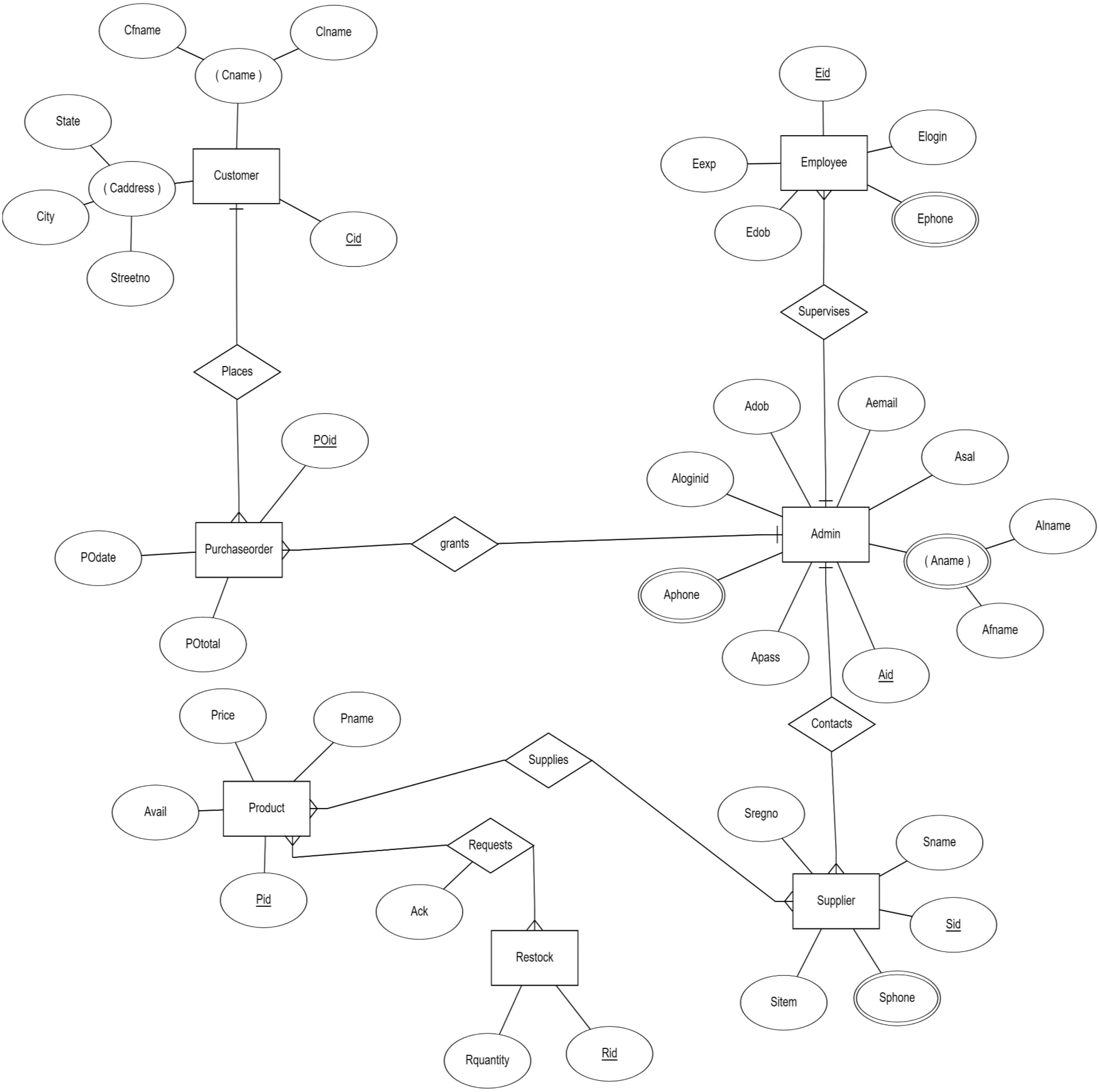
----------------------------------- Grocery Management System by 20pw31,20pw25-------------------------- --------------------------



```
CREATE TABLE Admin
(
   Aid varchar(10) primary key,
   Aloginid varchar(50),
   Apass varchar(50),
   Afname char(30),
   Alname char(30),
   Adob date,
   Asal float,
   check(Asal>0)
);

CREATE TABLE Supplier
(
   Aid varchar(10),
          Sid varchar(10) primary key,
   Sname char(30),
   Sitem varchar(50),
   Sregno int unique,
   foreign key (Aid) references Admin(Aid)
);

CREATE TABLE Suppno
(
          Sid varchar(10),
          Sphone int,
   foreign key(Sid) references Supplier(Sid),
   primary key(Sid,Sphone)
);
```

```
CREATE TABLE Customerr
(
            Cid varchar(10) primary key,
   Cfname char(30),
   Clname char(30),
            state char(100),
   city char(100),
   streetno varchar(100)
);

CREATE TABLE Purchaseorder
(
            Cid varchar(10),
   Aid varchar(10),
   POid varchar(10) primary key,
   POtotal int,
   POdate date,
   foreign key (Aid) references Admin(Aid),
   foreign key (Cid) references Customerr(Cid),
   check(POtotal>0)
);

CREATE TABLE Prod
(
            Pid varchar(10) primary key,
            Pname char(30) ,
            Price int not null,
            Avail int,
   check(price>0)
);
```

```
CREATE TABLE Employee
(
  Aid varchar(10),
       Eid varchar(10) primary key,
       Ename char(30),
       Elogin varchar(50),
  Edob date,
  Esal float,
  Eexp int,
  foreign key (Aid) references Admin(Aid),
  check(Esal>0)
);
drop table employee;

CREATE TABLE Empno
(
       Eid varchar(10),
       Ephone int,
  foreign key(Eid) references Employee(Eid),
  primary key(Eid,Ephone)
);

CREATE TABLE Supplies
(
  Pid varchar(10),
  Sid varchar(10),
  foreign key(Pid) references Prod(Pid),
  foreign key(Sid) references Supplier(Sid),
  Primary key(Pid,Sid)
);
```

```
CREATE TABLE Restock
(
   Rid varchar(10) primary key,
   Rquantity int,
   Rdate date
);

CREATE TABLE Requests
(
   Pid varchar(10),
   Rid varchar(10),
   Ack varchar(4),
   foreign key(Pid) references Prod(Pid),
   foreign key(Rid) references Restock(Rid),
   Primary key(Pid,Rid)
);

insert into admin values ('A1','ASHOK09','HIDDENSEAT','ASHOK','RAJ','10-OCT-1990',200000);
insert into admin values ('A2','LOGAN8KO','IUNETETT','LOGESH','MILKO','11-JAN-1990',220000);
insert into admin values ('A3','MUGEN88U','CARKEYS','MUGEN','ROBERT','15-MAY-1993',150000);
insert into admin values ('A4','SHIV09GG','LOPLICE','SHIV','GAGON','21-DEC-1979',180000);
insert into admin values ('A5','MITAL_KK','MILLSDET','MILAN','TALI','12-OCT-1991',200000);
insert into admin values ('A6','ALBERT_TOSS','JHEESHUS','ALBERT','RAJ','29-MAY-1988',190000);
insert into admin values ('A7','MIJAAL','LAJAK','MISHRA','AAJAL','07-JUN-1985',250000);
insert into admin values ('A8','KABEEEER','ARJRED','KABIR','SINGH','09-OCT-1995',140000);
insert into admin values ('A9','PRAGYAN_M007','PRAGMART','PRAGYA','MARTIN','11-FEB-
1992',180000);
insert into admin values ('A10','XAVIER_ZONKS','XAVISS09','XAVIER','ARAJ','01-JAN-1990',250000);
insert into admin values ('A11','LIMARR976','NATMNMTMMPVPSM','LIJAL','MARR','05-MAR-
1988',195000);
insert into admin values ('A12','SHILPAVT00U','HAMILTONIAN','SHILPA','PAVILYAT','10-OCT-
1980',300000);
```

```
insert into admin values ('A13','LIPSIA_TANN6','HAMSTERRAPTOR','LIPSIA','STILTON','11-MAY-1999',120000);

insert into admin values ('A14','HAMERIA_JUANGG','HAMCHINATO','HAMERIA','JUANGG','20-SEP-1982',270000);

insert into admin values ('A15','BARISTIA_BORE011','EPKKKKPNK','BARISTIA','POPE','17-JUN-1996',165000);


insert into Customerr values ('C1','Prem','Reddy','Andhra Pradesh','Amaravati','Ashoka Metropolitian,Street 3');

insert into Customerr values ('C2','Radha','Krishnan','Tamil Nadu','Coimbatore','Peelamedu,Coimbatore,Rk Nagar Street 7');

insert into Customerr values ('C3','Sandeep','Khanna','Tamil Nadu','Chennai','Bangaru Naidu Colony,K.K. Nagar street 3');

insert into Customerr values ('C4','Ajay','Kumar','Tamil Nadu','Chennai','Nkm street,12th Main Road,street 4');

insert into Customerr values ('C5','Vijay','Krishnan','Tamil Nadu','Chennai','Raghavendra Street, Srinivasa Nagar,street 2');

insert into Customerr values ('C6','Pravin','Balaji','Tamil Nadu','Coimbatore','Anandha backery,Hudo colony street 7');

insert into Customerr values ('C7','Arun','Vijay','Tamil Nadu','Chennai','Valasaravakkam,Street 3');

insert into Customerr values ('C8','Karthick','Manian','Tamil Nadu','Coimbatore','Bharathiyar road,ganapathy colony street 5');

insert into Customerr values ('C9','Siva','Reddy','Andhra Pradesh','Amaravati','Begumpet Begumpet,S.P.Road,,Street 8');

insert into Customerr values ('C10','Raj','Reddy','Andhra Pradesh','Hyderabad','Inorbit Mall, Mind Space,Street 9');

insert into Customerr values ('C11','Aniruddha','Jatkar','Karnataka','Bangalore','Pheonix Market City,Whitefield Road,Street 1');

insert into Customerr values ('C12','Karthick','Pranav','Tamil Nadu','Coimbatore','Peelamedu funmall,street 6');

insert into Customerr values ('C13','Abhishek','Varman','Karnataka','Bangalore','Gopalan Innovation Mall, Street 8');

insert into Customerr values ('C14','Trisha','Krishnan','Tamil Nadu','Chennai','BNT Connection, Nelson Manickam,Road,Street 9');

insert into Customerr values ('C15','Hari','Balagi','Maharashtra','Mumbai','Milan Mall, Milan Subway Road,Street 6');
insert into Prod values ('P1','Canned vegetables',56,200);
insert into Prod values ('P2','Cup noodles',46,150);
insert into Prod values ('P3','Chips',26,50);
insert into Prod values ('P4','Curd',30,20);
insert into Prod values ('P5','Milk',46,18);
insert into Prod values ('P6','Flour',200,50);
insert into Prod values ('P7','Shampoo',100,150);
insert into Prod values ('P8','Bread',46,160);
insert into Prod values ('P9','Biscuits',10,50);
insert into Prod values ('P10','Corn flakes',36,70);
insert into Prod values ('P11','Ketchup',10,80);
insert into Prod values ('P12','Dishwash liquid',76,90);
insert into Prod values ('P13','Ice cream',40,90);
insert into Prod values ('P14','Laundry Detergent',160,200);
insert into Prod values ('P15','Body lotion',120,300);

insert into restock values ('R1',20,'10-OCT-2020');
insert into restock values ('R2',30,'11-JAN-2020');
insert into restock values ('R3',17,'15-MAY-2020');
insert into restock values ('R4',15,'21-DEC-2020');
insert into restock values ('R5',22,'12-OCT-2021');
insert into restock values ('R6',23,'29-MAY-2021');
insert into restock values ('R7',40,'07-JUN-2020');
insert into restock values ('R8',15,'09-OCT-2021');
insert into restock values ('R9',33,'11-FEB-2021');
insert into restock values ('R10',17,'01-JAN-2021');
insert into restock values ('R11',19,'05-MAR-2020');
insert into restock values ('R12',39,'10-OCT-2020');
insert into restock values ('R13',35,'11-MAY-2021');
insert into restock values ('R14',14,'20-SEP-2022');
insert into restock values ('R15',10,'17-JUN-2022');
insert into supplier values ('A1','S1','Raghu','Bread',1);

insert into supplier values ('A4','S2','Gautham','Corn flakes',2);

insert into supplier values ('A9','S3','Krishnan','Chips',3);

insert into supplier values ('A7','S4','Ajay','Ice cream',4);

insert into supplier values ('A5','S5','Naveen','Laundry Detergent',5);

insert into supplier values ('A6','S6','Karthick','Biscuits',6);

insert into supplier values ('A12','S7','Varun','Corn flakes',7);

insert into supplier values ('A10','S8','Kishor','Shampoo',8);

insert into supplier values ('A9','S9','Raghavan','Cup noodles',9);

insert into supplier values ('A5','S10','Mathialagan','Body lotion',10);

insert into supplier values ('A6','S11','Rakshith','Dishwash liquid',11);

insert into supplier values ('A14','S12','Sanjith','Milk',12);

insert into supplier values ('A15','S13','Sukumar','Ketchup',13);

insert into supplier values ('A13','S14','Gokul','Laundry Detergent',14);

insert into supplier values ('A11','S15','Vineeth','Canned vegetables',15);


insert into purchaseorder values ('C6','A4','PO1',200,'10-JAN-2021');

insert into purchaseorder values ('C9','A9','PO2',50,'19-JUN-2021');

insert into purchaseorder values ('C1','A7','PO3',150,'15-MAR-2021');

insert into purchaseorder values ('C8','A12','PO4',250,'27-MAY-2021');

insert into purchaseorder values ('C11','A5','PO5',180,'15-AUG-2021');

insert into purchaseorder values ('C13','A7','PO6',90,'01-JAN-2021');

insert into purchaseorder values ('C10','A15','PO7',50,'18-SEP-2021');

insert into purchaseorder values ('C14','A8','PO8',140,'02-DEC-2021');

insert into purchaseorder values ('C15','A3','PO9',230,'04-NOV-2021');

insert into purchaseorder values ('C8','A8','PO10',300,'18-SEP-2021');

insert into purchaseorder values ('C3','A1','PO11',160,'13-FEB-2021');

insert into purchaseorder values ('C7','A4','PO12',80,'21-JUL-2021');

insert into purchaseorder values ('C4','A9','PO13',200,'29-JUL-2021');

insert into purchaseorder values ('C8','A7','PO14',250,'29-SEP-2021');
insert into purchaseorder values ('C11','A2','PO15',230,'24-SEP-2021');

insert into employee values ('A5','E1','Sam','Sam@gmail.com','24-SEP-2001',60000,5);
insert into employee values ('A3','E2','Ram','Ram@gmail.com','25-OCT-2001',50000,7);
insert into employee values ('A5','E3','Siddharth','Siddharth@gmail.com','19-JAN-2002',80000,3);
insert into employee values ('A3','E4','Vishal','Vishal@gmail.com','22-MAR-2003',40000,2);
insert into employee values ('A1','E5','Vimal','Vimal@gmail.com','23-JUL-2002',56000,4);
insert into employee values ('A6','E6','Murali','Murali@gmail.com','25-JUN-2003',45000,5);
insert into employee values ('A4','E7','Nakul','Nakul@gmail.com','21-NOV-2004',59000,3);
insert into employee values ('A7','E8','Kannan','Kannan@gmail.com','27-OCT-2002',54000,6);
insert into employee values ('A6','E9','Akil','Akil@gmail.com','15-OCT-2003',52000,2);
insert into employee values ('A4','E10','Kavin','Kavin@gmail.com','12-MAR-2002',65000,4);
insert into employee values ('A7','E11','Sukhil','suk2@yahoo','10-SEP-2002',50000,4);
insert into employee values ('A9','E12','Sugu','suGU@yahoo','19-AUG-2001',60000,5);
insert into employee values ('A8','E13','Shilmini','shill@yahoo','29-OCT-2002',72000,6);
insert into employee values ('A15','E14','Geetha','gi8@yahoo','10-SEP-2000',80000,3);
insert into employee values ('A2','E15','Lapraja','lapraj@yahoo','01-AUG-2001',65000,4);

insert into supplies values('P1','S2');
insert into supplies values('P2','S3');
insert into supplies values('P3','S4');
insert into supplies values('P4','S5');
insert into supplies values('P6','S5');
insert into supplies values('P1','S3');
insert into supplies values('P7','S8');
insert into supplies values('P9','S7');
insert into supplies values('P10','S1');
insert into supplies values('P11','S8');
insert into supplies values('P12','S5');
insert into supplies values('P7','S12');
insert into supplies values('P9','S8');
insert into supplies values('P9','S1');
insert into supplies values('P8','S10');

insert into requests values('P1','R2','YES');
insert into requests values('P2','R3','NO');
insert into requests values('P3','R4','YES');
insert into requests values('P4','R5','YES');
insert into requests values('P6','R5','NO');
insert into requests values('P1','R3','NO');
insert into requests values('P7','R8','YES');
insert into requests values('P9','R7','YES');
insert into requests values('P10','R1','NO');
insert into requests values('P11','R8','NO');
insert into requests values('P12','R5','YES');
insert into requests values('P7','R12','NO');
insert into requests values('P9','R8','NO');
insert into requests values('P9','R1','YES');
insert into requests values('P8','R10','NO');

insert into empno values ('E1',9865255153);
insert into empno values ('E2',9235855133);
insert into empno values ('E3',8865275153);
insert into empno values ('E4',7865258153);
insert into empno values ('E5',8885259453);
insert into empno values ('E6',9645255132);
insert into empno values ('E7',9445355113);
insert into empno values ('E8',8875257953);
insert into empno values ('E9',9443478092);
insert into empno values ('E10',8883179829);
insert into empno values ('E11',7894552391);
insert into empno values ('E12',9789125698);
insert into empno values ('E13',9943215430);
insert into empno values ('E14',9893274723);
insert into empno values ('E15',9589320132);
insert into empno values ('E1',9871274572);
insert into empno values ('E2',9234875125);
insert into empno values ('E3',8862247753);
insert into empno values ('E4',9875248253);
insert into empno values ('E5',9873548430);

insert into suppno values('S1',666211);
insert into suppno values('S3',214563);
insert into suppno values('S4',821779);
insert into suppno values('S2',635892);
insert into suppno values('S8',888251);
insert into suppno values('S2',168947);
insert into suppno values('S5',256125);
insert into suppno values('S10',105415);
insert into suppno values('S1',346112);
insert into suppno values('S7',141515);
insert into suppno values('S10',546546);
insert into suppno values('S6',148754);
insert into suppno values('S12',148375);
insert into suppno values('S8',748768);
insert into suppno values('S12',274178);
insert into suppno values('S6',925822);
insert into suppno values('S9',378148);
insert into suppno values('S11',275768);
insert into suppno values('S13',274468);
insert into suppno values('S14',284685);
insert into suppno values('S15',250444);
```

```
-- QUERIES  FOR DBMS PACKAGE

--1)Find the customer who purchased more number of product

set serveroutput on;

Declare

   cnt int :=0;

Begin

for i in (SELECT * FROM customerr natural join (SELECT cid from customerr natural join
purchaseorder group by cid order by count(cid) desc))

   loop

      dbms_output.put_line('Customer who purchased more product is : ' ||i.cfname||i.clname);

      cnt := cnt+1;

      exit when cnt=1;

   end loop;

end;


--2)Find the employee who gets low salary so update his salary by 15 percent

Begin

   update employee

   set Esal = Esal + (Esal * 0.15) Where Esal = (select min(Esal) from employee);

end;


--3)Delete the supplier who havent purchase any products

Begin

   delete supplier

   where sid in (select d1.sid from supplier d1 left join supplies d2 on d2.sid = d1.sid

             where   d2.sid IS NULL);

end;

--4)Find the product which has more number of restock and with acknowledgement 'Yes'

Declare

   cursor cou is (SELECT * from (SELECT * from prod natural join requests natural join (select rid from
requests natural join restock group by rid order by (rid)  desc)));

Begin
```

```
   dbms_output.put_line(' the product which has more number of restock and with acknowledgement Yes');
   for i in cou
   loop
      if(i.ack='YES') then

         dbms_output.put_line(i.pname||i.rid);
      end if;
   end loop;
end;
--5)List top 5 products name,id should be granted by admin to the store such that product should be supplied by more supplier
Begin
    dbms_output.put_line('Top 5 products granted by admin to the store');
   for i in (SELECT pname,pid from prod natural join (select pid from supplies group by pid order by count(pid) desc) where rownum<6)
   loop
       dbms_output.put_line(i.pname||i.pid);
   end loop;
end;

-- 6)most efficient admin
select aid,afname,alname from admin where
aid in(select aid from employee having count(*)=(select max(count(*)) from employee group by aid )group by aid)
and
aid in(select aid from purchaseorder having count(*)=(select max(count(*)) from purchaseorder group by aid )group by aid)

--7)maximum restocks needed for each product available
select p.pid,r.rid,p.avail,r.rquantity,ceil(p.avail/r.rquantity) from prod p,restock r,requests
where p.pid=requests.pid and r.rid=requests.rid and requests.ack='YES'
```

```
--8)finding the number of suppliers each admin monitors

select aid,count(aid) from(select * from admin natural join supplier) group by aid order by count(aid) desc
```

```
--9)finding the contribution of each product to the total return


declare

aver int;

cursor cp is select avail*price t from prod;

begin

dbms_output.put_line('Finding contribution of each product');

select sum(avail*price) into aver from prod;

for i in cp

loop

dbms_output.put_line(i.t||'        '||100*(i.t/aver));

end loop;

end;
```

```
--10)how many products does each supplier supplies

select s.sid,count(s.sitem) from supplier s,supplies p,prod pr where

s.sid=p.sid

and

pr.pid=p.pid

group by s.sid
```

```
--11)Find the best employee of the year

Begin

   dbms_output.put_line('The best employee of the Year is ');

   for i in (SELECT * from (SELECT * from employee natural join admin natural join (select aid from admin natural join purchaseorder group by aid order by (aid)  desc)) where rownum=1)

   loop
```

```
         dbms_output.put_line(i.ename||i.eid);
    end loop;
end;
--12) choose one employee as admin
Begin
     dbms_output.put_line('The employee choose to be admin is ');
    for i in (select * from employee natural join admin where eexp = (select max(eexp) from
employee) )
    loop
        Update admin set afname = i.ename, aid = i.eid where aid = i.eid;

         dbms_output.put_line(i.afname||i.aid);
    end loop;
end;
--13)Find the amount of expensive product brought by customer
set serveroutput on;
Begin
    for i in (select * from purchaseorder natural join customerr  where pototal = (Select max(pototal)
from purchaseorder))
    loop
         dbms_output.put_line('Customer who bought expensive product is '||i.cfname||'and price is
'||i.pototal);
    end loop;
end;

--14)Increase the salary of employee who have experience greater than 3
create or replace procedure Sal(amt in int,p out int)
is
begin
    for i in (select * from employee where eexp =6)
    loop
        p:=i.esal+amt;
    end loop;
end;

Declare
    c int;
Begin
    sal(5000,c);
    dbms_output.put_line('The updated salary is '||c);
end;

--15)Expection if customer id is zero or not
DECLARE
   c_cid customerr.cid%type :=&cid;
   c_fname customerr.cfname%type;
   c_lname customerr.clname%type;
   c_state customerr.state%type;
   c_city customerr.city%type;
   c_streetno customerr.streetno%type;
   -- user defined exception
   ex_invalid_id  EXCEPTION;
BEGIN
   IF c_cid <= 0 THEN
      RAISE ex_invalid_id;
   ELSE
      SELECT  cfname,clname,state,city,streetno INTO  c_fname,c_lname,c_state,c_city,c_streetno
      FROM customerr
      WHERE cid = c_cid;
      DBMS_OUTPUT.PUT_LINE ('First Name: '||  c_fname);
      DBMS_OUTPUT.PUT_LINE ('Last Name: '||  c_lname);
      DBMS_OUTPUT.PUT_LINE ('State: '||  c_state);
      DBMS_OUTPUT.PUT_LINE ('City: ' || c_city);
      DBMS_OUTPUT.PUT_LINE ('Steert no: ' || c_streetno);
   END IF;

    EXCEPTION
        WHEN ex_invalid_id THEN
            dbms_output.put_line('CID must be greater than zero!');
        WHEN no_data_found THEN
            dbms_output.put_line('No such customer!');
        WHEN others THEN
            dbms_output.put_line('Error!');
END;
--16)acknowledging availability
update requests set ack='YES' where pid='&pid' and rid='&rid' and ack='NO'

--17)total purchase by customers from each city
select c.city,sum(p.pototal) from customerr c,purchaseorder p where c.cid=p.cid group by c.city
order by sum(p.pototal)

--18)calculating loss if product is given at 10% discount
declare
    apr int;
    dpr int;
    loss int;
begin
    select sum(price) into apr from prod ;
    select sum(0.9*price) into dpr from prod;
    loss:=round(apr-dpr);
    dbms_output.put_line('Loss : '||loss);
end;

--19)if admins distribute their salary to employees under them,check whether profit or loss occurs
```

```
--considering actual income to be 1000*sum of purchase_order_total
declare
   asa int;
   tic int;
   chk int;
begin
   select 1000*sum(pototal) into tic from purchaseorder ;
   select sum(asal) into asa from admin;
   chk:=tic-asa;
   if chk>0 then
      dbms_output.put_line('Profit');
   elsif chk<0 then
      dbms_output.put_line('Loss');
   end if;
end;

--20)finding number of purchases more than the average amount

declare
   cursor c is (select sum(pototal) as t from purchaseorder group by cid);
   av int;
   cnt int:=0;
begin
   select avg(pototal) into av from purchaseorder;
   for i in c
   loop
      if i.t>av then
         cnt:=cnt+1;
      end if;
   end loop;
   dbms_output.put_line(cnt||' purchases');
```

```
end;

-------------------------------------- Trigger in PLSQL ----------------------------------------------------------------------
------
--1) CREATING TRIGGER FOR PRODUCT TO BE RESTOCKED , CANT BE RESTOCKED IF
ACKNOWLEDGEMENT IS NO
create or replace trigger Req
before insert
on requests
for each row
begin
   if :new.ack = 'NO' then
      raise_application_error(-20001,'The product cant be restocked for acknowledgement is NO ');
   end if;
end;

insert into requests values('P17','R15','NO');

--2)Trigger to check the employee salary more than 5000
create or replace trigger Esal
before insert
on employee
for each row

begin
   if :new.esal<5000 then
      raise_application_error(-20001,'The Salary cant be less than 5000');
   end if;
end;
insert into employee values ('A6','E16','Harsha','Harsha@gmail.com','23-JUL-2002',4000,2);
-------------------------------------------------Procedure in PLSQL--------------------------------------------------------
-----------
```

```
--3)Update employee salary by experience
create or replace procedure empsalary is
   cursor i is select * from employee;
   t i%ROWTYPE;
Begin
   open i;
   loop
      fetch i into t;
      exit when i%notfound;
      if(t.eexp>5 and t.eexp<7) then
         update employee set esal = esal*0.25 +esal where eid=t.eid;
      elsif(t.eexp>3 and t.eexp<5) then
         update employee set esal = esal*0.15 +esal where eid=t.eid;
      end if;
   end loop;
   close i;
end;

execute empsalary;
```