```
CREATE TABLE DOCENTI(
D_CODD VARCHAR2(20) NOT NULL,
D_NOME VARCHAR2(20),
D_COGNOME VARCHAR2(20),
D_FASCIA VARCHAR2(20),
D_PUNTEGGIO NUMBER,
D_RAGGRUPPAMENTO VARCHAR2(20),
CONSTRAINT "SYS_C004620" PRIMARY KEY(D_CODD));

CREATE TABLE RIVISTE(
R_NOME VARCHAR2(20) NOT NULL,
R_EDITORE VARCHAR2(20),
R_IMPACTFACTOR NUMBER,
CONSTRAINT "SYS_C004629" PRIMARY KEY(R_NOME));

CREATE TABLE PUBBLICAZIONI(
P_CODP VARCHAR2(20) NOT NULL,
P_TITOLO VARCHAR2(20),
P_RIVISTA VARCHAR2(20),
P_ANNO VARCHAR2(20),
CONSTRAINT "SYS_C004645" PRIMARY KEY(P_CODP),
CONSTRAINT "SYS_C004646" FOREIGN KEY(P_RIVISTA) REFERENCES RIVISTE(R_NOME));

CREATE TABLE DOC_PUBL(
DP_DOCENTE VARCHAR2(20) NOT NULL,
DP_PUBBLICAZIONE VARCHAR2(20) NOT NULL,
CONSTRAINT "SYS_C004672" PRIMARY KEY(DP_DOCENTE,DP_PUBBLICAZIONE),
CONSTRAINT "SYS_C004673" FOREIGN KEY(DP_DOCENTE) REFERENCES DOCENTI(D_CODD),
CONSTRAINT "SYS_C004674" FOREIGN KEY(DP_PUBBLICAZIONE) REFERENCES
PUBBLICAZIONI(P_CODP));
```

Soluzione con 3 cursori

```
CREATE OR REPLACE PROCEDURE RATING(v_raggruppamento
docenti.d_raggruppamento%type, v_anno pubblicazioni.p_anno%type) is

cursor cursoredocente is
  select d_codd
  from docenti
  where d_raggruppamento = v_raggruppamento;

cursor rivistedocente(coddocente in docenti.d_codd%type) is
  select dp_pubblicazione, r_impactfactor
  from doc_publ, riviste, pubblicazioni
  where dp_docente = coddocente and
        p_codp =dp_pubblicazione and
        r_nome = p_rivista and
        p_anno = v_anno;

cursor cursorefinale is
  select d_codd
  from docenti
  order by d_punteggio desc;

rigacursoredocente cursoredocente%rowtype;
rigarivistedocente rivistedocente%rowtype;
rigacursorefinale cursorefinale%rowtype;
numeroautori number;
numdocenti number;
percentuale float;
cont number := 1;
punteggioparziale docenti.d_punteggio%type;

begin
open cursoredocente;
  loop
    fetch cursoredocente into rigacursoredocente;
    exit when cursoredocente%notfound;

    punteggioparziale := 0;
    open rivistedocente(rigacursoredocente.d_codd);
      loop
         fetch rivistedocente into rigarivistedocente;
         exit when rivistedocente %notfound;
         select count(dp_docente) into numeroautori
         from doc_publ
         where dp_pubblicazione = rigarivistedocente.dp_pubblicazione;

         punteggioparziale := (rigarivistedocente.r_impactfactor)/numeroautori;
      end loop;
    close rivistedocente;
    update docenti set d_punteggio = punteggioparziale
    where d_codd = rigacursoredocente.d_codd;
  end loop;
close cursoredocente;
select count(d_codd) into numdocenti from docenti;

open cursorefinale;
   loop
     fetch cursorefinale into rigacursorefinale;
     exit when cursorefinale%notfound;
     if cont < numdocenti/3 then
       update docenti set d_fascia = 'A' where d_codd =rigacursorefinale.d_codd;
```

```
     elsif cont < numdocenti*2/3 then
      update docenti set d_fascia = 'B' where d_codd = rigacursorefinale.d_codd;
     else
       update docenti set d_fascia = 'C' where d_codd =rigacursorefinale.d_codd;
      end if;
     cont:=cont+1;
    end loop;
  close cursorefinale;
end;
```

Soluzione con 2 cursori

```
CREATE OR REPLACE PROCEDURE RATING (p_raggruppamento varchar2(20),p_anno
varchar2(20)) IS

CURSOR crs_pudDocenti IS
  SELECT CodD,CodP,ImpactFactor
  FROM PUBBLICAZIONE,DOC_PUBL,RIVISTE r,Docenti d
  WHERE Pubblicazione=CodP AND Anno=p_anno AND r.Nome=Rivista
    AND d.CodD=Docente AND p_raggruppamento=d.Raggruppamento
  ORDER BY CodD;

CURSOR crs_Docenti IS
  SELECT CodD
  FROM DOCENTI
  WHERE Raggruppamento=p_raggruppamento
  ORDER BY Punteggio;
```

Sfruttando l’ordinamento su CodD vengono elencate in sequenza tutte le pubblicazioni di un certo docente, quindi non è necessario il cursore con parametro sul docente. Attenzione però alla gestione del passaggio da un docente all’altro nel ciclo di scanzione del cursore.

```
explain plan for select sum(PS_SUPPLYCOST)
from PART,PARTSUPP
where P_PARTKEY=PS_PARTKEY
and P_TYPE='SMALL BURNISHED STEEL';
@?/RDBMS/ADMIN/UTLXPLS
```

```
Plan Table
--------------------------------------------------------------------------------
| Operation                 |  Name    |  Rows | Bytes|  Cost  | Pstart| Pstop |
--------------------------------------------------------------------------------
| SELECT STATEMENT          |          |     1 |   35 |   4009 |       |       |
|  SORT AGGREGATE           |          |     1 |   35 |        |       |       |
|   NESTED LOOPS            |          |     5K|  182K|   4009 |       |       |
|    TABLE ACCESS BY INDEX R|PART      |     1K|   32K|      7 |       |       |
|     INDEX RANGE SCAN      |P_TYPE    |     1K|      |      1 |       |       |
|    TABLE ACCESS BY INDEX R|PARTSUPP  |   800K|    7M|      3 |       |       |
|     INDEX RANGE SCAN      |IX_PART_P |   800K|      |      2 |       |       |
```

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

Costo di accesso a PART tramite indice unclustered su P_TYPE (**150 valori distinti**)

$NL_{P\_TYPE} = \lceil (150 \cdot 4) + 200.000 \cdot 4) / (4096 \times 0{,}69) \rceil = 284$

Costo di accesso a PART= 2 - 1 + $\lceil 1/ 150 \cdot 284 \rceil + 1 \cdot \Phi(200.000 / 150, 9.412) = 1+2+1.244=$**1.247**

Numero di tuple residue = 200.000 / 150 = 1.333

Costo di accesso a PARTSUPP tramite indice unclustered su PS_PARTKEY (**200.000 valori distinti**)

$NP_{PARTSUPP} = \lceil 800.000 \times 142/(4096 \times 0{,}69) \rceil = 40.195$
$NL_{PS\_PARTKEY} = \lceil (200.000 \times 4) + 800.000 \times 4) / (4096 \times 0{,}69) \rceil = 1.416$

Costo di accesso a PARTSUPP per un singolo valore di chiave = 2 - 1 + $\lceil 1/ 200.000 \times 1.416 \rceil + 1 \times \Phi(800.000 / 200.000, 40.195) = 1+1+4=$**6**

Costo totale del join = 1.247 + 1.333 · 6 = **9.245**

$NT_{PSRT'+PARTSUPP'} = 1.333 \times 4 = 5.332$

$NP_{PSRT'+PARTSUPP'} = \lceil 5.332 \times (133+142)/(4096 \times 0{,}69) \rceil =$ **519** (se non proitetto)
$NP_{PSRT'+PARTSUPP'} = \lceil 5.332 \times (4)/(4096 \times 0{,}69) \rceil =$ 8 (se proietto su PS_SUPPLYCOST)

Group by (PART'+PARTSUPP) = $2 \times 519 \times (\lceil \log_{100} \lceil 519/100 \rceil \rceil + 1) = 1.038 \times (1 + 1) =$ **2.076**

Costo finale = 9.245+ 2.076= **11.321**