

Sono possibili più soluzioni, se ne propone una con utilizzo di un cursore:

```
(p_CF char, p_obiettivo varchar2, p_gruppomuscolare varchar2) is
  infoutente utenti%ROWTYPE;
  infoperf performance%ROWTYPE;
  infoeser esercizi%ROWTYPE;

cursor perfor is
  select performance, eta, peso, e.nome
  from performancep, esercizi e, utenti u
  where esercizio=e.nome and gruppomuscolare = p_gruppomuscolare
    and obiettivo=p_obiettivo and CF=p_CF
    and eta BETWEEN etal and eta2
    and eta > etaminima
  order by performance desc;

cmt c_perfor%ROWTYPE;

begin

  open c_perfor();
  fetch c_perfor into cmt;
  if c_main%FOUND then
    declare
      bonta float;
      nserie integer;
    begin
      if p_obiettivo='Forza' then
        bonta := (cmt.performance * cmt.peso) / cmt.eta;
        nserie := ceil(bonta/3);
        for i in 1..nserie
          loop
            insert into serie values(p_CF, cmt.nome, i,
              ceil(cmt.ripetizionimedie-i*2), ceil(cmt.pesomedio+2*bonta));
          end loop;
        elsif p_obiettivo='Resistenza' then
          bonta := (cmt.performance * cmt.peso) / (cmt.eta * 2);
          nserie := ceil(bonta/3);
          for i in 1..nserie
            loop
              insert into serie values(p_CF, cmt.nome, i,
                ceil(cmt.ripetizionimedie+i*2), ceil(cmt.pesomedio-2*bonta));
            end loop;
          end if;
        end;
      else
        DBMS_OUTPUT.PUT_LINE('nessun esercizio trovato che soddisfi i
requisiti!');
      end if;
      close c_main;
    end;
  end;
```

Si noti che:

- la massimizzazione della bontà è indipendente da peso ed età che sono fissati dall'utente
- Il cursore può restituire molte tuple ma viene recuperate solo la prima
- Tutte le condizioni possono essere inserite nel cursore se si accede anche alla tabella utenti per la quale NON viene utilizzato un equi-join

Si poteva evitare l'uso del cursore assumendo che tutti i valori di performance siano diversi fissato l'obiettivo e la fascia di età. In questo caso era sufficiente una query innestata per recuperare le informazioni di interesse.

```
explain plan for select R_NAME, count(*) AS NCUST
from REGION,NATION,CUSTOMER
where R_REGIONKEY=N_REGIONKEY AND N_NATIONKEY=C_NATIONKEY
GROUP BY R_NAME,R_REGIONKEY
ORDER BY NCUST;
@?/RDBMS/ADMIN/UTLXPLS
```

Plan Table

Operation	Name	Rows	Bytes	Cost	Pstart	Pstop
SELECT STATEMENT		18	684	15785		
SORT ORDER BY		18	684	15785		
SORT GROUP BY		18	684	15785		
HASH JOIN		150K	5M	1094		
HASH JOIN		25	875	3		
TABLE ACCESS FULL	REGION	5	145	1		
TABLE ACCESS FULL	NATION	25	150	1		
TABLE ACCESS FULL	CUSTOMER	150K	439K	1059		

$$NP_{REGION} = \lceil 5 \times 113 / (4096 \times 0,69) \rceil = 1$$

$$NP_{NATION} = \lceil 25 \times 105 / (4096 \times 0,69) \rceil = 1$$

$$NP_{REGION+NATION} = \lceil 25 \times (113+105) / (4096 \times 0,69) \rceil = 2$$

$$NP_{CUSTOMER} = \lceil 150.000 \times 158 / (4096 \times 0,69) \rceil = 8.386$$

$$NP_{REGION+NATION+CUSTOMER} = \lceil 150.000 \times (113+105+158) / (4096 \times 0,69) \rceil = 19.956$$

$$\text{Costo Group by} = 2 \times 19.956 \times (\lceil \log_{100} \lceil 19.956 / 101 \rceil \rceil + 1) = 2 \times 19.956 \times (2 + 1) = 119.736$$

$$NP_{RISULTATO} = \lceil 5 \times 29 / (4096 \times 0,69) \rceil = 1$$

Sort del risultato = 2

Per calcolare il costo totale è necessario ipotizzare se il sistema utilizza o meno il pipeline

Senza pipeline il costo deve considerare anche le scritture su disco dei risultati intermedi che nel calcolo sottostante sono sottolineate:

$$\text{Costo totale } 1+1+\underline{2}+2+8.386+\underline{19.956}+119.736+2 = 148.086$$