DBMS

1) a) Sllect sum(qty) from purchase;

b) Select amount as p.price * s.qty from product p, sales s where p.proid = s.proid;

c) update product set stock = stock+10;

d) Select prod~~gc~~ from product where stock <15 order by stock;

e) Select prodid, prodesc, price from product where prodid in (Select proid from purchase where supname="ABC");

f) create view V as (select proidid from purchase intersect select prodid from sales);

2. a) alter table purchase ~~[illegible]~~ add column { custname varchar(15) };

b) select prodid from product where prodid not in (select pro~~[illegible]~~did from purchase);

c) select prodid, prodesc, price from product where custname="Mathew";

d) Select distinct (custname) from purchase;

e) Select sum(qty), proid from purchase group by proid;

f) alter table product add column ReorderQty varchar (50) ;, where ReorderQty = 50;

3. a) Select ~~distinct~~ count (proid), proid from purchase group by proid having count (proid) > 5;

b) select supname from purchase where proid = "DAL23";

c) select supname from purchase intersect select custname from sales;

d) create view V as select sum(qty), proid from sales group by proid;

e) Select prodesc from product where prodid in (select proid from sales where salesid = "SA234");

f) —

4. a) Select custid from customer
where custid not in (select custid
from loan);

b) Select loanid, Amount, custid from
loan where custid in (select custid
from customer where custname = "JOHN");

c) Select count (*) from loan;

d) Select sum(amount), custid from
loan ~~[illegible]~~ . group by custid;

e) Select custid, count (custid) from
loan group by. custid having
count (custid) ~~>~~ < 2;

f) Create Procedure disp (PID IN
Loan. ~~Load~~ . Loanid %. type)

```
IS
cursor . emp_cur is
Select custid, amount from loan
where loanid = pID;
BEGIN
For emp_rec in emp_cur
Loop
  dbms_output.put_line (emp_rec.
      custid || ' ' || emp_rec. amount);
End loop;
End;
/
```

Query: [Set serveroutput on;
Execute disp(2);]

5. a) update HLoan set amount = amount
- 5000;

b) Select count(*) from VLoan group by
custid having custid = 2;

c) select custid, custname, age, phno
from customer. where custid in
(select custid from HLoan);

d) Select sum(amount), custid
from VLoan group by custid;

e) Select custid, custname, Age, phno
from customer where custid in
(select custid from HLoan
intersect select custid from VLoan);

f) create ~~Producre~~ Procedure disp
(PID in HLoan.custid %.type)

```
IS
  cursor emp-cur is
  select Hloanid, amount from HLoan
        where custid = PID;
emp-rec.emp-cur %. row type;
 Begin
  for emp-rec in emp-cur
  loop
   dbms-output.put-line(emp-rec.
        HLoanid || ' ' || emp-rec.amount);
  End loop;
 end;
 /
```

6. a) select custname from customer
where custname [illegible] like '%.singh';

b) select custid , max(accbal) from
account group by custid;

c) Select * from account where
~~customer~~ custid in ('c01', 'c02', 'c03');

d) select custid , custname. from
customer Order by custid Desc;

e) (Select custid, custname from customer
where custname = 'Leena') and custid in
(select sum(emi), custid from
loan group by custid);

f) create Function emp-dtl-func
return account. accdetail %type;
~~Begi~~
IS
acc-bal account.accbal%.type;
Begin
select sum(accbal) into acc-bal from
account where custid = 2;
Return acc_bal;
End;
/

7. a) select * from customer;

b) createview v as select sum(amount) from H Loan group by custid having custid = (select custid from H Loan intersect select custid from V Loan);

c) select custname from customer where custid in (select custid from H Loan intersect select custid from V Loan);

d) Select sum(amount) from V Loan group by custid;

e) create Function cust_det
Return cust_details . cust det %. type
Is
custid . customer . cust_det %. type
Begin
select * from customer;
return cust_det;
End;
/

8. a) select * from boat where BType = 's'
and colour = 'red';

b) create view v as ~~boatList~~
Select Bid, BName from boat;

c) Select max (SID) from sails;

~~d) select * from sailor group by sid~~

d) create procedure disp (sid IN
sails. sid %. type)

```
IS
cursor emp-cur is
select sid, name, Age, gender, Rating
  from sailor;
 Begin
 For emp-rec in emp-cur
Loop
   dbms-output. put-line('Details : '||
            emp rec. sailor);
End loop;
End;
/
```

9) a) select count (name) from sailor
where gender = 'female';

b) select sid , name from sailor where
name = 'sam' and Bid in (select Bname,
Btype from boat);

c) select count (name) from sailor
where gender = 'male' and rating = 'fair';

d) select Bname from boat ~~[illegible]~~ in
(select name from sailor) where
Boat . Bid = sails . Bid and sailor . sid = sails . sid
;

e) create trigger boat trigger
Before update of BName
on boat
for each row
Begin
insert into Boat value (:old . bid,
:old . Btype , :old . Bname , :old . colour);
End;

10) a) select name from sailer where
name like '%c';

b) select age from sailor where
age >18;

c) Ans: 9c

d) create table sails (shift String(5),
sid int foreign key references sailor (sid),
Bid int foreign key references Boat (Bid));

e) ~~Ans~~: 9e.

11) a) Eno int primary key not null
Dept_no " " " " "
proj_no " " " " "

b). ~~S~~ create sequence employee (eno)
starts with 1;
increament by 1;
minvalue 1;
max value 10;
no cycle;

c) select * from employee where
name like "%Raju";

d) select dept_no from department
as in eno (select count (Name) from
employee) group by dept_no;

e) Select * from employee where salary < avg
(salary);

12) a) select * from department as in name ~~where~~. (select count (eno) ; narm from Employee where count (eno)>6 group by dept_no);

b) select count (name) from employee as in (select proj_no , name from project group by project);

c) select name from department having max (names) from project;

d) select name from employee as in. proj_no (select name from project) having count (hours)>5 ;

e) select name, designation, salary from employee group by eno;

13).

a) select eno, name from employee natural join project ~~nam~~ where proj_no = 4;

b) select * from department where dept_no not in (select dept_no from project);

c) select name from department having avg (salary) > 20000;

d) select name from employee natural join (select name from project natural join) (select hours from work for);

e) create index employees on employee (eno); ~~index~~

14)a) Refer joins in notes.

b)
```
create trigger cust trigger.
Before update of age
On customer
For each row
Begin
insert into customerss
Values
(:old.cust_id; :old.name, :old.age,
    :old, address);
End ;
```