DBMS

*Lab*

**PART - B**

1. 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. A supplier can supply more than one type of items.

Write the SQL queries for the following:

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

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

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

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

create table Supplier(

Supplier\_no varchar2(6) primary key,

Sname varchar2(25) not null,

Saddress varchar2(15) not null,

Scity varchar2(15) not null);

Table created.

insert into Supplier(Supplier\_no,Sname,Saddress,Scity)values('S1002','cats','serivce road','mangalore');

1 row(s) inserted.

select\*from Supplier;

|  |  |  |  |
| --- | --- | --- | --- |
| **SUPPLIER\_NO** | **SNAME** | **SADDRESS** | **SCITY** |
| S1001 | microtech | market road | bangalore |
| S1002 | cats | serivce road | mangalore |
| S1003 | electrotech | kalasipalya | bangalore |
| S1004 | DOT | gpuc | nittur |
| S1005 | cats | flower market | udupi |

create table Computer\_item(

item\_no varchar2(6) primary key,

Supplier\_no varchar2(6) references Supplier(Supplier\_no)on delete cascade,

item\_name varchar2(10) not null,

iquantity varchar2(10) not null);

Table created.

insert into Computer\_item(item\_no,Supplier\_no,item\_name,iquantity)values('i1001','S1002','keyboard','5');

1 row(s) inserted.

select\*from Computer\_item;

|  |  |  |  |
| --- | --- | --- | --- |
| **ITEM\_NO** | **SUPPLIER\_NO** | **ITEM\_NAME** | **IQUANTITY** |
| i1001 | S1002 | keyboard | 5 |
| i1002 | S1002 | mouse | 4 |
| i1003 | S1003 | keyboard | 5 |
| i1004 | S1004 | monitar | 10 |
| i1005 | S1001 | keyboard | 4 |
| i1006 | S1005 | cpu | 1 |
| i1007 | S1003 | pendrive | 5 |
| i1008 | S1004 | keyboard | 8 |
| i1009 | S1005 | mouse | 7 |
| i10010 | S1001 | printer | 8 |

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

select Supplier.Supplier\_no,Sname,Saddress,Scity,item\_no,item\_name,iquantity

from Supplier,Computer\_item

WHERE Supplier.Supplier\_no=Computer\_item.Supplier\_no

order by Scity,iquantity desc;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SUPPLIER\_NO** | **SNAME** | **SADDRESS** | **SCITY** | **ITEM\_NO** | **ITEM\_NAME** | **IQUANTITY** |
| S1001 | microtech | market road | bangalore | i10010 | printer | 8 |
| S1003 | electrotech | kalasipalya | bangalore | i1007 | pendrive | 5 |
| S1003 | electrotech | kalasipalya | bangalore | i1003 | keyboard | 5 |
| S1001 | microtech | market road | bangalore | i1005 | keyboard | 4 |
| S1002 | cats | serivce road | mangalore | i1001 | keyboard | 5 |
| S1002 | cats | serivce road | mangalore | i1002 | mouse | 4 |
| S1004 | DOT | gpuc | nittur | i1008 | keyboard | 8 |
| S1004 | DOT | gpuc | nittur | i1004 | monitar | 10 |
| S1005 | cats | flower market | udupi | i1009 | mouse | 7 |
| S1005 | cats | flower market | udupi | i1006 | cpu | 1 |

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

select Sname,Saddress,Scity from Supplier,Computer\_item where Supplier.Supplier\_no=Computer\_item.Supplier\_no and item\_name='keyboard';

|  |  |  |
| --- | --- | --- |
| **SNAME** | **SADDRESS** | **SCITY** |
| microtech | market road | bangalore |
| cats | serivce road | mangalore |
| electrotech | kalasipalya | bangalore |
| DOT | gpuc | nittur |

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

select Sname,item\_name from

Supplier,Computer\_item

where Supplier.Supplier\_no=Computer\_item.Supplier\_no

and(Sname='electrotech' or Sname='cats');

|  |  |
| --- | --- |
| **SNAME** | **ITEM\_NAME** |
| cats | keyboard |
| cats | mouse |
| electrotech | keyboard |
| cats | cpu |
| electrotech | pendrive |
| cats | mouse |

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

create table neworder as

select Supplier.Supplier\_no,Sname,Saddress,Scity,Computer\_item.item\_no,item\_name,iquantity from

Supplier,Computer\_item

where Supplier.Supplier\_no=Computer\_item.Supplier\_no and iquantity <5;

Table created.

select\*from neworder;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SUPPLIER\_NO** | **SNAME** | **SADDRESS** | **SCITY** | **ITEM\_NO** | **ITEM\_NAME** | **IQUANTITY** |
| S1001 | microtech | market road | bangalore | i1005 | keyboard | 4 |
| S1002 | cats | serivce road | mangalore | i1002 | mouse | 4 |
| S1005 | cats | flower market | udupi | i1006 | cpu | 1 |

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

EMPLOYEE\_MASTER (EMP\_ID, EMP\_NAME, EMP\_ADDRS, PHONE) ATTENDANCE (EMP\_ID, MONTH, WOM, MHRS, THRS, WHRS, TRHRS, FHRS, SHRS, SUHRS). (Valid values for WOM<=5, MONTH can be 1-12). Apply appropriate constraints. Consider 3 employees. And attendance records for at least two months.

Write the SQL queries for the following:

a) Display EMP\_ID, EMP\_NAME and EMAIL\_ID of all employees who are working on every Sunday of 2nd or 4th week in a month.

b) Display total hours worked by each employee in each month with EMP\_ID,

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

d) Display the employee name, month, week, total hours worked for employees who have total number of hours more than 20 hours a week.

create table Emp\_Master(

Emp\_ID varchar(10) primary key,

Emp\_Name varchar(15) not null,

Emp\_Adderss varchar(15) not null,

emp\_Phno number(15) not null,

Email\_ID varchar(20) not null);

Table created.

insert into Emp\_Master(Emp\_ID,Emp\_Name,Emp\_Adderss,emp\_Phno,Email\_ID)values('E1002','vignesh','nittur',6439982445,'vignesh80@gmail.com');

1 row(s) inserted.

select\*from emp\_master

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMP\_ID** | **EMP\_NAME** | **EMP\_ADDERSS** | **EMP\_PHNO** | **EMAIL\_ID** |
| E1001 | suresh | malpe | 7413542975 | suresh100@gmail.com |
| E1002 | vignesh | nittur | 6439982445 | vignesh80@gmail.com |
| E1003 | ravija | manipal | 8951936987 | ravijaravi@gmail.com |
| E1004 | ganesh | manipal | 9741704877 | ganesh101@gmail.com |
| E1005 | ritheesh | kundapura | 6364564578 | ritheesh12@gmail.com |
| E1006 | razik | udupi | 8648865745 | razikrazi@gmail.com |

create table Attedance(

Emp\_ID varchar(25),

Month number(2),

WOM number(1),

MONhrs number(2),

TUEhrs number(2),

WEhrs number(2),

THhrs number(2),

FRHrs number(2),

SAThrs number(2),

SUNhrs number(2),

foreign key(Emp\_ID)references Emp\_Master(Emp\_ID),

constraint mw check(WOM<=5 and Month BETWEEN 1 and 12));

Table created.

insert into Attedance(Emp\_ID,Month,WOM,MONhrs,TUEhrs,WEhrs,THhrs,FRHrs,SAThrs,SUNhrs)values('E1001',04,4,8,5,3,0,3,6,9);

1 row(s) inserted.

select \* from Attedance;

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **EMP\_ID** | **MONTH** | **WOM** | **MONHRS** | **TUEHRS** | **WEHRS** | **THHRS** | **FRHRS** | **SATHRS** | **SUNHRS** |
| E1001 | 11 | 5 | 4 | 3 | 7 | 3 | 5 | 8 | 5 |
| E1003 | 12 | 0 | 3 | 2 | 1 | 3 | 4 | 4 | 5 |
| E1004 | 6 | 3 | 2 | 3 | 8 | 3 | 9 | 5 | 2 |
| E1005 | 4 | 0 | 8 | 5 | 3 | 8 | 3 | 6 | 9 |
| E1001 | 4 | 4 | 8 | 5 | 3 | 0 | 3 | 6 | 9 |
| E1001 | 5 | 2 | 3 | 5 | 6 | 0 | 3 | 6 | 9 |
| E1004 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E1004 | 8 | 4 | 2 | 6 | 8 | 3 | 9 | 4 | 2 |
| E1003 | 11 | 5 | 3 | 6 | 7 | 3 | 4 | 2 | 5 |
| E1003 | 10 | 4 | 3 | 1 | 1 | 3 | 3 | 4 | 5 |

a) Display EMP\_ID, EMP\_NAME and EMAIL\_ID of all employees who are working on every Sunday of 2nd or 4th week in a month.

select e.Emp\_ID,e.Emp\_Name from Emp\_Master e,Attedance a where e.Emp\_ID=a.Emp\_ID and a.SUNhrs>0 and a.WOM in(2,4);

|  |  |
| --- | --- |
| **EMP\_ID** | **EMP\_NAME** |
| E1001 | suresh |
| E1001 | suresh |
| E1003 | ravija |
| E1004 | ganesh |

b) Display total hours worked by each employee in each month with EMP\_ID,

select a.Emp\_ID,sum(MONhrs+TUEhrs+WEhrs+THhrs+FRHrs+SAThrs+SUNhrs)from Emp\_Master e,Attedance a where e.Emp\_ID=a.Emp\_ID group by a.Emp\_ID;

|  |  |
| --- | --- |
| **EMP\_ID** | **SUM(MONHRS+TUEHRS+WEHRS+THHRS+FRHRS+SATHRS+SUNHRS)** |
| E1005 | 42 |
| E1001 | 101 |
| E1004 | 66 |
| E1003 | 72 |

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

select Emp\_Name from Emp\_Master where Emp\_ID not in(select Emp\_ID from Attedance);

|  |
| --- |
| **EMP\_NAME** |
| vignesh |
| razik |

d) Display the employee name, month, week, total hours worked for employees who have total number of hours more than 20 hours a week.

select emp\_name,wom,sum(MONhrs+TUEhrs+WEhrs+THhrs+FRHrs+SAThrs+SUNhrs) from emp\_master e, attedance a where e.emp\_id=a.emp\_id group by emp\_name,wom having sum(MONhrs+TUEhrs+WEhrs+THhrs+FRHrs+SAThrs+SUNhrs)>20;

|  |  |  |
| --- | --- | --- |
| **EMP\_NAME** | **WOM** | **SUM(MONHRS+TUEHRS+WEHRS+THHRS+FRHRS+SATHRS+SUNHRS)** |
| suresh | 4 | 34 |
| ganesh | 3 | 32 |
| ritheesh | 0 | 42 |
| suresh | 2 | 32 |
| suresh | 5 | 35 |
| ravija | 0 | 22 |
| ravija | 5 | 30 |
| ganesh | 4 | 34 |

3 Create the following tables by identifying primary and foreign keys, specify the not null property for mandatory keys.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRODUCT\_DETAIL | | | | |
| P\_No | P\_Name | QtyAvailable | Price | Profit (%) |
| P0001 | Monitor | 10 | 3000 | 20 |
| P0002 | Pen Drives | 50 | 650 | 5 |
| P0003 | CD Drive | 100 | 10 | 3 |
| P0004 | Keyboard | 25 | 600 | 10 |

|  |  |  |
| --- | --- | --- |
| PURCHASED\_DETAIL | | |
| CustNo | P\_No | QtySold |
| C1 | P0003 | 2 |
| C2 | P0002 | 4 |
| C3 | P0002 | 10 |
| C4 | P0001 | 3 |
| C1 | P0004 | 2 |
| C2 | P0003 | 2 |
| C4 | P0004 | 1 |

Write the SQL queries for the following:

a) Display total amount spent by C2.

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

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

d) How much Profit does the shopkeeper get on C1’s purchase?

e) How many ‘Pen Drives’ have been sold?

create table product\_detail(

p\_no varchar2(5) primary key check(p\_no like 'p%'),

p\_name varchar2(20) not null,

qtyavailable number(5) not null,

price number(6) not null,

profit number(3) not null);

Table created.

insert into product\_detail values('p0001','Monitor',10,3000,20);

insert into product\_detail values('p0002','Pen Drives',50,650,5);

insert into product\_detail values('p0003','CD Drive',100,10,3);

insert into product\_detail values('p0004','Keyboard',25,600,10);

1 row(s) inserted.

select \* from product\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P\_NO** | **P\_NAME** | **QTYAVAILABLE** | **PRICE** | **PROFIT** |
| p0001 | Monitor | 10 | 3000 | 20 |
| p0002 | Pen Drives | 50 | 650 | 5 |
| p0003 | CD Drive | 100 | 10 | 3 |
| p0004 | Keyboard | 25 | 600 | 10 |

create table purchased\_detail(

custno varchar2(2) not null check(custno like 'C%'),

p\_no varchar2(8) references product\_detail(p\_no) on delete cascade,

qtysold number(2) not null);

Table created.

insert into purchased\_detail values('C1','p0003',2);

insert into purchased\_detail values('C2','p0002',4);

insert into purchased\_detail values('C3','p0002',10);

insert into purchased\_detail values('C4','p0001',3);

insert into purchased\_detail values('C1','p0004',2);

insert into purchased\_detail values('C2','p0003',2);

insert into purchased\_detail values('C4','p0004',1);

1 row(s) inserted.

select \* from purchased\_detail

|  |  |  |
| --- | --- | --- |
| **CUSTNO** | **P\_NO** | **QTYSOLD** |
| C1 | p0003 | 2 |
| C2 | p0002 | 4 |
| C3 | p0002 | 10 |
| C4 | p0001 | 3 |
| C1 | p0004 | 2 |
| C2 | p0003 | 2 |
| C4 | p0004 | 1 |

a) Display total amount spent by C2.

select

sum(qtysold\*price)"Total Amount" from product\_detail,purchased\_detail

where

product\_detail.p\_no=purchased\_detail.p\_no and custno='C2';

|  |
| --- |
| **Total Amount** |
| 2620 |

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

select

p\_name from product\_detail where qtyavailable<30

union

select

p\_name from product\_detail where p\_no in(select p\_no from purchased\_detail group by p\_no having sum(qtysold)<5);

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |
| --- |
| **P\_NAME** |
| CD Drive |
| Keyboard |
| Monitor |

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

select

p\_name,qtysold from product\_detail,purchased\_detail

where

product\_detail.p\_no=purchased\_detail.p\_no and custno='C4';

|  |  |
| --- | --- |
| **P\_NAME** | **QTYSOLD** |
| Monitor | 3 |
| Keyboard | 1 |

d) How much Profit does the shopkeeper get on C1’s purchase?

select

(qtysold\*price\*profit)/100 "Profit Amount" from product\_detail,purchased\_detail

where

product\_detail.p\_no=purchased\_detail.p\_no and (custno='C1');

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |
| --- |
| **Profit Amount** |
| .6 |
| 120 |

e) How many ‘Pen Drives’ have been sold?

select

sum(qtysold)"Total Pen Drives sold" from product\_detail,purchased\_detail

where

product\_detail.p\_no=purchased\_detail.p\_no and (p\_name='Pen Drives');

|  |
| --- |
| **Total Pen Drives sold** |
| 14 |

4.Create table STUDENT\_PROFILE that includes Rollno, name, class, ECCC (Extra/Cocurricular he belongs to such as SPORTS, NSS, etc.) and another table MARKS\_REPORT that includes Rollno, Internal\_Test, Marks1, Marks2, Marks3 and ECCC\_marks.

Constraints

* Internal\_Test can be either 1 or 2.
* Each mark can be 0 – 100. Absence in the test can be entered as -1.
* Consider at least 3 classes.

Apply suitable datatype and constraints to each column.

Insert 5 students marks report in the both the tests.

Write the SQL queries for the following:

a) Find number of students failed class-wise.

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

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

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

create table student\_profile(

roll\_no varchar(4) primary key,

sname varchar(20) not null,

class varchar(20) not null,

ECCC varchar(20) not null);

insert into student\_profile values('s102','Vignesh','1BCA','Rovers');

1 row(s) inserted.

select \* from student\_profile;

|  |  |  |  |
| --- | --- | --- | --- |
| **ROLL\_NO** | **SNAME** | **CLASS** | **ECCC** |
| s101 | Nidhi | 1BCA | sports |
| s102 | Vignesh | 1BCA | Rovers |
| s103 | Vijay | 2BCA | NSS |
| s104 | Sultan | 3BCA | NSS |
| s105 | Ravija | 2BCA | Sports |

create table Marks\_report(

roll\_no varchar(4),

internal\_test number(1)check(internal\_test in(1,2)),

mark1 number(3),

mark2 number(3),

mark3 number(3),

ECCmark number(3),

constraints mark check ((mark1 between -1 and 100)and (mark2 between -1 and 100)and(mark3 between -1 and 100)),

constraints pk\_mr primary key (roll\_no ,internal\_test),

foreign key(roll\_no)references student\_profile(roll\_no));

insert into Marks\_report values('s101',1,75,80,90,87);

1 row(s) inserted.

select \* from Marks\_report;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ROLL\_NO** | **INTERNAL\_TEST** | **MARK1** | **MARK2** | **MARK3** | **ECCMARK** |
| s101 | 1 | 75 | 80 | 90 | 87 |
| s101 | 2 | 89 | 90 | 90 | 97 |
| s102 | 1 | 90 | 85 | 96 | 97 |
| s102 | 2 | 99 | 94 | 97 | 98 |
| s103 | 1 | 30 | 32 | 32 | 27 |
| s103 | 2 | 49 | 56 | 57 | 78 |
| s104 | 1 | 76 | 98 | 65 | 46 |
| s104 | 2 | 29 | 26 | 32 | 34 |
| s105 | 1 | 78 | 88 | 75 | 66 |
| s105 | 2 | 89 | 67 | 87 | 84 |

a) Find number of students failed class-wise.

select class,count(\*)from student\_profile s,marks\_report m where s.roll\_no=m.roll\_no and (mark1<35 or mark2<35 or mark3<35) group by class;

|  |  |
| --- | --- |
| **CLASS** | **COUNT(\*)** |
| 2BCA | 1 |
| 3BCA | 1 |

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

select student\_profile.roll\_no,sname,class,internal\_test,ECCC,mark1,mark2,mark3,ECCmark from student\_profile,marks\_report where student\_profile.roll\_no=marks\_report.roll\_no and ((mark1+mark2+mark3)/3)>=70 and class='1BCA';

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ROLL\_NO** | **SNAME** | **CLASS** | **INTERNAL\_TEST** | **ECCC** | **MARK1** | **MARK2** | **MARK3** | **ECCMARK** |
| s101 | Nidhi | 1BCA | 1 | sports | 75 | 80 | 90 | 87 |
| s101 | Nidhi | 1BCA | 2 | sports | 89 | 90 | 90 | 97 |
| s102 | Vignesh | 1BCA | 1 | Rovers | 90 | 85 | 96 | 97 |
| s102 | Vignesh | 1BCA | 2 | Rovers | 99 | 94 | 97 | 98 |

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

select class,max(mark1+mark2+mark3)from student\_profile,marks\_report where student\_profile.roll\_no=marks\_report.roll\_no and internal\_test=2 group by class;

|  |  |
| --- | --- |
| **CLASS** | **MAX(MARK1+MARK2+MARK3)** |
| 2BCA | 243 |
| 3BCA | 87 |
| 1BCA | 290 |

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

select student\_profile.roll\_no,sname,class from student\_profile,marks\_report where student\_profile.roll\_no=marks\_report.roll\_no and (mark1<35 and mark2<35 and mark3<35) and internal\_test=1

union

select student\_profile.roll\_no,sname,class from student\_profile,marks\_report where student\_profile.roll\_no=marks\_report.roll\_no and (mark1<35 or mark2<35 or mark3<35) and internal\_test=2;

|  |  |  |
| --- | --- | --- |
| **ROLL\_NO** | **SNAME** | **CLASS** |
| s103 | Vijay | 2BCA |
| s104 | Sultan | 3BCA |