

```
CREATE TABLE NAZIONALI(  
Nome varchar2(20),  
Allenatore varchar2(50),  
Girone char(1),  
primary key (Nome));
```

```
CREATE TABLE CALCIATORI(  
Cod varchar2(20),  
Nome varchar2(20),  
Cognome varchar2(20),  
Ruolo varchar2(10),  
Nazione varchar2(20),  
primary key (Cod),  
foreign key (Nazione) references NAZIONALI(Nome));
```

```
CREATE TABLE PARTITE(  
SquadraA varchar2(20),  
SquadraB varchar2(20),  
Fase varchar2(10),  
Data date,  
GoalsqA integer,  
GoalsqB integer,  
primary key (SquadraA,SquadraB,Fase),  
foreign key (SquadraA) references NAZIONALI(Nome),  
foreign key (SquadraB) references NAZIONALI(Nome));
```

```
CREATE TABLE FORMAZIONI(  
Squadra varchar2(20),  
Avversari varchar2(20),  
Fase varchar2(10),  
Calciatore varchar2(10),  
MinutiGiocati integer,  
Voto number,  
GoalSegnati number,  
primary key (Squadra,Avversari,Fase),  
foreign key (Squadra) references NAZIONALI(Nome),  
foreign key (Avversari) references NAZIONALI(Nome),  
foreign key (Calciatore) references CALCIATORI(Cod));
```

```

CREATE OR REPLACE PROCEDURE BestPlayer(v_Ruolo CALCIATORI.Ruolo%TYPE) IS

CURSOR crs_CalciatoriInRuolo IS
    SELECT Cod,SUM(Voto*MinutiGiocati) as VotiMinuti, sum(MinutiGiocati) as
        Minuti
    FROM CALCIATORI,FORMAZIONI
    WHERE Ruolo=v_ruolo AND Cod=Calciatore
GROUP BY Cod;

v_CalciatoriInRuolo crs_CalciatoriInRuolo%ROWTYPE;
v_bestplayer CALCIATORI.Cod%TYPE;
v_goal integer;
v_votofinale number;
v_votomax integer;

BEGIN

v_votomax:=0;
OPEN crs_CalciatoriInRuolo;

loop
    fetch crs_CalciatoriInRuolo into v_CalciatoriInRuolo;
    EXIT WHEN crs_CalciatoriInRuolo%NOTFOUND;

    SELECT SUM(GoalSegnati) INTO v_goal
    FROM FORMAZIONI F,PARTITE P
    WHERE Calciatore=v_CalciatoriInRuolo.Cod
        AND (Squadra=SquadraA AND Avversari=SquadraB AND F.Fase=P.Fase
            AND GoalSqA >= GoalSqB)
        OR (Squadra=SquadraB AND Avversari=SquadraA AND F.Fase=P.Fase
            AND GoalSqB >= GoalSqA);

    v_votofinale := v_CalciatoriInRuolo.VotiMinuti / v_CalciatoriInRuolo.Minuti
        + 0.2*v_goal;
    if (v_votofinale>v_votomax) then
        v_votomax:=v_votofinale;
        v_bestplayer:=v_CalciatoriInRuolo.Cod;
    end if;
end loop;
close crs_CalciatoriInRuolo;

    DBMS_OUTPUT.PUT_LINE('Il miglior giocatore nel ruolo: ' || v_ruolo || ' e': '
|| v_bestplayer);
end;

```

```

explain plan for
select P_NAME, sum(L_QUANTITY)
from PART,LINEITEM
where P_PARTKEY=L_PARTKEY and P_NAME ='dodger orchid puff peru mint'
group by P_PARTKEY,P_NAME;

```

```
@?/RDBMS/ADMIN/UTLXPLS;
```

Plan Table

Operation	Name	Rows	Bytes	Cost	Pstart	Pstop
SELECT STATEMENT		1	44	1228		
SORT GROUP BY		1	44	1228		
NESTED LOOPS		60	2K	1226		
TABLE ACCESS FULL	PART	2	72	1162		
TABLE ACCESS BY INDEX R	LINEITEM	6M	45M	32		
INDEX RANGE SCAN	IX_PART_LI	6M		2		

$$NP_{PART} = \lceil 200.000 \times 133 / (4096 \times 0,69) \rceil = 9.412$$

$$NP_{LI} = \lceil 6.001.215 \times 113 / (4096 \times 0,69) \rceil = 239.944$$

$SEL(P_NAME=' \dots ') = 1 / 199.995$ poteva essere usto anche $1/200.000$ assumendo che tutte la parti avessero nomi diversi

$$NL_{LI(P_PARTKEY)} = \lceil (6.001.215 \times 4 + 200.000 \times 4) / (4096 \times 0,69) \rceil = 8.777$$

$$\begin{aligned} \text{Costo di accesso a LI} &= 2 - 1 + \lceil (1 / 199.995 \times 8.777) + 1 \times \Phi(6.001.215 / 199.995, 239.944) \rceil \\ &= 1 + 1 + \Phi(30, 239.944) = 32 \end{aligned}$$

Il nested loop viene implementato accedendo alle sole 30 tuple di LINEITEM che sono in relazione con la tupla di part selezionata.

$$NP_{LI+PART} = \lceil 30 \times 246 / (4096 \times 0,69) \rceil = 3$$

Avendo a disposizione 100 buffer l'ordinamento può essere effettuato in memoria centrale, quindi basta leggere e scrivere le pagine

$$\text{Costo finale } 9.412 + 32 + 6 = 9.450$$