```
/*****************************************************************************
Program No      : 01
Program Name : Create a table EMPLOYEE using SQL command to store details of employees
                       such as EMPNO,NAME,DESIGNATION,DEPARTMENT,GENDER AND
                       SALARY. Specify Primary Key and Not NULL constraints on the table.
Date               :13/12/2022
*****************************************************************************/
SQL> create table workers
(empno varchar2(6)primary key,
name varchar2(20)not null,
desig varchar2(15)not null,
dept varchar2(15)not null,
gender varchar2(2)check(gender in('M','F')),
Salary number(10));

Table created.

SQL> insert into workers values('E001','BharathRaj','analyst','IT','M',50000);

1 row created.

SQL> insert into workers values('E002','Sharathraj','analyst','IT','M',45000);

1 row created.

SQL>  insert into workers values('E003','PruthviRaj','manager','accounts','M',45000);

1 row created.

SQL> insert into workers values('E004','chintu','accountant','accounts','F',40000);

1 row created.

SQL> insert into workers values('E005','Ranju','analyst','IT','F',38000);

1 row created.

SQL> insert into workers values('E006','Anju','Supervisor','sales','F',35000);
```

```
1 row created.

SQL> insert into workers values('E007','Sanju','analyst','IT','F',30000);

1 row created.

SQL>desc workers;
 Name                          Null?                  Type
 ------------------------------------- -------- ---------------------------
 EMPNO                        NOT NULL        VARCHAR2(6)
 NAME                          NOT NULL       VARCHAR2(20)
 DESIG                         NOT NULL        VARCHAR2(15)
 DEPT                         NOT NULL        VARCHAR2(15)
 GENDER                                        VARCHAR2(2)
 SALARY                                         NUMBER(10)

SQL> select * from workers;

EMPNO      NAME       DESIG          DEPT           GE        SALARY
------------ ------------ ---------------       ---------        -----------   ----------
E001        BharathRaj     analyst           IT              M          50000
E002        Sharathraj      analyst           IT              M          45000
E003        PruthviRaj      manager        accounts         M          45000
E004        chintu          accountant     accounts         F          40000
E005         Ranju           analyst           IT            F          38000
E006         Anju            Supervisor       sales           F          35000
E007         Sanju           analyst           IT            F          30000

7 rows selected.
```

a) Display EMPNO,NAME and DESIGNATION of all employees whose name ends with RAJ.

```
SQL> select empno,name,desig from workers where name like '%Raj';

EMPNO     NAME            DESIG
------------ --------------      ---------------
E001        BharathRaj        analyst
E003        PruthviRaj        manager
```

b) Display the details of all female employees who is earning salary within the range 20000 to 40000 in SALES or IT departments.

```
SQL> select * from workers where gender='F' and salary between 20000 and 40000 and deptin('IT','sales');

EMPNO   NAME        DESIG        DEPT       GE      SALARY
----------- ------------ ------------ ------------ -------- ----------
E005    Ranju       analyst       IT         F       38000
E006    Anju        Supervisor   sales        F       35000
E007    Sanju       analyst       IT         F       30000
```

c) List the different DEPARTMENTs with the DESIGNATIONs in that department.

```
SQL> select DISTINCT dept,desig from workers;

DEPT          DESIG
-------------- ---------------
sales          Supervisor
accounts      manager
accounts      accountant
IT              analyst
```

d) Display the department name, total, average, maximum, minimum salary of the DEPARTMENT only if the total salary given in that department is more than 30000.

```
SQL> select dept,sum(salary)"sum",max(salary)"maximum",min(salary)"minimum",avg(salary)"average" from workers group by dept having sum(salary)>30000;

DEPT          sum       maximum      minimum      average
--------------  ----------   ----------    ----------    ----------
accounts      85000      45000        40000        42500
IT            163000      50000        30000        40750
sales         35000       35000        35000        35000
```

e) List the departments which have more than 2 employees.

```
SQL> select dept,count(empno)"no of workers"from workers group by dept having count(empno)>2;
DEPT          no of workers
-------------- -------------
IT                4
```

/**

Program No :02

Program Name: Create a table CLIENT to store CLIENT_NO,NAME, ADDRESS, STATE, BAL_DUE. Client no must start with ‘C’. Apply the suitable structure for the columns. Specify Primary Key and NOT NULL constraint on the table.

Date : 13/12/2022

**/

```
SQL> create table client
(clientno varchar2(10)check(clientno like('C%'))primary key,
name varchar2(10)not null,
address1 varchar2(10)not null,
address2 varchar2(10)not null,
state varchar2(10),
bal_due number(10,2));

Table created.

SQL>desc client;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLIENTNO                                  NOT NULL VARCHAR2(10)
 NAME                                      NOT NULL VARCHAR2(10)
 ADDRESS1                                  NOT NULL VARCHAR2(10)
 ADDRESS2                                  NOT NULL VARCHAR2(10)
 STATE                                              VARCHAR2(10)
 BAL_DUE                                            NUMBER(10,2)


SQL> insert into Client values('C1001','Suchi','Nagara','Puttur','Karnataka',20000);

1 row created.

SQL> insert into Client values('C1002','Sinchu','Tumbe','Bantwal','Karnataka',30000);

1 row created.

SQL> insert into Client values('C1003','Sanju','Bayar','Muguli','Kerala',10000);

1 row created.
```

```
SQL> insert into Client values('C1004','Suhas','Uppala','Kasaragod','Kerala',10000);

1 row created.

SQL> insert into Client values('C1005','Ram','Nagara','Puttur','Karnataka',12000);

1 row created.

SQL> insert into Client values('C1006','Ashu','Darbe','Puttur','Karnataka',20000);

1 row created.

SQL> insert into Client values('C1007','Raj','K C Road','Bantwala','Karnataka',10000);

1 row created.

SQL> insert into Client values('C1008','Anju','Dasarakodi','Kalladka','Karnataka',12000);

1 row created.

SQL> insert into Client values('C1009','Sushma','BCRoad','Bantwala','Karnataka',13000);

1 row created.

SQL> insert into Client values('C1010','Shoury','Melkar','BCRoad','Karnataka',14000);

1 row created.

SQL> select * from client;

CLIENTNO   NAME       ADDRESS1   ADDRESS2   STATE       BAL_DUE
---------- ---------- ---------- ---------- ---------- ----------
C1001      Suchi      Nagara     Puttur     Karnataka    20000
C1002      Sinchu     Tumbe      Bantwal    Karnataka    30000
C1003      Sanju      Bayar      Muguli     Kerala       10000
C1004      Suhas      Uppala     Kasaragod  Kerala       10000
C1005      Ram        Nagara     Puttur     Karnataka    12000
C1006      Ashu       Darbe      Puttur     Karnataka    20000
```

```
C1007      Raj        K C Road     Bantwala    Karnataka    10000
C1008      Anju       Dasarakodi   Kalladka    Karnataka    12000
C1009      Sushma     BCRoad       Bantwala    Karnataka    13000
C1010      Shoury     Melkar       BCRoad      Karnataka    14000

10 rows selected.
```

a) From the table CLIENT, create a new table CLIENT1 that contains only CLIENT_NO and NAME,BAL_DUE from specifies STATE. Accept the state during run time.

```
SQL> create table client1 as select clientno,name,bal_due from client where state='&state';
Enter value for state: Karnataka
old   1: create table client1 as select clientno,name,bal_due from client where state='&state'
new   1: create table client1 as select clientno,name,bal_due from client where state='Karnataka'

Table created.

SQL> select * from client1;

CLIENTNO   NAME      BAL_DUE
----------  ----------  ----------
C1001      Suchi      20000
C1002      Sinchu     30000
C1005      Ram        12000
C1006      Ashu       20000
C1007      Raj        10000
C1008      Anju       12000
C1009      Sushma     13000
C1010      Shoury     14000

8 rows selected.
```

b) Create a new table CLIENT2 that has the same structure as CLIENT but with no records. Display the structure and records.

```
SQL> create table client2
as select clientno,name,address1,address2,state,bal_due
from client
where 1=2;

Table created.
```

```
SQL>desc client;
Name                           Null?          Type
-------------------------------- -------- ----------------------------
CLIENTNO                  NOT NULL  VARCHAR2(10)
NAME                      NOT NULL  VARCHAR2(10)
ADDRESS1                  NOT NULL  VARCHAR2(10)
ADDRESS2                  NOT NULL  VARCHAR2(10)
STATE                               VARCHAR2(10)
BAL_DUE                             NUMBER(10,2)
```

c) Add a new column by name PENALTY number(10,2) to the CLIENT.

```
SQL> alter table client
add(Penalty number(10,2));

Table altered.

SQL>desc client;
Name                           Null?          Type
-------------------------------- -------- ----------------------------
CLIENTNO                  NOT NULL  VARCHAR2(10)
NAME                      NOT NULL  VARCHAR2(10)
ADDRESS1                  NOT NULL  VARCHAR2(10)
AD                        NOT NULL  VARCHAR2(10)
STATE                               VARCHAR2(10)
BAL_DUE                             NUMBER(10,2)
PENALTY                             NUMBER(10,2)
```

d)Assign Penalty as 10% of BAL_DUE for the clients C1002,C1005,C1009 and for others 8%. Display Records.

```
SQL> update client
set penalty=bal_due*0.1
where clientno in('C1002','C1005','C1009');

3 rows updated.
```

```
SQL> select * from client;

CLIENTNO   NAME      ADDRESS1   ADDRESS2   STATE       BAL_DUE   PENALTY
---------- ---------- ---------- ---------- ---------- ---------- ----------
C1001      Suchi      Nagara     Puttur     Karnataka    20000
C1002      Sinchu     Tumbe      Bantwal    Karnataka    30000      3000
C1003      Sanju      Bayar      Muguli     Kerala       10000
C1004      Suhas      Uppala     Kasaragod  Kerala       10000
C1005      Ram        Nagara     Puttur     Karnataka    12000      1200
C1006      Ashu       Darbe      Puttur     Karnataka    20000
C1007      Raj        K C Road   Bantwala   Karnataka    10000
C1008      Anju       Dasarakodi Kalladka   Karnataka    12000
C1009      Sushma     BCRoad     Bantwala   Karnataka    13000      1300
C1010      Shoury     Melkar     BCRoad     Karnataka    14000

10 rows selected.

SQL> update client
set penalty=bal_due*0.08
where clientno not in('C1002','C1005','C1009');

7 rows updated.

SQL> select * from client;

CLIENTNO   NAME      ADDRESS1   ADDRESS2   STATE       BAL_DUE   PENALTY
---------- ---------- ---------- ---------- ---------- ---------- ----------
C1001      Suchi      Nagara     Puttur     Karnataka    20000      1600
C1002      Sinchu     Tumbe      Bantwal    Karnataka    30000      3000
C1003      Sanju      Bayar      Muguli     Kerala       10000      800
C1004      Suhas      Uppala     Kasaragod  Kerala       10000      800
C1005      Ram        Nagara     Puttur     Karnataka    12000      1200
C1006      Ashu       Darbe      Puttur     Karnataka    20000      1600
C1007      Raj        K C Road   Bantwala   Karnataka    10000      800
C1008      Anju       Dasarakodi Kalladka   Karnataka    12000      960
C1009      Sushma     BCRoad     Bantwala   Karnataka    13000      1300
C1010      Shoury     Melkar     BCRoad     Karnataka    14000      1120

10 rows selected.
```

e) Change the name of CLIENT1 as NEW_CLIENT.

```
SQL> rename client1 to new_client;

Table renamed.

SQL> select * from new_client;

CLIENTNO   NAME     BAL_DUE
----------  ----------  ----------
C1001       Suchi       20000
C1002       Sinchu      30000
C1005       Ram         12000
C1006       Ashu        20000
C1007       Raj         10000
C1008       Anju        12000
C1009       Sushma      13000
C1010       Shoury      14000

8 rows selected.
```

d) Delete the table CLIENT2.

```
SQL> drop table client2;

Table dropped.
```

/***
Program No   :03
Program Name: Create a table BOOK using SQL command to store Accession No, TITLE, AUTHOR, PUBLISHER, YEAR,PRICE. Apply the suitable structure for the columns. Specify Primary Key and NOT NULL Constraints on the table.
Date   :20/12/2022
***/

```
SQL> create table book
(acc_novarchar2(10)primary key,
title varchar2(15)not null,
author varchar2(15)not null,
publisher varchar2(15)not null,
year number(4),
price number(8,2));
Table created
SQL>desc book
Name                              Null?             Type
--------------------------------- --------       ----------------------------
ACC_NO                       NOT NULL     VARCHAR2(10)
TITLE                        NOT NULL     VARCHAR2(15)
AUTHOR                       NOT NULL     VARCHAR2(15)
PUBLISHER                    NOT NULL     VARCHAR2(15)
YEAR                                      NUMBER(4)
PRICE                                     NUMBER(8,2)
SQL> insert into book values('a1001','Networking','Ivan','Microsoft',1989,500);

1 row created.

SQL> insert into book values('a1002','RDBMS','Ivan','Navaneet',2008,350);

1 row created.

SQL> insert into book values('a1003','FIT','Suraj','Microsoft',2007,560);

1 row created.

SQL> insert into book values('a1004','Statistic','Suraj','Himani',2010,580);

1 row created.
```

```
SQL> insert into book values('a1005','C','Balaguru','Navaneet',2007,600);

1 row created.

SQL> insert into book values('a1006','Java','Balaguru','Microsoft',2009,650);

1 row created.

SQL> insert into book values('a1007','DS','Ivan','Himani',2003,500);

1 row created.

SQL> insert into book values('a1008','DBMS','Suraj','Himani',2004,700);

1 row created.

SQL> insert into book values('a1009','Operator','Ivan','Microsoft',2005,350);

1 row created.

SQL> insert into book values('a1010','OOPs','Ivan','Microsoft',2002,450);

1 row created.

SQL> select * from book;
ACC_NO    TITLE          AUTHOR          PUBLISHER      YEAR      PRICE
----------  ---------------  ---------------  ---------------  ----------  ----------
a1001     Networking     Ivan            Microsoft      1989      500
a1002     RDBMS          Ivan            Navaneet       2008      350
a1003     FIT            Suraj           Microsoft      2007      560
a1004     Statistic      Suraj           Himani         2010      580
a1005     C              Balaguru        Navaneet       2007      600
a1006     Java           Balaguru        Microsoft      2009      650
a1007     DS             Ivan            Himani         2003      500
a1008     DBMS           Suraj           Himani         2004      700
a1009     Operator       Ivan            Microsoft      2005      350
a1010     OOPs           Ivan            Microsof       2002      450
10 rows selected.
```

a) List the details of publisher having ‘a’ as the second character in their names.

```
SQL> select * from book
where publisher like'_a%';
ACC_NO     TITLE          AUTHOR         PUBLISHER    YEAR     PRICE
-----------   --------------      ---------------      ---------------     ----------   ----------
a1002          RDBMS          Ivan              Navaneet       2008      350
a1005            C            Balaguru          Navaneet       2007      600
```

b) Display Accession No, TITLE, AUTHOR and YEAR of the books published by the specified author before 2010 in the descending order of YEAR. Accept author during run time.

```
SQL> select acc_no,title,publisher,year from book
where author='&author'
and year<2010
order by year desc;
Enter value for author: Ivan
old   2: where author='&author'
new   2: where author='Ivan'
ACC_NO     TITLE          PUBLISHER      YEAR
----------      ---------------     --------------      ----------
a1002          RDBMS         Navaneet        2008
a1009         Operator        Microsoft       2005
a1007          DS             Himani          2003
a1010         OOPs           Microsoft        2002
a1001         Networking     Microsoft       1989
```

c) Modify the size of TITLE to increase the size 5 characters more.

```
SQL> alter table book
modify(title varchar2(20));
Table altered.
SQL>desc book;
Name                              Null?           Type
-------------------------------- -------- ---------------------------
ACC_NO                      NOT NULL  VARCHAR2(10)
TITLE                       NOT NULL  VARCHAR2(20)
AUTHOR                      NOT NULL  VARCHAR2(15)
PUBLISHER                   NOT NULL  VARCHAR2(15)
YEAR                                  NUMBER(4)
PRICE                                 NUMBER(8,2)
```

d) Display the details of the books other than Microsoft press publisher.

```
SQL> select * from book
where publisher not in('Microsoft');

ACC_NO   TITLE    AUTHOR   PUBLISHER   YEAR   PRICE
------------   ----------- --------------- ---------------    ---------  -----------
  a1002      RDBMS      Ivan        Navaneet       2008     350
  a1004       Statistic     Suraj        Himani         2010     580
  a1005          C         Balaguru    Navaneet       2007     600

ACC_NO   TITLE    AUTHOR   PUBLISHER   YEAR    PRICE
------------- ------------ -------------- ---------------   ----------  ----------
  a1007         DS            Ivan          Himani         2003     500
  a1008       DBMS      Suraj           Himani          2004    700
```

e) Remove the records of the books published before 1990.

```
SQL> delete from book
where year<1990;
1 row deleted.

SQL> select * from book;
ACC_NO    TITLE   AUTHOR   PUBLISHER   YEAR   PRICE
------------- ----------- -------------- ------------------ ---------- ----------
  a1002      RDBMS     Ivan          Navaneet      2008      350
  a1003         FIT        Suraj        Microsoft      2007      560
  a1004       Statistic     Suraj           Himani        2010      580

ACC_NO   TITLE    AUTHOR    PUBLISHER  YEAR    PRICE
------------- ------------ -------------- ----------------- ----------  ---------
  a1005          C         Balaguru      Navaneet      2007       600
  a1006       Java      Balaguru     Microsoft      2009       650
  a1007        DS          Ivan            Himani       2003       500

ACC_NO   TITLE   AUTHOR  PUBLISHER   YEAR   PRICE
------------- ----------- -------------- ------------------ ---------- ----------
  a1008       DBMS      Suraj          Himani          2004      700
  a1009     Operator    Ivan          Microsoft       2005      350
  a1010       OOPs       Ivan          Microsoft        2002      450
9 rows selected.
```

```
/****************************************************************************
Program No     :04
Program Name: Create a table SALES with columns  SNO,SNAME,MANAGER_NAME,
                    JOIN_DATE,DATE_BIRTH,SALARY,SALES_AMOUNT and
                    COMMISSION. Minimum age for joining the company must be 18 Yrs.
 Date               : 20/12/2022
****************************************************************************/
SQL> create table sales
(sno varchar2(6)primary key,
sname varchar2(10)not null,
mno varchar2(6),
doj date,
dob date,
salary number(10,2),
s_amount number(10),
commission number(10)default(0),
check((doj-dob)/365>=18));

Table created.

SQL>desc sales;
 Name                                      Null?          Type
 -------------------------------- -------- ---------------------------
 SNO                               NOT NULL VARCHAR2(6)
 SNAME                             NOT NULL VARCHAR2(10)
 MNO                                        VARCHAR2(6)
 DOJ                                        DATE
 DOB                                        DATE
 SALARY                                     NUMBER(10,2)
 S_AMOUNT                                   NUMBER(10)
 COMMISSION                                 NUMBER(10)
SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(111,'suchi',222,'30jan-1964','30
-jan-1944',10000,20000);

1 row created.

SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(222,'sinchu',' ','20jan-1984','3
0-jan-1964',15000,20000);
```

```
1 row created.

SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(333,'sanju','222','20jan-1984','
30-jan-1964',20000,15000);

1 row created.

SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(444,'Sushma','333','30jan-1984',
'30-jan-1964',20000,20000);

1 row created.

SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(555,'suchi','333','30jan-1984','
30-jan-1964',10000,20000);

1 row created.

SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(666,'Sonu','333','30jan-2020','3
0-jan-1964',15000,20000);

1 row created.

SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(777,'Sonu','444','30jan-2020','3
0-jan-1964',13000,25000);

1 row created.

SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(888,'Ramu','111','12jan-2022','3
0-jan-1989',15000,12000);

1 row created.

SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(999,'Savi','555','12jan-2022','3
0-jan-1989',16000,12000);

1 row created.

SQL> insert into sales(sno,sname,mno,DOJ,DOB,salary,s_amount)values(110,'Pavi','555','12jan-2022','3
0-jan-1989',16000,12000);
```

```
1 row created.

SQL> select * from sales;

SNO    SNAME   MNO      DOJ          DOB       SALARY   S_AMOUNT    COMMISSION
------ ---------- ------   ---------     ---------    ----------    ----------       ----------
111    suchi     222    30-JAN-64  30-JAN-44    10000      20000           0
222    sinchu           20-JAN-84  30-JAN-64    15000      20000           0
333    sanju     222    20-JAN-84  30-JAN-64    20000      15000           0
444    Sushma   333     30-JAN-84  30-JAN-64    20000      20000           0
555    suchi     333    30-JAN-84  30-JAN-64    10000      20000           0
666     Sonu      333   30-JAN-20  30-JAN-64    15000      20000           0
777     Sonu     444    30-JAN-20  30-JAN-64    13000      25000           0
888     Ramu     111   12-JAN-22   30-JAN-89    15000      12000           0
999     Savi     555   12-JAN-22   30-JAN-89    16000       12000          0
110     Pavi     555   12-JAN-22   30-JAN-89    16000       12000          0

10 rows selected.
```

a) Display the details of Sales Persons whose salary is more than Average salary in the company.

```
SQL> select * from sales
where salary>(select avg(salary)from sales);
SNO    SNAME   MNO    DOJ          DOB        SALARY   S_AMOUNT    COMMISSION
------ ---------- ------   ---------     ---------     ----------     ----------       ----------
333    sanju     222   20-JAN-84  30-JAN-64     20000       15000          0
444    Sushma   333   30-JAN-84  30-JAN-64     20000       20000          0
999    Savi      555   12-JAN-22  30-JAN-89     16000        12000         0
110     Pavi     555   12-JAN-22  30-JAN-89     16000        12000         0
```

b) Update commission as 20% of Sales Amount.

```
SQL> update sales
set commission=s_amount*0.2;

10 rows updated.

SQL> select * from sales;
```

```
SNO  SNAME MNO    DOJ          DOB       SALARY   S_AMOUNT  COMMISSION
------ ---------- ------  ---------       ---------      ---------     ----------         ----------
111    suchi     222    30-JAN-64  30-JAN-44    10000     20000           4000
222    sinchu           20-JAN-84  30-JAN-64    15000     20000           4000
333    sanju     222    20-JAN-84  30-JAN-64    20000     15000           3000
444    Sushma   333    30-JAN-84  30-JAN-64    20000     20000          4000
555    suchi     333    30-JAN-84  30-JAN-64    10000     20000          4000
666    Sonu      333    30-JAN-20  30-JAN-64    15000     20000          4000
777    Sonu      444    30-JAN-20  30-JAN-64    13000     25000          5000
888    Ramu      111    12-JAN-22  30-JAN-89    15000     12000          2400
999    Savi      555    12-JAN-22  30-JAN-89    16000     12000          2400
110    Pavi      555    12-JAN-22  30-JAN-89    16000     12000          2400

10 rows selected.
```

c) Display SNO,SNAME,MANAGER_NAME, JOIN_DATE,DATE_BIRTH,SALARY of the sales persons getting sum of salary and commission more than salary of manager.(Self join)

```
SQL>  selects.sno as sno,s.sname as sname,s.mno as mno,s.salary as salary,s.commission as
commission, m.salary as manager_salary
from sales m,sales s
where s.mno=m.sno
and (s.salary+s.commission)>m.salary;
```

```
SNO  SNAME  MNO   SALARY   COMMISSION MANAGER_SALARY
------ ----------   ------     ----------        ----------           --------------
333    sanju      222      20000           3000            15000
444    Sushma  333      20000           4000            20000
888    Ramu      111      15000           2400            10000
999    Savi      555      16000           2400            10000
110    Pavi      555      16000           2400            10000
```

d) Display the records of employees who finished the service of 10 years.

```
SQL> select * from sales
where (sysdate-doj)>10;
```

```
SNO  SNAME  MNO    DOJ        DOB        SALARY   S_AMOUNT   COMMISSION
------ ----------   ------     ---------    ---------         ----------  ----------             ----------
111    suchi      222   30-JAN-64  30-JAN-44     10000     20000              4000
222    sinchu           20-JAN-84  30-JAN-64     15000     20000              4000
```

```
333   sanju    222   20-JAN-84   30-JAN-64     20000     15000              3000
444   Sushma   333   30-JAN-84   30-JAN-64     20000     20000              4000
555   suchi    333   30-JAN-84   30-JAN-64     10000     20000              4000
666   Sonu     333   30-JAN-20   30-JAN-64     15000     20000              4000
777   Sonu     444   30-JAN-20   30-JAN-64     13000     25000              5000
888   Ramu     111   12-JAN-22   30-JAN-89     15000     12000              2400
999   Savi     555   12-JAN-22   30-JAN-89     16000     12000              2400
110   Pavi     555   12-JAN-22   30-JAN-89     16000     12000              2400

10 rows selected.
```

/**
Program No :05
Program Name: Create a table Sales_Details with the columns SNO, MONTH, TARGET and QTY_SOLD to store the Sales Details of one year. Specify the composite primary key to the columns SNO and MONTH. TARGET and SALES must be positive numbers.
Date :27/12/2022
**/

```
SQL> create table sales_detail
(sno varchar(10),
month varchar(10),
target number(10) check(target>0),
qtysold number(10),
sales number(10,2) check(sales>0),
primary key(sno,month));

Table created.

SQL>descsales_detail;
 Name                              Null?      Type
 -------------------------------- -------- ----------------------------
 SNO                              NOT NULL  VARCHAR2(10)
 MONTH                            NOT NULL  VARCHAR2(10)
 TARGET                                     NUMBER(10)
 QTYSOLD                                    NUMBER(10)
 SALES                                      NUMBER(10,2)

SQL> insert into sales_detail values('s0001','dec',40,23,10);

1 row created.

SQL> insert into sales_detail values('s0002','jan',70,4,11);

1 row created.

SQL> insert into sales_detail values('s0003','feb',10,2,22);

1 row created.
```

```
SQL> insert into sales_detail values('s0004','mar',66,77,15);

1 row created.

SQL> insert into sales_detail values('s0005','jun',6,100,33);

1 row created.

SQL> insert into sales_detail values('s0006','jun',40,10,33);

1 row created.

SQL> select * from sales_detail;

SNO   MONTH   TARGET    QTYSOLD    SALES
-------- ----------     ---------       ----------      ----------
s0001     dec       40              23           10
s0002     jan       70               4           11
s0003     feb       10               2           22
s0004     mar       66              77           15
s0005     jun        6             100           33
s0006     jun        40             10           33

6 rows selected.
```

a) Display the total sales by each sales person considering only those months sales where target was reached.

```
SQL> Select sno,sum(qtysold)
From sales_detail
Where qtysold>target
Group by sno;

  SNO          SUM(QTYSOLD)
----------      ------------
s0004             77
s0005            100
```

b) If a commission of RS.50 provided for each item after reaching target, calculate and display the total commission for each sales person.

```
SQL> select sno,sum(qtysold-target)*50 "commission earned"
 From sales_detail
 Where qtysold>target
 Group by sno;

SNO        commission earned
----------     -----------------
s0004               550
s0005              4700
```

c) Display the SNO of those who never reached the target.

```
SQL> Select sno from sales_detail
Where target>qtysold;

SNO
----------
s0001
s0002
s0003
s0006
```

d) Display the SNO, MONTH and QTY_SOLD of the sales persons with SNO S0001 or S0003

```
SQL> select sno,month,qtysold
 From sales_detail
 Where sno='s0001' or sno='s0003';

SNO        MONTH    QTYSOLD
---------- ----------      ----------
s0001        dec             23
s0003         feb              2
```

```
/*****************************************************************************
Program No    :06
Program Name: Create a table Bank with the columns ACNO, ACT_NAME, ACT_TYPE and
              BAL. Specify the Primary Key. Initial BAL must be greater than 500. Write
              a PL/SQL program to perform debit operation by providing acct_no and amount
              required.
Date          :27/12/2022
*****************************************************************************/
```

```
SQL> create table bank1
(acno number(7) primary key,
act_name varchar2(10),
act_type varchar2(10),
bal number(10));

Table created.

SQL>desc bank1;
 Name                              Null?      Type
 --------------------------------- -------- ----------------------------
 ACNO                          NOT NULL   NUMBER(7)
 ACT_NAME                                  VARCHAR2(10)
 ACT_TYPE                                  VARCHAR2(10)
 BAL                                       NUMBER(10)

SQL> insert into bank1 values(101,'suchi','RD',3000);

1 row created.

SQL> insert into bank1 values(102,'Sinchu','SB',600);

1 row created.

SQL>  insert into bank1 values (103,'Uday','FD',3000);

1 row created.
SQL>  insert into bank1 values (104,'Sanju','RD',4000);
SQL> select * from bank1;

ACNO   ACT_NAME   ACT_TYPE    BAL
---------       ----------         ----------      ----------
101           suchi              RD            3000
102          Sinchu             SB             600
103           Uday               FD            3000
104          Sanju              RD            4000

SQL>ed bank;
Declare
l_account_no number:= &account_no;
l_amount number:= &amount;
Lbalance number(10,2);
Lcount number(10):= 0;
min_bal_constraint exception;
begin
select count(*) into Lcount from bank1 where acno=l_account_no;
ifLcount<> 0 then
if(l_amount> 100 and l_amount< 20000)then
select bal into Lbalance from bank1 where acno = l_account_no;
Lbalance:=Lbalance-l_amount;
if (Lbalance> 500) then
update bank1
set bal=Lbalance
where acno=l_account_no;
dbms_output.put_line('balance updated');
else
dbms_output.put_line('no sufficient amount1');
raisemin_bal_constraint;
end if;
else
dbms_output.put_line('the amount must be greater than 100 and lesser than 20000');
end if;
else
dbms_output.put_line('no such account');
end if;
exception
```

```
whenmin_bal_constraint then
dbms_output.put_line('minimum balance should be 500');
end;
```

```
SQL> set serveroutput on;
SQL> @bank;
32  /
Enter value for account_no: 1
old   2: l_account_no number:= &account_no;
new   2: l_account_no number:= 1;
Enter value for amount: 100
old   3: l_amount number:= &amount;
new   3: l_amount number:= 100;
no such account

PL/SQL procedure successfully completed.

SQL> @bank;
32  /
Enter value for account_no: 101
old   2: l_account_no number:= &account_no;
new   2: l_account_no number:= 101;
Enter value for amount: 400
old   3: l_amount number:= &amount;
new   3: l_amount number:= 400;
balance updated

PL/SQL procedure successfully completed.

SQL> /
Enter value for account_no: 101
old   2: l_account_no number:= &account_no;
new   2: l_account_no number:= 101;
Enter value for amount: 4000
old   3: l_amount number:= &amount;
new   3: l_amount number:= 4000;
no sufficient amount1
minimum balance should be 500
```

```
PL/SQL procedure successfully completed.
/******************************************************************************
Program No    :07
Program Name: Create a table STOCK_DETAIL with the columns PNO,PNAME and
              QTY_AVL to store stock details of computer accessories. Specify Primary Key
              and NOT NULL constraint on the table.
Date          :27/12/2022
******************************************************************************/
SQL> create table stock_detail
(pno number(5) primary key,
pname varchar2(10) not null,
qty_avl number(5) check(qty_avl>0) not null);

 Table created.

SQL>descstock_detail;
 Name                              Null?       Type
 --------------------------------- --------  ----------------------------
 PNO                         NOT NULL    NUMBER(5)
 PNAME                       NOT NULL    VARCHAR2(10)
 QTY_AVL                    NOT NULL     NUMBER(5)

SQL> insert into stock_detail values(101,'Monitor',300);

1 row created.

SQL> insert into stock_detail values(102,'CPU',230);

1 row created.

SQL> insert into stock_detail values(103,'Keyboard',150);

1 row created.

SQL> insert into stock_detail values(104,'Printer',100);

1 row created.

SQL> insert into stock_detail values(105,'Mouse',140);
```

```
1 row created.

SQL> select * from stock_detail;

 PNO    PNAME    QTY_AVL
--------   ----------     ----------
 101       Monitor        300
 102       CPU            230
 103       Keyboard       150
 104       Printer        100
 105       Mouse          140

SQL>ed stock;
prompt enter the product number;
acceptproduct_number;
prompt enter the quantity required;
acceptquantity_required;
declare
l_productnum number:=&product_number;
l_quantityreq number:=&quantity_required;
l_quantityavailable number;
 LOW_STOCK exception;
begin
dbms_output.put_line('the product number you have entered is:'||l_productnum);
dbms_output.put_line('the quantity you have entered is:'||l_quantityreq);
selectqty_avl into l_quantityavailable from stock_detail where pno=l_productnum;
if(l_quantityavailable<|_quantityreq) then
raise LOW_STOCK;
else
dbms_output.put_line('the quantity value before purchase:'||l_quantityavailable);
l_quantityavailable:=l_quantityavailable-l_quantityreq;
updatestock_detail set qty_avl=l_quantityavailable where pno=l_productnum; dbms_output.put_line('the
quantity value after purchase:'||l_quantityavailable);
end if;
exception when LOW_STOCK then
dbms_output.put_line('NO SUFFICIENT STOCK');
end;
SQL> set serveroutput on;
SQL> @stock;
enter the product number
101
enter the quantity required
20
21  /
old   2: l_productnum number:=&product_number;
new   2: l_productnum number:=101;
old   3: l_quantityreq number:=&quantity_required;
new   3: l_quantityreq number:=20;
the product number you have entered is:101
the quantity you have entered is:20
the quantity value before purchase:280
the quantity value after purchase:260

PL/SQL procedure successfully completed.
```

```
/*************************************************************************
Program No    :08
Program Name: Create the following tables by identifying primary and foreign keys. Specify the
                  not null property for mandatory keys. SUPPLIERS (SUPPLIER_NO,SNAME,
                  SADDRESS,SCITY)     COMPUTER _ITEMS(ITEM_NO, SUPPLIER_NO,
                  ITEM_NAME,IQUANTITY) Consider three suppliers.
 Date              :03/01/2023
*************************************************************************/
```

```
SQL>create table suppliers
(sup_no varchar2(6) primary key,
sname varchar2(15)not null,
saddress varchar2(15)not null,
scity varchar2(15)not null);
Table created.
SQL>desc suppliers;
 Table created.
 Name                              Null?           Type
 ------------------------------- -------- ---------------------------
 SUP_NO                   NOT NULL   VARCHAR2(6)
 SNAME                    NOT NULL    VARCHAR2(15)
 SADDRESS                 NOT NULL    VARCHAR2(15)
 SCITY                    NOT NULL   VARCHAR2(15)

SQL>create table c_items
(item_no varchar2(6),
sup_no varchar2(6) references suppliers(sup_no),
item_name varchar2(10) not null,
iqty number(3));
   Table created.

SQL>descc_items;
 Name                                Null?           Type
 --------------------------------- -------- ---------------------------
 ITEM_NO                                      VARCHAR2(6)
 SUP_NO                                       VARCHAR2(6)
 ITEM_NAME                 NOT NULL   VARCHAR2(10)
 IQTY                                         NUMBER(3)
```

```
SQL> insert into suppliers values('s001','Microtech','Bolwar','Puttur');
1 row created.
SQL> insert into suppliers values('s002','Cats','Bayar','Uppala');
1 row created.
SQL> insert into suppliers values('s003','Electrotech','Darbe','Puttur');
1 row created.
SQL> insert into suppliers values('s004','Dell','Manchi','Konaje');
1 row created.
SQL> insert into suppliers values('s005','Intex','Muguli','Vitla');
1 row created.
SQL> insert into suppliers values('s006','HP','Nagara','Kelinja');
1 row created
SQL> select * from suppliers;

SUP_NO   SNAME      SADDRESS    SCITY
------     --------------- --------------- ---------------
s001       Microtech    Bolwar        puttur
s002       Cats          Bayar        Uppala
s003      Electrotech    Darbe        Puttur
s004      Dell            Manchi       Konaje
s005      Intex          Muguli       Vitla
s006       HP             Nagara       Kelinja

6 rows selected.
SQL> insert into c_items values('1001','s001','Keyboard',10);
1 row created.
SQL> insert into c_items values('1002','s001','mouse',5);
1 row created.
SQL> insert into c_items values('1003','s002','monitor',4);
1 row created.
SQL> insert into c_items values('1004','s003','printer',7);
1 row created.
SQL> insert into c_items values('1005','s004','Keyboard',8);
1 row created.
SQL> insert into c_items values('1006','s005','scanner',6);
1 row created.
SQL> select * from c_items;
```

```
ITEM_N    SUP_NO  ITEM_NAME     IQTY
------      ----------    ----------        ----------
1003          s002         monitor            4
1004          s003         printer            7
1005          s004        Keyboard          8
1006          s005        scanner            6
1001          s001        Keyboard         10
1002          s001        mouse                5

6 rows selected.
```

a) List ITEMS and SUPPLIER details in alphabetical order of city name and in each city decreasing order of IQUANTITY.

```
SQL> select sname,saddress,scity,iqty,item_name,item_no
from suppliers s,c_items c
where c.sup_no=s.sup_no
order by scity,iqtydesc;
```

```
SNAME    SADDRESS   SCITY   IQTY     ITEM_NAME  ITEM_N
------------  --------------- ------------ ----------     ----------          ------
Dell           Manchi       Konaje       8          Keyboard        1005
Electrotech Darbe          Puttur        7          printer             1004
Cats            Bayar         Uppala       4         monitor             1003
Intex          Muguli        Vitla           6          scanner           1006
Microtech   Bolwar        puttur       10          Keyboard         1001
Microtech    Bolwar       puttur         5          mouse                1002

6 rows selected.
```

b) List the name, city, and address of the suppliers who are supplying keyboard.

```
SQL> select sname,saddress,scity
from suppliers s,c_items c
where s.sup_no=c.sup_no
and item_name='Keyboard';
```

```
SNAME    SADDRESS   SCITY
-----------    --------------- ---------------
```

```
Dell          Manchi       Konaje
Microtech   Bolwar       puttur
```

c) List the supplier name, items supplied by the suppliers 'Cats' and 'Electrotech'.

```
SQL> select sname,item_name
from suppliers s,c_items c
where s.sup_no=c.sup_no
and sname in('Cats','Electrotech');

SNAME      ITEM_NAME
------------     ----------
Cats              monitor
Electrotech      printer
```

d) Find the items having quantity less than 5 and insert the details of supplier and item of these, into another table NEWORDER.

```
SQL> create table neworder
as(select s.sup_no,s.sname,s.saddress,c.item_no,c.item_name,c.iqty
from suppliers s,c_items c
where s.sup_no=c.sup_no
and iqty<5);
Table created
SQL> select * from neworder;

SUP_NO   SNAME     SADDRESS    ITEM_NAME     IQTY
------        ------------   ---------------      ----------          ----------
s002         Cats          Bayar               monitor             4
```

```
/****************************************************************************
Program No    :09
Program Name: Create the following tables by identifying primary and foreign keys. Specify the
               not null property for mandatory keys. EMPLOYEE_MASTER (EMP_ID,
               EMP_NAME,EMAIL_ID,EMP_ADDRS,PHONE)   ATTENDANCE (EMP_ID
               MONTH,WOM,MHRS,THRS,WHRS,TRHRS,FHRS,SHRS,SUHRS).
Date            : 03/01/2023
****************************************************************************/
SQL> create table emp_master
(emp_id varchar2(6)primary key,
emp_name varchar2(10)not null,
emp_addr varchar2(15)not null,
email varchar2(15),
phone number(10)not null);

Table created.

SQL>descemp_master;
 Name                            Null?        Type
 ------------------------------- -------- ----------------------------
 EMP_ID                     NOT NULL    VARCHAR2(6)
 EMP_NAME                   NOT NULL    VARCHAR2(10)
 EMP_ADDR                   NOT NULL    VARCHAR2(15)
 EMAIL                                  VARCHAR2(15)
 PHONE                      NOT NULL    NUMBER(10)


SQL> create table attendance
(emp_id varchar2(6)references emp_master(emp_id)ON DELETE CASCADE,
Month number(1),
WOM number(1),
MHrs number(1),
Thrs number(1),
Whrs number(1),
Trhrs number(1),
Fhrs number(1),
Shrs number(1),
Suhrs number(1),
check(WOM<=5),
check(month between 1 and 12));

Table created.

SQL> insert into emp_master values('e001','Rashmi','Puttur','rash@gmail.com',9464567891);

1 row created.

SQL> insert into emp_mastervalues('e002','Bunny','Vittal','bunny@gmail.com',9448780611);

1 row created.

SQL> insert into emp_master values('e003','Mahesh','Bayar','mahe@gmail.com',9867564390);

1 row created.

SQL> insert into emp_master values('e004','Sanju','Alike','sanju@gmail.com',9678976543);

1 row created.

SQL> insert into emp_master values('e005','Sinchu','Bantwal','sincu@gmail.com',9448780611);

1 row created.

SQL> insert into emp_master values('e006','Achu','Manglore','achu@gmail.com',9448780611);

1 row created.

SQL> select * from emp_master;

EMP_ID  EMP_NAME   EMP_ADDR      EMAIL            PHONE
------     ----------     ---------------   ---------------       ----------
e001       Rashmi        Puttur      rash@gmail.com     9464567891
e002       Bunny          Vittal      bunny@gmail.com    9448780611
e003      Mahesh         Bayar       mahe@gmail.com    9867564390
e004      Sanju          Alike       sanju@gmail.com    9678976543
e005      Sinchu        Bantwal      sinchu@gmail.com   9448780611
```

```
e006     Achu          Manglore    achu@gmail.com     9448780611

6 rows selected.

SQL> insert into attendance values('e001',2,2,4,3,2,2,2,5,4);

1 row created.

SQL> insert into attendance values('e002',2,3,2,5,6,7,4,5,6);

1 row created.

SQL> insert into attendance values('e001',3,4,3,2,5,4,3,2,1);

1 row created.

SQL> insert into attendance values('e003',4,3,4,1,3,2,4,3,1);

1 row created.

SQL> insert into attendance values('e004',4,2,5,2,2,4,5,6,7);

1 row created.

SQL> insert into attendance values('e005',4,2,0,0,0,0,0,0,0);

1 row created.

SQL> select * from attendance;

EMP_ID MONTH   WOM   MHRS  THRS  WHRS  TRHRS  FHRS  SHRS  SUHRS
------------ ----------   ---------- ---------- ---------- ---------- --------- ---------  --------- ----------
  e001      2       2       4       3       2       2       2       5       4
  e002      2       3       2       5       6       7       4       5       6
  e001      3       4       3       2       5       4       3       2       1

EMP_ID MONTH   WOM   MHRS  THRS  WHRS  TRHRS  FHRS  SHRS  SUHRS
------------ ----------   ---------- ---------- ---------- ---------- --------- ---------  --------- ----------
  e003      4       3       4       1       3       2       4       3       1
  e004      4       2       5       2       2       4       5       6       7
  e005      4       2       0       0       0       0       0       0       0
6 rows selected.
```

a) Display EMP_ID,EMP_NAME and EMAIL_ID of all employees who are working on every Sunday of 2nd and 4th week in a month.

```
SQL> select e.emp_id,e.email,e.emp_name
from emp_mastere,attendance a
where e.emp_id=a.emp_id
and a.suhrs>0
and(a.wom=2 or a.WOM=4);
```

```
EMP_ID     EMAIL          EMP_NAME
-----------  ---------------   --------------
e001      rash@gmail.com    Rashmi
e001      rash@gmail.com    Rashmi
e004      sanju@gmail.com   Sanju
```

b) Display total hours worked by each employee in each month with EMP_ID.

```
SQL> select month,emp_id,sum(mhrs+thrs+whrs+trhrs+fhrs+shrs+suhrs)"total hrs"
from attendance
group by month,emp_id;
```

```
MONTH    EMP_ID    totalhrs
----------    ------    ----------
  2        e001       22
  4        e003       18
  4        e005        0
  4        e004       31
  3        e001       20
  2        e002       35

6 rows selected.
```

c) Display the names of the employees who never attended the duty so far(Attendances not given so far).

```
SQL> select emp_name from emp_mastere,attendance a
where e.emp_id=a.emp_id
and(mhrs+thrs+whrs+trhrs+fhrs+shrs+suhrs=0);
```

```
EMP_NAME
----------
Sinchu
```

d) Display the employee name, month, week, total hours worked for employees who have total no. of hours more than 20 hrs. a week.

```
SQL> select emp_name,month,wom,sum(mhrs+thrs+whrs+trhrs+fhrs+shrs+suhrs)"total hrs"
from emp_mastere,attendance a
where e.emp_id=a.emp_id
group by emp_name,month,wom
having sum(mhrs+thrs+whrs+trhrs+fhrs+shrs+suhrs)>=20;
```

```
EMP_NAME    MONTH    WOM  totalhrs
----------  ----------  -------- ----------
Bunny          2          3      35
Rashmi         2          2      22
Rashmi         3          4      20
Sanju          4          2      31
```

/***

Program No :10

Program Name: Create the following tables by identifying primary and foreign keys. Specify the not null property for mandatory keys.

Date :10/01/2023

***/

```
SQL> create table product
(pno varchar2(6)primary key,
pname varchar2(10)not null,
qty_avail number(3)not null,
price number(6),
profit number(3));

Table created.

SQL>desc product;
 Name                            Null?    Type
 -------------------------------- -------- ---------------------------
 PNO                             NOT NULL VARCHAR2(6)
 PNAME                           NOT NULL VARCHAR2(10)
 QTY_AVAIL                       NOT NULL NUMBER(3)
 PRICE                                    NUMBER(6)
 PROFIT                                   NUMBER(3)


SQL> create table purchase
(cno varchar2(15),
pno varchar2(6)references product(pno),
qty_sold number(3));

Table created.

SQL>desc purchase;
 Name                            Null?    Type
 -------------------------------- -------- ---------------------------
 CNO                                      VARCHAR2(15)
 PNO                                      VARCHAR2(6)
 QTY_SOLD                                NUMBER(3)

SQL> insert into product values('P001','Monitor',10,3000,20);

1 row created.

SQL> insert into product values('P002','Pendrive',50,650,5);

1 row created.

SQL> insert into product values('P003','CDdrive',100,10,3);

1 row created.

SQL> insert into product values('P004','Keyboard',25,600,10);

1 row created.

SQL> select * from product;

PNO   PNAME     QTY_AVAIL     PRICE    PROFIT
------ ----------    ----------       ----------  ----------
P001   Monitor        10            3000      20
P002   Pendrive       50             650       5
P003   CDdrive       100              10       3
P004   Keyboard       25             600      10

SQL> insert into purchase values('c1','P003',2);

1 row created.

SQL> insert into purchase values('c2','P002',4);

1 row created.

SQL> insert into purchase values('c3','P002',10);

1 row created.
```

```
SQL> insert into purchase values('c4','P001',3);

1 row created.

SQL> insert into purchase values('c1','P004',2);

1 row created.

SQL> insert into purchase values('c2','P003',2);

1 row created.

SQL> insert into purchase values('c4','P004',1);

1 row created.

SQL> select * from purchase;
CNO    PNO   QTY_SOLD
-------  ------     ----------
c1       P003         2
c2       P002         4
c3       P002        10
c4       P001         3
c1       P004         2
c2       P003         2
c4       P004         1
7 rows selected.
```

a) Display total amount spent by C2.

```
SQL> select sum(p.price*r.qty_sold)"total"
from product p,purchase r
where p.pno=r.pno
and r.cno='c2';
   total
----------
      2620
```

b) Display the names of product for which either QtyAvailable is less than 30 or total QtySold is less than 5(USE UNION).

```
SQL> select pname from product
where pno in(select pno from product
where qty_avail<30 UNION
select pno from purchase where qty_sold<15);
PNAME
----------
Monitor
Pendrive
CDdrive
Keyboard
```

c) Display the name of products and quantity purchased by C4.

```
SQL> select p.pname,p.qty_avail
from product p,purchase r
where p.pno=r.pno
and r.cno='c4';
PNAME      QTY_AVAIL
----------   ----------
Monitor        10
Keyboard       25
```

d)How much Profit does the shopkeeper gets on C1's purchase?

```
SQL> select sum(p.profit*r.qty_sold)"profit"
from product p,purchase r
where p.pno=r.pno
and r.cno='c1';
profit
----------
   26
```

e) How many 'Pen Drives' have been sold?

```
SQL> select sum(r.qty_sold)"pendrive"
from product p,purchase r
where p.pno=r.pno and
p.pname='Pendrive';
pendrive
----------
      14
```

```
/*************************************************************************
Program No    :11
Program Name: Create table STUDENT_PROFILE includes Roll no, name, class, ECCC
              (Extra-Co curricular he belongs to such. as SPORTs, NSS etc) and another table
              MARKS_REPORT includes Roll no, Internal_Test,Marks1,Marks2,Marks3 and
              ECCC_marks.
Date          : 10/01/2023
*************************************************************************/
SQL>  create table student_profile
(rollno number(10) primary key,
name varchar2(10) not null,
class varchar2(10) not null,
Eccc varchar2(10));

Table created.

SQL>desc student_profile;
 Name                                      Null?        Type
 ----------------------------------------- -------- ----------------------------
 ROLLNO                                NOT NULL   NUMBER(10)
 NAME                                  NOT NULL   VARCHAR2(10)
 CLASS                                NOT NULL   VARCHAR2(10)
 ECCC                                             VARCHAR2(10)

SQL> insert into student_profile values('18901','Alan','II BCA','NSS');

1 row created.

SQL> insert into student_profile values('18902','Aron','II BCA','NCC');

1 row created.

SQL> insert into student_profile values('18903','Dhanya','II BCA','NSS');

1 row created.

SQL>  insert into student_profile values('17901','Alex','III BCA','');

1 row created.
```

```
SQL> insert into student_profile values('17902','Bipin','III BCA','');

1 row created.

SQL>  insert into student_profile values('17903','Simran','III BCA','');

1 row created.

SQL> insert into student_profile values('19901','Arun','I BCA','NSS');

1 row created.

SQL> insert into student_profile values('19902','Bindu','I BCA','ECO CLUB');

1 row created.

SQL>  insert into student_profile values('19903','Deepa','I BCA','NCC');

1 row created.

SQL>  select * from student_profile;

ROLLNO NAME     CLASS      ECCC
----------    ---------- ----------    ----------
18901         Alan       II BCA     NSS
18902         Aron       II BCA     NCC
18903         Dhanya   II BCA     NSS
17901         Alex       III BCA
17902         Bipin      III BCA
17903        Simran    III BCA
19901         Arun       I BCA      NSS
19902        Bindu      I BCA      ECO CLUB
19903        Deepa      I BCA      NCC

9 rows selected.

SQL> create table marks_report
(Roll_no number(5)references student_profile,
internal_test number(1) not null,
marks1 number(3) not null,
marks2 number(3) not null,
marks3 number(3) not null,
ECCM number(3),
check(internal_test in(1,2)),
check(marks1 between 1 and 100),
check(marks2 between 1 and 100),
check(marks3 between 1 and 100),
check(ECCM between 0 and 50));

Table created.

SQL>desc marks_report;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ROLL_NO                                            NUMBER(5)
 INTERNAL_TEST                             NOT NULL NUMBER(1)
 MARKS1                                    NOT NULL NUMBER(3)
 MARKS2                                    NOT NULL NUMBER(3)
 MARKS3                                    NOT NULL NUMBER(3)
 ECCM                                               NUMBER(3)

SQL> insert into marks_report values('18901',1,20,50,40,20);

1 row created.

SQL>  insert into marks_report values('18902',2,50,40,90,30);

1 row created.

SQL> insert into marks_report values('18903',1,70,56,38,25);

1 row created.

SQL>  insert into marks_report values('17901',1,66,56,67,'');

1 row created.
```

```
SQL> insert into marks_report values('17901',2,66,77,78,'');

1 row created.

SQL> insert into marks_repor tvalues('17902',2,89,86,98,'');

1 row created.

SQL> insert into marks_report values('17902',1,66,56,59,'');

1 row created.

SQL> insert into marks_report values('17903',1,66,56,59,'');

1 row created.

SQL> insert into marks_report values('17903',2,26,36,31,'');

1 row created.

SQL> insert into marks_report values('19901',1,26,36,57,23);

1 row created.

SQL> insert into marks_report values('19901',2,66,36,57,23);

1 row created.

SQL> insert into marks_report values('19902',1,56,36,57,23);

1 row created.

SQL> insert into marks_report values('19903',2,89,86,77,25);

1 row created.

SQL> select * from marks_report;
```

```
 ROLL_NO  INTERNAL_TEST   MARKS1    MARKS2    MARKS3     ECCM
 ----------  --------------------  -----------   ---------   ----------   ----------
    18901        1             20         50         40        20
    18902        2             50         40         90        30
    18903        1             70         56         38        25
    17901        1             66         56         67
    17901        2             66         77         78
    17903        1             66         56         59
    17903        2             26         36         31
    19901        1             26         36         57        23
    19901        2             66         36         57        23
    17902        2             89         86         98
    17902        1             66         56         59

 ROLL_NO INTERNAL_TEST   MARKS1    MARKS2    MARKS3     ECCM
----------      -------------        ----------   ----------    ----------   ----------
    19902           1            56         36         57        23
    19903           2            89         86         77        25

13 rows selected.
```

a)Find number of students failed class-wise.

```
SQL> select class,count(marks_report.roll_no)"classwise fail"
from student_profile,marks_report
where student_profile.rollno=marks_report.roll_no
and (marks1<35 or marks2<35 or marks3<35)
group by class;

CLASS      classwise fail
----------    --------------
III BCA             1
I BCA               1
II BCA              1
```

b) Display the complete details of the students secured distinction(Percentage>=70)in I BCA.

```
SQL> select student_profile.rollno,name,class,eccc,internal_test,marks1,marks2,marks3,eccm
from student_profile,marks_report
where student_profile.rollno=marks_report.roll_no
and (marks1+marks2+marks3)/3>=70 and class='I BCA';
```

```
ROLLNO  NAME   CLASS  ECCC  INTERNAL_TEST   MARKS1  MARKS2
------------ ---------- ---------- ---------- --------------------     ----------    ----------
 MARKS3  ECCM
------------  ---------
  19903      Deepa     I BCA     NCC           2                     89           86
     77          25
```

c) Display class and highest total marks in second internals in each class.

```
SQL> select class,max(marks1+marks2+marks3)"class wise max marks"
from student_profile,marks_report
where student_profile.rollno=marks_report.roll_no
and internal_test=2
group by class;
CLASS      class wise max marks
----------         --------------------
III BCA                  273
I BCA                   252
II BCA                   180
```

d)Display the student name with rollno and class of those who passed in I internals and failed in II internals.

```
SQL> select student_profile.rollno,name,class
from student_profile,marks_report
where student_profile.rollno=marks_report.roll_no
and internal_test=1
and (marks1>=35 or marks2>=35 or marks3>=35)
intersect
select student_profile.rollno,name,class
from student_profile,marks_report
where student_profile.rollno=marks_report.roll_no
and internal_test=2
and (marks1<35 or marks2<35 or marks3<35);

ROLLNO NAME       CLASS
----------    ----------     ----------
    17903   Simran     III BCA
```

```
/*****************************************************************************
Program No    :12
Program Name: Write a PL/SQL program to compute the selling price of books  depending on
               the book code and category. Use Open, Fetch and Close.The Book_detail
               table contains columns: Book Code, Author, Title, Category  and Price.
Date           :17/01/2023
*****************************************************************************/
SQL> create table book1
( bookcode varchar2(1) not null check(bookcode in('A','B','C','D')),
author varchar2(20) not null,
title varchar2(20) not null,
category varchar2(10) not null,
price number(8,2) not null  );

Table created.
SQL> insert into book1 values('A','rashmi','infotechnology','Novels',125);

1 row created.

SQL> insert into book1 values('A','likith','microtechnology','technology',900);

1 row created.

SQL> insert into book1 values('B','gagan','ecommerce','commerce',1500);

1 row created.

SQL> insert into book1 values('B','vijeth','relatives','Science',2500);

1 row created.

SQL> insert into book1 values('C','jerry','go ask','song',500);

1 row created.

SQL> insert into book1 values('C','tom','mind','Sports',400);

1 row created.
SQL> insert into book1 values('D','henry','mask','Others',3000);
```

```
1 row created.
SQL> select * from book1;

B      AUTHOR        TITLE          CATEGORY    PRICE
---    -------------- ------------------    ----------      ----------
A      rashmi        infotechnology     Novels        125
A      likith         microtechnology   technology     900
B       gagan         ecommerce         commerce     1500
B      vijeth          relatives          Science        2500
C       jerry           go ask              song            500
C       tom             mind                Sports          400
D      henry           mask                Others          3000
7 rows selected.

SQL>ed book;
declare
cursor book_cur is select bookcode,author,title,category,price from book1;
code book1.bookcode%type;
auth book1.author%type;
tit book1.title%type;
cate book1.category%type;
rate book1.price%type;
sellprice book1.price%type;
disamt number(10,2);
discount varchar2(10);
pr number(10,3);
begin
dbms_output.put_line(' ---------------------------------------------------------');
dbms_output.put_line('Bookcode Category Title Author Discount dis_amt sellingprice');
dbms_output.put_line(' ---------------------------------------------------------');
open book_cur;
loop
fetch book_cur into code,auth,tit,cate,rate;
exit when book_cur% notfound;
if(code='A'and cate='novels')then
pr:=0.1;
elsif(code='A'and cate='technology')then
pr:=0.125;
elsif(code='B'and cate='commerce')then
```

```
pr:=0.18;
elsif(code='B'and cate='science')then
pr:=0.19;
elsif(code='C'and cate='song')then
pr:=0.25;
elsif(code='C'and cate='sports')then
pr:=0.24;
elsif(code='D'and cate='others')then
pr:=0.28;
end if;
discount:=rate*pr;
sell price:=rate-discount;
dbms_output.put_line(code||' '||cate||' '||tit||' '||auth||' '||rate||' '||pr*100||'
'||discount||' '||sellprice||' ');
end loop;
close book_cur;
end;
SQL> set serveroutput on;
SQL> @book;
42  /
----------------------------------------------------------
Bookcode Category Title Author Discount dis_amtsellingprice
----------------------------------------------------------
A Novels infotechnology rashmi 125 25
31.25 93.75
A technology microtechnology likith 900 12.5
112.5 787.5
B commerce ecommerce gagan 1500 18
270 1230
B Science relatives vijeth 2500 18
450 2050
C song go ask jerry 500 25
125 375
C Sports mind tom 400 25
100 300
D Others mask henry 3000 25
750 2250
PL/SQL procedure successfully completed.
```

```
/**************************************************************************
Program No    :13
Program Name: Write a PL/SQL program to display employee pay bill (using Cursor For loop)
              Use a Procedure to receive basic pay and to compute DA, HRA, Tax, PF, Gross
              Pay and Net Pay(Use OUT).Base table contains the following columns empnum,
              empname, basic pay.
Date          :24/01/2023
**************************************************************************/
SQL> create table employe
(empno varchar2(14) primary key,
ename varchar2(20) not null,
basic number(10,2) not null);
Table created.
SQL>desc employe;
 Name                            Null?    Type
 -------------------------------- -------- ---------------------------
 EMPNO                           NOT NULL VARCHAR2(14)
 ENAME                           NOT NULL VARCHAR2(20)
 BASIC                           NOT NULL NUMBER(10,2)

SQL> insert into employe values('E1001','radha',20000);

1 row created.

SQL> insert into employe values('E1002','vishwa',30000);

1 row created.

SQL> insert into employe values('E1003','krish',5000);

1 row created.

SQL> insert into employe values('E1004','naveen',45000);

1 row created.

SQL> insert into employe values('E1005','munna',5000);

1 row created.
```

```
SQL> select * from employe;

EMPNO       ENAME    BASIC
------------- -------------- ----------
E1001        radha         20000
E1002        vishwa        30000
E1003        krish         5000
E1004        naveen       45000
E1005        munna         5000

SQL>ed employee;
declare
cursor emp_cur is select empno,ename,basic from employe;
eno employe.empno%type;
name employe.ename%type;
bs employe.basic%type;
da number(10,2);
hra number(10,2);
pt number(10,2);
pf number(10,2);
gross number(10,2);
np number(10,2);
begin
open emp_cur;
loop
fetch emp_cur into eno,name,bs;
exit when emp_cur%notfound;
ifbs<=20000 then
da:=bs*0.35;
hra:=bs*0.8;
elsif(bs>20000 and bs<=30000)then
da:=bs*0.38;
hra:=bs*0.9;
elsif(bs>30000 and bs<=40000)then
da:=bs*0.40;
hra:=bs*0.10;
elsif(bs>40000)then
da:=bs*0.45;
hra:=bs*0.10;
```

```
end if;
gross:=bs+da+hra;
pf:=gross*0.12;
ifpf>=2000 then
pf:=2000;
end if;
if gross<=25000 then
pt:=100;
else
pt:=200;
end if;
np:=gross-(pf+pt);
dbms_output.put_line('========== ===PAY SLIP===============');
dbms_output.put_line('Emp no :'||eno||'Emp name :'||name)
;
dbms_output.put_line('Basicpay :'||bs||'PF :'||pf);
dbms_output.put_line('DA :'||da||'PT :'||pt);
dbms_output.put_line('H.R.A :'||hra);
dbms_output.put_line('Gross :'||gross||'NET pay:'||np);
dbms_output.put_line('***********************************');
end loop;
close emp_cur;
end;
SQL> @employee;
52  /

PL/SQL procedure successfully completed.

SQL> set serveroutput on;
SQL> @employee;
52  /
SQL> @C:/Users/acer/Desktop/emp.sql;
========== ===PAY SLIP==============
Empno :E1001Emp name :radha
Basicpay :20000PF :2000
DA :7000PT :200
H.R.A :16000
Gross :43000NET pay:40800
**********************************
```

```
========== ===PAY SLIP==============
Empno :E1002Emp name :vishwa
Basicpay :30000PF :2000
DA :11400PT :200
H.R.A :27000
Gross :68400NET pay:66200
*********************************
========== ===PAY SLIP==============
Empno :E1003Emp name :krish
Basicpay :5000PF :1290
DA :1750PT :100
H.R.A :4000
Gross :10750NET pay:9360
*********************************
========== ===PAY SLIP==============
Empno :E1004Emp name :naveen
Basicpay :45000PF :2000
DA :20250PT :200
H.R.A :4500
Gross :69750NET pay:67550
*********************************
========== ===PAY SLIP==============
Empno :E1005Emp name :munna
Basicpay :5000PF :1290
DA :1750PT :100
H.R.A :4000
Gross :10750NET pay:9360
*********************************

PL/SQL procedure successfully completed.
```

/***

Program No :14

Program Name: Create a package PCK_ITEM includes a function CHK_ITEM and a procedure PROC_ITEM. Function CHK_ITEM gets one arguments itemno and is used to check whether the parameter itemno exits in ITEM_MASTER and should return 1 if exit. Otherwise 0 and displays proper message.

Date :24/01/2023

***/

```
SQL> create table item_master
(item_number number(5) primary key,
name varchar2(15),
qty number(4),
price number(8,2),
check(qty>0),
check(price>0));

Table created.

SQL>desc item_master;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_NUMBER                               NOT NULL NUMBER(5)
 NAME                                               VARCHAR2(15)
 QTY                                                NUMBER(4)
 PRICE                                              NUMBER(8,2)

SQL> create table item_trans
(item_number number(5),
qty number(4),
trans_date date);

Table created.

SQL>desc item_trans;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ITEM_NUMBER                                        NUMBER(5)
 QTY                                                NUMBER(4)
 TRANS_DATE                                         DATE
```

```
SQL> INSERT INTO item_master values(1,'Ball pen',500,15);

1 row created.

SQL> INSERT INTO item_master values(2,'Cello ball pen',125,10);

1 row created.

SQL> INSERT INTO item_master values(3,'Link blue pen',200,20);

1 row created.

SQL> INSERT INTO item_master values(4,'Camlin pencil',500,7);

1 row created.

SQL> INSERT INTO item_master values(5,'Classmates200',500,600);

1 row created.

SQL>  INSERT INTO item_master values(6,'Classmates100',1500,45);

1 row created.

SQL> INSERT INTO item_master values(7,'A4 ruled',1500,10);

1 row created.

SQL> INSERT INTO item_master values(8,'A4 unruled',1500,10);

1 row created.

SQL> INSERT INTO item_master values(9,'Drawingsheet',450,16);

1 row created.

SQL> INSERT INTO item_master values(10,'JK A4 sheets',450,230);
```

```
1 row created.

SQL> select * from item_master;

ITEM_NUMBER    NAME           QTY     PRICE
    -----------         -------------        ----------  ----------
        1            Ball pen             500        15
        2            Cello ball pen     125        10
        3            Link blue pen      200        20
        4            Camlin pencil       500         7
        5            Classmates200    500        600
        6            Classmates100   1500       45
        7             A4 ruled            1500       10
        8             A4 unruled         1500       10
        9             Drawingsheet      450        16
       10            JK A4 sheets       450        230

10 rows selected.

SQL>ed pkg;
create or replace package Pck_item as
function Chk_item(ino in number)return number;
procedure Proc_item(ino in number,rqty in number,nm out varchar2,up in out number,amt out
number);
end pck_item;
SQL> set serveroutput on;
SQL> @pkg;
6  /

Package created.

SQL>ed pkgb;
create or replace package body Pck_item as
function Chk_item(ino in number) return number is
it_no number(5);
begin
select item_number into it_no from item_master where
item_number=ino;
return 1;
```

```
exception when NO_DATA_FOUND then return 0;
end;
procedure Proc_item(ino in number,rqty in number,nm out varchar2,up in out number,amt out
number)is
rqa number(5);
begin
select qty,price,name into rqa,up,nm from item_master where
item_number=ino;
if rqa<rqty then
dbms_output.put_line('required quantity is more than the available quantity');
else
insert into item_trans values(ino,rqty,sysdate);
update item_master set qty=qty-rqty
where item_number=ino;
dbms_output.put_line('record updated successfully');
amt:=rqty*up;
end if;
end;
end Pck_item;
SQL> @pkgb
27  /

Package body created.

SQL>ed main;
declare
ino item_master.item_number%type;
found_item number(5);
nm item_master.name%type;
qr item_master.qty%type;
up item_master.price%type;
amt number(9,2);
begin
ino:=&ino;
qr:=&qr;
found_item:=Pck_item.Chk_item(ino);
if found_item=1 then
Pck_item.Proc_item(ino,qr,nm,up,amt);
dbms_output.put_line('item:'||nm||'Qunatity:'||qr||'Price:'||up||'total:'||amt);
```

```
else
dbms_output.put_line('item not found');
end if;
end;
SQL> @main;
19  /
Enter value for ino: 1
old   9: ino:=&ino;
new   9: ino:=1;
Enter value for qr: 100
old  10: qr:=&qr;
new  10: qr:=100;
record updated successfully
item:Ball penQunatity:100Price:15total:1500

PL/SQL procedure successfully completed.

SQL> select * from item_trans;

ITEM_NUMBER    QTY      TRANS_DAT
  -----------          ----------        ---------
        1                100          17-JAN-23
```