

```

(v_da consegne.c_data%type, v_a consegne.c_data%type) is

cursor cur_cittadino is
  select u_id, u_nome, u_cognome, u_zona, sum(m_compensapeso*c_peso) as
pesocompensato
  from utenti, materia, consegne
  where u_id=c_utente and m_id = c_materia and c_data >= v_da and c_data <= v_a
  group by u_id, u_nome, u_cognome, u_zona;

rigacittadino cur_cittadino%rowtype;
v_totale number(10,2);

begin
  open cur_cittadino;
  loop
    fetch cur_cittadino into rigacittadino;
    exit when cur_cittadino%notfound;

    v_totale:=rigacittadino.pesocompensato;

    if(rigacittadino.u_zona='Centro') then
      v_totale:=v_totale*1.2;
    elsif (rigacittadino.u_zona='Periferia') then
      v_totale:=v_totale*1.3;
    else
      v_totale:=v_totale*1.1;
    end if;
    dbms_output.put_line('Cittadino ' || rigacittadino.u_nome || ' ' ||
rigacittadino.u_cognome || ' Bonus= ' || v_totale || ' ');

  end loop;
  close cur_cittadino;
end;

```

```

explain plan for select distinct(S_NAME)
from SUPPLIER, PART,LINEITEM
where S_SUPPKEY=L_SUPPKEY AND
      P_PARTKEY=L_PARTKEY AND P_TYPE LIKE 'STANDARD%';
@?/RDBMS/ADMIN/UTLXPLS;

```

| Operation | Name | Rows | Bytes | Cost | Pstart | Pstop |
|-----------------------|-----------|------|-------|-------|--------|-------|
| SELECT STATEMENT | | 10K | 625K | 47844 | | |
| SORT UNIQUE | | 10K | 625K | 47844 | | |
| HASH JOIN | | 40K | 2M | 45319 | | |
| TABLE ACCESS FULL | SUPPLIER | 10K | 292K | 65 | | |
| NESTED LOOPS | | 40K | 1M | 43850 | | |
| TABLE ACCESS FULL | PART | 1K | 32K | 1162 | | |
| TABLE ACCESS BY INDEX | LINEITEM | 6M | 51M | 32 | | |
| INDEX RANGE SCAN | IX_PART_L | 6M | | 2 | | |

Il DBMS esegue inizialmente il nested loop tra PART e LINEITEM che è a tabella interna e viene acceduta tramite indice unclustered sul campo L_PARTKEY. Il risultato del join è quindi messo nuovamente in join con la tabella SUPPLIER. Infine al risultato è ordinato per eliminare i valori duplicati

Per conoscere la selettività esatta del predicato su P_TYPE è necessario eseguire la seguente interrogazione

```

select count(*)
from PART
where P_TYPE LIKE 'STANDARD%';

```

L'interrogazione

```

select count(distinct P_TYPE)
from PART
where P_TYPE LIKE 'STANDARD%';

```

non è del tutto corretta rispetto ai requisiti dell'esercizio poiché comporta di applicare l'ipotesi di distribuzione uniforme delle parti sui diversi tipi. In effetti la prima interrogazione restituisce il valore 33.434 parti, mentre la seconda il valore 25 tipi il che porta a stimare $33.333 = 200.000 * (25 / 150)$

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

$$NP_{LINEITEM} = \lceil 6.001.215 \times 113 / (4.096 \times 0,69) \rceil = 239.944$$

$$ET_{PART} = 333.434$$

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

Accesso a LINEITEM per ogni singola parte

$$= 2-1 + \lceil 1/200.000 \times 8.777 \rceil + 1 \times \Phi(6.001.215 / 200.000, 239.944) = 1 + 1 + 31 = 33$$

$$\text{Costo Nested Loop} = 9412 + 33.434 \times 33 = \mathbf{1.112.734}$$

$$NR_{\text{LINEITEM+PART}} = \lceil 6.001.215 \times 33.434 / 200.000 \rceil = 1.003.224$$

$$NP_{\text{LINEITEM+PART}} = \lceil 1.003.224 \times (113+133) / (4.096 \times 0,69) \rceil = 87.323$$

$$NP_{\text{SUPPLIER}} = \lceil 10.000 \times 143 / (4.096 \times 0,69) \rceil = 506$$

$$\text{Costo Hash Join con SUPPLIER} = 506 + 87.323 = \mathbf{87.829}$$

Dopo il join il numero di tuple nel risultato parziale rimane invariato perchè non ci sono condizioni di selezione su SUPPLIER e per ogni riga d'ordine (LINEITEM) c'è un solo fornitore (SUPPLIER)

$$NP_{\text{LINEITEM + PART + SUPPLIER}} = \lceil 1.003.224 \times (113+133+143) / (4.096 \times 0,69) \rceil = 138.083$$

$$\begin{aligned} \text{Sort LINEITEM+PART+SUPPLIER} &= 2 \times 138.083 \times (\lceil \log_{100} 138.083 \rceil + 1) = 276.166 \times (3+1) \\ &= \mathbf{1.104.664} \end{aligned}$$

$$\text{Costo totale} = \mathbf{1.112.734 + 87.829 + 1.104.664 = 2.305.227}$$