1.Write PL/SQL block which will give a raise in salary to employees as per the following:

If sal+comm<5000	**increase by 10% of sal**
If sal+comm>=5000	**increase by 500 + 12% of above 7000.**

ANS:-

```
declare
 cursor c1 is
   select ename,sal,comm from emp;
 newsal number(4);
begin

 for c2 in c1
  loop
   if c2.sal+c2.comm<5000 then
    newsal:=c2.sal+c2.sal*0.10;
   elsif c2.sal+c2.comm>3000 then
    newsal:=c2.sal+500;
   else
    newsal:=c2.sal+500+c2.sal*0.12;
   end if;
   dbms_output.put_line(c2.ename||' '||c2.sal||' '||c2.comm||' '||newsal);
  end loop;
end;
```

output:-

```
KING            5000               6100
BLAKE           2850               3692
CLARK           2450               3244
JONES           2975               3832
SCOTT           3000               3860
FORD            3000               3860
SMITH           800                1396
ALLEN           1600        300    1760
WARD            1250        500    1375
MARTIN          1250        1400   1375
TURNER          1500        0      1650
ADAMS           1100               1732
JAMES           950                1564
MILLER          1300               1956

Statement processed.
```

2.Suppose you have created the following three tables in your database:

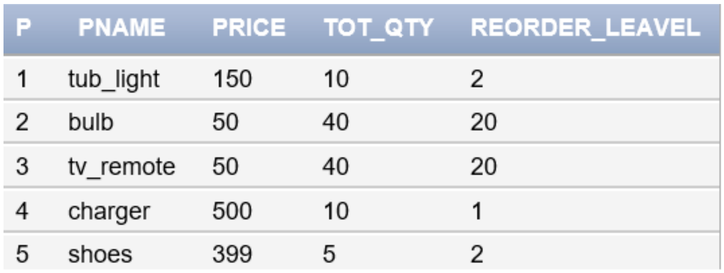
PART_MASTER(<u>P#</u>,PNAME,PRICE,TOT_QTY,REORDER_LEVEL)
PART_TRANS(<u>O#,</u>P#,QTY_REQ),where O# is the order no.
TEMP(P#,QTY)

Write a PL/SQL block which will read records one by one from the transaction table and will check whether after issuing this part, the total quantity of this part is going below the reorder level in the

master table or not. If it is then that part is not to be issued, hence give appropriate message and enter the details of this part in TEMP table. If the total quantity after issuing that part is not going below the reorder level then modify the master table accordingly.

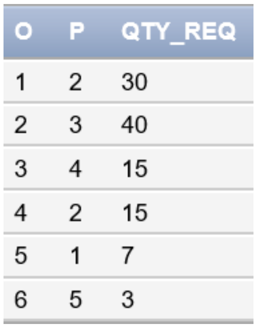
Ans:-

```
CREATE TABLE PART_MASTER(P NUMBER(2) PRIMARY KEY,PNAME VARCHAR2(30),PRICE
NUMBER(3),TOT_QTY NUMBER(3),REORDER_LEAVEL NUMBER(3));

insert into part_master values(1,'tub_light',150,10,2);
insert into part_master values(2,'bulb',50,40,20);
insert into part_master values(3,'tv_remote',50,40,20);
insert into part_master values(4,'charger',500,10,1);
insert into part_master values(5,'shoes',399,5,2);
```



P	PNAME	PRICE	TOT_QTY	REORDER_LEAVEL
1	tub_light	150	10	2
2	bulb	50	40	20
3	tv_remote	50	40	20
4	charger	500	10	1
5	shoes	399	5	2

```
CREATE TABLE PART_TRANS(O NUMBER(2),P NUMBER(2) ,QTY_REQ NUMBER(2),PRIMARY KEY
(P,O),FOREIGN KEY (P) REFERENCES PART_MASTER(P));

insert into part_trans values(1,2,30);
 insert into part_trans values(2,3,40);
 insert into part_trans values(3,4,15);
 insert into part_trans values(4,2,15);
 insert into part_trans values(5,1,7);
 insert into part_trans values(6,5,3);
```



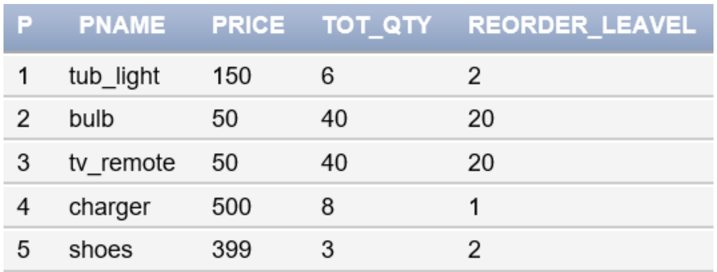
O	P	QTY_REQ
1	2	30
2	3	40
3	4	15
4	2	15
5	1	7
6	5	3

```
CREATE TABLE TEMP(P NUMBER(2),QTY_REQ NUMBER (2),FOREIGN KEY (P) REFERENCES
PART_MASTER(P));
```

Plsql:-

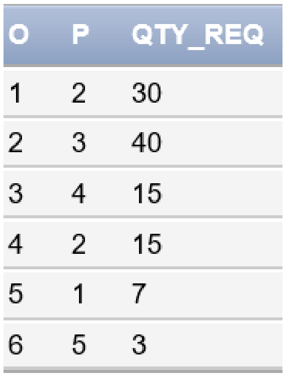
```
declare
        p number(2);
        tq number(3);
        ro number(3);
        qr number(2);
        n number(2);
        cursor c is select a.p,a.tot_qty,a.REORDER_LEAVEL,b.qty_req from part_master a,part_trans b
where a.p=b.p;
begin
        open c;
        loop
                fetch c into p,tq,ro,qr;
                exit when c%notfound;
                n:=tq-ro;
                if n<0 then
                        dbms_output.put_line(' we dont have that much items');
                        insert into temp values(p,qr);
                else
                        if n<ro then
                                dbms_output.put_line(' we are not able to provide you this no. of parts
');
                                insert into temp values(p,qr);
                        end if;
                        if n>ro then
                                update part_master set tot_qty=n where p=part_master.p;
                        end if;
                end if;
        end loop;
        close c;
end;
```

master_table after execution:-



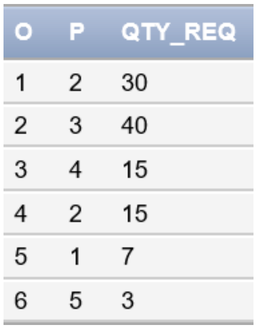
P	PNAME	PRICE	TOT_QTY	REORDER_LEAVEL
1	tub_light	150	6	2
2	bulb	50	40	20
3	tv_remote	50	40	20
4	charger	500	8	1
5	shoes	399	3	2

Trans table after execution:-



O	P	QTY_REQ
1	2	30
2	3	40
3	4	15
4	2	15
5	1	7
6	5	3

Temp table after execution:-



O	P	QTY_REQ
1	2	30
2	3	40
3	4	15
4	2	15
5	1	7
6	5	3

3.Given a table ISSUE(ROLLNO,BOOKNO,ISSUE_DATE,RETURN_DATE), check which students have to pay fine and the amount of fine to be paid. A Student can keep the book for fifteen days. If the number of days has exceeded 15 then fine is calculated as follow:

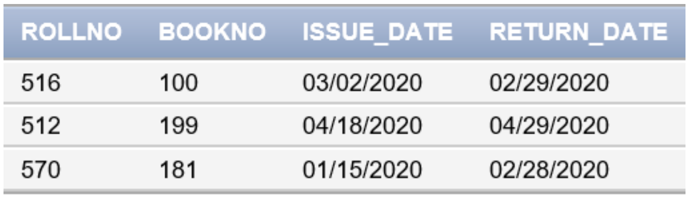
Upto 7 days	**50 paise per day**
8-15	**Rs.1 per day from 8th day onwards**
More than 15 days	**Rs. 1.5per day from 16th day onwards**

After calculating the fine store the required information in the FINE table so that a report can later on printed out.

Ans:-

```
create table issue(rollno number(3),bookno number(3) primary key,issue_date date,return_date date);
```

```
insert into issue values(570,181,'01/15/2020','02/28/2020');
insert into issue values(512,199,'04/18/2020','04/29/2020');
insert into issue values(516,100,'03/02/2020','02/29/2020');
```



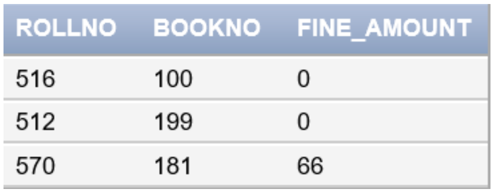
ROLLNO	BOOKNO	ISSUE_DATE	RETURN_DATE
516	100	03/02/2020	02/29/2020
512	199	04/18/2020	04/29/2020
570	181	01/15/2020	02/28/2020

```
create table fine(rollno number(3) ,bookno number(3) ,fine_amount number(4,2),foreign key(bookno) references issue(bookno));
```

PLSQL:-

```
declare
 rollno number(3);
 bookno number(3);
 issue_date date;
 return_date date;
 fine number(4,2);
 days_up number(2);
 cursor c2 is select rollno,bookno,issue_date,return_date from issue;
begin
 open c1;
 loop
  fetch c1 into rollno,bookno,issue_date,return_date;
  exit when c2%notfound;
  fine:=0;
  days_up:=return_date-issue_date;
  if days_up>15 and days_up<22 then
   fine:=days_up*0.5;
  else if days_up>15 and days_up<30 then
   fine:=days_up;
  else if days_up>15 and days_up>30 then
   fine:=days_up*1.5;
  else
   fine:=0;
  end if;
 insert into fine values(rollno,bookno,fine);
 endloop;
 close c2;
end;
```

FINE TABLE AFTER EXECUTION:-



ROLLNO	BOOKNO	FINE_AMOUNT
516	100	0
512	199	0
570	181	66