Sir -------> 8738970899 , 9453318798

Database Notes Dated

(29-10-2022)

SQL :-

SQL stands for structured query language. It is used to perform database operations.

Based on database operations sql is divided in four parts:-

1. DDL (Data Definition Language)
2. DML (Data Manipulation Language)
3. DCL (Data Control Language)
4. TCL (Transaction Control Language)

Commands of DDL:-

~~~~~~~~~~~~~~~~~~

i.) Create :- create command is used to create a new database object.

create table table_name

create view view_name

create user user_name

ii.) alter :- alter command is used to modify structure of database object.

iii.) drop :- drop command is used to delete database object.
~~~~~~~~~~~~~~~~~~

iv.) truncate :- truncate command is used to delete all data from database object.

v.) backup :- backup command is used to take backup of database.

vi.) restore :- restore command is used to reconstruct database from its backup.

Commands of DML:-

~~~~~~~~~~~~~~~~~~

i.) insert :- The insert command is used to insert record into database object (table,view).

ii.) delete :- The delete command is used to delete record from database object.

iii.) update :- The update command is used to modify record in database object.

iv.) select :- The select command is used to select records from database object.

Commands of DCL:-

~~~~~~~~~~~~~~~~~~

i.) grant :- grant command is used to give rights to database user.

ii.) revoke :- revoke command is used to take off rights from database user.

It is just opposit to grant command.

iii.) rename:- rename command is used to change name of database object.

Commands of TCL:-

~~~~~~~~~~~~~~~~~~

i.) commit :- commit command is used to save transaction.

ii.) rollback :- rollback command is work like undo.
~~~~~~~~~~~~~~~~~~

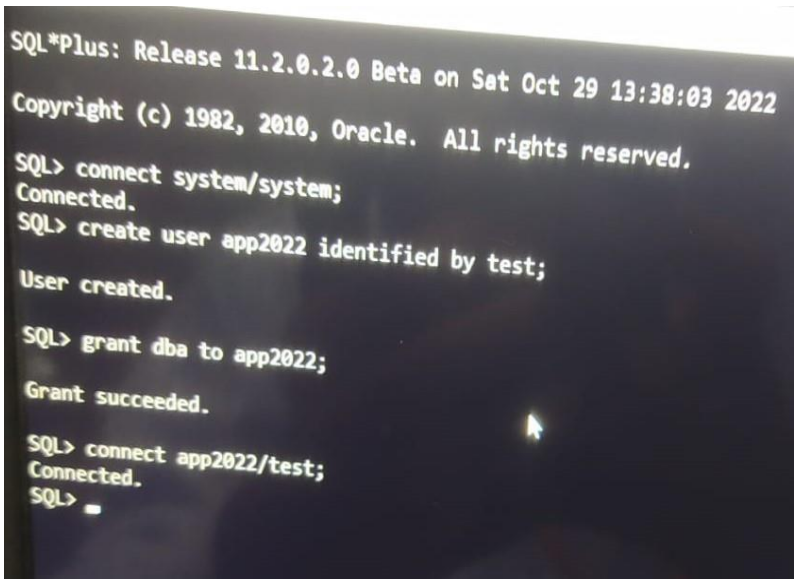
Use of create command to create a new user:-

Syntax:-

create user <username> identified by <password>;

E.g.

create user app2022 identified by test;



Use of grant command to give rights to user:-

Syntax:-

grant <rights> to <username>;

E.g.

grant dba to app2022;

Use of create command to create a new table :-

.....

Table :- Table is a collection of rows and columns.

Rows are called tuple and columns are called Attributes.

Syntax to create Table :

```
create table <tablename>
(
<column1><datatype>,
<column2><datatype>,
<column3><datatype>
) ;
```

Table Name :- employee

empid	int	primary key
empname	varchar2(30)	
department	varchar2(20)	
salary	int	

Primary Key :- Primary key is a field in a table , which is used to identify each record uniquely . It is atomic and not null.

```
create table employee
(
empid  int primary key ,
empname  varchar2(30) ,
department varchar2(20) ,
salary   int
);
```

Run SQL Command Line

```
SQL*Plus: Release 11.2.0.2.0 Beta on Sun Oct 30 11:55:07 2022

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> connect app2022/test;
Connected.
SQL>   create table employee
  2  (
  3  empid  int primary key ,
  4  empname  varchar2(30) ,
  5  department varchar2(20) ,
  6  salary   int
  7  );

Table created.
```

NOTES :-

[sable rows select krne ----->{ * } (select*from employee;)]

(table dekhne) cmd -----> description employee;

desc employee;

1. connect table →connect<username>/<password> => connect app2022/test;

2. desc employee;

```
SQL*Plus: Release 11.2.0.2.0 Beta on Sun Oct 30 11:55:07 2022

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> connect app2022/test;
Connected.
SQL>   create table employee
  2  (
  3  empid  int primary key ,
  4  empname  varchar2(30) ,
  5  department varchar2(20) ,
  6  salary   int
  7  );

Table created.

SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPID                                     NOT NULL NUMBER(38)
 EMPNAME                                            VARCHAR2(30)
 DEPARTMENT                                         VARCHAR2(20)
 SALARY                                             NUMBER(38)
```

Use of insert command to insert record into table :-

.....

Syntax :-

insert into <tablename> values

(<value1> , <value2> , <value3>);

For Eg :-

insert into employee values (1001, 'soni', 'CSE' , 50000);

```
SQL> insert into employee values (1001,'soni','CSE',50000);

1 row created.

SQL> insert into employee values(1002,'soumya','IT',20000);

1 row created.

SQL> select*from employee;

     EMPID EMPNAME                        DEPARTMENT              SALARY
---------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                      50000
      1002 soumya                         IT                       20000
```

Use of select command to select records from table :-

.....

select*from employee;

```
SQL> select*from employee;

     EMPID EMPNAME                        DEPARTMENT              SALARY
---------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                      50000
      1002 soumya                         IT                       20000
```

Ques:-

EMPID	EMPNAME	DEPARTMENT	SALARY
1001	soni	CSE	50000
1002	soumya	IT	20000
1003	Ajay	management	80000
1004	Nisha	HR	40000
1005	Shikha	development	50000

Sol :-

insert into employee values(1001,'soni','CSE' ,50000);

insert into employee values(1003,'ajay','management' ,80000);

insert into employee values(1004,'nisha','HR' ,40000);

insert into employee values(1005,'Shikha','development' ,50000);

```
SQL> insert into employee values(1003,'Ajay','management',80000);

1 row created.

SQL> insert into employee values(1004,'Nisha','HR' , 40000);

1 row created.

SQL> insert into employee values(1005,'Shikha','development' ,50000);

1 row created.

SQL> select*from employee;

     EMPID EMPNAME                        DEPARTMENT               SALARY
--------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                       50000
      1002 soumya                         IT                        20000
      1003 Ajay                           management                80000
      1004 Nisha                          HR                        40000
      1005 Shikha                         development               50000
```

Use of insert command to insert data in specific columns if table :-

……

Syntax :-

insert into <tablename> (<column1> , <column2>)

values(<values1> , <values2>) ;

Eg :

insert into employee (empid , empname , department)

values (1006 , 'mehak' , 'development');

```
SQL>
SQL>
SQL>
SQL>  insert into employee ( empid , empname , department)
  2   values (1006 , 'mehak' , 'development');

1 row created.

SQL> select*from employee
  2
SQL> select*from employee;

     EMPID EMPNAME                        DEPARTMENT               SALARY
---------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                       50000
      1002 soumya                         IT                        20000
      1003 Ajay                           management                80000
      1004 Nisha                          HR                        40000
      1005 Shikha                         development               50000
      1006 mehak                          development

6 rows selected.
```

Use of select command :-

~~~~~~~~~~~~~~~~~~~~~

1. *Use of select command to select all columns of table.*

select*from<tablename>;

select*from employee ;

2. ***Use of select command to select specific columns.***

select <column1> , <column2> from <tablename>;

select empid , empname ,salary from employee;

```
SQL> select empid, empname,salary from employee;

     EMPID EMPNAME                          SALARY
---------- ------------------------------ ----------
      1001 soni                                50000
      1002 soumya                              20000
      1003 Ajay                                80000
      1004 Nisha                               40000
      1005 Shikha                              50000
      1006 mehak

6 rows selected.

SQL> commit;

Commit complete.

SQL> select*from employee where empid=1001;

     EMPID EMPNAME                        DEPARTMENT               SALARY
---------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                       50000
```

Use of where clause :-

Where clause is used to specify condition in SQL statement .

select*from <tablename> where <condition>;

Eg :

select*from employee where empid=1001;

{* ----> sabhi column ko select krna}

```
SQL> select*from employee where empid=1001;

     EMPID EMPNAME                        DEPARTMENT              SALARY
---------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                      50000
```

……

```
SQL> select*from employee where empname='mehak';

     EMPID EMPNAME                        DEPARTMENT              SALARY
---------- ------------------------------ -------------------- ----------
      1006 mehak                          development
```

Operators in SQL

=	equality
>	greator than
<	less than
>=	greator than or equal to
<=	less than or equal to
<>	not equal
in	Compare value in given values
between	Check whether given value is available in given range or not
like	This operator is used for pattern matching .

Use of delete command to delete record from table :-

Syntax :

delete from <tablename> where <condition>;

Eg : -

delete from empid where empid=1002;

```
      1006 mehak                          development

SQL> delete from employee where empid=1002;

1 row deleted.

SQL> select*from employee;

     EMPID EMPNAME                        DEPARTMENT               SALARY
---------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                       50000
      1003 Ajay                           management                80000
      1004 Nisha                          HR                        40000
      1005 Shikha                         development               50000
      1006 mehak                          development
```

Use of UPDATE command to modify record of table :-

Syntax :

update <tablename> set <column1>=<value1>,

<column2>=<value2> where <condition>;

Eg :

update employee set salary=40000 where empid=1006;

```
SQL> update employee set salary=40000 where empid=1006;

1 row updated.

SQL> select*from employee;

     EMPID EMPNAME                        DEPARTMENT               SALARY
---------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                       50000
      1003 Ajay                           management                80000
      1004 Nisha                          HR                        40000
      1005 Shikha                         development               50000
      1006 mehak                          development               40000
```

```
SQL> update employee set salary=40000 where empid=1006;

1 row updated.

SQL> select*from employee;

     EMPID EMPNAME                        DEPARTMENT               SALARY
---------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                       50000
      1003 Ajay                           management                80000
      1004 Nisha                          HR                        40000
      1005 Shikha                         development               50000
      1006 mehak                          development               40000

SQL> update employee set salary=60000 where empid=1005;

1 row updated.

SQL> select*from employee;

     EMPID EMPNAME                        DEPARTMENT               SALARY
---------- ------------------------------ -------------------- ----------
      1001 soni                           CSE                       50000
      1003 Ajay                           management                80000
      1004 Nisha                          HR                        40000
      1005 Shikha                         development               60000
      1006 mehak                          development               40000
```

Use of truncate command :-

Syntax :-

truncate table <tablename>;

truncate table employee;

```
SQL>  truncate table employee;

Table truncated.

SQL> select*from employee;

no rows selected

SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPID                                     NOT NULL NUMBER(38)
 EMPNAME                                            VARCHAR2(30)
 DEPARTMENT                                         VARCHAR2(20)
 SALARY                                             NUMBER(38)
```

```
SQL>  truncate table employee;

Table truncated.

SQL> select*from employee;

no rows selected

SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPID                                     NOT NULL NUMBER(38)
 EMPNAME                                            VARCHAR2(30)
 DEPARTMENT                                         VARCHAR2(20)
 SALARY                                             NUMBER(38)

SQL>  drop table employee ;

Table dropped.

SQL> select*from employee;
select*from employee
            *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Use of drop :-

Syntax :

drop table<tablename>;

drop table employee ;

```
SQL>  drop table employee ;

Table dropped.

SQL> select*from employee;
select*from employee
            *
ERROR at line 1:
ORA-00942: table or view does not exist
```

commit; -------> for save the data.

TASK :----

QUES :-

ID Task Signatures with Date

Student Coordinator

1.1 i) Create a database user with user name “mydb and password “mydb”.

ii) Give privileges of DBA to the user “mydb”.

1.2 Connect the database user “mydb” and create a table cust_info with following structure:-

Field Name Datatype Constraint

Cust_Id Number(5) Primary Key

Cust_Name Varchar(20)

Cust_Address Varchar(100)

Cust_connect_date Date

Contact_No Varchar(15)

1.3 i) Insert the following records in cust_info table:-

Cust_Id	Cust_Name	Cust_Address	Cust_connect_date	ContactNo
101	Rajeev Singh	Lucknow	12-dec-2012	05226565114
102	Jitendra Verma	Sitapur	01-Jan-2013	9838505980
103	Ravi Singh	barabanki	15-Jan-2013	9936652039
104	priya Singh	Lucknow	16-Jan-2013	9936390301
105	Brijesh Mishra	barabanki	16-Feb-2013	8738970899
106	Amit Singh	Lucknow	18-mar-2013	0548-2202798

ii) Perform commit command.

1.4 Perform select operation based on following instruction:-

i) Select all record with its all fields.

ii) Select all records with its Cust_Id and Cust_Name field.

iii) Select those record in which the Cust_connect_Date is after 15-Jan-2013, select

only Cust_Id Cust_Name and Cust_connect_Date fields.

iv) Select those records in which the Custmoer belongs from “Lucknow”.

v) Display all records of Cust_info table in ascending order on Cust_Name field.

vi) Display all records of Cust_info table in descending order on Cust_Name field.

1.5 i) Delete the record of Cust_info table having CustId 105.

ii) Update the record of Cust_info table having Cust_Id 106(Update ContactNo with new value 0522-6767144).

iii) Perform truncate command with Cust_info table.

iv) Perform drop command to delete Cust_info table

Solution :-

~~~~~~~~~~
~~~~~~~~~~

~~~~~~~~~~~~~~ Create .....

a.) connect system/system;

```
SQL> connect system/system;
Connected.
SQL>
```

b.) create user mydb identified by mybd;

c.) grant dba to mydb;

~~~~~~~~~~~~~~~ connect .....

1.1) connect mydb/mydb;

```
SQL> connect mydb/mydb;
Connected.
```

1.2) create table cust_info

(

Cust_Id Number(5) Primary Key,

Cust_Name Varchar(20) ,

Cust_Address Varchar(100) ,

Cust_connect_ date Date,

Contact_No Varchar(15)

);

```
SQL> create table cust_info (
  2   Cust_Id Number(5) Primary Key,
  3  Cust_Name Varchar(20) ,
  4  Cust_Address Varchar(100) ,
  5  Cust_connect_date Date,
  6  Contact_No Varchar(15)
  7   );

Table created.
```

--→ desc cust_info;

```
SQL> desc cust_info;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 CUST_ID                                   NOT NULL NUMBER(5)
 CUST_NAME                                          VARCHAR2(20)
 CUST_ADDRESS                                       VARCHAR2(100)
 CUST_CONNECT_DATE                                  DATE
 CONTACT_NO                                         VARCHAR2(15)
```

1.3 (i)

insert into cust_info values (101,' Rajeev Singh', 'Lucknow', '12-dec-2012', '05226565114');

insert into cust_info values (102 ,'Jitendra Verma', 'Sitapur' , '01-Jan-2013' , '9838505980');

insert into cust_info values (103 ,'Ravi Singh' ,'barabanki' , '15-Jan-2013', '9936652039');

insert into cust_info values (104 , 'priya Singh' , 'Lucknow' , '16-Jan-2013', '9936390301');

insert into cust_info values (105, 'Brijesh Mishra' ,'barabanki' , '16-Feb-2013' , '8738970899');

insert into cust_info values (106 , 'Amit Singh' ,'Lucknow' , '18-mar-2013' , '0548-2202798');

```
SQL>  insert  into cust_info values ( 101,' Rajeev Singh', 'Lucknow', '12-dec-2012', '05226565114');

1 row created.

SQL>  insert  into cust_info values (102 ,'Jitendra Verma', 'Sitapur' , '01-Jan-2013' , '9838505980');

1 row created.

SQL>  insert  into cust_info values (103 ,'Ravi Singh' ,'barabanki' , '15-Jan-2013', '9936652039');

1 row created.

SQL>  insert  into cust_info values (104 , 'priya Singh' , 'Lucknow' , '16-Jan-2013', '9936390301');

1 row created.

SQL>  insert  into cust_info values (105, 'Brijesh Mishra' ,'barabanki' , '16-Feb-2013' , '8738970899');

1 row created.

SQL>  insert  into cust_info values (106 , 'Amit Singh' ,'Lucknow' , '18-mar-2013' , '0548-2202798');

1 row created.
```

(ii). commit;

```
SQL> commit;

Commit complete.
```

4(i). select*from cust_info;

```
SQL> select*from cust_info;

   CUST_ID CUST_NAME
---------- --------------------
CUST_ADDRESS
------------------------------------
CUST_CONN CONTACT_NO
--------- ---------------
       101  Rajeev Singh
Lucknow
12-DEC-12 05226565114

       102 Jitendra Verma
Sitapur
01-JAN-13 9838505980

   CUST_ID CUST_NAME
---------- --------------------
CUST_ADDRESS
------------------------------------
CUST_CONN CONTACT_NO
--------- ---------------

       103 Ravi Singh
barabanki
15-JAN-13 9936652039

       104 priya Singh
Lucknow

   CUST_ID CUST_NAME
---------- --------------------
CUST_ADDRESS
------------------------------------
CUST_CONN CONTACT_NO
--------- ---------------
16-JAN-13 9936390301

       105 Brijesh Mishra
barabanki
16-FEB-13 8738970899

       106 Amit Singh

   CUST_ID CUST_NAME
```

```
       106 Amit Singh

   CUST_ID CUST_NAME
---------- --------------------
CUST_ADDRESS
------------------------------------
CUST_CONN CONTACT_NO
--------- ---------------
Lucknow
18-MAR-13 0548-2202798


6 rows selected.
```

(ii). select Cust_Id , Cust_Name from cust_info;

```
SQL> select Cust_Id , Cust_Name from cust_info;

   CUST_ID CUST_NAME
---------- --------------------
       101  Rajeev Singh
       102 Jitendra Verma
       103 Ravi Singh
       104 priya Singh
       105 Brijesh Mishra
       106 Amit Singh

6 rows selected.
```

(iii). select Cust_Id , Cust_Name , Cust_connect_date from cust_info where cust_connect_date>'15-jan-2013';

```
SQL> select Cust_Id , Cust_Name , Cust_connect_date from cust_info  where cust_connect_date>'15-jan-2013' ;

   CUST_ID CUST_NAME            CUST_CONN
---------- -------------------- ---------
       104 priya Singh          16-JAN-13
       105 Brijesh Mishra       16-FEB-13
       106 Amit Singh           18-MAR-13
```

(iv). select*from cust_info where CUST_ADDRESS = 'Lucknow';

```
SQL> select*from cust_info where CUST_ADDRESS = 'Lucknow';

   CUST_ID CUST_NAME
---------- --------------------
CUST_ADDRESS
--------------------------------------------------------------------------------
CUST_CONN CONTACT_NO
--------- ---------------
       101  Rajeev Singh
Lucknow
12-DEC-12 05226565114

       104 priya Singh
Lucknow
16-JAN-13 9936390301

   CUST_ID CUST_NAME
---------- --------------------
CUST_ADDRESS
--------------------------------------------------------------------------------
CUST_CONN CONTACT_NO
--------- ---------------

       106 Amit Singh
Lucknow
18-MAR-13 0548-2202798


SQL> _
```

Order By :Order by is used to display records in ascending or decending order.

(v). select cust_id , cust_name from cust_info order by cust_name;

```
SQL> select  cust_id , cust_name from cust_info order by cust_name;

   CUST_ID CUST_NAME
---------- --------------------
       101  Rajeev Singh
       106 Amit Singh
       105 Brijesh Mishra
       102 Jitendra Verma
       103 Ravi Singh
       104 priya Singh

6 rows selected.
```

For decending Order :-

(vi) select cust_id , cust_name from cust_info order by cust_name desc;

```
SQL> select  cust_id , cust_name from cust_info order by cust_name desc;

   CUST_ID CUST_NAME
---------- --------------------
       104 priya Singh
       103 Ravi Singh
       102 Jitendra Verma
       105 Brijesh Mishra
       106 Amit Singh
       101  Rajeev Singh

6 rows selected.
```

5.

(i). delete from cust_info where cust_id=105;

```
SQL> delete from cust_info where cust_id=105;

1 row deleted.
```

(ii). update cust_info set contact_no='0522-6565114'

where cust_id=106;

```
SQL> select*from cust_info;

   CUST_ID CUST_NAME
---------- --------------------
CUST_ADDRESS
--------------------------------------------
CUST_CONN CONTACT_NO
--------- ---------------
       101  Rajeev Singh
Lucknow
12-DEC-12 05226565114

       102 Jitendra Verma
Sitapur
01-JAN-13 9838505980

   CUST_ID CUST_NAME
---------- --------------------
CUST_ADDRESS
--------------------------------------------
CUST_CONN CONTACT_NO
--------- ---------------

       103 Ravi Singh
barabanki
15-JAN-13 9936652039

       104 priya Singh
Lucknow

   CUST_ID CUST_NAME
---------- --------------------
CUST_ADDRESS
--------------------------------------------
CUST_CONN CONTACT_NO
--------- ---------------
16-JAN-13 9936390301

       106 Amit Singh
Lucknow
18-MAR-13 0548-2202798
```

(iii). truncate table cust_info;

```
SQL> truncate table cust_info;

Table truncated.

SQL> select*from cust_info;

no rows selected

SQL>
```

(iv). drop table cust_info;

```
SQL> drop table cust_info;

Table dropped.
```

```
SQL> drop table cust_info;

Table dropped.

SQL> select*from cust_info;
select*from cust_info
             *
ERROR at line 1:
ORA-00942: table or view does not exist
```

31-Oct-2022

Order By :- Order by is used to display records in ascending or decending order.

Use of alter command :-

alter :- The alter command is used to modify structure of database object.

create table login

(

userid varchar2(20) primary key ,

password varchar2(20)

);

```
SQL> connect app2022/test;
Connected.
SQL>  create table login
  2   (
  3   userid  varchar2(20) primary key ,
  4   password varchar2(20)
  5  );

Table created.

SQL> desc login;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERID                                    NOT NULL VARCHAR2(20)
 PASSWORD                                           VARCHAR2(20)

SQL>  alter table login  add usertype varchar2(20);

Table altered.
```

Use of alter command to ADD a NEW column :-

.....

Syntax :-

alter table <tablename> add <column_name>

<data_type>;

Eg :-

alter table login add *usertype* varchar2(20);

```
SQL>  alter table login  add usertype varchar2(20);

Table altered.

SQL> desc login;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERID                                    NOT NULL VARCHAR2(20)
 PASSWORD                                           VARCHAR2(20)
 USERTYPE                                           VARCHAR2(20)
```

Use of alter command to DELETE column :-

.....

Syntax :

alter table <table_name> drop column <column_name>;

Eg :

alter table login drop column usertype;

```
USERTYPE                                          VARCHAR2(20)

SQL>  alter table login  drop column usertype;

Table altered.

SQL> desc login;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERID                                    NOT NULL VARCHAR2(20)
 PASSWORD                                           VARCHAR2(20)
```

Use of alter command to Modify column :

.....

* datatype change ----> modify

Syntax :

alter table <tablename> modify <column_name> <data_type>;

Eg :-

alter table login modify password varchar2(10);

```
SQL> alter table login modify password varchar2(10);
```

```
SQL> alter table login modify password varchar2(10);

Table altered.

SQL> desc login;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERID                                    NOT NULL VARCHAR2(20)
 PASSWORD                                           VARCHAR2(10)
```

Use of alter command to RENAME column :-

.....

Syntax :

alter table <tablename> rename column <old_name> to <new_name>;

Eg :

alter table login rename column password to paswd;

```
SQL> alter table login rename column password to paswd;

Table altered.

SQL> desc login;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERID                                    NOT NULL VARCHAR2(20)
 PASWD                                              VARCHAR2(10)
```

Use of RENAME command to change name of database object :-

.....

Syntax :

rename <old_name> to <new_name>;

Eg :

rename login to logininfo;

```
SQL>  rename login to logininfo;

Table renamed.

SQL> desc logininfo;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERID                                    NOT NULL VARCHAR2(20)
 PASWD                                              VARCHAR2(10)
```

[kitni table h hamare database me ,

Table ----> select*from tab;

]

```
SQL>  select*from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
LOGININFO                      TABLE
```

QUES :-

create table employee

(

empid number (5) primary key ,

empname varchar2(30) ,

salary number(8)

);

```
SQL> create table employee
  2  (
  3  empid number (5) primary key ,
  4  empname varchar2(30) ,
  5  salary number(8)
  6  );

Table created.

SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPID                                     NOT NULL NUMBER(5)
 EMPNAME                                            VARCHAR2(30)
 SALARY                                             NUMBER(8)
```

insert into employee values (1001,'soni', '35000');

insert into employee values (1002,'priya', '40000');

insert into employee values (1003,'muskan', '30000');

```
SQL>
SQL> insert into employee values (1001,'soni', '35000');

1 row created.

SQL> insert into employee values (1002,'priya', '40000');

1 row created.

SQL> insert into employee values (1003,'muskan', '30000');

1 row created.
```

```
SQL> commit;
```

```
SQL> select*from employee;

     EMPID EMPNAME                          SALARY
---------- ------------------------------ ----------
      1001 soni                                35000
      1002 priya                               40000
      1003 muskan                              30000
```

```
SQL*Plus: Release 11.2.0.2.0 Beta on Tue Nov 1 11:09:50 2022

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> connect app2022/test;
Connected.
SQL> select*from tab;

TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
EMPLOYEE                       TABLE
LOGININFO                      TABLE
```

SQL FUNCTIONS

SQL has many built-in functions.

These functions are used with select command.

1.) **Sum() :-** sum() function is used to find sum of values of given column.

Eg :

select sum(salary) from employee;

select sum(salary) "Total salary" from employee;

```
SQL>   select sum(salary) from employee;

SUM(SALARY)
-----------
     105000
```

```
SQL>     select sum(salary) "Total salary" from employee;

Total salary
------------
      105000
```

```
SQL>   select sum(salary) from employee;

SUM(SALARY)
-----------
     105000

SQL>     select sum(salary) "Total salary" from employee;

Total salary
------------
      105000
```

2.) **max() :-** max() function is used to find maximum value in given column.

Eg :

select max(salary) "Maximum salary" from employee ;

```
SQL>  select max(salary) "Maximum salary" from employee ;

Maximum salary
--------------
         40000
```

3.) **min() :-** min() function is used to find **minimun** value in given column .

Eg :

select **min(salary)** "Minimum salary" from employee ;

```
SQL> select min(salary) "Minimum salary" from employee ;

Minimum salary
--------------
         30000
```

Date : 1-Nov-2022

……..

1.) Count() :- The count() function in Sql is used to count number of rows in a table.

eg :

select count(*) "No. of rows" from employee;

```
SQL>  select count(*) "No. of rows" from employee;

No. of rows
-----------
          3
```

2.) **Upper()** :- The upper() function convert string data into upper case .

eg :

select **upper**(empname) "Employee Name" from employee ;

```
SQL> select upper(empname) "Employee Name" from employee ;

Employee Name
------------------------------
SONI
PRIYA
MUSKAN
```

3.) **lower()** :- The lower() function convert string data into lower case .

eg :

select **lower**(empname) "Employee Name" from employee ;

```
SQL> select lower(empname) "Employee Name" from employee ;

Employee Name
------------------------------
soni
priya
muskan
```

4.) **avg()** :- The avg() function is used to find average of values in given column .

eg :

select **avg**(salary) "Average salary " from employee;

```
SQL> select avg(salary) "Average salary " from employee;

Average salary
---------------
          35000
```

Nested Query : - If you use a sql query inside another sql query , then it is called nested query .

It is also called subquery .

Example : 1. Write a SQL statement to find record with **largest** salary in employee table .

select *from employee **where** salary = (select max(salary) from employee);

```
SQL> select *from employee where salary = ( select max(salary) from employee );

     EMPID EMPNAME                          SALARY
---------- ------------------------------ ----------
      1002 priya                               40000
```

Example : 2. Write a SQL statement to find record with **second largest** salary in employee table.

select * from employee where salary = (select max (salary) from employee **where** salary< (select max (salary) from employee));

```
SQL>
SQL> select * from employee where salary = ( select max (salary) from employee where salary< (select max (salary) from employee ) );

     EMPID EMPNAME                          SALARY
---------- ------------------------------ ----------
      1001 soni                               35000
```

Ex : 3. Write a SQL statement to find record with **minimun** salary in employee table .

select*from employee where salary = (select min(salary) from employee);

```
SQL> select*from employee where salary = ( select min(salary) from employee);

     EMPID EMPNAME                          SALARY
---------- ------------------------------ ----------
      1003 muskan                              30000
```

Ex : 4. Write a SQL statement to find record with **second minimum** salary in employee table.

select*from employee where salary = (select min (salary) from employee where salary > (select min (salary) from employee));

```
SQL> select*from employee where salary = ( select min (salary) from employee where salary>(select min (salary) from employee));

    EMPID EMPNAME                          SALARY
--------- ------------------------------ ----------
     1001 soni                               35000
```

Join Operation :

********** If you want to select data from two tables , then you need perform join operation .

For join operation atleast one column should be common in both tables.

Foreign key : Foreign key is a field in a tbale which works as primary key in another table.

Foreign key is used to establish relationship in two tables .

NOTES : There can be more than one foreign key in a table.

Ques :-

empinfo

empid (p.k)	empname
1001	John
1002	Brown
1003	Smith
1004	Lily

product

pid(p.k)	pname	empidf.k
101	Printer	1001
102	Scanner	1002
103	Plotter	1002
104	Laptop	1003
105	Projector	

i) Create the table empinfo & product :--

```
create table empinfo (
empid        number(5)    primary key,
empname   varchar2(20)   );
```

```
SQL> create table empinfo
  2  (
  3  empid         number(5)    primary key,
  4  empname   varchar2(20)
  5  );

Table created.
```

```
create table product
(
pid         number(5) primary key ,
pname    varchar2(20),
empid     number(5),
foreign key (empid) references empinfo(empid)
);
```

```
SQL> create table product
  2  (
  3  pid          number(5) primary key ,
  4  pname    varchar2(20),
  5  empid     number(5),
  6  foreign key (empid) references empinfo(empid)
  7  );

Table created.
```

ii) Insert the record in empinfo tbale & product :-

insert into empinfo values (1001 , 'john');

insert into empinfo values (1002 , 'Brown');

insert into empinfo values (1003 , 'Smith');

insert into empinfo values (1004 , 'lily');

```
SQL> insert into empinfo values (1001 , 'john');

1 row created.

SQL> insert into empinfo values (1002 , 'Brown');

1 row created.

SQL> insert into empinfo values (1003 , 'Smith');

1 row created.

SQL> insert into empinfo values (1004 , 'lily');

1 row created.
```

```
SQL> commit;

Commit complete.
```

```
SQL> select*from empinfo;

     EMPID EMPNAME
---------- --------------------
      1001 john
      1002 Brown
      1003 Smith
      1004 lily
```

```
SQL> insert into empinfo values (1001 , 'john');

1 row created.

SQL> insert into empinfo values (1002 , 'Brown');

1 row created.

SQL> insert into empinfo values (1003 , 'Smith');

1 row created.

SQL> insert into empinfo values (1004 , 'lily');

1 row created.

SQL> commit;

Commit complete.

SQL> select*from empinfo;

     EMPID EMPNAME
---------- --------------------
      1001 john
      1002 Brown
      1003 Smith
      1004 lily
```

insert into product values (101 , 'printer' , 1001);

insert into product values (102 , 'Scanner' , 1002);

insert into product values (103 , 'Plotter', 1002);

insert into product values (104 , 'laptop' , 1003);

insert into product (pid ,pname) values (105,'projector');

```
SQL> insert into product values (101 , 'printer' , 1001);

1 row created.

SQL> insert into product values (102 , 'Scanner' , 1002);

1 row created.

SQL> insert into product values (103 , 'Plotter', 1002);

1 row created.

SQL> insert into product values (104 , 'laptop' , 1003);

1 row created.

SQL> insert into product(pid ,pname) values (105,'projector');

1 row created.
```

```
SQL> commit;
```

```
SQL> select*from product;

       PID PNAME                     EMPID
---------- -------------------- ----------
       101 printer                    1001
       102 Scanner                    1002
       103 Plotter                    1002
       104 laptop                     1003
       105 projector
```

```
SQL> insert into product values (101 , 'printer' , 1001);

1 row created.

SQL> insert into product values (102 , 'Scanner' , 1002);

1 row created.

SQL> insert into product values (103 , 'Plotter', 1002);

1 row created.

SQL> insert into product values (104 , 'laptop' , 1003);

1 row created.

SQL> insert into product(pid ,pname) values (105,'projector');

1 row created.

SQL> commit;

Commit complete.

SQL> select*from product;

       PID PNAME                    EMPID
---------- -------------------- ----------
       101 printer                    1001
       102 Scanner                    1002
       103 Plotter                    1002
       104 laptop                     1003
       105 projector
```

Natual Join Operation :

****

select empname , pname from empinfo , product

where empinfo.empid=product.empid;

```
SQL> select empname , pname from empinfo , product
  2  where empinfo.empid=product.empid;

EMPNAME              PNAME
-------------------- --------------------
john                 printer
Brown                Scanner
Brown                Plotter
Smith                laptop
```

Left Join Operation :

..... When you perform left join operation then all records of left table are displayed and matching records of right table are displayed .

select empname,pname from empinfo left join product on empinfo.empid=product.empid;

```
SQL> select empname,pname from  empinfo left join product on empinfo.empid=product.empid;

EMPNAME              PNAME
-------------------- --------------------
john                 printer
Brown                Scanner
Brown                Plotter
Smith                laptop
lily
```

Rigth join Operation :

..... When you perform right join operation then all records of right table are displayed and matching records of left table are displayed .

select empname,pname from empinfo right join product on empinfo.empid=product.empid;

```
SQL> select empname,pname from  empinfo right join product on empinfo.empid=product.empid;

EMPNAME              PNAME
-------------------- --------------------
john                 printer
Brown                Plotter
Brown                Scanner
Smith                laptop
                     projector
```

Data Base TASK - 02

<table>
<tr><th rowspan="2">ID</th><th rowspan="2">Task</th><th colspan="2">Signatures with Date</th></tr>
<tr><th>Student</th><th>Coordinator</th></tr>
<tr><td>2.1</td><td>Connect the database user “mydb” and create a table login_info having following structure:-
Field_name Data_Type Constraints
User_id Number(5) Primary Key
Passwd Varchar2(10) Not Null</td><td></td><td></td></tr>
<tr><td>2.2</td><td>i) Use alter command to add new field HINT_QUES with data type VARCHAR2 (30), in LOGIN_INFO table and view the table structure.
ii) Use alter command drop field HINT_QUES and view the table structure.
iii) Use alter command modify PASSWD field of LOGIN_INFO table with Data Type VARCHAR (15) and view the table structure.</td><td></td><td></td></tr>
<tr><td>2.3(i)</td><td><u>EMPLOYEES</u>
<table><tr><th>Field/Column Name</th><th>Type</th><th>size</th><th>Constraint</th></tr><tr><td>Employee_id</td><td>number</td><td>5</td><td>NotNull</td></tr><tr><td>Employee_Name</td><td>varchar2</td><td>20</td><td>null</td></tr></table><u>ORDERS</u>
<table><tr><th>Field/Column Name</th><th>Type</th><th>size</th><th>Constraint</th></tr><tr><td>Product_Id</td><td>number</td><td>5</td><td>NotNull</td></tr><tr><td>product</td><td>Varchar2</td><td>20</td><td>Null</td></tr><tr><td>Employee</td><td>Number</td><td>5</td><td>null</td></tr></table></td><td></td><td></td></tr>
<tr><td>(ii)</td><td>Insert the record into table Employee and Orders with following specification:-
<table><tr><th>Employee_Id</th><th>Employee_Name</th></tr><tr><td>1001</td><td>Karan</td></tr><tr><td>1002</td><td>shikhar</td></tr><tr><td>1003</td><td>rajan</td></tr></table></td><td></td><td></td></tr>
</table>

	Product_Id Product Employee_Id 1 Table 1001 2 Chair 1002 3 Printer 1003 .		
2.4	i) Perform select operation to select Employee_Name Employee Table and Product from Orders Table based on Employee_Id. ii) Perform select operation using left join to select Employee_Name from Employees table and product from Orders table based on Employee_Id. iii) Perform select operation using right join to select Employee_Name Employees table and product from Orders table based on Employee_Id.		

Solution :-

2.1) connect mydb/mydb;

```
SQL> connect mydb/mydb;
Connected.
```

ii) create table login_info

(

User_id number(5) primary key,

passwd varchar2(10) not null

);

```
SQL> create login_info
  2
SQL> create table login_info
  2  (User_id  number(5) primary key,
  3  passwd  varchar2(10) not null
  4  );

Table created.
```

iii) Desc login_info;

```
SQL> desc login_info;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------
 USER_ID                                   NOT NULL NUMBER(5)
 PASSWD                                             VARCHAR2(10)
```

```
SQL*Plus: Release 11.2.0.2.0 Beta on Mon Nov 7 18:43:47 2022

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> connect system/system;
Connected.
SQL> connect mydb/mydb;
Connected.
SQL> create table login_info
  2         (
  3         User_id  number(5) primary key,
  4         passwd  varchar2(10) not null
  5         );

Table created.

SQL> desc login_info;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USER_ID                                   NOT NULL NUMBER(5)
 PASSWD                                    NOT NULL VARCHAR2(10)
```

2.2)

i) alter table login_info add hint_ques varchar2(30);

```
SQL> alter table login_info add hint_ques varchar2(30);

Table altered.
```

desc login_info;

```
SQL> desc login_info;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 USER_ID                                   NOT NULL NUMBER(5)
 PASSWD                                             VARCHAR2(10)
 HINT_QUES                                          VARCHAR2(30)
```

ii) alter table login_info drop column hint_ques;

```
SQL> alter table login_info drop column hint_ques;

Table altered.
```

desc login_info;

```
SQL> desc login_info;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USER_ID                                   NOT NULL NUMBER(5)
 PASSWD                                    NOT NULL VARCHAR2(10)
```

==➔

```
SQL> alter table login_info drop column hint_ques;

Table altered.

SQL> desc login_info;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USER_ID                                   NOT NULL NUMBER(5)
 PASSWD                                             VARCHAR2(10)
```

iii) alter table login_info *modify passwd* varchar(15);

```
SQL> alter table login_info modify passwd varchar(15);

Table altered.
```

```
SQL>  alter table login_info modify passwd varchar(15);

Table altered.

SQL> desc login_info;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 USER_ID                                   NOT NULL NUMBER(5)
 PASSWD                                             VARCHAR2(15)
```

2.3) EMPLOYEES :-

i)

create table employees

(

employee_id number(5) not null ,

employee_name varchar2(20)

);

⇨ desc login;

```
SQL> desc employees;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 EMPLOYEE_ID                               NOT NULL NUMBER(5)
 EMPLOYEE_NAME                                      VARCHAR2(20)
```

a) ORDERS :-

```
create table orders
(
product_id  number(5) not null ,
product varchar2(20) ,
employee number(5) null
);
```

⇨ desc orders;

```
SQL> desc orders;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PRODUCT_ID                                NOT NULL NUMBER(5)
 PRODUCT                                            VARCHAR2(20)
 EMPLOYEE                                           NUMBER(5)
```

ii)a.

```
insert into employees values (1001 , 'karan');
insert into employees values (1002 , 'shikhar');
insert into employees values (1003 , 'rajan');
```

```
SQL> insert into employees values (1001 , 'karan');

1 row created.

SQL>        insert into employees values (1002 , 'shikhar');

1 row created.

SQL>        insert into employees values (1003 , 'rajan');

1 row created.
```

commit ;

```
SQL> commit;

Commit complete.
```

select*from employees;

```
SQL> select*from employees;

EMPLOYEE_ID EMPLOYEE_NAME
----------- --------------------
       1001 karan
       1002 shikhar
       1003 rajan
```

```
SQL>
SQL> insert into employees values (1001 , 'karan');

1 row created.

SQL>          insert into employees values (1002 , 'shikhar');

1 row created.

SQL>          insert into employees values (1003 , 'rajan');

1 row created.

SQL>
SQL> commit;

Commit complete.

SQL> select*from employees;

EMPLOYEE_ID EMPLOYEE_NAME
----------- --------------------
       1001 karan
       1002 shikhar
       1003 rajan
```

b) insert into orders values (1, 'table',1001);

insert into orders values (2, 'chair' ,1002) ;

insert into orders values (3, 'printer' ,1003);

⇨ commit;

⇨ select*from orders;

```
SQL> insert into orders values (1, 'table',1001 );

1 row created.

SQL>        insert into orders values ( 2,  'chair' ,1002 ) ;

1 row created.

SQL>        insert into orders values (3, 'printer' ,1003 );

1 row created.

SQL> commit;

Commit complete.

SQL> select*from orders;

PRODUCT_ID PRODUCT                EMPLOYEE
---------- -------------------- ----------
         1 table                      1001
         2 chair                      1002
         3 printer                    1003

SQL>
```

2.4)

i) select employee_name,product from employees , orders

where employees.employee_id=orders.employee_id;

ii) select employee_name,product from employees left join orders

on employees.employee_id=orders.employee_id;

iii) select employee_name,product from employees right join orders

on employees.employee_id=orders.employee_id;

2-Nov-2022

****

View:- View is a logical table, which is created from another table (main table). Main table is affected

with the change in view.

We create a table with name spiemp having fields empid, empname, grade and salary.

```
create table spiemp
(
empid number(5) primary key,
empname varchar2(30),
grade varchar2(2),
salary number(8)
);
```

Now we create a view with fields empid and empname.

```
create view emp as (select empid,empname from spiemp);
insert into emp values(1001,'Brijesh');
insert into emp values(1002,'Prashant');
insert into emp values(1003,'Seema');
insert into emp values(1004,'Shubham');
delete from emp where empid=1004;
```

Use of like operator:-

like operator is used to match a pattern in data values.

```
create table student
(
rollno number(5) primary key,
name varchar2(30),
branch varchar2(20)
);

insert into student values(1001,'Ajay Singh','CS');
insert into student values(1002,'Priya Singh','IT');
insert into student values(1003,'Brijesh Mishra','CS');
insert into student values(1004,'Prashant Seth','CS');
```

i.) select records of students with 'Singh' surname.

```
select * from student where name like '%Singh';
```

ii.) select records of students whose name is started from 'P'.

```
select * from student where name like 'P%';
```

Composit Key or Candidate Key:- If you use more than one fields for identification of record uniquely.

Then resultant key is called as composit key or candidate key.

```
create table shipment
(
S# varchar2(5),
P# varchar2(5),
QTY number(8),

primary key(S#,P#)
);
```

Check:- Check constraint is used to apply validation in table.

Example:-

Create a table with name 'staff' with following validations.

fieldname datatype validation

empid varchar2(10) Check empid must start with 'SPI'

empname varchar2(20) Check empname must be in upper case.

country varchar2(5) Check country must be either 'India' or 'Nepal'

salary number(6) Check salary not more than 100000.

```
create table staff
(
empid varchar2(10) check(empid like 'SPI%'),
empname varchar2(20) check(empname=upper(empname)),
country varchar2(5) check(country in ('India','Nepal')),
salary number(6) check(salary<=100000),
primary key(empid)
);
```

Database Tasks - 03

<table>
<tr><th rowspan="2">ID</th><th rowspan="2">Task</th><th colspan="2">Signatures with Date</th></tr>
<tr><th>Student</th><th>Coordinator</th></tr>
<tr><td>1.1</td><td>Create a table described below:-
Table name:client_master
<table>
<tr><th>Column name</th><th>Data type</th><th>Size</th></tr>
<tr><td>Client_no</td><td>Varchar2</td><td>6</td></tr>
<tr><td>Name</td><td>Varchar2</td><td>20</td></tr>
<tr><td>City</td><td>Varchar2</td><td>15</td></tr>
<tr><td>State</td><td>Varchar2</td><td>15</td></tr>
<tr><td>Bal_due</td><td>Number</td><td>10,2</td></tr>
</table>
Attributes:-
⇨ Data values being inserted into the client_no Must start with the capital letter 'C'.
⇨ Data values being inserted into the column name should be In upper case only.
⇨ Only allow "Bombay", "Delhi", "Madras" and "Calcutta" as legitimate values for column city.</td><td></td><td></td></tr>
<tr><td>1.2</td><td>Create a table sales_order_details as given structure
<table>
<tr><th>Column_name</th><th>Data_type</th><th>Size</th></tr>
<tr><td>Detlorder_no</td><td>Varchar2</td><td>6</td></tr>
<tr><td>Product_no</td><td>Varchar2</td><td>6</td></tr>
<tr><td>Qty_ordered</td><td>Number</td><td>8</td></tr>
<tr><td>Qty_disp</td><td>Number</td><td>8</td></tr>
<tr><td>Product_rate</td><td>Number</td><td>8</td></tr>
</table>
Attribute:-
⇨ Detlorder_no work as primary key.
⇨ Detlorder _no work as foreign key in sales_order table.</td><td></td><td></td></tr>
<tr><td>1.3</td><td>Create a table sales_order with following structure:-
Order_no varchar2(6)
Client_no varchar2(6)
Order_date date
Detlorder_no varchar2(6) foreign key</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

1.4 These are two tables client_master and salesman_master as given below:-

Table name:-client_master

Client no	Name	City
C00001	Ashok mehra	Mumbai
C00002	Vishal Parikh	Delhi
C00003	Ajay Mehta	Mumbai
C00004	Rohit Roy	Calcutta
C00005	Nalini deewan	Mumbai
C00006	Prem iyer	Delhi
C00007	Rahul desai	Baroda

Table name :sales man_master

Salesman no	Name	City
S00001	Manish Patel	Mumbai
S00002	Kiran dixit	Delhi
S00003	Nitesh khanna	Mumbai
S00004	Mahesh patel	Calcutta

⇨ Now retrieve the names of all the client and salesman in the city of 'Mumbai' from Client –master and salesman_master.

connect system/system;

connect app2022/test;

1.1)

```
create table client_master
(
client_no     varchar2(6),
name          varchar2(20),
city          varchar2(15),
state         varchar2(15),
bal_due       number(10,2)
);
```

➔

```
SQL*Plus: Release 11.2.0.2.0 Beta on Mon Nov 7 21:03:03 2022

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> connect system/system;
Connected.
SQL> connect app2022/test;
Connected.
SQL>     create table client_master
  2       (
  3       client_no  varchar2(6),
  4       name         varchar2(20),
  5       city            varchar2(15),
  6       state          varchar2(15),
  7       bal_due     number(10,2)
  8      );

Table created.

SQL> desc client_master;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 CLIENT_NO                                          VARCHAR2(6)
 NAME                                               VARCHAR2(20)
 CITY                                               VARCHAR2(15)
 STATE                                              VARCHAR2(15)
 BAL_DUE                                            NUMBER(10,2)

SQL> _
```

1.2)

```
create table sales_order_details
 (
  detlorder_no    varchar2(6),
  product_no      varchar2(6),
  QTY ordered     number(8),
  QTY_disp        number(8),
  product_rate    number(8)
 );
```

1.3)

```
create table sales_order
 (
  Order_no        varchar2(6),
  Client_no                varchar2(6),
  Order_date      date
  Detlorder_no    varchar2(6) foreign key
);
```

1.4)

PL/SQL

Pl/SQL is a

1.) Write a PL/SQL code to print Hello World on Screen.

```
SQL*Plus: Release 11.2.0.2.0 Beta on Thu Nov 3 11:53:57 2022

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> connect app2022/test;
Connected.
```

```
begin
dbms_output.put_line('Hello World');
end;
 /
```

```
Connected.
SQL> begin
  2  dbms_output.put_line('Hello World');
  3  end;
  4
  5  /

PL/SQL procedure successfully completed.
```

```
set  serveroutput on
 begin
dbms_output.put_line('Hello World');
end;
 /
```

```
SQL> set serveroutput on
SQL> begin
  2  dbms_output.put_line('Hello World');
  3  end;
  4  /
Hello World
```

2.) How to create variable in PL/SQL.

variablename datatype;

a int;

b int;

How to store value in variable?

a:=100;

b:=100;

1.) Write a PL/SQL code to find sum two numbers.

Sol :-

```
declare
 a int;
 b int;
 begin
 a:=100;
 b:=200;
dbms_output.put_line(a+b);
end;
 /
```

```
SQL>          declare
  2              a int;
  3              b int;
  4      begin
  5             a:=100;
  6             b:=200;
  7       dbms_output.put_line(a+b);
  8        end;
  9           /
300

PL/SQL procedure successfully completed.
```

=> declare

a int;

b int;

begin

a:=100;

b:=200;

dbms_output.put_line('Summation='||(a+b));

end;

/

```
SQL>        declare
  2             a int;
  3             b int;
  4      begin
  5             a:=100;
  6             b:=200;
  7       dbms_output.put_line('Summation='||(a+b));
  8        end;
  9           /
Summation=300

PL/SQL procedure successfully completed.
```

2.) Write a PL/SQL code to make a simple calculator.

Sol :-

```
declare
  a int;
  b int;
begin
  a:=&a;
  b:=&b;
dbms_output.put_line('Summation='||(a+b));
dbms_output.put_line('Subtraction ='||(a-b));
dbms_output.put_line('Multiplication ='||(a*b));
dbms_output.put_line(' Division ='||(a/b));
end;
 /
```

```
SQL>     declare
  2        a int;
  3        b int;
  4      begin
  5        a:=&a;
  6        b:=&b;
  7      dbms_output.put_line('Summation='||(a+b));
  8      dbms_output.put_line('Subtraction ='||(a-b));
  9      dbms_output.put_line('Multiplication ='||(a*b));
 10      dbms_output.put_line(' Division ='||(a/b));
 11      end;
 12       /
Enter value for a:
```

----------> Enter value for a :

--------> Enter value for b :

```
SQL>       begin
  2           a:=&a;
  3           b:=&b;
  4        dbms_output.put_line('Summation='||(a+b));
  5        dbms_output.put_line('Subtraction ='||(a-b));
  6        dbms_output.put_line('Multiplication ='||(a*b));
  7        dbms_output.put_line(' Division ='||(a/b));
  8        end;
  9         /
Enter value for a: 5
old   2:          a:=&a;
new   2:          a:=5;
Enter value for b: _
```

```
SQL>   declare
  2        a int;
  3        b int;
  4      begin
  5        a:=&a;
  6        b:=&b;
  7      dbms_output.put_line('Summation='||(a+b));
  8      dbms_output.put_line('Subtraction ='||(a-b));
  9      dbms_output.put_line('Multiplication ='||(a*b));
 10      dbms_output.put_line(' Division ='||(a/b));
 11      end;
 12       /
Enter value for a: 5
old   5:       a:=&a;
new   5:       a:=5;
Enter value for b: 10
old   6:       b:=&b;
new   6:       b:=10;
Summation=15
Subtraction =-5
Multiplication =50
Division =.5

PL/SQL procedure successfully completed.
```

-------> User se input lene ke liye a:=&b; ------> & => input ---

----> Here **b** = jo value user input karega..

----> a = jisme value save hogi…

3.) Write a PL/SQL code to calculate area or perimeter of area of rectangle.

<u>Sol :-</u>

Type 1 :-

```
declare
    a int;
    b int;
 begin
    a:=&a;
    b:=&b;
dbms_output.put_line('Area  of rectangle = ' ||(a*b));
dbms_output.put_line('Perimeter  of rectangle = ' ||(2*(a+b)));
    end;
     /
```

```
SQL>         declare
  2    a int;
  3    b int;
  4  begin
  5           a:=&a;
  6           b:=&b;
  7          dbms_output.put_line('Area  of rectangle = ' ||(a*b));
  8           dbms_output.put_line('Perimeter  of rectangle = ' ||(2*(a+b)));
  9           end;
 10   /
Enter value for a:
```

-----> Enter value for a:

```
SQL>          declare
  2    a int;
  3    b int;
  4  begin
  5           a:=&a;
  6           b:=&b;
  7          dbms_output.put_line('Area  of rectangle = ' ||(a*b));
  8           dbms_output.put_line('Perimeter  of rectangle = ' ||(2*(a+b)));
  9           end;
 10   /
Enter value for a: 10
old   5:          a:=&a;
new   5:          a:=10;
Enter value for b:
```

-----> Enter value for b:

Final result :-

```
SQL>          declare
  2    a int;
  3    b int;
  4  begin
  5           a:=&a;
  6           b:=&b;
  7          dbms_output.put_line('Area  of rectangle = ' ||(a*b));
  8           dbms_output.put_line('Perimeter  of rectangle = ' ||(2*(a+b)));
  9           end;
 10   /
Enter value for a: 10
old   5:          a:=&a;
new   5:          a:=10;
Enter value for b: 20
old   6:          b:=&b;
new   6:          b:=20;
Area  of rectangle = 200
Perimeter  of rectangle = 60

PL/SQL procedure successfully completed.
```

Type 2:-

```
declare
      l int;
       b int;
       a int;
       p int;
 begin
     l:=&l;
     b:=&b;
     a:=l*b;
     p:=2*(l+b);
dbms_output.put_line('Area = '||a);
dbms_output.put_line('Perimeter = '||p);
 end;
   /
```

```
SQL>
SQL>                    declare
  2                       l int;
  3     b int;
  4     a int;
  5     p int;
  6                 begin
  7                     l:=&l;
  8                     b:=&b;
  9                     a:=l*b;
 10                     p:=2*(l+b);
 11                dbms_output.put_line('Area = '||a);
 12      dbms_output.put_line('Perimeter = '||p);
 13       end;
 14         /
Enter value for l:
```

-----> Enter value for l:

```
SQL>         declare
  2             l int;
  3     b int;
  4     a int;
  5     p int;
  6          begin
  7             l:=&l;
  8             b:=&b;
  9             a:=l*b;
 10             p:=2*(l+b);
 11          dbms_output.put_line('Area = '||a);
 12  dbms_output.put_line('Perimeter = '||p);
 13  end;
 14  /
Enter value for l: 4
old   7:             l:=&l;
new   7:             l:=4;
Enter value for b:
```

-----> Enter value for b:

Final result :

```
SQL>         declare
  2             l int;
  3     b int;
  4     a int;
  5     p int;
  6          begin
  7             l:=&l;
  8             b:=&b;
  9             a:=l*b;
 10             p:=2*(l+b);
 11          dbms_output.put_line('Area = '||a);
 12  dbms_output.put_line('Perimeter = '||p);
 13  end;
 14  /
Enter value for l: 4
old   7:             l:=&l;
new   7:             l:=4;
Enter value for b: 6
old   8:             b:=&b;
new   8:             b:=6;
Area = 24
Perimeter = 20

PL/SQL procedure successfully completed.
```

Decision Control : - Decision controls are used for decision making.

i) If statement : if is a keyword , which works as decision control . we attach a condition with if statement , if condition is true then code will excuted and if given condition is false then code will not excuted .

Syntax :

.....

```
if condition then
 /* code */
  end if;
```

ii) If-Else Statement : if-else is a variation of if statement . we attach a condition with if statement . If given condition is true then if block code will excuted and if given condition is false then else block code will excuted.

Syntax :

```
if condition then
  /*code1*/
else
  /*code2*/
end if;
```

4.) Write a PL/SQL code to check given number is even or Odd .

```
declare
    n int;
    begin
    n:=&n;
 if n mod 2=0 then
dbms_output.put_line('Number is even');
 else
dbms_output.put_line('Number is odd');
 end if;
 end;
  /
```

```
SQL> declare
  2    n int;
  3  begin
  4     n:=&n;
  5           if n mod 2=0 then
  6          dbms_output.put_line('Number is even');
  7           else
  8  dbms_output.put_line('Number is odd');
  9   end if;
 10           end;
 11  /
Enter value for n: _
```

```
SQL> declare
  2    n int;
  3  begin
  4     n:=&n;
  5           if n mod 2=0 then
  6          dbms_output.put_line('Number is even');
  7           else
  8  dbms_output.put_line('Number is odd');
  9   end if;
 10           end;
 11  /
Enter value for n: 2
old   4:    n:=&n;
new   4:    n:=2;
Number is even

PL/SQL procedure successfully completed.
```

3-Nov-2022

****

Database SessionLecture - 6

****

Introduction To PL/SQL

****

The PL/SQL programming language was developed by Oracle Corporation in the late 1980s as procedural extension language for SQL and the Oracle relational database. Following are certain notable facts about PL/SQL −

❖PL/SQL is a completely portable, high-performance transaction-processing language.

❖PL/SQL provides a built-in, interpreted and OS independent programming environment.

❖PL/SQL can also directly be called from the command-line SQL*Plus interface.

❖Direct call can also be made from external programming language calls to database.

PL/SQL Blocks

****

Every PL/SQL statement ends with a semicolon (;). PL/SQL blocks can be nested within other PL/SQL blocks using BEGIN and END. Following is the basic structure of a PL/SQL block –

```
DECLARE

<declarations section>


BEGIN

<executable command(s)>


EXCEPTION

<exception handling>

END;
```

Hello World Example

.....

```
DECLARE

message varchar2(20):= 'Hello, World!';

 BEGIN

dbms_output.put_line(message);

END;

/
```

The above code will display Hello, World! Message.

Simple Calculator Code In PL/SQL DECLARE

.....

```
a int;
b int;
BEGIN
a:=&a;
b:=&b;
dbms_output.put_line('Summation='||(a+b));
dbms_output.put_line('Subtraction='||(a-b));
dbms_output.put_line('Multiplication='||(a*b));
dbms_output.put_line('Division='||(a/b));
END;
/
```

Syntax Of If & If Else Statements

.....

Syntax of If Statement:-

```
If condition then
/* If block code*/
End if;
```

Syntax of If Else Statement:-

.....

```
If condition then
/* If block code */
Else
/* Else block code*/
End if;
```

Example Application - 1

Develop a PL/SQL code to check given no. is even or odd.

```
DECLARE
n int;
BEGIN
n:=&n;
If n mod 2==0 then
dbms_output.put_line('The number is even');
Else
dbms_output.put_line('The number is odd');
End if;
END;
/
```

Syntax Of Ladder If Else If condition1 then

.....

```
/* Code 1 */
Elsif condition2 then
/* Code 2 */
Elsif condition3 then
/* Code 3*/
Else
/* Code 4 */
End If;
```

Example Application - 2

.....

```
/*

Write a PL/SQL code to make a electricity bill calculator.

Unit                          Bill
1-150                         2.40/unit
For next 151-300              3.00/unit
For next more than 300        3.20/unit

*/
```

_____
_____

Example Application – 2 (cont..)

```
DECLARE
unit number(5,2);
bill number(10,2);

BEGIN
unit:=&unit;

If unit<=150 then
bill:=unit*2.40;

Elsif unit>150 and unit<=300 then
bill:=(150*2.40)+(unit-150)*3.00;

Else
bill:=(150*2.40)+(150*3.00)+(unit-300)*3.20; End if;
dbms_output.put_line('Your bill is : '||bill); END;
/
```

Loops In PL/SQL

.....

PL/SQL Basic Loop:-

.....

In this loop structure, sequence of statements is enclosed between the LOOP and the END LOOP statements. At each iteration, the sequence of statements is executed and then control resumes at the top of the loop.

PL/SQL While Loop:-

.....

Repeats a statement or group of statements while a given condition is true. It tests the Condition before executing the loop body.

PL/SQL For Loop:-

.....

Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

PL/SQL Basic Loop

****

Syntax of PL/SQL basic Loop:-

.....

```
Loop
/* Code */
End Loop;
```

Example Application - 3

~~~~~~~~~~~~~~~~~~~~

Develop a PL/SQL code to print 1-5 numbers using basic loop.

```
declare
a number(5):=1;
begin
dbms_output.put_line('Program started');
Loop
dbms_output.put_line(a);
a:=a+1;
Exit when a>5;
End loop;
dbms_output.put_line('Program completed');
End;
/
```
~~~~~~~~~~~~~~~~~~~~

PL/SQL While Loop

……

It works like an entry-check loop in which execution block will not even be executed once if the condition is not satisfied, as the exit condition is checking before execution part. It does not require keyword 'EXIT' explicitly to exit from the loop since it is validating the condition implicitly each time of the loop.

Syntax Of While Loop:-

.....

```
WHILE <EXIT condition>
LOOP
<execution block starts>
.
.
.
<execution_block_ends>
END LOOP;
```

<u>Example Application - 4</u>

```
DECLARE
a NUMBER :=1;
```

```
BEGIN
dbms_output.put_line('Program started');  WHILE (a <= 5)
LOOP
dbms_output.put_line(a);
a:=a+1;
END LOOP;
dbms_output.put_line(‘Program completed' );  END;
/
```

PL/SQL For Loop :

……..

"FOR LOOP" statement is best suitable when you want to execute a code for a known number of times rather than based on some other conditions.

In this loop, the lower limit and the higher limit will be specified and as long as the loop variable is in between this range, the loop will be executed.

```
FOR <loop_variable> in <lower_limit> .. <higher_limit>
LOOP <execution block starts>
.
.
.
<execution_block_ends>
END LOOP;
```

Example Application - 5

```
BEGIN
dbms_output.put_line(‘Program started.' );  FOR a IN 1 .. 5 LOOP
dbms_output.put_line(a);
END LOOP;
dbms_output.put_line('Program completed.');  END;
/
```

[**Note** :- connect system

Then server connect { **set serveroutput on** }

then code…….

]

4-Nov-2022

****

1.) Write a PL/SQL code to find greatest number in three unique numbers.

[a, b, c

a>b and a>c -----> a

b>a and b>c -----> b]

Example:--

```
declare
a int;
b int;
c int;
begin
a:=&a;
b:=&b;
c:=&c;
if a>b and a>c then
dbms_output.put_line('Greatest Number ='||a);
elsif b>a and b>c then
dbms_output.put_line('Greatest no. = '||b);
else
```

dbms_output.put_line('Greatest number = '||c);

end if;

end;

/

```
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 and a>c then
 10  dbms_output.put_line('Greatest Number ='||a);
 11  elsif b>a and b>c then
 12  dbms_output.put_line('Greatest no. = '||b);
 13  else
 14  dbms_output.put_line('Greatest number = '||c);
 15  end if;
 16  end;
 17  /
Enter value for a: 10
old   6: a:=&a;
new   6: a:=10;
Enter value for b: 20
old   7: b:=&b;
new   7: b:=20;
Enter value for c: 30
old   8: c:=&c;
new   8: c:=30;
Greatest number = 30

PL/SQL procedure successfully completed.
```

SALARY calculator :-

.....

ba ---> basic salary

gs ---> gross salary

hra ---> house rent allownsence

da --->

create table account

(

empid number(5) primary key,

bs number(10,5),

hra number(10,5),

da number(10,5),

gs number(10,5)

);

```
SQL>
SQL> create table account
  2  (
  3  empid number(5) primary key,
  4  bs number(10,5),
  5  hra number(10,5),
  6  da number(10,5),
  7  gs number(10,5)
  8  );

Table created.
```

```
SQL> desc account;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------
 EMPID                                     NOT NULL NUMBER(5)
 BS                                                 NUMBER(10,5)
 HRA                                                NUMBER(10,5)
 DA                                                 NUMBER(10,5)
 GS                                                 NUMBER(10,5)
```

i.) *Write PL/SQL code to take basic salary as input and calculate hra , da and gs based on given parameters. now save value of bs , hra , da and gs in account table.*

Basic	HRA	DA
1-4000	10%	50%
4000-8000	20%	60%
8000-12000	25%	70%
More than 12000	30%	80%

```
declare
empid number(5);
bs number(10,5);
hra number(10,5);
da number(10,5);
gs number(10,5);
begin
empid:=&empid;
bs:=&bs;
if bs<=4000 then
hra:=(bs*10)/100;
da:=(bs*50)/100;
elsif bs>4000 and bs<=8000 then
hra:=(bs*20)/100;
```

```
da:=(bs*60)/100;
elsif bs>8000 and bs<=12000 then
hra:=(bs*25)/100;
da:=(bs*70)/100;
else
hra:=(bs*30)/100;
da:=(bs*80)/100;
end if;
gs:=bs+hra+da;
dbms_output.put_line('Gross Salary ='||gs);
insert into account values(empid,bs,hra,da,gs);
commit;
end;
/
```

```
SQL>  desc account;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPID                                     NOT NULL NUMBER(5)
 BS                                                 NUMBER(10,5)
 HRA                                                NUMBER(10,5)
 DA                                                 NUMBER(10,5)
 GS                                                 NUMBER(10,5)

SQL> declare
  2  empid number(5);
  3  bs number(10,5);
  4  hra number(10,5);
  5  da number(10,5);
  6  gs number(10,5);
  7  begin
  8  empid:=&empid;
  9  bs:=&bs;
 10  if bs<=4000 then
 11  hra:=(bs*10)/100;
 12  da:=(bs*50)/100;
 13  elsif bs>4000 and bs<=8000 then
 14  hra:=(bs*20)/100;
 15  da:=(bs*60)/100;
 16  elsif bs>8000 and bs<=12000 then
 17  hra:=(bs*25)/100;
 18  da:=(bs*70)/100;
 19  else
 20  hra:=(bs*30)/100;
 21  da:=(bs*80)/100;
 22  end if;
 23  gs:=bs+hra+da;
 24  dbms_output.put_line('Gross Salary ='||gs);
 25  insert into account values(empid,bs,hra,da,gs);
 26  commit;
 27  end;
 28  /
Enter value for empid: 1001
old   8: empid:=&empid;
new   8: empid:=1001;
Enter value for bs: 5000
old   9: bs:=&bs;
new   9: bs:=5000;
Gross Salary =9000
```

3.) Write PL/SQL code to print numbers from 1-10 .

```
declare
n number(3);
begin
n:=1;
loop
dbms_output.put_line(n);
n:=n+1;
exit when n>10;
end loop;
end;
/
```

```
SQL> declare
  2  n number(3);
  3  begin
  4  n:=1;
  5  loop
  6  dbms_output.put_line(n);
  7  n:=n+1;
  8  exit when n>10;
  9  end loop;
 10  end;
 11  /
1
2
3
4
5
6
7
8
9
10
```

While loop :-

.....

Write a PL/SQL code to print table of given number.

```
declare
n int;
i int;
t int;
begin
n:=&n;
i:=1;
while i<=10
loop
t:=n*i;
dbms_output.put_line(t);
i:=i+1;
end loop;
end;
/
```

```
SQL> declare
  2  n int;
  3  i int;
  4  t int;
  5  begin
  6  n:=&n;
  7  i:=1;
  8  while i<=10
  9  loop
 10  t:=n*i;
 11  dbms_output.put_line(t);
 12  i:=i+1;
 13  end loop;
 14  end;
 15  /
Enter value for n: 5
old   6: n:=&n;
new   6: n:=5;
5
10
15
20
25
30
35
40
45
50

PL/SQL procedure successfully completed.
```

SOL

Database Task – 05

1. Write a PL/SQL code to find sum of digits of given number.
2. 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 area in a table Areas.
   Table Name : Areas

radius	area

3. Write a PL/SQL code to find factorial of given number.
4. Write a PL/SQL code to print series of even number from 1-50.

Sol :-

05-Nov-2022

.....

1.)

```
create table  Accounts
(
accountID   varchar2(10)  check(accountID like 'AC%' ),
name        varchar2(30),
balance int ,
primary key(accountID)
);
```

```
SQL*Plus: Release 11.2.0.2.0 Beta on Sat Nov 5 13:00:34 2022

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> connect app2022/test;
Connected.
SQL>  create table  Accounts
  2   (
  3   accountID   varchar2(10)  check(accountID like 'AC%' ),
  4   name           varchar2(30),
  5   balance int ,
  6   primary key(accountID)
  7   );

Table created.

SQL> desc Accounts;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACCOUNTID                                 NOT NULL VARCHAR2(10)
 NAME                                               VARCHAR2(30)
 BALANCE                                            NUMBER(38)
```

```
insert into Accounts values ('AC001','Ajay',5000);
insert into Accounts values ('AC002','Robert',10000);
insert into Accounts values ('AC003','Mita',5000);
insert into Accounts values ('AC004','Sunita',15000);
insert into Accounts values ('AC005','Melba',10000);
```

```
SQL>    insert into Accounts values ('AC001','Ajay',5000);

1 row created.

SQL>    insert into Accounts values ('AC002','Robert',10000);

1 row created.

SQL>    insert into Accounts values ('AC003','Mita',5000);

1 row created.

SQL>    insert into Accounts values ('AC004','Sunita',15000);

1 row created.

SQL>    insert into Accounts values ('AC005','Melba',10000);

1 row created.

SQL> commit;

Commit complete.

SQL> select*from Accounts;

ACCOUNTID  NAME                           BALANCE
---------- ------------------------------ ----------
AC001      Ajay                                5000
AC002      Robert                             10000
AC003      Mita                                5000
AC004      Sunita                             15000
AC005      Melba                              10000
```

1.) Write a PL/SQL code block that will accept an account number from user and debit an amount of RS. 2000 from the account if the account has a minimun balance of 2000 after the amount is debited . This process is to be fired on accounts table.

```
declare
  acno varchar2(10);
   bal int;
begin
 acno:=&acno;
 select balance into bal from accounts where AccountID=acno;
 bal:=bal-2000;
```

```
if bal>=2000 then
update accounts set balance=bal where accountID=acno;
dbms_output.put_line( 'Amount is Debited');
 commit;
 else
 dbms_output.put_line('Unsufficient balance');
 end if;
 end;
    /
```

```
SQL>
SQL> declare
  2  acno varchar2(10);
  3  bal int;
  4          begin
  5          acno:=&acno;
  6          select balance into bal from accounts where AccountID=acno;
  7          bal:=bal-2000;
  8          if bal>=2000 then
  9           update accounts set balance=bal where accountID=acno;
 10          dbms_output.put_line( 'Amount is Debited');
 11           commit;
 12           else
 13           dbms_output.put_line('Unsufficient balance');
 14           end if;
 15   end;
 16    /
Enter value for acno: 'AC001'
old   5:         acno:=&acno;
new   5:         acno:='AC001';

PL/SQL procedure successfully completed.
```

```
SQL> /
Enter value for acno: 'AC002'
old   5:         acno:=&acno;
new   5:         acno:='AC002';
Amount is Debited

PL/SQL procedure successfully completed.

SQL> select*from Accounts;

ACCOUNTID  NAME                           BALANCE
---------- ------------------------------ ----------
AC001      Ajay                                 3000
AC002      Robert                               8000
AC003      Mita                                 5000
AC004      Sunita                              15000
AC005      Melba                               10000

SQL> /

ACCOUNTID  NAME                           BALANCE
---------- ------------------------------ ----------
AC001      Ajay                                 3000
AC002      Robert                               8000
AC003      Mita                                 5000
AC004      Sunita                              15000
AC005      Melba                               10000
```

2.) create table exp

```
(
 empid varchar2(10) check(empid like'E%'),
empname varchar2(30),
 salary number(6),
 primary key(empid)
);
```

➔ desc exp;

```
  6   primary key(empid)
  7   );

Table created.

SQL> desc exp;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 EMPID                                     NOT NULL VARCHAR2(10)
 EMPNAME                                            VARCHAR2(30)
 SALARY                                             NUMBER(6)
```

insert into exp values ('E001','Harry',5000);

insert into exp values ('E002','Blake',1000);

insert into exp values ('E003','Jack',5000);

insert into exp values ('E004','Clark',1000);

```
SQL>
SQL>          insert into exp values ('E001','Harry',5000);

1 row created.

SQL>          insert into exp values ('E002','Blake',1000);

1 row created.

SQL>      insert into exp values ('E003','Jack',5000);

1 row created.

SQL>      insert into exp values ('E004','Clark',1000);

1 row created.

SQL> select*from exp;

EMPID      EMPNAME                          SALARY
---------- ------------------------------ ----------
E001       Harry                                5000
E002       Blake                                1000
E003       Jack                                 5000
E004       Clark                                1000

SQL> _
```

Database Machine Test Phase – 1

1. Create a database user with name testdb with password test.
2. Give permission to database user testdb.
3. Connect database user testdb.
4. Create a table with name customer with following structure:-

Column Name	Data Type
Id	Varchar2(10) primary key, check id like 'C%'
Name	Varchar2(30), check name in upper case.
City	Varchar2(8), city in 'Lucknow', 'Kanpur'
Contactno	Varchar2(10)

5. Insert following records in customer table

Id	Name	City	Contactno
C1001	BRIJESH MISHRA	Lucknow	9453318798
C1002	RAJAT SINGH	Lucknow	9450150719
C1003	DISHA SINGH	Kanpur	9936652039
C1004	AJAY VERMA	Kanpur	9838505980

6. Do following operations in customer table
   i.) Select those customers which belongs to Lucknow.
   ii.) Select those customers which Last Name is SINGH.
   iii.) Select Id, Name of customers.
   iv.) Select Customer in ascending order with Name.
   v.) Select Customer in descending order with Name.
7. Create table Product with following structure:-

Column Name	Data Type
Pid	Number(5) primary key
Pname	Varchar2(30)
Id	Varchar2(10) foreign key to Customer table

8. Insert following records in Product table

Pid	Pname	Id
101	Laptop	C1001
102	Projector	C1002
103	Scanner	C1002
104	Plotter	C1003
105	Printer	

9. Perform following operations:-
   i.) Select name and pname from customer and product table based on id (Natural Join).
   ii.) Select name and pname from customer and product table based on id (Left Join).
   iii.) Select name and pname from customer and product table based on id (Right Join).
10. Create a table with name staff as following structure:-

Column Name	Data Type
Staffid	Number(5) primary key
Name	Varchar2(30)
Salary	Number(8)

11. Create a view with name stf in which use staffid and name.
12. Test view stf with inserting some records.
13. Write a PL/SQL code to find greatest number in three unequal numbers.
14. Write a PL/SQL code to check given number is prime or not.

Database Machine Test Solution Phase

****

Solution of Database Machine Test Phase - 1 :-

➔ connect system/system;

1.)

create user testdb identified by test;

2.)

grant dba to testdb;

3.)

connect testdb/test;

4.)

```
create table customer
(
id varchar2(10) check(id like 'C%'),
name varchar2(30) check(name=upper(name)),
city varchar2(8) check(city in('Lucknow','Kanpur')),
contactno varchar2(10),
primary key(id)
);
```

```
SQL> connect system/system;
Connected.
SQL> connect testdb/test;
Connected.
SQL> create table customer *
  2  (
  3  id varchar2(10) check(id like 'C%'),
  4  name varchar2(30) check(name=upper(name)),
  5  city varchar2(8) check(city in('Lucknow','Kanpur')),
  6  contactno varchar2(10),
  7  primary key(id)
  8  );

Table created.

SQL> desc customer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL VARCHAR2(10)
 NAME                                               VARCHAR2(30)
 CITY                                               VARCHAR2(8)
 CONTACTNO                                          VARCHAR2(10)

SQL>
```

5.)

insert into customer values('C1001','BRIJESHMISHRA','Lucknow','9453318798');

insert into customer values('C1002','RAJAT SINGH','Lucknow','9450150719');

insert into customer values('C1003','DISHA SINGH','Kanpur','9936652039');

insert into customer values('C1004','AJAY VERMA','Kanpur','9838505980');

```
SQL> insert into customer values('C1001','BRIJESHMISHRA','Lucknow','9453318798');

1 row created.

SQL> insert into customer values('C1002','RAJAT SINGH','Lucknow','9450150719');

1 row created.

SQL> insert into customer values('C1003','DISHA SINGH','Kanpur','9936652039');

1 row created.

SQL> insert into customer values('C1004','AJAY VERMA','Kanpur','9838505980');

1 row created.

SQL> commit;

Commit complete.

SQL> select*from customer;

ID         NAME                           CITY     CONTACTNO
---------- ------------------------------ -------- ----------
C1001      BRIJESHMISHRA                  Lucknow  9453318798
C1002      RAJAT SINGH                    Lucknow  9450150719
C1003      DISHA SINGH                    Kanpur   9936652039
C1004      AJAY VERMA                     Kanpur   9838505980
```

6.)

i.)

select * from customer where city='Lucknow';

```
SQL> select * from customer where city='Lucknow';

ID         NAME                           CITY     CONTACTNO
---------- ------------------------------ -------- ----------
C1001      BRIJESHMISHRA                  Lucknow  9453318798
C1002      RAJAT SINGH                    Lucknow  9450150719

SQL>
```

ii.)

select * from customer where name like '%SINGH';

```
SQL> select * from customer where name like '%SINGH';

ID         NAME                           CITY     CONTACTNO
---------- ------------------------------ -------- ----------
C1002      RAJAT SINGH                    Lucknow  9450150719
C1003      DISHA SINGH                    Kanpur   9936652039
```

iii.)

select id,name from customer;

```
SQL> select id,name from customer;

ID         NAME
---------- ----------------------------
C1001      BRIJESHMISHRA
C1002      RAJAT SINGH
C1003      DISHA SINGH
C1004      AJAY VERMA

SQL>
```

iv.)

select * from customer order by name;

```
SQL> select * from customer order by name;

ID         NAME                           CITY     CONTACTNO
---------- ------------------------------ -------- ----------
C1004      AJAY VERMA                     Kanpur   9838505980
C1001      BRIJESHMISHRA                  Lucknow  9453318798
C1003      DISHA SINGH                    Kanpur   9936652039
C1002      RAJAT SINGH                    Lucknow  9450150719
```

v.)

select * from customer order by name desc;

```
SQL>     select * from customer order by name desc;

ID         NAME                           CITY     CONTACTNO
---------- ------------------------------ -------- ----------
C1002      RAJAT SINGH                    Lucknow  9450150719
C1003      DISHA SINGH                    Kanpur   9936652039
C1001      BRIJESHMISHRA                  Lucknow  9453318798
C1004      AJAY VERMA                     Kanpur   9838505980
```

```
SQL> select * from customer where city='Lucknow';

ID         NAME                           CITY     CONTACTNO
---------- ------------------------------ -------- ----------
C1001      BRIJESHMISHRA                  Lucknow  9453318798
C1002      RAJAT SINGH                    Lucknow  9450150719

SQL> select * from customer where name like '%SINGH';

ID         NAME                           CITY     CONTACTNO
---------- ------------------------------ -------- ----------
C1002      RAJAT SINGH                    Lucknow  9450150719
C1003      DISHA SINGH                    Kanpur   9936652039

SQL> select id,name from customer;

ID         NAME
---------- ------------------------------
C1001      BRIJESHMISHRA
C1002      RAJAT SINGH
C1003      DISHA SINGH
C1004      AJAY VERMA

SQL> select * from customer order by name;

ID         NAME                           CITY     CONTACTNO
---------- ------------------------------ -------- ----------
C1004      AJAY VERMA                     Kanpur   9838505980
C1001      BRIJESHMISHRA                  Lucknow  9453318798
C1003      DISHA SINGH                    Kanpur   9936652039
C1002      RAJAT SINGH                    Lucknow  9450150719

SQL>     select * from customer order by name desc;

ID         NAME                           CITY     CONTACTNO
---------- ------------------------------ -------- ----------
C1002      RAJAT SINGH                    Lucknow  9450150719
C1003      DISHA SINGH                    Kanpur   9936652039
C1001      BRIJESHMISHRA                  Lucknow  9453318798
C1004      AJAY VERMA                     Kanpur   9838505980

SQL>
```

7.)

```
create table product
(
pid number(5) primary key,
pname varchar2(30),
id varchar2(10),
foreign key(id) references customer(id)
);
```

```
SQL>
SQL> create table product
  2  (
  3  pid number(5) primary key,
  4  pname varchar2(30),
  5  id varchar2(10),
  6  foreign key(id) references customer(id)
  7  );

Table created.

SQL> desc product;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 PID                                       NOT NULL NUMBER(5)
 PNAME                                              VARCHAR2(30)
 ID                                                 VARCHAR2(10)
```

8.)

```
insert into product values(101,'Laptop','C1001');
insert into product values(102,'Projector','C1002');
insert into product values(103,'Scanner','C1002');
insert into product values(104,'Plotter','C1003');
insert into product(pid,pname) values(105,'Printer');
```

```
Connected.
SQL> insert into product values(101,'Laptop','C1001');

1 row created.

SQL> insert into product values(102,'Projector','C1002');

1 row created.

SQL> insert into product values(103,'Scanner','C1002');

1 row created.

SQL> insert into product values(104,'Plotter','C1003');

1 row created.

SQL> insert into product(pid,pname) values(105,'Printer');

1 row created.

SQL> commit;

Commit complete.

SQL> select*from product;

       PID PNAME                          ID
---------- ------------------------------ ----------
       101 Laptop                         C1001
       102 Projector                      C1002
       103 Scanner                        C1002
       104 Plotter                        C1003
       105 Printer
```

9.

i.)

select name,pname from customer, product where customer.id = product.id;

```
SQL> select name,pname from customer, product where customer.id = product.id;

NAME                           PNAME
------------------------------ ------------------------------
BRIJESHMISHRA                  Laptop
RAJAT SINGH                    Projector
RAJAT SINGH                    Scanner
DISHA SINGH                    Plotter
```

ii.)

select name, pname from customer left join product on customer.id=product.id;

```
SQL> select name, pname from customer left join product on customer.id=product.id;

NAME                           PNAME
------------------------------ ------------------------------
BRIJESHMISHRA                  Laptop
RAJAT SINGH                    Projector
RAJAT SINGH                    Scanner
DISHA SINGH                    Plotter
AJAY VERMA
```

iii.)

select name, pname from customer right join product on customer.id=product.id;

```
SQL> select name, pname from customer right join product on customer.id=product.id;

NAME                           PNAME
------------------------------ ------------------------------
BRIJESHMISHRA                  Laptop
RAJAT SINGH                    Scanner
RAJAT SINGH                    Projector
DISHA SINGH                    Plotter
                               Printer
```

➔

```
SQL> select*from product;

       PID PNAME                          ID
---------- ------------------------------ ----------
       101 Laptop                         C1001
       102 Projector                      C1002
       103 Scanner                        C1002
       104 Plotter                        C1003
       105 Printer

SQL> select name,pname from customer, product where customer.id = product.id;

NAME                           PNAME
------------------------------ ------------------------------
BRIJESHMISHRA                  Laptop
RAJAT SINGH                    Projector
RAJAT SINGH                    Scanner
DISHA SINGH                    Plotter

SQL> select name, pname from customer left join product on customer.id=product.id;

NAME                           PNAME
------------------------------ ------------------------------
BRIJESHMISHRA                  Laptop
RAJAT SINGH                    Projector
RAJAT SINGH                    Scanner
DISHA SINGH                    Plotter
AJAY VERMA

SQL> select name, pname from customer right join product on customer.id=product.id;

NAME                           PNAME
------------------------------ ------------------------------
BRIJESHMISHRA                  Laptop
RAJAT SINGH                    Scanner
RAJAT SINGH                    Projector
DISHA SINGH                    Plotter
                               Printer
```

10.)

```
create table staff
(
staffid number(5) primary key,
name varchar2(30),
salary number(8)
);
```

```
SQL> create table staff
  2  (
  3  staffid number(5) primary key,
  4  name varchar2(30),
  5  salary number(8)
  6  );

Table created.

SQL> desc staff;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STAFFID                                   NOT NULL NUMBER(5)
 NAME                                               VARCHAR2(30)
 SALARY                                             NUMBER(8)
```

11.)

create view stf as (select staffid,name from staff);

```
SQL> create view stf as (select staffid,name from staff);

View created.
```

12.)

insert into stf values(1001,'Nisha');

insert into stf values(1002,'Ravi');

```
SQL>
SQL> insert into stf values(1001,'Nisha');

1 row created.

SQL> insert into stf values(1002,'Ravi');

1 row created.

SQL>
SQL> commit;

Commit complete.

SQL> select*from staff;

   STAFFID NAME                           SALARY
---------- ------------------------------ ----------
      1001 Nisha
      1002 Ravi
```

```
SQL> create view stf as (select staffid,name from staff);

View created.

SQL>
SQL> insert into stf values(1001,'Nisha');

1 row created.

SQL> insert into stf values(1002,'Ravi');

1 row created.

SQL>
SQL> commit;

Commit complete.

SQL> select*from staff;

   STAFFID NAME                           SALARY
---------- ------------------------------ ----------
      1001 Nisha
      1002 Ravi

SQL> select*from stf;

   STAFFID NAME
---------- ------------------------------
      1001 Nisha
      1002 Ravi

SQL> _
```

13.)

```
declare
a int;
b int;
c int;
begin
a:=&a;
```

```
b:=&b;
c:=&c;
if a>b and a>c then
dbms_output.put_line('Greatest No='||a);
elsif b>a and b>c then
dbms_output.put_line('Greatest No='||b);
else
dbms_output.put_line('Greatest No='||c);
end if;
end;
/
```

```
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 and a>c then
 10  dbms_output.put_line('Greatest No='||a);
 11  elsif b>a and b>c then
 12  dbms_output.put_line('Greatest No='||b);
 13  else
 14  dbms_output.put_line('Greatest No='||c);
 15  end if;
 16  end;
 17  /
Enter value for a: 40
old   6: a:=&a;
new   6: a:=40;
Enter value for b: 50
old   7: b:=&b;
new   7: b:=50;
Enter value for c: 160
old   8: c:=&c;
new   8: c:=160;

PL/SQL procedure successfully completed.

SQL>
```

14.

```
declare
n int;
i int;
c int;
begin
n:=&n;
c:=0;
i:=1;
while i<=n loop
if n mod i=0 then
c:=c+1;
end if;
i:=i+1;
end loop;
if c=2 then
dbms_output.put_line('Prime');
else
dbms_output.put_line('Non-prime');
end if;
end;
/
```

```
SQL> declare
  2  n int;
  3  i int;
  4  c int;
  5  begin
  6  n:=&n;
  7  c:=0;
  8  i:=1;
  9  while i<=n loop
 10  if n mod i=0 then
 11  c:=c+1;
 12  end if;
 13  i:=i+1;
 14  end loop;
 15  if c=2 then
 16  dbms_output.put_line('Prime');
 17  else
 18  dbms_output.put_line('Non-prime');
 19  end if;
 20  end;
 21  /
Enter value for n:
```

```
SQL> declare
  2  n int;
  3  i int;
  4  c int;
  5  begin
  6  n:=&n;
  7  c:=0;
  8  i:=1;
  9  while i<=n loop
 10  if n mod i=0 then
 11  c:=c+1;
 12  end if;
 13  i:=i+1;
 14  end loop;
 15  if c=2 then
 16  dbms_output.put_line('Prime');
 17  else
 18  dbms_output.put_line('Non-prime');
 19  end if;
 20  end;
 21  /
Enter value for n: 60
old   6: n:=&n;
new   6: n:=60;

PL/SQL procedure successfully completed.

SQL> /
Enter value for n: 2
old   6: n:=&n;
new   6: n:=2;

PL/SQL procedure successfully completed.
```

7-nov-2022

…….

Machine Phase Test -2

Database Machine Test Phase – 2

1. Connect database user testdb.
2. Create a table with name employee with following structure:-

Column Name	Data Type
Empid	Number(5) primary key
Empname	Varchar2(30)
Department	Varchar(20)
Salary	Number(8)

3. Insert following records in employee table:-

Empid	Empname	Department	Salary
1001	Ravi Singh	Management	80000
1002	Brijesh Mishra	Development	45000
1003	Rajat Verma	Management	50000
1004	Krishna	Development	35000
1005	Nisha Singh	HR	38000

4. Now perform following operations on employee table:-
   i.) Select records of employees of Development department.
   ii.) Delete record of employee with empid 1005.
   iii.) Update department with Management and salary with 60000 of record with empid 1002.
   iv.) Write sql statement to show record with maximum salary.
   v.) Write sql statement to show record with second largest salary.
   vi.) Truncate table employee.
   vii.) Drop table employee.
5. Create a table elect_bill with following structure:-

Column Name	Data Type
Id	Number(5) primary key
Unit	Number(5)
Bill	Number(10,5)

6. Write PL/SQL code to take id and unit as input and calculate bill based on following parameters:-

Unit	Bill/Unit
1-150	2.40
For next 151-300	3.00
For next more than 300	3.20

Now insert id, unit and bill in elect_bill table.

7. Write PL/SQL code to make a temperature convertor based on user choice. E.g. If user input 1 then convert temperature from centigrade to Fahrenheit and if user input 2 then convert temperature from Fahrenheit to centigrade.
8. Write PL/SQL code to find sum of digits of given number.
9. Write PL/SQL code to print table of given number.
10. Write PL/SQL code to convert binary number to its decimal equivalent.

SOL :-

1.)
connect testdb/test;

2.)

```
create table employee
(
empid number(5) primary key,
empname varchar2(30),
department varchar2(20),
salary number(8)
);
```

```
SQL> create table employee
  2  (
  3  empid number(5) primary key,
  4  empname varchar2(30),
  5  department varchar2(20),
  6  salary number(8)
  7  );

Table created.

SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 EMPID                                     NOT NULL NUMBER(5)
 EMPNAME                                            VARCHAR2(30)
 DEPARTMENT                                         VARCHAR2(20)
 SALARY                                             NUMBER(8)
```

3.)

insert into employee values(1001,'Ravi Singh','Management',80000);
insert into employee values(1002,'Brijesh Mishra','Development',45000);
insert into employee values(1003,'Rajat Verma','Management',50000);
insert into employee values(1004,'Krishna','Development',35000);
insert into employee values(1005,'Nisha Singh','HR',38000);

4.)

i.)
select * from employee where department='Development';
ii.)
delete from employee where empid=1005;
iii.)
update employee set department='Management', salary=60000 where empid=1002;
iv.)
select * from employee where salary=(select max(salary) from employee);
v.)
select * from employee where salary=(select max(salary) from employee where salary<(select max(salary) from employee));
vi.)

truncate table employee;
vii.)
drop table employee;

5.)

```
create table elect_bill
(
id number(5) primary key,
unit number(5),
bill number(10,5)
);
```

6.)

```
declare
id number(5);
unit number(5);
bill number(10,5);
begin
id:=&id;
unit:=&unit;
if unit<=150 then
bill:=unit*2.40;
elsif unit>150 and unit<=300 then
bill:=(150*2.40)+(unit-150)*3.00;
else
bill:=(150*2.40)+(150*3.00)+(unit-300)*3.20;
end if;
dbms_output.put_line('Your Bill='||bill);
insert into elect_bill values(id, unit, bill);
commit;
end;
/
```

7.)

```
declare
ch int;
c number(10,5);
f number(10,5);
```

```
begin
dbms_output.put_line('Enter 1 for c to f');
dbms_output.put_line('Enter 2 for f to c');
ch:=&ch;
if ch=1 then
c:=&c;
f:=(9*c)/5+32;
dbms_output.put_line('Temperature in f='||f);
elsif ch=2 then
f:=&f;
c:=(f-32)*5/9;
dbms_output.put_line('Temperature in c='||c);
else
dbms_output.put_line('Invalid choice');
end if;
end;
```

Database Task – 05

1. Write a PL/SQL code to find sum of digits of given number.
2. 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 area in a table Areas.

   Table Name : Areas

radius	area

3. Write a PL/SQL code to find factorial of given number.
4. Write a PL/SQL code to print series of even number from 1-50.

Max time : 1 Hour
Test Time : SQL
Type : written
Max Marks : 50 Marks

Rules as given Below :-

1. All Questions are mendatory
2. There is a -1 Negative marking for wrong Question.
3. For Correct Option You will be Awarded +2 Marks.

Q1 :- Write a Query for doing following Queries
1. Make a student table with, following field set
stdId,stdname,class,marks,gender as enumeration field

Schema is given as following:-
1. StdId : integer size 11
2. student Name : varchar 191
3. class : varchar 191
4. marks : varchar 191
5. gender : enumeration of male|female|others

StdId : NOT NUll, AutoIncrement, Primary Key and Name should be Unique
Also Add index Key on the marks column.
2. Make sure you are doing This task of student on `Softpro_db` under this database only.
3. Write the Alter Query to Add New Column with Mobile Number.
4. Write the Query to Insert two Record with Student Details, Make sure that Student Have same name but different Other Details.

5. Write the Query for Those student whose mobile is empty
6. Write the Query to Delete the marks of the student whose Id is 1002.
7. Now Write the Bulk insert Quert for Inserting 10 Student At Once
where 3 Are girls and others are Boys.
2nd : Marks can be Anything not greater than 100

8. if All the girls donot Appear for the Exam then There marks Are not available hence,
write a Query, to change there marks to 0 Again, because there marks where added by mistakes.
9. Now Insert A Record whose name is Kuldeep and class is 12th and Marks NUll.
10. Write a Query to Delete All the Student Whose Roll No is 1001
11. Write a Query to Display total Number of student whos marks are Null
12. Write the Query on the Overall sex Ratio of Boys and Girl on the Class.

13. write the Query total marks Acquired by Boys and Total Marks Acquired by Girls

14. Alter the Table and Add, New Field with Percentage Column on student Tables.
15. Write the Select Query to find the percentage of the All the student with columns Alias as Total Percentage of Student
16. Write the Query to find the Duplicate Student Names.
17. write the Query to find the Insert the percentage of all the student on percentage columns
18. On Result Day, On Student Told teacher that there was problem, on Question Paper and every one should get 5 marks bonus hence you need to increment the Bonus of 5 marks on All student who has got marks less than 95
19. perform the Average of All Student to girl and Boys on the behalf of marks using dual table.
20. Make a Another Table named Hostel with following Schema

Hostel(id,stdId,types(A|B|C))
****
id not null, primary key Auto Increment,
stdId Id
types varchar(255)

21. Insert 5 Record with Student Id and Type A and B and C
1001 A
1002 A
1003 B
1004 C
1005 A

write the performing Join for All the student who are girl and have taken the hostel facility with Type A
write the performing Join for the All the student who are girl but have not taken hostel facility
write the Query for All the Student grouped by Boy and Girls with Hostel Facility.

22. Write the Query for Truncating the table, of Hostel, without Truncate Table
23. Write a Query for Making a virtual Table such that, Type Hostel is Hidden, but one can see How many student
have Availed the hostel facility
24 write the Query to drop the type column of the hostel
25 a)write the Qyery to Drop student Id column from hostel table
b)write the Query to drop hostel Id column from hostel Table

SOL :-

1.)

```
Create table student
  (
  stdid     int(11) not null primary key auto_increment,
  stdname varchar(191) not null,
  class varchar(191) not null ,
  marks varchar(191) not null ,
  gender enum( 'male' , 'female' , 'others') not null
  unique(stdname),
  index(marks)
  )
```

Add Index key on Marks

OR

Note emun => check constraint

2.) create database softpro_db;

use softpro_db;

3.) Alter table student Add [column]

mobileno bigint(20)/varchar(50);

4.) insert [into] student values ('1001', 'vijay' , '12th' , '85' , 'male' , '9876543210')

insert [into] student values ('1002', 'vijay dina nath chahuhan' , '11th' , '90' , 'male' , '9876543211');

5.) select*from student where mobileno = ' ' ;

OR

select*from student where mobileno is Null ;

Null ----> 4bit

6.) update student set marks= ' ' where stdid='1002' ;

7.) insert into student values (Null , 'ramani' , '10th' , '50' , 'female' , '9876543210') ,

(Null , 'yashi' , '11th' , '05' , 'female' , '9876543212') ,

(Null , 'rohit' , '12th' , '20' , 'male' , '9876543214') ,

(Null , 'brijesh' , '12th' , '50' , 'male' , '9876543215') ,

(Null , 'awnish' , '12th' , '0' , 'male' , '9876543216') ;

OR

insert into student values

(' ' , 'ramani' , '10th' , '50' , 'female' , '9876543210') ,

(' ' , 'yashi' , '11th' , '05' , 'female' , '9876543212') ,

(' ' , 'rohit' , '12th' , '20' , 'male' , '9876543214') ,

(' ' , 'brijesh' , '12th' , '50' , 'male' , '9876543215') ,

(' ' , 'awnish' , '12th' , '0' , 'male' , '9876543216') ;

OR

insert into student(student, class, marks, gender, mobileno) values

(' ' , 'ramani' , '10th' , '50' , 'female' , '9876543210') ,

(' ' , 'yashi' , '11th' , '05' , 'female' , '9876543212') ,

(' ' , 'rohit' , '12th' , '20' , 'male' , '9876543214') ,

(' ' , 'brijesh' , '12th' , '50' , 'male' , '9876543215') ,

(' ' , 'awnish' , '12th' , '0' , 'male' , '9876543216') ;

8.) update student set marks='0' where gender='female' ;

9.) insert into student (name, class, gender , mobile) values ('kuldeep' , '12th' , 'male' , '8976543210') ;

10.) delete from student where stdid=1001;

11.) select count(stdid) as "Total Number of Student" from student where marks is Null;

12.) select tmp.no_of_boys/tmp.no_of_girls as "sex ratio of Boys to girls " from tmp

(select count(stdid) as no_of_boys from student where gender = 'male'

UNION

select count(stdid) as no_of_girls from student where gender = 'female') as tmp

OR

select (count(stdid) as no_of_boys from student where gender = 'male') / (count(stdid) as no_of_girls from student where gender = 'female') from student limit 1 ;

OR

select gender , count(gender) from student group by gender ;

13.