PROCEDURA CHE MEMORIZZA ATTRIBUTI DI UNA TABELLA CON UN CURSORE E LI STAMPA

create or replace PROCEDURE STAMPATAB

is

-- dichiaro le variabili

c\_id CHAR(30);

c\_ditta Char(30);

c\_prezzo number;

-- popolo il cursore

CURSOR c\_farmaci is

SELECT codice\_aic, ditta, prezzo FROM DATABASE\_TESI.FARMACI ;

BEGIN

--apro il cursore

OPEN c\_farmaci;

LOOP

-- inserisco il contenuto del cursore in delle variabili

FETCH c\_farmaci into c\_id, c\_ditta, c\_prezzo;

EXIT WHEN c\_farmaci%notfound;

dbms\_output.put\_line(c\_id || ' ' || c\_ditta || ' ' || c\_prezzo);

--IF c\_farmaci%found THEN

--dbms\_output.put\_line(c\_id || ' ' || c\_ditta || ' ' || c\_prezzo);

--ELSE

--EXIT;

--END IF;

-- chiudo il ciclo e il cursore

END LOOP;

CLOSE c\_farmaci;

END STAMPATAB;

PROCEDURA DI CALCOLO DEL RICOVERO PIU’ COSTOSO

--creazione di una procedura che ricava il ricovero più costoso nell'arco di tempo selezionato dall'utente

create or replace procedure Costi(v\_DataI date, v\_DataF date) IS

CURSOR ricoveri\_cursor IS --creo il cursore che immagazzina i dati per ogni paziente nell'arco di tempo selezionato

select R.CODICE\_FISC,R.DATA\_INGRESSO,R.DATA\_USCITA,sum(P.TARIFFA)

from RICOVERI R,RICHIESTAPRESTAZIONI RP,PRESTAZIONI P,VISITE V

where R.CODICE\_FISC=V.CF\_PAZIENTE and V.ID\_VISITA=RP.ID\_VISITA and RP.CODPREST=P.CODICE

and R.DATA\_INGRESSO BETWEEN v\_DataI and v\_DataF and R.DATA\_USCITA BETWEEN v\_DataI and v\_DataF

group by R.CODICE\_FISC,R.DATA\_INGRESSO,R.DATA\_USCITA;

-- dichiaro le variabili

v\_costo float;

v\_durata INT;

v\_DurataMax INT;

v\_max float :=0;

v\_Data date;

v\_CF RICOVERI.CODICE\_FISC%TYPE;

c\_CF RICOVERI.CODICE\_FISC%TYPE;

c\_DataI RICOVERI.DATA\_INGRESSO%TYPE;

c\_DataF RICOVERI.DATA\_USCITA%TYPE;

c\_Costo PRESTAZIONI.TARIFFA%TYPE;

r\_paziente GENERALE%rowtype;

BEGIN

open ricoveri\_cursor; --ciclo il cursone per fare il calcolo del costo

loop

--calcolo per ogni paziente selezionato la funzione di costo sulla base della durata del ricovero

fetch ricoveri\_cursor into c\_CF,c\_DataI,c\_DataF,c\_Costo;

exit when ricoveri\_cursor%NOTFOUND;

v\_Durata:=c\_DataF-C\_DataI;

--DBMS\_OUTPUT.PUT\_LINE(c\_CF || 'DURATA ' || v\_Durata);

if (v\_Durata<7) then

v\_costo:=20 + c\_Costo -2\*(v\_Durata);

--DBMS\_OUTPUT.PUT\_LINE('COSTO ' || v\_costo);

else

v\_costo:=15 + c\_Costo -1.5\*(v\_Durata);

--DBMS\_OUTPUT.PUT\_LINE('COSTO ' || v\_costo);

end if;

if (v\_costo>v\_max) then -- immagazzino il costo maggiore e i dati del paziente

v\_max:=v\_costo;

v\_DurataMax:=v\_Durata;

v\_CF:=c\_CF;

v\_Data:=c\_DataI;

end if;

end loop;

close ricoveri\_cursor;

-- stampo i dati relativi al ricovero

DBMS\_OUTPUT.PUT\_LINE('Il ricovero piu costoso e durato ' || v\_DurataMax || ' giorni, il costo e ' || v\_max || ' in data ' || v\_Data);

-- estraggo i dati relativi al paziente usando il codice fiscale

select \* INTO r\_paziente

from GENERALE G

where G.CODICE\_FISC=c\_CF;

+-- stampo i dati del paziente col ricovero più costoso

DBMS\_OUTPUT.PUT\_LINE('Il ricovero e relativo a ' || r\_paziente.nome || ' ' || r\_paziente.cognome);

END Costi;

PROCEDURA PER IL CALCOLODELL’OPERATORE MIGLIORE E PEGGIORE

Dopo aver creato le seguenti tabelle

create table CLIENTI (

C\_ID int,

C\_NOME varchar2(20),

C\_COGNOME varchar2(20),

C\_DATAN date,

C\_TELEFONO varchar2(50),

C\_TIPOCONTR varchar2(20),

primary key (C\_ID));

create table OPERATORI (

O\_ID int,

O\_NOME varchar2(20),

C\_COGNOME varchar2(20),

O\_DATAN date,

O\_LIVELLO int,

primary key (O\_ID));

create table PROBLEMI (

P\_COD varchar2(16),

P\_DESCRIZIONE varchar2(50),

P\_CLASSE varchar2(20),

P\_DIFF float,

primary key (P\_COD));

create table CHIAMATE(

C\_ID int,

C\_CLIENTE int,

C\_OPERATORE int,

C\_PROBLEMA varchar2(16),

C\_APERTURA date,

C\_RISULTATO int,

primary key (C\_ID),

foreign key (C\_CLIENTE) references CLIENTI(C\_ID),

foreign key (C\_OPERATORE) references OPERATORI(O\_ID),

foreign key (C\_PROBLEMA) references PROBLEMI(P\_COD));

create table OPERAZIONI(

R\_CHIAMATA int,

R\_DATAORA date,

R\_DURATA int,

R\_DESCRIZIONE varchar2(50),

primary key (R\_CHIAMATA,R\_DATAORA),

foreign key (R\_CHIAMATA) references CHIAMATE(C\_ID));

create or replace PROCEDURE RATING\_OPERATORI (v\_Data date) IS

--definisco il cursore

CURSOR cursor\_operatori IS

select distinct C\_OPERATORE

from CHIAMATE

where C\_APERTURA>=v\_Data and C\_RISULTATO!=0;

--definisco le variabili

c\_OP CHIAMATE.C\_OPERATORE%TYPE;

v\_pos float;

v\_neg float;

v\_valutazione float;

v\_max float :=0;

v\_min float :=0;

v\_codiceP RICOVERI.CODICE\_FISC%TYPE;

v\_codiceN RICOVERI.CODICE\_FISC%TYPE;

BEGIN

-- apro il cursore

open cursor\_operatori;

loop

-- per ogni riga del cursore calcolo i parametri della funzione di costo

fetch cursor\_operatori into c\_OP;

exit when cursor\_operatori%NOTFOUND;

v\_pos :=0;

select sum(R\_DURATA\*P\_DIFF) into v\_pos

from CHIAMATE, OPERAZIONI,PROBLEMI

where C\_ID=R\_CHIAMATA and C\_PROBLEMA=P\_COD and C\_RISULTATO=1

and C\_OPERATORE=c\_OP and C\_APERTURA>=v\_Data;

v\_neg :=0;

select sum(R\_DURATA\*P\_DIFF) into v\_neg

from CHIAMATE, OPERAZIONI,PROBLEMI

where C\_ID=R\_CHIAMATA and C\_PROBLEMA=P\_COD and C\_RISULTATO=-1

and C\_OPERATORE=c\_OP and C\_APERTURA>=v\_Data;

-- calcolo la funzione di costo

if (v\_Pos+V\_Neg<>0) then

v\_Valutazione:=(v\_Pos-V\_Neg)/(v\_Pos+V\_Neg);

else

v\_Valutazione:=0;

end if;

-- estrapolo i dati dell'operatore peggiore

if (v\_Valutazione<v\_min) then

v\_min:=v\_Valutazione;

v\_codiceN:=c\_OP;

-- estrapolo i dati dell'operatore migliore

end if;

if (v\_Valutazione>v\_max) then -- immagazzino il costo maggiore

v\_max:=v\_Valutazione;

v\_codiceP:=c\_OP;

end if;

end loop;

close cursor\_operatori;

-- stampo i dati ottenuti

DBMS\_OUTPUT.PUT\_LINE('L''operatore ' || v\_codiceN || ' ha ottenuto il peggior punteggio=' || v\_min);

DBMS\_OUTPUT.PUT\_LINE('L''operatore ' || v\_codiceP || ' ha ottenuto ha ottenuto il miglior punteggio=' || v\_max);

END RATING\_OPERATORI;

FUNZIONE PER IL CALCOLO DELLA DITTA CHE PRODUCE PIU’ FARMACI

create or replace FUNCTION BIGDITTA

RETURN CHAR IS

nomeDitta CHAR(30) := 0;

BEGIN

SELECT DITTA into nomeDitta

FROM DATABASE\_TESI.FARMACI

GROUP BY DITTA

HAVING count(\*)= (select max(COUNT(\*)) FROM DATABASE\_TESI.FARMACI GROUP BY DITTA);

RETURN nomeDitta;

END;

--CHIAMATA DA LINEA DI COMANDO

--DECLARE

-- ditta CHAR(30);

--BEGIN

-- ditta := BIGDITTA();

-- dbms\_output.put\_line('Ditta che fa più farmaci: ' || ditta);

--END;

FUNZIONE CHE CALCOLA IL FARMACO PIU’ COSTOSO

create or replace FUNCTION MAXPRICE

RETURN number IS

maxPrice number(2,1) := 0;

BEGIN

SELECT max(PREZZO) into maxPrice

FROM DATABASE\_TESI.FARMACI;

RETURN maxPrice;

END;

--DECLARE

-- c number(2,1);

--BEGIN

-- c := MAXPRICE();

-- dbms\_output.put\_line('Max price farmaci: ' || c);

--END;

TRIGGER CHE VISUALIZZA UN MESSAGGIO IN CASO DI INSERIMENTO, AGGIORNAMENTO O ELIMINAZIONE DI UNA RIGA DELLA TABELLA

create or replace TRIGGER design\_trig

BEFORE DELETE OR INSERT OR UPDATE ON FARMACI

FOR EACH ROW

BEGIN

if deleting then

dbms\_output.put\_line('Before delete triggers fired');

elsif inserting then

dbms\_output.put\_line('Before insert triggers fired');

elsif updating then

dbms\_output.put\_line('Before update triggers fired');

END IF;

END;

TRIGGER PER VISUALIZZARE L’AGGIORNAMENTO DEL PREZZO DI UN FARMACO: STAMPA IL PREZZO PRECEDENTE, IL PREZZO NUOVO E LA LORO DIFFERENZA

create or replace TRIGGER display\_price\_changes

BEFORE DELETE OR INSERT OR UPDATE ON FARMACI

FOR EACH ROW

WHEN (NEW.codice > 0)

DECLARE

price\_diff number;

BEGIN

price\_diff := :NEW.prezzo - :OLD.prezzo;

dbms\_output.put\_line('Old price: ' || :OLD.prezzo);

dbms\_output.put\_line('New price: ' || :NEW.prezzo);

dbms\_output.put\_line('Price difference: ' || price\_diff);

END;