

ESERCIZIO 1

Il sistema informativo di una palestra deve gestire il programma dei corsi:

SALE (S_IDSala, S_NomeSala, S_Capienza)

ISTRUTTORI (I_IDIstr, I_CodFisc, I_Nome, I_Cognome, I_DataNascita, I_Email, I_Telefono)

CORSI (C_IDCorso, C_DataInizio, C_DataFine, C_Nome, C_Tipo, C_Livello)

PROGRAMMA (P_IDIstr:ISTRUTTORI, P_Giorno, P_Orainizio, P_Durata, P_IDSala:SALE, P_IDCorso:CORSI)

La tabella programma registra lo storico delle lezioni dei corsi su base settimanale: P_Giorno in [1..7], P_Orainizio in [8..20], la durata è espressa in ore.

L'occupazione giornaliera delle aule deve essere estratta considerando i soli corsi attivi in una data (e calcolando a quale giorno della settimana corrisponde la data).

Si usi la seguente funzione per estrarre da una data il numero (1..7) relativo al giorno settimanale: to_number(to_char(vData, 'D'))

Si scriva la procedura **proglstr(vIstruttore, vData)** che stampa il programma settimanale di un istruttore (7 giorni a partire dalla data indicata). Non è necessario stampare nulla nei giorni liberi.

Login: SID: ESAMESI NOMEHOST:si-oracle-11.csr.unibo.it

Esempio

proglstr(3, '14-APR-21')

Giorno: Lun

Dalle: 10 Alle: 11 in Aula3

Giorno: Mer

Dalle: 10 Alle: 11 in Aula3

Giorno: Ven

Dalle: 10 Alle: 11 in Aula3

SOLUZIONE 1

```
create or replace procedure
```

```
A20210615_proglstr(vIstruttore number, vDataI date) is
```

```
cursor cProg is
```

```
select *
```

```
from A20210615_programma join A20210615_corsi on (p_idcorso = c_idcorso)
```

```
where c_dataInizio<=vDataI+7 and c_datafine>vDataI
```

```
and P_IDIstr=vIstruttore
```

```
order by P_giorno-TO_NUMBER(to_char(vDataI, 'D')), P_oraInizio;
```

```
V_FLAG NUMBER := 0;
```

```
vGiorno number;
```

```
type array_t is varray(7) of varchar2(10);
```

```

dayArray array_t := array_t('Lun', 'Mar', 'Mer','Gio','Ven','Sab','Dom');

begin
  vGiorno:=-10;
  for vPro in cProg
    LOOP
      V_FLAG := 1;
      if (vGiorno != vPro.P_giorno) then
        dbms_output.put_line('Giorno: ' ||
dayArray(vPro.P_giorno) );
        vGiorno :=vPro.P_giorno;
      end if;
      dbms_output.put_line(' Da: ' || vPro.P_OraInizio || ' A: ' ||
(vPro.P_oraInizio+ vPro.P_Durata) || ' in ' || vPro.P_IDSala );

      END LOOP;

  end;

```

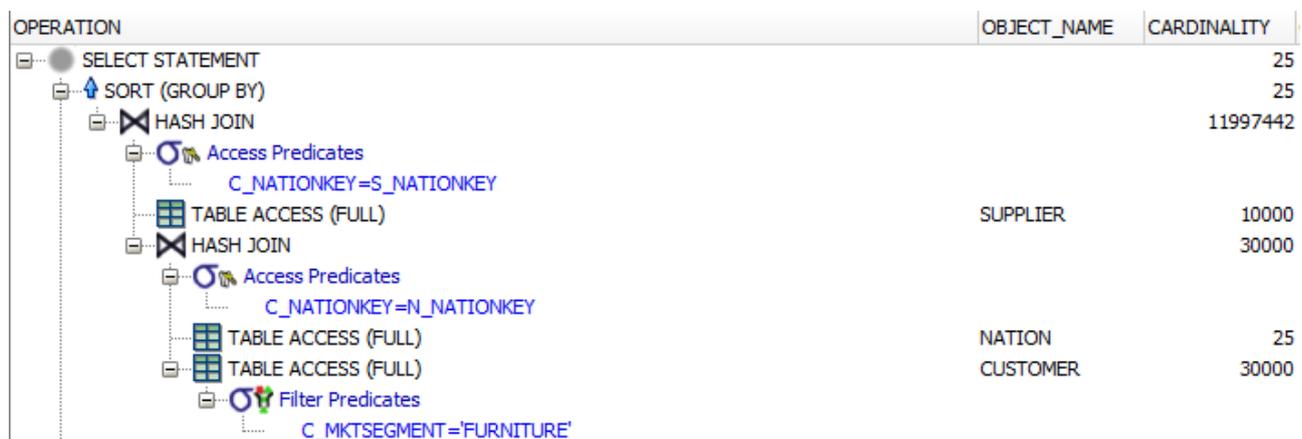
ESERCIZIO 2

Utilizzando il database TPCD si **disegni l'albero** di esecuzione proposto da ORACLE e si **calcoli il costo di accesso** della seguente query:

```

select n_name, count(*)
from TPCD.CUSTOMER,TPCD.NATION,TPCD.SUPPLIER
where C_NATIONKEY=S_NATIONKEY and C_NATIONKEY=N_NATIONKEY and C_MKTSEGMENT=
'FURNITURE'
group by N_NAME;

```



Si facciano le seguenti assunzioni e si estraggano dal DB eventuali dati mancanti:

$D = 4096$ byte $\text{len}(P) = \text{len}(K) = 4$ byte $NB = 101$ $u = 0.69$

Si assuma inoltre che ORACLE non applichi proiezioni sui risultati intermedi e che non esegua operazioni in pipeline.

Login: USER: USERSI PASSWORD: usersi SID: TPCD NOMEHOST:si-oracle-11.csr.unibo.it

ESERCIZIO 3

Data la collezione [Movies](#) scrivere le seguenti **query**:

- a) Aggiorna il titolo di uscita di "Doktor M.H. - Kes on Marie Johansson" a "Doktor M.H. - Kes on Marie Johanssons" .
- b) Mostrare il titolo di tutti i film in cui recitano (almeno uno) Angelina Jolie e Tom Cruise
- c) Trovare i 10 film con rating IMDB più alto del 2014 (titolo, attori e rating)

Consegnare le query in un file Es3<cognome>.js

SOLUZIONE 3

```
db.getCollection('Movies').update({"title" : "Doktor M.H. - Kes on Marie Johansson"},{ $set: {"title" : "Doktor M.H. - Kes on Marie Johanssons"}})
```

```
db.getCollection("Movies").aggregate( [
  { "$match" : { "actors" : { $in: ["Angelina Jolie", "Tom Cruise"] } } },
  { "$project" : { "title" : 1.0, "_id" : 0.0 } }
]);
```

```
db.getCollection("Movies").aggregate( [
  { "year" : 2014.0 },
  { "$sort" : { "imdb.rating" : -1.0 } },
  { "$project" : { "_id" : 0, "title" : 1, "actors" : 1, "imdb.rating" : 1
}}
  { "$limit" : 10.0 }
]);
```