SQL> select sid,sname from student where age>10 and age<20;

no rows selected

SQL> select sid,sname from student where age<20;

no rows selected

SQL> select sid,sname from student where age>=21;

SID SNAME

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

501 akash

502 thanmayi

SQL> select sid,sname from student where age<=30;

SID SNAME

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

501 akash

502 thanmayi

SQL> select sid,sname from student where age<>50;

SID SNAME

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

501 akash

502 thanmayi

**Logical Operators:**

SQL> select sid,sname from student where age>20 and age<30;

SID SNAME

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

501 akash

502 thanmayi

SQL> select sid,sname from student where age>=21 or age<25;

SID SNAME

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

501 akash

502 thanmayi

SQL> select sid,sname from student where age between 20 and 25;

SID SNAME

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

501 akash

502 thanmayi

SQL> select sid,sname from student where age between 20 and 23;

SID SNAME

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

501 akash

SQL> select sid,sname from student where not age=21;

SID SNAME

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

502 thanmayi

**Like Operator:**

SQL> create table customers

2 (

3 name varchar2(30),

4 city varchar2(30),

5 );

Table created.

SQL> desc customers

Name Null? Type

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

NAME VARCHAR2(20)

CITY VARCHAR2(30)

SQL> insert into customers values('ajay','perry ridge');

1 row created.

SQL> insert into customers values('pavani','downtown');

1 row created.

SQL> insert into customers values('ravi','paris');

1 row created.

SQL> select \* from student;

NAME CITY

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

ajay perry ridge

pavani downtown

ravi paris

SQL> select \* from customers where city like '%idge%';

NAME CITY

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

ajay perry ridge

SQL> select \* from customers where city like 'p\_\_\_\_';

NAME CITY

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

ravi paris

**Is null and Is not null:**

SQL> desc student

Name Null? Type

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

SID VARCHAR2(20)

SNAME VARCHAR2(30)

AGE NUMBER(38)

SQL> insert into student values(503,'ajay',NULL);

1 row created.

SQL> insert into student values(504,'suma',NULL);

1 row created.

SQL> select \* from student;

SID SNAME AGE

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

501 akash 21

502 thanmayi 24

503 ajay

504 suma

SQL> select sid,sname from student where age IS NULL;

SID SNAME

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

503 ajay

504 suma

SQL> select sid,sname from student where age IS NOT NULL;

SID SNAME

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

501 akash

502 thanmayi

**4.1) Queries using Group By, Order By, and Having Clauses.**

SQL> create table company

2 (

3 companyn varchar2(30),

4 amount integer

5 );

Table created.

SQL> insert into company values('&companyn',&amount);

Enter value for companyn: wipro

Enter value for amount: 5000

old 1: insert into company values('&companyn',&amount)

new 1: insert into company values('wipro',5000)

1 row created.

SQL> ibm

SP2-0042: unknown command "ibm" - rest of line ignored.

SQL> /

Enter value for companyn: ibm

Enter value for amount: 9000

old 1: insert into company values('&companyn',&amount)

new 1: insert into company values('ibm',9000)

1 row created.

SQL> /

Enter value for companyn: dell

Enter value for amount: 10000

old 1: insert into company values('&companyn',&amount)

new 1: insert into company values('dell',10000)

1 row created.

SQL> /

Enter value for companyn: wipro

Enter value for amount: 4000

old 1: insert into company values('&companyn',&amount)

new 1: insert into company values('wipro',4000)

1 row created.

SQL> /

Enter value for companyn: dell

Enter value for amount: 6000

old 1: insert into company values('&companyn',&amount)

new 1: insert into company values('dell',6000)

1 row created.

Find the sum of amount of each company.

SQL> select companyn,sum(amount) from company group by companyn;

COMPANYN SUM(AMOUNT)

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

wipro 9000

dell 16000

ibm 9000

2.find the count of all the rows grouped by each company name.

SQL> select companyn,count(\*) from company group by companyn;

COMPANYN COUNT(\*)

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

wipro 2

dell 2

ibm 1

SQL>

3.find the minimum amount of each company.

SQL> select companyn,min(amount) from company group by companyn;

COMPANYN MIN(AMOUNT)

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

wipro 4000

dell 6000

ibm 9000

SQL>

4.find the max amount of each company.

SQL> select companyn,max(amount) from company group by companyn;

COMPANYN MAX(AMOUNT)

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

wipro 5000

dell 10000

ibm 9000

SQL>

5.find the count of all the rows grouped by each company name and having count greater than 1.

SQL> select companyn,count(\*) from company group by companyn having count(\*)>1;

COMPANYN COUNT(\*)

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

wipro 2

dell 2

SQL>

6.find sum of amount of each company and having sum of amount greater than 10000.

SQL> select companyn,sum(amount) from company group by companyn having sum(amount)>10000;

COMPANYN SUM(AMOUNT)

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

dell 16000

**4.2) Queries on Controlling Data: Commit, Rollback, and Save point.**

SQL> create table th1

2 (

3 rno integer,

4 name varchar2(30),

5 marks integer

6 );

Table created.

SQL> insert into th1 values(&rno,'&name',&marks);

Enter value for rno: 501

Enter value for name: abhi

Enter value for marks: 50

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(501,'abhi',50)

1 row created.

SQL> /

Enter value for rno: 502

Enter value for name: ravi

Enter value for marks: 40

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(502,'ravi',40)

1 row created.

SQL> /

Enter value for rno: 503

Enter value for name: suma

Enter value for marks: 30

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(503,'suma',30)

1 row created.

SQL> /

Enter value for rno: 504

Enter value for name: raju

Enter value for marks: 35

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(504,'raju',35)

1 row created.

SQL> /

Enter value for rno: 505

Enter value for name: ramu

Enter value for marks: 45

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(505,'ramu',45)

1 row created.

**Commit:**

SQL> commit;

Commit complete.

**Savepoint:**

SQL> insert into th1 values(504,'bhanu',14);

1 row created.

SQL> insert into th1 values(533,'pavani',15);

1 row created.

SQL> insert into th1 values(549,'harika',15);

1 row created.

SQL> select \* from th1;

RNO NAME MARKS

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

501 abhi 50

502 ravi 40

503 suma 30

504 raju 35

505 ramu 45

504 bhanu 14

533 pavani 15

549 harika 15

8 rows selected.

SQL> insert into th1 values(546,'harshitha',15);

1 row created.

SQL> savepoint a;

Savepoint created.

SQL> insert into th1 values(555,'sita',14);

1 row created.

SQL> savepoint b;

Savepoint created.

SQL> insert into th1 values(565,'ramya',15);

1 row created.

SQL> savepoint c;

Savepoint created.

SQL> select \* from th1;

RNO NAME MARKS

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

501 abhi 50

502 ravi 40

503 suma 30

504 raju 35

505 ramu 45

504 bhanu 14

533 pavani 15

549 harika 15

5\*\* bavya 15

546 harshitha 15

555 sita 14

RNO NAME MARKS

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

565 ramya 15

12 rows selected.

**Rollback:**

SQL> rollback to b;

Rollback complete.

SQL> select \* from th1;

RNO NAME MARKS

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

501 abhi 50

502 ravi 40

503 suma 30

504 raju 35

505 ramu 45

504 bhanu 14

533 pavani 15

549 harika 15

5\*\* bavya 15

546 harshitha 15

555 sita 14

11 rows selected.

**5. Queries on Joins and Correlated Sub-queries.**

SQL> create table th1

2 (

3 rno integer,

4 name varchar2(30),

5 marks integer

6 );

Table created.

SQL> create table th2

2 (

3 rno integer,

4 fee integer

5 );

Table created.

SQL> insert into th1 values(&rno,'&name',&marks);

Enter value for rno: 501

Enter value for name: abhi

Enter value for marks: 50

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(501,'abhi',50)

1 row created.

SQL> /

Enter value for rno: 502

Enter value for name: ravi

Enter value for marks: 40

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(502,'ravi',40)

1 row created.

SQL> /

Enter value for rno: 503

Enter value for name: suma

Enter value for marks: 30

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(503,'suma',30)

1 row created.

SQL> /

Enter value for rno: 504

Enter value for name: raju

Enter value for marks: 35

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(504,'raju',35)

1 row created.

SQL> /

Enter value for rno: 505

Enter value for name: ramu

Enter value for marks: 45

old 1: insert into th1 values(&rno,'&name',&marks)

new 1: insert into th1 values(505,'ramu',45)

1 row created.

SQL> insert into th2 values(&rno,&fee);

Enter value for rno: 501

Enter value for fee: 3000

old 1: insert into th2 values(&rno,&fee)

new 1: insert into th2 values(501,3000)

1 row created.

SQL> /

Enter value for rno: 502

Enter value for fee: 2000

old 1: insert into th2 values(&rno,&fee)

new 1: insert into th2 values(502,2000)

1 row created.

SQL> /

Enter value for rno: 503

Enter value for fee: 1500

old 1: insert into th2 values(&rno,&fee)

new 1: insert into th2 values(503,1500)

1 row created.

SQL> /

Enter value for rno: 504

Enter value for fee: 4000

old 1: insert into th2 values(&rno,&fee)

new 1: insert into th2 values(504,4000)

1 row created.

SQL> select \* from th1;

RNO NAME MARKS

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

501 abhi 50

502 ravi 40

503 suma 30

504 raju 35

505 ramu 45

SQL> select \* from th2;

RNO FEE

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

501 3000

502 2000

503 1500

504 4000

**Inner Join:**

SQL> select \* from th1 inner join th2 on th1.rno=th2.rno;

RNO NAME MARKS RNO FEE

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

501 abhi 50 501 3000

502 ravi 40 502 2000

503 suma 30 503 1500

504 raju 35 504 4000

SQL> select \* from th1 join th2 on th1.rno=th2.rno;

RNO NAME MARKS RNO FEE

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

501 abhi 50 501 3000

502 ravi 40 502 2000

503 suma 30 503 1500

504 raju 35 504 4000

**Outer Join:**

**Left outer join:**

SQL> select \* from th1 left outer join th2 on th1.rno=th2.rno;

RNO NAME MARKS RNO FEE

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

501 abhi 50 501 3000

502 ravi 40 502 2000

503 suma 30 503 1500

504 raju 35 504 4000

505 ramu 45

**Right outer join:**

SQL> select \* from th2 right outer join th1 on th2.rno=th1.rno;

RNO FEE RNO NAME MARKS

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

501 3000 501 abhi 50

502 2000 502 ravi 40

503 1500 503 suma 30

504 4000 504 raju 35

505 ramu 45

**Natural Join:**

SQL> select \* from th1 natural join th2;

RNO NAME MARKS FEE

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

501 abhi 50 3000

502 ravi 40 2000

503 suma 30 1500

504 raju 35 4000

SQL> select \* from th1 cross join th2;

RNO NAME MARKS RNO FEE

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

501 abhi 50 501 3000

502 ravi 40 501 3000

503 suma 30 501 3000

504 raju 35 501 3000

505 ramu 45 501 3000

501 abhi 50 502 2000

502 ravi 40 502 2000

503 suma 30 502 2000

504 raju 35 502 2000

505 ramu 45 502 2000

501 abhi 50 503 1500

RNO NAME MARKS RNO FEE

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

502 ravi 40 503 1500

503 suma 30 503 1500

504 raju 35 503 1500

505 ramu 45 503 1500

501 abhi 50 504 4000

502 ravi 40 504 4000

503 suma 30 504 4000

504 raju 35 504 4000

505 ramu 45 504 4000

20 rows selected.

SQL> select \* from th1,th2;

RNO NAME MARKS RNO FEE

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

501 abhi 50 501 3000

502 ravi 40 501 3000

503 suma 30 501 3000

504 raju 35 501 3000

505 ramu 45 501 3000

501 abhi 50 502 2000

502 ravi 40 502 2000

503 suma 30 502 2000

504 raju 35 502 2000

505 ramu 45 502 2000

501 abhi 50 503 1500

RNO NAME MARKS RNO FEE

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

502 ravi 40 503 1500

503 suma 30 503 1500

504 raju 35 503 1500

505 ramu 45 503 1500

501 abhi 50 504 4000

502 ravi 40 504 4000

503 suma 30 504 4000

504 raju 35 504 4000

505 ramu 45 504 4000

20 rows selected.

SQL> select t1.rno,t1.name from th1 t1,th2 t2 where t1.rno=t2.rno;

RNO NAME

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

501 abhi

502 ravi

503 suma

504 raju

SQL> create table sailors

  2  (

  3  sid integer,

  4  sname varchar2(33),

  5  age number(3,1),

  6  rating integer,

  7  constraints pk\_sailors primary key(sid)

  8  );

Table created.

SQL> create table boats

  2  (

  3  bid integer,

  4  bname varchar2(20),

  5  bcolor varchar2(20),

  6  constraints pk\_boats primary key(bid)

  7  );

Table created.

SQL> create table reserves

  2  (

  3  sid integer,

  4  bid integer,

  5  rdate date,

  6  constraints fk\_sailors foreign key (sid) references sailors(sid),

  7  constraints fk\_boats foreign key(bid) references boats(bid)

  8  );

Table created.

SQL> insert into sailors values(&sid,'&sname',&age,&rating);

Enter value for sid: 22

Enter value for sname: dustin

Enter value for age: 45

Enter value for rating: 7

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(22,'dustin',45,7)

1 row created.

SQL> /

Enter value for sid: 29

Enter value for sname: brutus

Enter value for age: 33

Enter value for rating: 1

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(29,'brutus',33,1)

1 row created.

SQL> /

Enter value for sid: 31

Enter value for sname: lubber

Enter value for age: 55.5

Enter value for rating: 8

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(31,'lubber',55.5,8)

1 row created.

SQL> /

Enter value for sid: 32

Enter value for sname: andy

Enter value for age: 25.5

Enter value for rating: 8

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(32,'andy',25.5,8)

1 row created.

SQL> /

Enter value for sid: 64

Enter value for sname: horatio

Enter value for age: 35

Enter value for rating: 7

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(64,'horatio',35,7)

1 row created.

SQL> /

Enter value for sid: 71

Enter value for sname: zobra

Enter value for age: 16

Enter value for rating: 10

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(71,'zobra',16,10)

1 row created.

SQL> /

Enter value for sid: 74

Enter value for sname: ravi

Enter value for age: 35

Enter value for rating: 9

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(74,'ravi',35,9)

1 row created.

SQL> /

Enter value for sid: 85

Enter value for sname: art

Enter value for age: 25

Enter value for rating: 3

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(85,'art',25,3)

1 row created.

SQL> /

Enter value for sid: 95

Enter value for sname: bob

Enter value for age: 63

Enter value for rating: 3

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(95,'bob',63,3)

1 row created.

SQL> /

Enter value for sid: 58

Enter value for sname: rusty

Enter value for age: 35

Enter value for rating: 10

old   1: insert into sailors values(&sid,'&sname',&age,&rating)

new   1: insert into sailors values(58,'rusty',35,10)

1 row created.

SQL> select \* from sailors;

       SID SNAME                                    AGE     RATING

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

        22 dustin                                    45          7

        29 brutus                                    33          1

        31 lubber                                  55.5          8

        32 andy                                    25.5          8

        64 horatio                                 35          7

        71 zobra                                     16         10

        74 ravi                                      35          9

        85 art                                       25          3

        95 bob                                       63          3

        58 rusty                                     35         10

10 rows selected.

SQL> insert into boats values(&bid,'&bname','&bcolor');

Enter value for bid: 101

Enter value for bname: interlake

Enter value for bcolor: blue

old   1: insert into boats values(&bid,'&bname','&bcolor')

new   1: insert into boats values(101,'interlake','blue')

1 row created.

SQL> /

Enter value for bid: 102

Enter value for bname: interlake

Enter value for bcolor: red

old   1: insert into boats values(&bid,'&bname','&bcolor')

new   1: insert into boats values(102,'interlake','red')

1 row created.

SQL> /

Enter value for bid: 103

Enter value for bname: clipper

Enter value for bcolor: green

old   1: insert into boats values(&bid,'&bname','&bcolor')

new   1: insert into boats values(103,'clipper','green')

1 row created.

SQL> /

Enter value for bid: 104

Enter value for bname: marine

Enter value for bcolor: red

old   1: insert into boats values(&bid,'&bname','&bcolor')

new   1: insert into boats values(104,'marine','red')

1 row created.

SQL> select \* from boats;

       BID BNAME                          BCOLOR

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

       101 interlake                      blue

       102 interlake                      red

       103 clipper                        green

       104 marine                         red

SQL> insert into reserves values(&sid,&bid,'&rdate');

Enter value for sid: 22

Enter value for bid: 101

Enter value for rdate: 10-oct-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(22,101,'10-oct-98')

1 row created.

SQL> /

Enter value for sid: 22

Enter value for bid: 102

Enter value for rdate: 10-oct-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(22,102,'10-oct-98')

1 row created.

SQL> /

Enter value for sid: 22

Enter value for bid: 103

Enter value for rdate: 10-aug-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(22,103,'10-aug-98')

1 row created.

SQL> /

Enter value for sid: 22

Enter value for bid: 104

Enter value for rdate: 10-july-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(22,104,'10-july-98')

1 row created.

SQL> /

Enter value for sid: 31

Enter value for bid: 103

Enter value for rdate: 11-jun-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(31,103,'11-jun-98')

1 row created.

SQL> /

Enter value for sid: 31

Enter value for bid: 104

Enter value for rdate: 11-dec-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(31,104,'11-dec-98')

1 row created.

SQL> /

Enter value for sid: 31

Enter value for bid: 102

Enter value for rdate: 11-oct-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(31,102,'11-oct-98')

1 row created.

SQL> /

Enter value for sid: 64

Enter value for bid: 101

Enter value for rdate: 09-may-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(64,101,'09-may-98')

1 row created.

SQL> /

Enter value for sid: 64

Enter value for bid: 102

Enter value for rdate: 09-aug-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(64,102,'09-aug-98')

1 row created.

SQL> /

Enter value for sid: 74

Enter value for bid: 103

Enter value for rdate: 09-aug-98

old   1: insert into reserves values(&sid,&bid,'&rdate')

new   1: insert into reserves values(74,103,'09-aug-98')

1 row created.

SQL> select \* from reserves;

       SID        BID RDATE

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

        22        101 10-OCT-98

        22        102 10-OCT-98

        22        103 10-AUG-98

        22        104 10-JUL-98

        31        103 11-JUN-98

        31        104 11-DEC-98

        31        102 11-OCT-98

        64        101 09-MAY-98

        64        102 09-AUG-98

        74        103 09-AUG-98

10 rows selected.

**Set Operators:**

**UNION:**

**Find the sailors name who have reserved a red boat or a green boat.**

SQL> select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and b.bcolor='red'

2 UNION

3 select s1.sname from sailors s1,boats b1,reserves r1 where s1.sid=r1.sid and b1.bid=r1.bid and b1.bcolor='green';

SNAME

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

dustin

horatio

lubber

ravi

**UNION ALL:**

**Find the sailors name who have reserved a red boat or a green boat.**

SQL> select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and b.bcolor='red'

2 UNION ALL

3 select s1.sname from sailors s1,boats b1,reserves r1 where s1.sid=r1.sid and b1.bid=r1.bid and b1.bcolor='green';

SNAME

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

dustin

dustin

lubber

lubber

horatio

dustin

lubber

ravi

8 rows selected.

**Without using union:**

**Find the sailors name who have reserved a red boat or a green boat.**

SQL> select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and (b.bcolor='red' or b.bcolor='green');

SNAME

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

dustin

dustin

dustin

lubber

lubber

lubber

horatio

ravi

8 rows selected.

**INTERSECT:**

**Find the sailors name who have reserved a red boat and a green boat.**

SQL> select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and b.bcolor='red'

2 INTERSECT

3 select s1.sname from sailors s1,boats b1,reserves r1 where s1.sid=r1.sid and b1.bid=r1.bid and b1.bcolor='green';

SNAME

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

dustin

lubber

**Without using intersect:**

SQL>select s.sname from sailors s,boats b, boats b1,reserves r,reserves r1 where s.sid=r.sid and (b.bid=r.bid and b1.bid=r1.bid)and (b.bcolor='red' and b1.bcolor='green';

SNAME

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

dustin

dustin

lubber

lubber

**MINUS:**

**Find the sailors name who have reserved a red boat but not a green boat.**

SQL> select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and b.bid=r.bid and b.bcolor='red'

2 MINUS

3 select s1.sname from sailors s1,boats b1,reserves r1 where s1.sid=r1.sid and b1.bid=r1.bid and b1.bcolor='green';

SNAME

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

horatio

**Nested Queries:**

**3)Find the name and age of youngest sailor .**

SQL> select s.sname ,s.age from sailors s

2 where s.age<=ALL(select s.age from sailors s);

SNAME AGE

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

Zobra 16

**4)Find the name of the sailors who has the highest rating.**

SQL> select s.sname from sailors s

2 where s.rating>=ALL(select s1.rating from sailors s1);

SNAME

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

Zobra

Rusty

**5)Find the names of the sailors whose rating is better than horatio**

SQL> select s.sname from sailors s

2 where s.rating>ALL(select s1.rating from sailors s1 where s1.sname='Horatio');

SNAME

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

lubber

Andy

Zobra

Ravi

Rusty

Leela

6 rows selected.

**6)Find the names of sailors whose rating is better than some sailor Horatio**

SQL> select s.sname from sailors s

2 where s.rating>ANY(select s1.rating from sailors s1 where s1.sname = 'Horatio');

SNAME

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

Zobra

Rusty

Ravi

Andy

Leela

Lubber

**MULTI NESTED**

**1)Find the names of sailors who have reserved a red boat using IN operator.**

SQL> select s.sname from sailors s

2 where s.sid IN(select r.sid from reserves r where r.bid IN(select b.bid from boats b where b.bcolor='red'));

SNAME

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

dustin

lubber

Horatio

**2) Find the names of sailors who does not have reserved a red boat using NOT IN operator.**

SQL> select s.sname from sailors s

2 where s.sid NOT IN(select r.sid from reserves r where r.bid IN(select b.bid from boats b where b.bcolor='red'));

SNAME

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

brutus

Andy

Zobra

Ravi

Art

bob

Rusty

Leela

**CORELEATED QUERIES:**

**1)Find the names of sailors who have reserved boat number: 103 using EXISTS**

SQL> select s.sname from sailors s

2 where EXISTS(select \* from reserves r where r.sid=s.sid and r.bid=103);

SNAME

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

dustin

lubber

Ravi

**2) Find the names of sailors who not have reserved boat number: 103 using NOT EXISTS**

SQL> select s.sname from sailors s

2 where NOT EXISTS(select \* from reserves r where r.sid=s.sid and r.bid=103);

SNAME

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

Zobra

Art

Horatio

Rusty

Andy

Leela

brutus

bob

8 rows selected.

**3)Find the count of distinct ratings of sailors whose age <40**

SQL> select COUNT(DISTINCT s.rating) from sailors s

2 where s.age<40;

COUNT(DISTINCTS.RATING)

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

6

**4)Find the count of distinct sailors who reserved the boats and age<40**

SQL> select COUNT(\*) from reserves r,sailors s where s.sid=r.sid and s.age<40;

COUNT(\*)

----------

3

**5)Find the bid of boats which are not reserved by sid=64**

SQL> select b.bid from boats b

2 MINUS

3 select r.bid from reserves r

4 where r.sid=64;

BID

----------

103

104

**6) Find the names of sailors having maximum rating.**

SQL> select s.sname from sailors s

2 where s.rating = (select MAX(s1.rating) from sailors s1);

SNAME

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

Zobra

Rusty

**7)Find the names of sailors who have reserved both red and green boats.**

SQL> select s.sname from sailors s , reserves r, boats b where s.sid=r.sid and b.bid=r.bid and b.bcolor='red'

2 AND s.sid IN(select r1.sid from sailors s1,boats b1,reserves r1 where r1.sid=s1.sid and b1.bid=r1.bid and b1.bcolor='green');

SNAME

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

dustin

dustin

lubber

lubber

**6. Queries on Working with Index, Sequence, Synonyms.**

**INDEX:**

SQL> create index ind on student(UPPER(sname));

Index created.

SQL> alter index ind rename to inde;

Index altered.

SQL> drop index inde;

Index dropped.

SEQUENCE:

Create sequence

SQL> create sequence s1

2 start with 1

3 increment by 1;

Sequence created.

Insert into sequence:

SQL> insert into student(sid,sname,age) values (s1.nextval,'Jungkook',24);

1 row created.

SQL> insert into student(sid,sname,age) values (s1.nextval,'Felix',19);

1 row created.

SQL> select \* from student;

SID SNAME LOGIN

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

AGE

----------

501 Harry harry@cs

20

1 Jungkook

24

503 Liam liam@cs

22

SID SNAME LOGIN

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

AGE

----------

2 Felix

19

Alter sequence:

SQL> alter sequence s1 maxvalue 200;

Sequence altered.

SQL> select \* from user\_sequences;

SEQUENCE\_NAME MIN\_VALUE MAX\_VALUE INCREMENT\_BY C O CACHE\_SIZE

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

LAST\_NUMBER

-----------

S1 1 200 1 N N 20

3

DROP sequence:

SQL> drop sequence s1;

Sequence dropped.

**7. Queries to Build Views.**

SQL> create view sail\_view as(select sid,sname from sailors);

View created.

SQL> desc sail\_view;

Name Null? Type

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

SID NOT NULL NUMBER(38)

SNAME VARCHAR2(33)

SQL> select \* from tab;

TNAME TABTYPE CLUSTERID

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

STUDENT TABLE

STUDENTS TABLE

FACULTY TABLE

STUDENT2 TABLE

COURSE1 TABLE

COURSES TABLE

STUD TABLE

FACULTY1 TABLE

STU1 TABLE

STU2 TABLE

STU3 TABLE

TNAME TABTYPE CLUSTERID

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

STU6 TABLE

STU7 TABLE

COURSE TABLE

STU8 TABLE

ENROLLED TABLE

EMPLOYEE TABLE

EMPLOYEES TABLE

DEPARTMENT TABLE

BOATS TABLE

SAILORS TABLE

RESERVERS TABLE

TNAME TABTYPE CLUSTERID

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

T2 TABLE

T3 TABLE

T1 TABLE

TH1 TABLE

TH2 TABLE

SAIL\_VIEW VIEW

28 rows selected.

SQL> insert into sail\_view values(1\*\*,'uday');

1 row created.

SQL> select \* from sail\_view;

SID SNAME

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

22 dustin

29 brutus

31 lubber

32 andy

64 andy

71 zobra

74 ravi

85 art

95 bob

58 rusty

1\*\* uday

11 rows selected.

SQL> select \* from sailors;

SID SNAME AGE RATING

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

22 dustin 45 7

29 brutus 33 1

31 lubber 55.5 8

32 andy \*\*.5 8

64 andy 35 7

71 zobra 16 10

74 ravi 35 9

85 art \*\* 3

95 bob 63 3

58 rusty 35 10

1\*\* uday

11 rows selected.

SQL> delete from sail\_view where sid=1\*\*;

1 row deleted.

SQL> select \* from sail\_view;

SID SNAME

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

22 dustin

29 brutus

31 lubber

32 andy

64 andy

71 zobra

74 ravi

85 art

95 bob

58 rusty

10 rows selected.

SQL> select \* from sailors;

SID SNAME AGE RATING

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

22 dustin 45 7

29 brutus 33 1

31 lubber 55.5 8

32 andy \*\*.5 8

64 andy 35 7

71 zobra 16 10

74 ravi 35 9

85 art \*\* 3

95 bob 63 3

58 rusty 35 10

10 rows selected.

SQL> create or replace view sail\_view as(select sid,rating from sailors);

View created.

SQL> select \* from sail\_view;

SID RATING

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

22 7

29 1

31 8

32 8

64 7

71 10

74 9

85 3

95 3

58 10

10 rows selected.

SQL> commit;

Commit complete.

SQL> create view sail\_stu as select rno,name from th1 WITH READ ONLY;

View created.

SQL> insert into sail\_stu values(45,'ooha');

insert into sail\_stu values(45,'ooha')

\*

ERROR at line 1:

ORA-01733: virtual column not allowed here

SQL> create force view vi\_stu as select \* from dummy;

Warning: View created with compilation errors.

SQL> update sail\_stu

2 set name='teja'

3 where rno=501;

set name='teja'

\*

ERROR at line 2:

ORA-01733: virtual column not allowed here

SQL> create view sail\_stu1 as select \* from th1 where marks<=100 WITH CHECK OPTION;

View created.

SQL> insert into sail\_stu1 values(12,'padm',191);

insert into sail\_stu1 values(12,'padm',191)

\*

ERROR at line 1:

ORA-01402: view WITH CHECK OPTION where-clause violation

SQL> create force view vi\_stu as select \* from dummy;

Warning: View created with compilation errors

**8.Write a PL/SQL Code using Basic Variables and Usage of Assignment**

**Write a PL/SQL program to print Hello World.**

SQL> set serveroutput on

SQL> begin

  2  dbms\_output.put\_line('Hello World');

  3  end;

  4  /

Hello World

PL/SQL procedure successfully completed.

**Write a PL/SQL program to add two numbers.**

SQL> declare

  2  a integer;

  3  b integer;

  4  c integer;

  5  begin

  6  a:=2;

  7  b:=3;

  8  c:=a+b;

  9  dbms\_output.put\_line('value of a is'||a);

 10  dbms\_output.put\_line('value of b is'||b);

 11  dbms\_output.put\_line('value of c is'||c);

 12  end;

 13  /

value of a is2

value of b is3

value of c is5

PL/SQL procedure successfully completed.

SQL> declare

  2  a integer;

  3  b integer;

  4  c integer;

  5  begin

  6  a:=2;

  7  b:=3;

  8  c:=a+b;

  9  dbms\_output.put\_line('sum of' || a|| 'and' || b || 'is' || c );

 10  end;

 11  /

sum of2and3is5

PL/SQL procedure successfully completed.

**Write a PL/SQL program to display student details.**

SQL> declare

  2  name varchar2(20);

  3  rollno integer;

  4  age integer;

  5  begin

  6  name:='Iswarya';

  7  rollno:=561;

  8  age:=19;

  9  dbms\_output.put\_line('Name is'||name);

 10  dbms\_output.put\_line('RollNo is'||rollno);

 11  dbms\_output.put\_line('Age is'||age);

 12  end;

 13  /

Name isIswarya

RollNo is561

Age is19

PL/SQL procedure successfully completed.

SQL>  declare

  2   name varchar2(20);

  3   rollno integer;

  4   age integer;

  5   begin

  6   name:='&name';

  7   rollno:=&rollno;

  8   age:=&age;

  9   dbms\_output.put\_line('Name is:'||name);

 10   dbms\_output.put\_line('RollNo is:'||rollno);

 11   dbms\_output.put\_line('Age is:'||age);

 12   end;

 13  /

Enter value for name: Iswarya

old   6:  name:='&name';

new   6:  name:='Iswarya';

Enter value for rollno: 561

old   7:  rollno:=&rollno;

new   7:  rollno:=561;

Enter value for age: 19

old   8:  age:=&age;

new   8:  age:=19;

Name is:Iswarya

RollNo is:561

Age is:19

PL/SQL procedure successfully completed.

**9. Write a PL/SQL Code to Bind and Substitute variables in PL/SQL.**

**Bind Variables:**

SQL> variable a number

SQL> begin

2 :a:=1;

3 end;

4 /

PL/SQL procedure successfully completed.

SQL> print a;

A

----------

1

SQL> exec:a:=2;

PL/SQL procedure successfully completed.

SQL> print a;

A

----------

2

**Substitue Variables:**

SQL> define name='ravi';

SQL> select '&&name' from dual;

old 1: select '&&name' from dual

new 1: select 'ravi' from dual

'RAV

----

ravi

SQL> undefine name;

SQL> select '&&name' from dual;

Enter value for name: ravi

old 1: select '&&name' from dual

new 1: select 'ravi' from dual

'RAV

----

Ravi

**10. Write a PL/SQL block using SQL and Control Structures.**

**If-Else:**

**Write a PL/SQL program to check number is even or odd.**

SQL> declare

2 n int;

3 begin

4 n:=&n;

5 if mod(n,2)=0 then

6 dbms\_output.put\_line('Even Number'||n);

7 else

8 dbms\_output.put\_line('Odd Number'||n);

9 end if;

10 end;

11 /

Enter value for n: 61

old 4: n:=&n;

new 4: n:=61;

Odd Number61

PL/SQL procedure successfully completed.

**Write a PL/SQL program to print the biggest of two numbers.**

SQL> declare

2 a int;

3 b int;

4 begin

5 a:=&a;

6 b:=&b;

7 if (a>b) then

8 dbms\_output.put\_line(a||' is bigger');

9 else

10 dbms\_output.put\_line(b||' is bigger');

11 end if;

12 end;

13 /

Enter value for a: 3

old 5: a:=&a;

new 5: a:=3;

Enter value for b: 5

old 6: b:=&b;

new 6: b:=5;

5 is bigger

PL/SQL procedure successfully completed.

**If-Elseif:**

**Write a PL/SQL program to take marks as input and print the status.**

SQL> declare

2 marks integer:=&marks;

3 begin

4 if(marks>=75)then

5 dbms\_output.put\_line('DISTINCTION');

6 elsif(marks>=60 and marks<75)then

7 dbms\_output.put\_line('FIRSTCLASS');

8 elsif(marks>=50 and marks<60)then

9 dbms\_output.put\_line('SECOND CLASS');

10 else

11 dbms\_output.put\_line('FAIL');

12 end if;

13 end;

14 /

Enter value for marks: 95

old 2: marks integer:=&marks;

new 2: marks integer:=95;

DISTINCTION

PL/SQL procedure successfully completed.

**Switch Case:**

**Write a PL/SQL program to take marks as input and print the status.**

SQL> declare

2 grade char(1):='&grade';

3 begin

4 case grade

5 when 'A' then dbms\_output.put\_line('Excellent');

6 when 'B' then dbms\_output.put\_line('Good');

7 when 'C' then dbms\_output.put\_line('Average');

8 when 'D' then dbms\_output.put\_line('Bad');

9 end case;

10 end;

11 /

Enter value for grade: A

old 2: grade char(1):='&grade';

new 2: grade char(1):='A';

Excellent

PL/SQL procedure successfully completed.

**Nested If:**

**Write a PL/SQL program to print the biggest of three numbers.**

SQL> declare

2 a int;

3 b int;

4 c int;

5 begin

6 a:=&a;

7 b:=&b;

8 c:=&c;

9 if (a>b)then

10 if(a>c)then

11 dbms\_output.put\_line(a||' is bigger');

12 else

13 dbms\_output.put\_line(c||' is bigger');

14 end if;

15 else

16 if(a>c)then

17 dbms\_output.put\_line(b||' is bigger');

18 else

19 dbms\_output.put\_line(c||' is bigger');

20 end if;

21 end if;

22 end;

23 /

Enter value for a: 3

old 6: a:=&a;

new 6: a:=3;

Enter value for b: 4

old 7: b:=&b;

new 7: b:=4;

Enter value for c: 5

old 8: c:=&c;

new 8: c:=5;

5 is bigger

PL/SQL procedure successfully completed.

**Loops:**

**Simple Loop:**

**Write a PL/SQL program to print Sequence of n numbers using simple loop.**

SQL> declare

2 a integer;

3 n integer;

4 begin

5 a:=1;

6 n:=&n;

7 loop

8 dbms\_output.put\_line(a);

9 a:=a+1;

10 exit when a>n;

11 end loop;

12 end;

13 /

Enter value for n: 6

old 6: n:=&n;

new 6: n:=6;

1

2

3

4

5

6

PL/SQL procedure successfully completed.

**While Loop:**

**Write a PL/SQL program to print Sequence of n numbers using while loop**.

SQL> declare

2 a int:=1;

3 n int:=&n;

4 begin

5 while(a<=n)loop

6 dbms\_output.put\_line(a);

7 a:=a+1;

8 end loop;

9 end;

10 /

Enter value for n: 6

old 3: n int:=&n;

new 3: n int:=6;

1

2

3

4

5

6

PL/SQL procedure successfully completed.

**For Loop:**

**Write a PL/SQL program to print Sequence of n numbers using for loop**.

SQL> declare

2 a int:=1;

3 n int:=&n;

4 begin

5 for a in 1..n

6 loop

7 dbms\_output.put\_line(a);

8 end loop;

9 end;

10 /

Enter value for n: 6

old 3: n int:=&n;

new 3: n int:=6;

1

2

3

4

5

6

PL/SQL procedure successfully completed.

**Write a PL/SQL program to calculate factorial of any given number**

SQL> declare

2 fact int:=1;

3 n int:=&n;

4 begin

5 while n > 0 loop

6 fac:=n\*fact;

7 n:=n-1;

8 end loop;

9 dbms\_output.put\_line(fac);

10 end;

11 /

Enter value for n: 5

old 3: n int:=&n;

new 3: n int:=5;

120

PL/SQL procedure successfully completed.

**Write a PL/SQL program to print multiplication table**

SQL> declare

2 i int:=1;

3 n int:=&n;

4 begin

5 for i in 1..10

6 loop

7 dbms\_output.put\_line(n || ' x ' || i || ' = ' || n\*i);

8 end loop;

9 end;

10 /

Enter value for n: 5old 6: n:=&n;new 6: n:=5;5 x 1 = 55 x 2 = 105 x 3 = 155 x 4 = 205 x 5 = 255 x 6 = 305 x 7 = 355 x 8 = 405 x 9 = 455 x 10 = 50

**11. Write a PL/SQL Code using Cursors, Exceptions and Composite Data Types**

**CURSORS :**

**i)write an explicit cursor using simple loop**.

**Program:**

SQL> set serveroutput on;

SQL> declare

2 cursor sail\_cur is

3 select sid,sname from sailors;

4 ab sail\_cur%rowtype;

5 begin

6 open sail\_cur;

7 loop

8 fetch sail\_cur into ab;

9 exit when sail\_cur%NOTFOUND;

10 dbms\_output.put\_line(ab.sid||' '||ab.sname);

11 end loop;

12 close sail\_cur;

13 end;

14 /

**Output:**

421 leela

22 dustin

29 brutus

31 lubber

32 andy

64 horatio

71 zobra

85 art

74 ravi

95 bob

58 rusty

PL/SQL procedure successfully completed.

**ii)write an explicit cursor using while loop**

**Program:**

SQL> declare

2 cursor sail\_cur is

3 select sid,sname from sailors;

4 ab sail\_cur%rowtype;

5 begin

6 open sail\_cur;

7 fetch sail\_cur into ab;

8 while sail\_cur%FOUND

9 loop

10 dbms\_output.put\_line(ab.sid||' '||ab.sname);

11 fetch sail\_cur into ab;

12 end loop;

13 close sail\_cur;

14 end;

15 /

**Output:**

421 leela

22 dustin

29 brutus

31 lubber

32 andy

64 horatio

71 zobra

85 art

74 ravi

95 bob

58 rusty

PL/SQL procedure successfully completed.

**iii) write an explicit cursor using for loop.**

**Program:**

SQL> declare

2 cursor sail\_cur is

3 select sid,sname from sailors;

4 ab sail\_cur%rowtype;

5 begin

6 for ab in sail\_cur

7 loop

8 dbms\_output.put\_line(ab.sid||' '||ab.sname);

9 end loop;

10 end;

11 /

**Output:**

421 leela

22 dustin

29 brutus

31 lubber

32 andy

64 horatio

71 zobra

85 art

74 ravi

95 bob

58 rusty

PL/SQL procedure successfully completed.

**EXCEPTIONS:**

**i)PL/SQL program to print divide by zero exception.**

**WITHOUT EXCEPTION:**

SQL> set serveroutput on;

SQL> declare

2 id number := 12;

3 BEGIN

4 id:=12/0;

5 end;

6 /

declare

\*

ERROR at line 1:

ORA-01476: divisor is equal to zero

ORA-06512: at line 4

**WITH EXCEPTION:**

SQL> declare

2 id number:=12;

3 BEGIN

4 id:=12/0;

5 exception

6 when zero\_divide then

7 dbms\_output.put\_line('Divide by zero');

8 end;

9 /

Divide by zero

PL/SQL procedure successfully completed.

**ii) PL/SQl program to print value error exception**.

**WITHOUT EXCEPTION:**

SQL> declare

2 num number:=&num;

3 BEGIN

4 dbms\_output.put\_line('Square root of '||num||' is '||sqrt(num));

5 end;

6 /

Enter value for num: -25

old 2: num number:=&num;

new 2: num number:=-25;

declare

\*

ERROR at line 1:

ORA-06502: PL/SQL: numeric or value error

ORA-06512: at line 4

**WITH EXCEPTION**:

SQL> declare

2 num number:=&num;

3 BEGIN

4 dbms\_output.put\_line('Square root of '||num||' is'||sqrt(num));

5 exception

6 when value\_error then

7 dbms\_output.put\_line('Value error');

8 end;

9 /

Enter value for num: -25

old 2: num number:=&num;

new 2: num number:=-25;

Value error

PL/SQL procedure successfully completed.

**iii) Write a PL/SQL program to print unique constraint.**

**WITHOUT EXCEPTION:**

SQL> declare

2 roll\_no int:=502;

3 sname varchar2(20):='Alekhya';

4 marks number:=99;

5 begin

6 insert into student values(roll\_no,sname,marks);

7 end;

8 /

declare

\*

ERROR at line 1:

ORA-00001: unique constraint (CSE205B0.SYS\_C005394) violated

ORA-06512: at line 6

**WITH EXCEPTION :**

SQL> declare

2 roll\_no int:=502;

3 sname varchar2(20):='Alekhya';

4 marks number:=99;

5 begin

6 insert into student values(roll\_no,sname,marks);

7 exception

8 when dup\_val\_on\_index then

9 dbms\_output.put\_line('Unique Constraint violated');

10 end;

11 /

Unique Constraint violated

PL/SQL procedure successfully completed.

**iv) PL/SQL program to print no data found exception.**

**WITHOUT EXCEPTION:**

SQL> declare

2 id int:=&id;

3 name varchar2(20);

4 begin

5 select sname into name from student where roll\_no=id;

6 dbms\_output.put\_line(name);

7 end;

8 /

Enter value for id: 503

old 2: id int:=&id;

new 2: id int:=503;

declare

\*

ERROR at line 1:

ORA-01403: no data found

ORA-06512: at line 5

**WITH EXCEPTION:**

SQL> declare

2 id int:=&id;

3 name varchar2(20);

4 begin

5 select sname into name from student where roll\_no=id;

6 dbms\_output.put\_line(name);

7 exception

8 when no\_data\_found then

9 dbms\_output.put\_line('No data found');

10 end;

11 /

Enter value for id: 503

old 2: id int:=&id;

new 2: id int:=503;

No data found

PL/SQL procedure successfully completed.

**v) Write a PL/SQL program to print case not found exception**

**WITHOUT EXCEPTION:**

SQL> declare

2 grade char(1):='&grade';

3 begin

4 case grade

5 when 'A' then dbms\_output.put\_line('Excellent');

6 when 'B' then dbms\_output.put\_line('Good');

7 when 'C' then dbms\_output.put\_line('Average');

8 when 'D' then dbms\_output.put\_line('Poor');

9 end case;

10 end;

11 /

Enter value for grade: K

old 2: grade char(1):='&grade';

new 2: grade char(1):='K';

declare

\*

ERROR at line 1:

ORA-06592: CASE not found while executing CASE statement

ORA-06512: at line 4

**WITH EXCEPTION:**

SQL> declare

2 grade char(1):='&grade';

3 begin

4 case grade

5 when 'A' then dbms\_output.put\_line('Excellent');

6 when 'B' then dbms\_output.put\_line('Good');

7 when 'C' then dbms\_output.put\_line('Average');

8 when 'D' then dbms\_output.put\_line('Poor');

9 end case;

10 exception

11 when case\_not\_found then

12 dbms\_output.put\_line('Case not found');

13 end;

14 /

Enter value for grade: J

old 2: grade char(1):='&grade';

new 2: grade char(1):='J';

Case not found

PL/SQL procedure successfully completed.

**COMPOSITE DATA TYPES:**

**Write a PL/SQL program to print the student name and marks using table type.**

**Program :**

SQL> declare

2 type namet is table of varchar2(20);

3 type grades is table of integer;

4 names namet;

5 marks grades;

6 total integer;

7 BEGIN

8 names:=namet('Shah','Mike','Maddi','Alex','Peter');

9 marks:=grades(92,87,98,97,78);

10 total:=names.count;

11 dbms\_output.put\_line('Total '||total||' Students');

12 for i in 1..total loop

13 dbms\_output.put\_line('Student: '||names(i)||' marks : '||marks(i));

14 end loop;

15 end;

16 /

**Output:**

Total 5 Students

Student: Shah marks : 92

Student: Mike marks : 87

Student: Maddi marks : 98

Student: Alex marks : 97

Student: Peter marks : 78

PL/SQL procedure successfully completed.

**Write a PL/SQL program to print student id and student name by using RECORD TYPE**

**Program:**

SQL> declare

2 type t\_name IS RECORD

3 (

4 sname student.sname%TYPE,roll\_no student.roll\_no%TYPE);

5 r\_name t\_name; --name record

6 n\_emp\_id student.roll\_no%TYPE:=502;

7 BEGIN

8 select sname,roll\_no INTO r\_name FROM student WHERE roll\_no=n\_emp\_id;

9 dbms\_output.put\_line(r\_name.sname||' , '||r\_name.roll\_no);

10 end;

11 /

**Output:**

Felix , 502

PL/SQL procedure successfully completed.

**Write a PL/SQL program to print student names and marks using VARRAYTYPE**.

**Program:**

SQL> declare

2 type namesarray IS VARRAY(5) OF varchar2(10);

3 type grades is VARRAY(5) OF integer;

4 names namesarray;

5 marks grades;

6 total integer;

7 BEGIN

8 names:=namesarray('Shah','Mike','Maddi','Alex','Peter');

9 marks:=grades(92,87,98,97,78);

10 total:=names.count;

11 dbms\_output.put\_line('Total'||total||'Students');

12 FOR i in 1..total LOOP

13 dbms\_output.put\_line('Student:'||names(i)||'marks:'||marks(i));

14 end loop;

15 end;

16 /

**Output:**

Total5Students

Student:Shahmarks:92

Student:Mikemarks:87

Student:Maddimarks:98

Student:Alexmarks:97

Student:Petermarks:78

PL/SQL procedure successfully completed.

**12. Write a PL/SQL Code using Procedures, Functions, Packages.**

**PROCEDURE:**

**i)Write a stored procedure to print hello message.**

SQL> create procedure greet

2 as

3 begin

4 dbms\_output.put\_line('Hello');

5 end;

6 /

Procedure created.

SQL> begin

2 greet;

3 end;

4 /

Hello

PL/SQL procedure successfully completed.

SQL> exec greet;

Hello

PL/SQL procedure successfully completed.

**ii)Write a stored procedure to find addition of two numbers.**

**Program:**

SQL> create procedure sum\_c(a in number,b in number, c out number)

2 as

3 begin

4 c:=a+b;

5 end;

6 /

Procedure created.

SQL> declare

2 d number;

3 begin

4 sum\_c(2,7,d);

5 dbms\_output.put\_line(d);

6 end;

7 /

9

PL/SQL procedure successfully completed.

**FUNCTION:**

**Write a stored function to find square of a given number.**

SQL> create or replace function sq(x in number)

2 return number

3 as

4 begin

5 return(x\*x);

6 end;

7 /

Function created.

SQL> begin

2 dbms\_output.put\_line(sq(7));

3 end;

4 /

49

PL/SQL procedure successfully completed.

SQL> select sq(7) from dual;

SQ(7)

----------

49

**Write a stored function to find addition of two numbers.**

SQL> create or replace function add\_c(a in number,b in number)

2 return number

3 as

4 c number;

5 begin

6 c:=a+b;

7 return c;

8 end;

9 /

Function created.

SQL> declare

2 d number;

3 begin

4 d:=add\_c(10,20);

5 dbms\_output.put\_line(d);

6 end;

7 /

30

PL/SQL procedure successfully completed.

SQL> select add\_c(20,30) from dual;

ADD\_C(20,30)

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

50

**PACKAGES:**

**Write a package to print hello message.**

SQL> create package ss1

2 as procedure greet;

3 end;

4 /

Package created.

SQL> create package body ss1

2 as procedure greet as

3 begin

4 dbms\_output.put\_line('Hello');

5 end;

6 end;

7 /

Package body created.

SQL> exec ss1.greet;

Hello

PL/SQL procedure successfully completed.

**Write a package that prints student name by passing student id.**

SQL> create or replace package pk as

2 function fun1(no in number)

3 return varchar2;

4 end;

5 /

Package created.

SQL> create or replace package body pk

2 is

3 function fun1(no in number) return varchar2

4 is

5 name varchar2(20);

6 begin

7 select sname into name from student where roll\_no = no;

8 return name;

9 end;

10 end;

11 /

Package body created.

SQL> select pk.fun1(501) from dual;

PK.FUN1(501)

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

Jungkook

**15. For a Faculty Database**

**EMPLOYEE (EMPID, FName, L Name, Address, Sex, Salary, Dept No)**

**DEPARTMENT (Dept No, D Name, HOD\_EMPID)**

**PROJECT (Proj No, P Name, Dept No)**

**WORKS\_ON (EMPID, Proj No, Hours)**

**Write SQL queries to**

**a. To Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.**

**b. Find the sum of the salaries of all employees of the ‘IT’ department, as well as the maximum salary, the minimum salary, and the average salary in this department.**

**Program:**

SQL> create table department

2 (

3 dept\_no integer,

4 dname varchar2(20),

5 hod\_empid integer,

6 primary key(dept\_no)

7 );

Table created.

SQL> create table employeee

2 (

3 empid integer,

4 fname varchar2(20),

5 lname varchar2(20),

6 address varchar2(30),

7 sex char(1),

8 dept\_no integer,

9 salary integer,

10 primary key(empid),

11 foreign key(dept\_no) references department(dept\_no)

12 );

Table created.

SQL> create table projectt

2 (

3 proj\_no integer,

4 pname varchar2(20),

5 dept\_no integer,

6 primary key(proj\_no),

7 foreign key(proj\_no) references department(dept\_no)

8 );

Table created.

SQL> create table works\_on

2 (

3 empid integer,

4 proj\_no integer,

5 hours integer,

6 foreign key(empid) references employeee(empid),

7 foreign key(proj\_no) references projectt (proj\_no)

8 );

Table created.

INSERTION INTO DEPARTMENT

SQL> insert into department values(1,&#39;CSE&#39;,1240);

1 row created.

SQL> insert into department values(2,&#39;IT&#39;,1245);

1 row created.

SQL> select \* from department;

DEPT\_NO DNAME HOD\_EMPID

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

1 CSE 1240

2 IT 1245

INSERTION INTO EMPLOYEEE

SQL> insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary);

Enter value for empid: 1201

Enter value for fname: Adarsh

Enter value for lname: Kumar

Enter value for address: Kakinada

Enter value for sex: F

Enter value for dept\_no: 1

Enter value for salary: 150000

old 1: insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary)

new 1: insert into employeee values(1201,'Adarsh','Kumar','Kakinada','F',1,150000)

1 row created.

SQL> /

Enter value for empid: 1240

Enter value for fname: Mahi

Enter value for lname: John

Enter value for address: Rajahmundry

Enter value for sex: F

Enter value for dept\_no: 1

Enter value for salary: 95000

old 1: insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary)

new 1: insert into employeee values(1240,'Mahi','John','Rajahmundry','F',1,95000)

1 row created.

SQL> /

Enter value for empid: 1245

Enter value for fname: Ramu

Enter value for lname: Murty

Enter value for address: Rajahmundry

Enter value for sex: M

Enter value for dept\_no: 2

Enter value for salary: 90000

old 1: insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary)

new 1: insert into employeee values(1245,'Ramu','Murty','Rajahmundry','M',2,90000)

1 row created.

SQL> /

Enter value for empid: 1234

Enter value for fname: Aditya

Enter value for lname: Surya

Enter value for address: Banglore

Enter value for sex: M

Enter value for dept\_no: 1

Enter value for salary: 80000

old 1: insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary)

new 1: insert into employeee values(1234,'Aditya','Surya','Banglore','M',1,80000)

1 row created.

SQL> /

Enter value for empid: 1247

Enter value for fname: Jack

Enter value for lname: Paul

Enter value for address: Banglore

Enter value for sex: M

Enter value for dept\_no: 2

Enter value for salary: 75000

old 1: insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary)

new 1: insert into employeee values(1247,'Jack','Paul','Banglore','M',2,75000)

1 row created.

SQL> /

Enter value for empid: 1235

Enter value for fname: Pradeep

Enter value for lname: Chitra

Enter value for address: Rajahmundry

Enter value for sex: M

Enter value for dept\_no: 1

Enter value for salary: 78000

old 1: insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary)

new 1: insert into employeee values(1235,'Pradeep ','Chitra','Rajahmundry','M',1,78000)

1 row created.

SQL> /

Enter value for empid: 1211

Enter value for fname: Srinivas

Enter value for lname: Kumar

Enter value for address: Hyderabad

Enter value for sex: M

Enter value for dept\_no: 1

Enter value for salary: 59000

old 1: insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary)

new 1: insert into employeee values(1211,'Srinivas','Kumar','Hyderabad','M',1,59000)

1 row created.

SQL> /

Enter value for empid: 1492

Enter value for fname: Gopala

Enter value for lname: Rao

Enter value for address: Kakinada

Enter value for sex: M

Enter value for dept\_no: 2

Enter value for salary: 65000

old 1: insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary)

new 1: insert into employeee values(1492,'Gopala','Rao','Kakinada','M',2,65000)

1 row created.

SQL> /

Enter value for empid: 1205

Enter value for fname: Eswari

Enter value for lname: Nirupama

Enter value for address: Kakinada

Enter value for sex: F

Enter value for dept\_no: 2

Enter value for salary: 65000

old 1: insert into employeee values(&empid,'&fname','&lname','&address','&sex',&dept\_no,&salary)

new 1: insert into employeee values(1205,'Eswari','Nirupama','Kakinada','F',2,65000)

1 row created.

SQL> select \* from employeee;

EMPID FNAME LNAME

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

ADDRESS S DEPT\_NO SALARY

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

1201 Adarsh Kumar

Kakinada F 1 150000

1240 Mahi John

Rajahmundry F 1 95000

1245 Ramu Murty

Rajahmundry M 2 90000

EMPID FNAME LNAME

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

ADDRESS S DEPT\_NO SALARY

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

1234 Aditya Surya

Banglore M 1 80000

1247 Jack Paul

Banglore M 2 75000

1235 Pradeep Chitra

Rajahmundry M 1 78000

EMPID FNAME LNAME

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

ADDRESS S DEPT\_NO SALARY

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

1211 Srinivas Kumar

Hyderabad M 1 59000

1492 Gopala Rao

Kakinada M 2 65000

1250 Eswari Nirupama

Kakinada F 2 65000

9 rows selected.

INSERTION INTO PROJECTT

SQL> insert into projectt values(&projno,'&pname',&dept\_no);

Enter value for projno: 100

Enter value for pname: IOT

Enter value for dept\_no: 1

old 1: insert into projectt values(&projno,'&pname',&dept\_no)

new 1: insert into projectt values(100,'IOT',1)

1 row created.

SQL> /

Enter value for projno: 101

Enter value for pname: CLOUD

Enter value for dept\_no: 1

old 1: insert into projectt values(&projno,'&pname',&dept\_no)

new 1: insert into projectt values(101,'CLOUD',1)

1 row created.

SQL> /

Enter value for projno: 102

Enter value for pname: BIGDATA

Enter value for dept\_no: 2

old 1: insert into projectt values(&projno,'&pname',&dept\_no)

new 1: insert into projectt values(103,'BIGDATA',2)

1 row created.

SQL> /

Enter value for projno: 103

Enter value for pname: NETWORKS

Enter value for dept\_no: 2

old 1: insert into projectt values(&projno,'&pname',&dept\_no)

new 1: insert into projectt values(103,'NETWORKS',2)

1 row created.

SQL> /

Enter value for projno: 104

Enter value for pname: IOT

Enter value for dept\_no: 2

old 1: insert into projectt values(&projno,'&pname',&dept\_no)

new 1: insert into projectt values(104,'IOT',2)

1 row created.

SQL> /

Enter value for projno: 105

Enter value for pname: NETWORKS

Enter value for dept\_no: 1

old 1: insert into projectt values(&projno,'&pname',&dept\_no)

new 1: insert into projectt values(105,'NETWORKS',1)

1 row created.

SQL> select \* from projectt;

PROJNO PNAME DEPT\_NO

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

100 IOT 1

101 CLOUD 1

102 BIGDATA 2

103 NETWORKS 2

104 IOT 2

105 NETWORKS 1

6 rows selected.

INSERTION INTO WORKS\_ON

SQL> insert into works\_on values(&empid,&projno,&hours);

Enter value for empid: 1245

Enter value for projno: 104

Enter value for hours: 16

old 1: insert into works\_on values(&empid,&projno,&hours)

new 1: insert into works\_on values(1245,104,16)

1 row created.

SQL> /

Enter value for empid: 1240

Enter value for projno: 101

Enter value for hours: 22

old 1: insert into works\_on values(&empid,&projno,&hours)

new 1: insert into works\_on values(1240,101,22)

1 row created.

SQL> /

Enter value for empid: 1201

Enter value for projno: 100

Enter value for hours: 31

old 1: insert into works\_on values(&empid,&projno,&hours)

new 1: insert into works\_on values(1201,100,31)

1 row created.

SQL> /

SQL> /

Enter value for empid: 1250

Enter value for projno: 102

Enter value for hours: 25

old 1: insert into works\_on values(&empid,&projno,&hours)

new 1: insert into works\_on values(1250,102,25)

1 row created.

SQL> /

Enter value for empid: 1492

Enter value for projno: 103

Enter value for hours: 25

old 1: insert into works\_on values(&empid,&projno,&hours)

new 1: insert into works\_on values(1492,103,25)

1 row created.

SQL> /

Enter value for empid: 1235

Enter value for projno: 105

Enter value for hours: 29

old 1: insert into works\_on values(&empid,&projno,&hours)

new 1: insert into works\_on values(1235,105,29)

1 row created.

SQL> select \* from works\_on;

EMPID PROJNO HOURS

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

1245 104 16

1240 101 22

1201 100 31

1492 103 25

1235 105 29

1250 102 25

6 rows selected.

To Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.

SQL> select e.empid,(e.salary+(e.salary\*0.1)) as salraise from employeee e,projectt p,work\_on w where p.pname='IOT' and w.projno=p.projno and e.empid=w.empid;

EMPID SALRAISE

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

1245 99000

1201 165000

Find the sum of the salaries of all employees of the ‘IT’ department, as well as the maximum salary, the minimum salary, and the average salary in this department.

SQL> select sum(e.salary) from employeee e,department d where e.dept\_no=d.dept\_no and d.dname='IT';

SUM(E.SALARY)

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

295000

SQL> select max(e.salary) from employeee e ,department d where e.dept\_no=d.dept\_no and d.dname='IT';

MAX(E.SALARY)

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

90000

SQL> select min(e.salary) from employeee e ,department d where e.dept\_no=d.dept\_no and d.dname='IT';

MIN(E.SALARY)

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

65000

SQL> select avg(e.salary) from employeee e ,department d where e.dept\_no=d.dept\_no and d.dname='IT';

AVG(E.SALARY)

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

73750

**16. For a Movie Database:**

**ACTOR (Act\_id, Act\_Name, Act\_Gender)**

**DIRECTOR (Dir\_id, Dir\_Name)**

**MOVIES (Mov\_id, Mov\_Title, Mov\_Year, Dir\_id)**

**MOVIE\_CAST (Act\_id, Mov\_id, Role)**

**RATING (Mov\_id, Rev\_Stars)**

**Write SQL queries to**

**1. List the titles of all movies directed by ‘STEVEN SPIELBERG’.**

**2. Find the movie names where one or more actors acted in two or more movies.**

**3. List all actors who acted in a movie before 2015 and also in a movie after 2015 (use JOIN operation).**

**4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.**

**TABLES CREATION:**

SQL> create table actor

2 (

3 act\_id integer,

4 act\_name varchar2(20),

5 act\_gender char(1),

6 primary key(act\_id)

7 );

Table created.

SQL> create table director

2 (

3 did integer,

4 dname varchar2(30),

5 primary key(did)

6 );

Table created.

SQL> create table movie

2 (

3 mid integer,

4 mtitle varchar2(20),

5 myear integer,

6 did integer,

7 primary key(mid),

8 foreign key(did) references director(did)

9 );

Table created.

SQL> create table movie\_cast

2 (

3 act\_id integer,

4 mid integer,

5 role varchar2(20),

6 primary key(act\_id,mid),

7 foreign key(act\_id) references actor(act\_id),

8 foreign key(mid) references movie(mid)

9 );

Table created.

SQL> create table rating

2 (

3 mid integer,

4 rev\_stars integer,

5 foreign key(mid) references movie(mid)

6 );

Table created.

INSERTION INTO ACTOR

SQL> insert into actor values(&act\_id,'&act\_name','&act\_gender');

Enter value for act\_id: 101

Enter value for act\_name: DICAPRIO

Enter value for act\_gender: M

old 1: insert into actor values(&act\_id,'&act\_name','&act\_gender')

new 1: insert into actor values(101,'DICAPRIO','M')

1 row created.

SQL> /

Enter value for act\_id: 102

Enter value for act\_name: KATE WINSLET

Enter value for act\_gender: F

old 1: insert into actor values(&act\_id,'&act\_name','&act\_gender')

new 1: insert into actor values(102,'KATE WINSLET ','F')

1 row created.

SQL> /

Enter value for act\_id: 103

Enter value for act\_name: SAM WORTHINGTON

Enter value for act\_gender: M

old 1: insert into actor values(&act\_id,'&act\_name','&act\_gender')

new 1: insert into actor values(103,'SAM WORTHINGTON','M')

1 row created.

SQL> /

Enter value for act\_id: 104

Enter value for act\_name: SAM NEIL

Enter value for act\_gender: M

old 1: insert into actor values(&act\_id,'&act\_name','&act\_gender')

new 1: insert into actor values(104,'SAM NEIL ','M')

1 row created.

SQL> /

Enter value for act\_id: 105

Enter value for act\_name: CATE BLANCHETT

Enter value for act\_gender: F

old 1: insert into actor values(&act\_id,'&act\_name','&act\_gender')

new 1: insert into actor values(105,'CATE BLANCHETT','F')

1 row created.

SQL> /

Enter value for act\_id: 106

Enter value for act\_name: CHRIS PRATT

Enter value for act\_gender: M

old 1: insert into actor values(&act\_id,'&act\_name','&act\_gender')

new 1: insert into actor values(106,'CHRIS PRATT','M')

1 row created.

SQL> /

Enter value for act\_id: 107

Enter value for act\_name: BRYCE DALLAS

Enter value for act\_gender: F

old 1: insert into actor values(&act\_id,'&act\_name','&act\_gender')

new 1: insert into actor values(107,'BRYCE DALLAS ','F')

1 row created.

SQL> /

Enter value for act\_id: 108

Enter value for act\_name: LAURA DERN

Enter value for act\_gender: F

old 1: insert into actor values(&act\_id,'&act\_name','&act\_gender')

new 1: insert into actor values(108,'LAURA DERN','F')

1 row created.

SQL> /

Enter value for act\_id: 109

Enter value for act\_name: DANIEL YORK

Enter value for act\_gender: F

old 1: insert into actor values(&act\_id,'&act\_name','&act\_gender')

new 1: insert into actor values(109,'DANIEL YORK','F')

1 row created.

SQL> select \* from actor;

ACT\_ID ACT\_NAME A

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

101 DICAPRIO M

102 KATE WINSLET F

103 SAM WORTHINGTON M

104 SAM NEIL M

105 CATE BLANCHETT F

106 CHRIS PRATT M

107 BRYCE DALLAS F

108 LAURA DERN F

109 DANIEL YORK F

9 rows selected.

INSERTION INTO DIRECTOR

SQL> insert into director values(&did,'&dname');

Enter value for did: 10

Enter value for dname: STEVEN SPIELBERG

old 1: insert into director values(&did,'&dname')

new 1: insert into director values(10,'STEVEN SPIELBERG')

1 row created.

SQL> /

Enter value for did: 11

Enter value for dname: JAMES CAMERON

old 1: insert into director values(&did,'&dname')

new 1: insert into director values(11,'JAMES CAMERON')

1 row created.

SQL> /

Enter value for did: 12

Enter value for dname: MARTIN SCORSESE

old 1: insert into director values(&did,'&dname')

new 1: insert into director values(12,'MARTIN SCORSESE')

1 row created.

SQL> /

Enter value for did: 13

Enter value for dname: BAZ LUHRMANN

old 1: insert into director values(&did,'&dname')

new 1: insert into director values(13,'BAZ LUHRMANN')

1 row created.

SQL> /

Enter value for did: 14

Enter value for dname: CHRISTOPHER NOLAN

old 1: insert into director values(&did,'&dname')

new 1: insert into director values(14,'CHRISTOPHER NOLAN')

1 row created.

SQL> /

Enter value for did: 15

Enter value for dname: COLIN TREVORROW

old 1: insert into director values(&did,'&dname')

new 1: insert into director values(15,'COLIN TREVORROW')

1 row created.

SQL> /

Enter value for did: 16

Enter value for dname: RIDLEY SCOTT

old 1: insert into director values(&did,'&dname')

new 1: insert into director values(16,'RIDLEY SCOTT')

1 row created.

SQL> select \* from director;

DID DNAME

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

10 STEVEN SPIELBERG

11 JAMES CAMERON

12 MARTIN SCORSESE

13 BAZ LUHRMANN

14 CHRISTOPHER NOLAN

15 COLIN TREVORROW

16 RIDLEY SCOTT

7 rows selected.

INSERTION INTO MOVIE

SQL> insert into movie values(&mid,'&mtitle',&myear,&did);

Enter value for mid: 1001

Enter value for mtitle: JURASSIC PARK

Enter value for myear: 1993

Enter value for did: 10

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(1001,'JURASSIC PARK',1993,10)

1 row created.

SQL> /

Enter value for mid: 1002

Enter value for mtitle: TITANIC

Enter value for myear: 1997

Enter value for did: 11

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(1002,'TITANIC',1997,11)

1 row created.

SQL> /

Enter value for mid: 1003

Enter value for mtitle: THE AVIATOR

Enter value for myear: 2004

Enter value for did: 12

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(1003,'THE AVIATOR',2004,12)

1 row created.

SQL> /

Enter value for mid: 1004

Enter value for mtitle: BODY OF LIES

Enter value for myear: 2008

Enter value for did: 16

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(1004,'BODY OF LIES ',2008,16)

1 row created.

SQL> /

Enter value for mid: 1005

Enter value for mtitle: AVATAR

Enter value for myear: 2009

Enter value for did: 11

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(1005,'AVATAR',2009,11)

1 row created.

SQL> /

Enter value for mid: 1006

Enter value for mtitle: INCEPTION

Enter value for myear: 2010

Enter value for did: 14

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(1006,'INCEPTION ',2010,14)

1 row created.

SQL> /

Enter value for mid: 1007

Enter value for mtitle: THE GREAT GATSBY

Enter value for myear: 2013

Enter value for did: 13

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(1007,'THE GREAT GATSBY',2013,13)

1 row created.

SQL>

SQL> /

Enter value for mid: 1008

Enter value for mtitle: JURASSIC WORLD

Enter value for myear: 2015

Enter value for did: 15

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(1008,'JURASSIC WORLD ',2015,15)

1 row created.

SQL> /

Enter value for mid: 1009

Enter value for mtitle: THE BFG

Enter value for myear: 2016

Enter value for did: 10

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(1009,'THE BFG',2016,10)

1 row created.

SQL> /

Enter value for mid: 1010

Enter value for mtitle: THE POST

Enter value for myear: 2017

Enter value for did: 10

old 1: insert into movie values(&mid,'&mtitle',&myear,&did)

new 1: insert into movie values(2010,'THE POST',2017,10)

1 row created.

SQL> select \* from movie;

MID MTITLE MYEAR DID

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

1001 JURASSIC PARK 1993 10

1002 TITANIC 1997 11

1003 THE AVIATOR 2004 12

1004 BODY OF LIES 2008 16

1005 AVATAR 2009 11

1006 INCEPTION 2010 14

1007 THE GREAT GATSBY 2013 13

1008 JURASSIC WORLD 2015 15

1009 THE BFG 2016 10

1010 THE POST 2017 10

10 rows selected.

INSERTION INTO MOVIE\_CAST

SQL> insert into movie\_cast values(&act\_id,&mid,'&role');

Enter value for act\_id: 104

Enter value for mid: 1001

Enter value for role: Hero

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(104,1001,'Hero')

1 row created.

SQL> /

Enter value for act\_id: 108

Enter value for mid: 1001

Enter value for role: Heroine

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(108,1001,'Heroine')

1 row created.

SQL> /

Enter value for act\_id: 101

Enter value for mid: 1002

Enter value for role: Hero

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(101,1002,'Hero')

1 row created.

SQL> /

Enter value for act\_id: 102

Enter value for mid: 1002

Enter value for role: Heroine

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(102,1002,'Heroine')

1 row created.

SQL> /

Enter value for act\_id: 101

Enter value for mid: 1003

Enter value for role: Hero

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(101,1003,'Hero')

1 row created.

SQL> /

Enter value for act\_id: 109

Enter value for mid: 1003

Enter value for role: Heroine

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(109,1003,'Heroine')

1 row created.

SQL> /

Enter value for act\_id: 101

Enter value for mid: 1004

Enter value for role: Hero

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(101,1004,'Hero')

1 row created.

SQL> /

Enter value for act\_id: 103

Enter value for mid: 1005

Enter value for role: Hero

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(103,1005,'Hero')

1 row created.

SQL> /

Enter value for act\_id: 101

Enter value for mid: 1006

Enter value for role: Hero

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(101,1006,'Hero')

1 row created.

SQL> /

Enter value for act\_id: 101

Enter value for mid: 1007

Enter value for role: Hero

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(101,1007,'Hero')

1 row created.

SQL> /

Enter value for act\_id: 106

Enter value for mid: 1008

Enter value for role: Hero

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(106,1008,'Hero')

1 row created.

SQL> /

Enter value for act\_id: 107

Enter value for mid: 1008

Enter value for role: Heroine

old 1: insert into movie\_cast values(&act\_id,&mid,'&role')

new 1: insert into movie\_cast values(107,1008,'Heroine')

1 row created.

SQL> select \* from movie\_cast;

ACT\_ID MID ROLE

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

104 1001 Hero

108 1001 Heroine

101 1002 Hero

102 1002 Heroine

101 1003 Hero

109 1003 Heroine

101 1004 Hero

103 1005 Hero

101 1006 Hero

101 1007 Hero

106 1008 Hero

ACT\_ID MID ROLE

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

107 1008 Heroine

12 rows selected.

INSERTION INTO RATING

SQL> insert into rating values(&mid,&rev\_stars);

Enter value for mid: 1001

Enter value for rev\_stars: 5

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1001,5)

1 row created.

SQL> /

Enter value for mid: 1002

Enter value for rev\_stars: 6

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1002,6)

1 row created.

SQL> /

Enter value for mid: 1003

Enter value for rev\_stars: 3

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1003,3)

1 row created.

SQL> /

Enter value for mid: 1004

Enter value for rev\_stars: 4

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1004,4)

1 row created.

SQL> /

Enter value for mid: 1005

Enter value for rev\_stars: 4

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1005,4)

1 row created.

SQL> /

Enter value for mid: 1006

Enter value for rev\_stars: 2

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1006,2)

1 row created.

SQL> /

Enter value for mid: 1007

Enter value for rev\_stars: 2

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1007,2)

1 row created.

SQL> /

Enter value for mid: 1008

Enter value for rev\_stars: 6

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1008,6)

1 row created.

SQL> /

Enter value for mid: 1009

Enter value for rev\_stars: 4

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1009,4)

1 row created.

SQL> /

Enter value for mid: 1010

Enter value for rev\_stars: 2

old 1: insert into rating values(&mid,&rev\_stars)

new 1: insert into rating values(1010,2)

1 row created.

SQL> select \* from rating;

MID REV\_STARS

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

1001 5

1002 6

1003 3

1004 4

1005 4

1006 2

1007 2

1008 6

1009 4

1010 2

10 rows selected.

**1.List the titles of all movies directed by ‘STEVEN SPIELBERG’.**

SQL> select mov\_title from movies where dir\_id IN(select dir\_id from director where

dir\_name='STEVEN SPIELBERG');

MOV\_TITLE

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

JURASSIC PARK

THE BFG

THE POST

**2.Find the movie names where one or more actors acted in two or more**

**movies.**

SQL> select a.act\_id,m.mov\_title from movies m,actor a,movie\_cast c where c.mov\_id=m.mov\_id and c.act\_id=a.act\_id and a.act\_id in(select act\_id from movie\_cast c group by c.act\_id having count(\*)>=2);

ACT\_ID MOV\_TITLE

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

101 TITANIC

101 THE AVIATOR

101 BODY OF LIES

101 INCEPTION

101 THE GREAT GATSBY

**3. List all actors who acted in a movie before 2015 and also in a movie**

**after 2015 (use JOIN operation).**

SQL> select act\_name,mov\_title,mov\_year from actor a join movie\_cast c on a.act\_id=c.act\_id join movies m on c.mov\_id=m.mov\_id where m.mov\_year not between 2000 and 2015;

ACT\_NAME MOV\_TITLE MOV\_YEAR

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

DICAPRIO TITANIC 1997

KATE WINSLET TITANIC 1997

SAM NEIL JURASSIC PARK 1993

LAURA DERN JURASSIC PARK 1993

**4. Find the title of movies and number of stars for each movie that has at**

**least one rating and find the highest number of stars that movie**

**received. Sort the result by movie title.**

SQL>select m.mov\_title,r.rev\_stars from movies m,rating r where m.mov\_id=r.mov\_id and

r.rev\_stars >= all(select r.rev\_stars from rating r) order by m.mov\_title;

MOV\_TITLE REV\_STARS

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

JURASSIC WORLD 6

TITANIC 6