

SQL> create table book

2 ( Bookcode varchar(1) not null check(bookcode in('A','B','C',’D’)),

3 Author varchar2(20) not null,

4 Title varchar2(30) not null,

5 Category varchar2(10) not null,

6 Price number(8,2) not null);

Table created.

SQL> select \* from book;

Bookcode AUTHOR TITLE CATEGORY PRICE

- -------- ------------ ------------------------------ ---------- ----------

A Tolstoy War and peace Novel 500

A Macmillan Info technoogy Technology 1500

B Sarika gupta E-commerce Commerce 800

B Albert Einstein Relativity science 900

C Beatrice sparks Go ask alice song 450

C Vishwanathan anand Mind master sports 650

D Rashmi Touch the sky others 200

7 rows selected.

set serveroutput on

declare

cursor book\_cur is select bookcode,author,title,category,price from book\_details;

code book\_details.bookcode%type;

auth book\_details.author%type;

tit book\_details.title%type;

cate book\_details.category%type;

rate book\_details.price%type;

sellprice book\_details.price%type;

disamt number(10,2);

discount varchar2(10);

pr number(10,3);

begin

dbms\_output.put\_line('--------------------------------------------------------------------------------------------------------');

dbms\_output.put\_line('Book code Category Title Author price Discount(%) dis\_amt selling price');

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='Novel') 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;

sellprice:=rate-discount;

dbms\_output.put\_line(code||' '||cate||' '||tit||' '||auth||' '||rate||' '||pr\*100||' '||discount||' '||sellprice||' ');

end loop;

close book\_cur;

end;

/

SQL> @C:/Users/User/Documents/f1.sql;

---------------------------------------------------------------------------------------------------------

Book code Category Title Author Discount dis\_amt selling price

---------------------------------------------------------------------------------------------------------

A Novel War and peace Tolstoy 10% 10 450

A Technology Info technoogy Macmillan 12.5% 12.5 1312.5

B Commerce E-commerce Sarika gupta 18% 18 656

B science Relativity Albert Einstein 19% 19 729

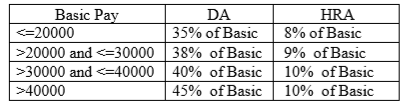
C song Go ask alice Beatrice sparks 25% 25 337.5

C sports Mind master Vishwanathan anand 24% 24 494

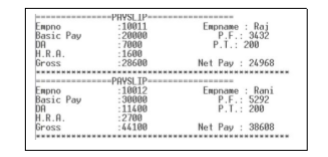
D others Touch the sky Rashmi 28% 28 144

PL/SQL procedure successfully completed.

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 columns empnum, empname, basic pay. Insert 3 records. Allowances are computed as follows:



Gross = Basic + DA + HRA; PF = 12% of Gross or Rs. 2000, whichever is minimum. PT = Rs. 100 upto Gross is 25,000; else Rs. 200, Net = Gross – (PF +PT) Print Pay slip as follows:



create table employe

( empno varchar2(14) primary key,

ename varchar2(20)not null,

basic number(10,2) not null

);

SQL> select \* from employe;

SQL> select \* from employe;

EMPNO ENAME BASIC

-------------- -------------------- ----------

E1001 Shubhada 20000

E1002 Kusuma 30000

E1003 Mamata 35000

E1004 Ravi 45000

E1005 Anusha 15000

E1006 Shanatha 5000

6 rows selected.

set serveroutput on

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;

if bs<=20000 then

da:=bs\*0.35;

hra:=bs\*0.08;

elsif(bs>20000 and bs<=30000) then

da:=bs\*0.38;

hra:=bs\*0.09;

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;

if pf>=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('Basic pay :'||bs||' P.F :'||pf);

dbms\_output.put\_line('DA :'||da||' P.T :'||pt);

dbms\_output.put\_line('H.R.A :'||hra);

dbms\_output.put\_line('Gross :'||gross);

dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

end loop;

close emp\_cur;

end;

/

SQL> @C:/Users/User/Documents/emp1.sql;

===============================PAY SLIP==============================================

Emp no :E1001 Emp name :Shubhada

Basic pay :20000 P.F :2000

DA :7000 P.T :200

H.R.A :16000

Gross :43000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

==============================PAY SLIP==============================================

Emp no :E1002 Emp name :Kusuma

Basic pay :30000 P.F :2000

DA :11400 P.T :200

H.R.A :27000

Gross :68400

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

===============================PAY SLIP==============================================

Emp no :E1003 Emp name :Mamata

Basic pay :35000 P.F :2000

DA :14000 P.T :200

H.R.A :3500

Gross :52500

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

===========================PAY SLIP==============================================

Emp no :E1004 Emp name :Ravi

Basic pay :45000 P.F :2000

DA :20250 P.T :200

H.R.A :4500

Gross :69750

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

===============================PAY SLIP==============================================

Emp no :E1005 Emp name :Anusha

Basic pay :15000 P.F :2000

DA :5250 P.T :200

H.R.A :12000

Gross :32250

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

============================PAY SLIP==============================================

Emp no :E1006 Emp name :Shanatha

Basic pay :5000 P.F :1290

DA :1750 P.T :100

H.R.A :4000

Gross :10750

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PL/SQL procedure successfully completed.

Consider the following tables: LIBRARY (Accession no, Title, Author, Publication, Status). Status can be A for available and I for Issued. Insert 3 records with status ‘A’ for all initially. ISSUE (Rollno, Accession no, Borrowdate, returndate). OUTDATED (Accession no, Title, Author, Publication, tdate).

Write the following Trigger programs.

i. Whenever the book is to be issued, insert a new record to ISSUE without having return date. When the record is inserted to ISSUE table, trigger TRIG\_ISSUE to be executed to update status in LIBRARY as ‘I’.

ii. Whenever book is returned, update return date of that record as todays date in ISSUE table. When the record is updated to ISSUE table, trigger TRIG\_ISSUE to be executed to update status in LIBRARY as ‘A’.

iii. Whenever the book is deleted by accepting Accession no. for status ‘A’ (at SQL >), trigger TRIG\_OUTDATE has to be executed to insert a record to OUTDATED.

Write a PL/SQL program to accept Rollno, Accession no. and transaction (B for Borrow and R for Return). Check for the existence of a given Accession no. and proceed as follows.

 If does not exist , display the message ‘Given accession no. is not available’

 If exist and transaction is B, check the status as ‘A’, then insert to ISSUE, and display the message with accno, author, title, publication and roll no to whom it is issued .

 If exist and transaction is R, then update return date as current system date in ISSUE by accepting Rollno and Aceession no (for the record having return date empty.)

If searched record is not available, raise the predefined exception.

create table item\_master(

item\_no number(5),

name varchar(10),

stock number(5),

unit\_price number(5),

constraint stock\_price check(stock>0 and unit\_price>0) );

create table item\_trans(

item\_no number(10),

qty number(5),

tdate date);

SQL> select \* from item\_master;

ITEM\_NO NAME STOCK UNIT\_PRICE

---------- ---------- ---------- ----------

1001 keyboard 50 360

1002 mouse 60 300

1003 hard disk 10 5000

create or replace package pck\_item as

function ch\_item( n number ) return number;

procedure proc\_item(n number,q number);

end pck\_item;

/

create or replace package body pck\_item as

function ch\_item(n in number) return number is

ino number(5);

begin

select item\_no into ino from item\_master where item\_no=n;

return 1;

exception when no\_data\_found then

return 0;

end ;

procedure proc\_item(n in number,q in number) is

ino number(5);

stk number(10);

begin

if(pck\_item.ch\_item(n)=1) then

select item\_no,stock into ino,stk from item\_master where item\_no=n;

if(stk<q) then

dbms\_output.put\_line('required stock not available');

else

insert into item\_trans values(ino,q,sysdate);

update item\_master set stock=stock-q where item\_no=n;

end if;

end if;

end;

end pck\_item;

/

set serveroutput on

prompt enter item number;

accept ino;

prompt enter qty;

accept qty;

begin

if(pck\_item.ch\_item(&ino)=1) then

dbms\_output.put\_line('item exists');

else

dbms\_output.put\_line('item does not exists');

end if;

pck\_item.proc\_item(&ino,&qty);

end;

/

SQL> @C:/Users/User/Documents/pck\_item.sql;

Package created.

SQL> @C:/Users/User/Documents/pck\_item\_body.sql;

Package body created.

SQL> @C:/Users/User/Documents/pck\_item\_pls.sql;

enter item number

1001

enter qty

10

old 3: if(pck\_item.ch\_item(&ino)=1) then

new 3: if(pck\_item.ch\_item(1001)=1) then

old 9: pck\_item.proc\_item(&ino,&qty);

new 9: pck\_item.proc\_item(1001,10);

item exists

PL/SQL procedure successfully completed.

SQL> select \* from item\_master;

ITEM\_NO NAME STOCK UNIT\_PRICE

---------- ---------- ---------- ----------

1001 keyboard 40 360

1002 mouse 60 300

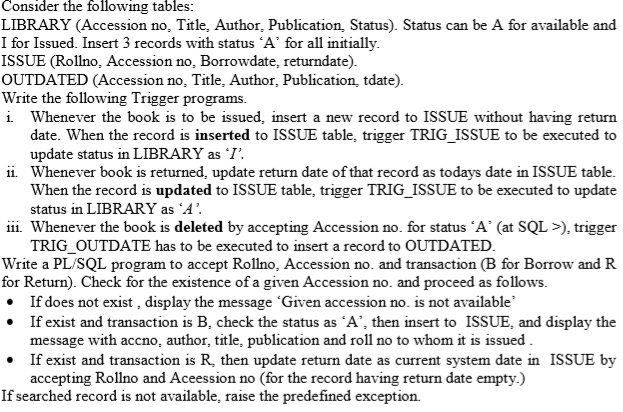
1003 hard disk 10 5000

SQL> select \* from item\_trans;

ITEM\_NO QTY TDATE

---------- ---------- ---------

1001 10 23-FEB-20



create table Library

(Ano number(6) primary key,

Title varchar2(30) not null,

Author varchar2(30) not null,

Publication varchar2(30) not null,

Status char(1) not null );

create table Issue

(Rollno varchar2(6) ,

ano number(6) not null,

borrowdate date,

returndate date,

foreign key (ano) references library(Ano) );

create table Outdated

(Ano number(6) not null,

Title varchar2(30) not null,

Author varchar2(30) not null,

Publication varchar2(30) not null,

tdate date );

SQL> select \* from library;

ANO TITLE AUTHOR PUBLICATION S

---------- ------------------------------ ------------------------------ ------------------------------ -

10001 E-commerce Sarika gupta CA publication A

10002 Mind master Anada SB publication A

10003 Relativity Albert Einstein AB publication A

10004 SQL,PL/SQL Ivan Bayross BPB publication A

SQL> select \* from issue;

no rows selected

SQL> select \* from outdated;

no rows selected

**trigger outdated**

set serveroutput on

create or replace trigger trig\_outdated before delete on library for each row

Begin

insert into outdated (ano,title,author,publication,tdate)values(:old.ano,:old.title,:old.author,:old.publication,sysdate);

end;

/

**trigger issue**

set serveroutput on

create or replace trigger trig\_iss before insert or update on issue for each row

declare

rl issue.rollno%type;

an issue.ano%type;

Begin

an:=:new.ano;

if inserting then

update library set Status='I' where ano=an;

elsif updating then

update library set Status='A' where ano=an;

end if;

end;

/

**Plsql code**

set serveroutput on

prompt enter Roll number;

accept rollno;

prompt enter an accession number;

accept accno;

prompt enter the transaction;

accept trans;

declare

acno library.ano%type;

tit library.title%type;

auth library.author%type;

publi library.publication%type;

stus library.status%type;

begin

select author,title,publication,status into auth,tit,publi,stus from library where ano=&accno;

if sql%found then

if('&trans'='B')then

if stus='A' then

insert into issue values('&rollno',&accno,sysdate,null);

end if;

dbms\_output.put\_line('accno :'||&accno);

dbms\_output.put\_line('author :'||auth);

dbms\_output.put\_line('title :'||tit);

dbms\_output.put\_line('publication :'||publi);

dbms\_output.put\_line('rollno :'||'&rollno');

elsif('&trans'='R') then

update issue set returndate=sysdate where rollno='&rollno';

end if;

end if;

exception when no\_data\_found then

dbms\_output.put\_line('ACC NUMBER DOES NOT EXIST');

end;

/

**Execution method**

SQL> @C:/Users/User/Documents/iss.sql;

Trigger created.

SQL> @C:/Users/User/Documents/isspl.sql;

enter Roll number

R1001

enter an accession number

10001

enter the transaction

B

old 11: acno:=&accno;

new 11: acno:=10001;

old 12: rlno:='&rollno';

new 12: rlno:='R1001';

old 16: transaction:='&trans';

new 16: transaction:='B';

hr

accno :10001

author :Sarika gupta

title :E-commerce

publication :CA publication

rollno :R1001

PL/SQL procedure successfully completed.

SQL> select \* from issue;

ROLLNO ANO BORROWDAT RETURNDAT

------ ---------- --------- ---------

R1001 10001 15-FEB-20

SQL> @C:/Users/User/Documents/isspl.sql;

enter Roll number

R1001

enter an accession number

10001

enter the transaction

R

old 11: acno:=&accno;

new 11: acno:=10001;

old 12: rlno:='&rollno';

new 12: rlno:='R1001';

old 16: transaction:='&trans';

new 16: transaction:='R';

hr

PL/SQL procedure successfully completed.

SQL> select \* from issue;

ROLLNO ANO BORROWDAT RETURNDAT

------ ---------- --------- ---------

R1001 10001 15-FEB-20 15-FEB-20

SQL> @C:/Users/User/Documents/trgout.sql;

Trigger created.

SQL> delete from library where ano=10002;

1 row deleted.

SQL> select \* from outdated;

ANO TITLE AUTHOR PUBLICATION TDATE

---------- ------------------------------ ------------------------------ ------------------------------ ---------

10002 Mind master Anada SB publication 15-FEB-20