

```

(p_anno int) IS

cursor csr_fatturatocliente IS
  select C_cod, sum(ART_prezzo*D_quantità) as fatturatoannuo
  from CLIENTI,ARTICOLI,DETTAGLIO,FATTURE
  where C_cod = F_cliente and F_cod = D_codfattura and F_anno = D_annofattura
        and D_articolo = ART_cod and F_anno = p_anno
  group by C_cod
  order by fatturatoannuo DESC;

v_FattXcli csr_fatturatocliente%ROWTYPE;
v_fatt_totale float;
v_fatt_parziale float := 0;

begin

  select sum(D_quantità*ART_prezzo) into v_fatt_totale
  from DETTAGLIO,ARTICOLI
  where D_articolo = ART_cod and D_annofattura = p_anno;

  open csr_fatturatocliente;
  loop
    fetch csr_fatturatocliente into v_FattXcli;
    exit when csr_fatturatocliente%NOTFOUND;

    v_fatt_parziale := v_fatt_parziale + v_FattXcli.fatturatoannuo;

    if (v_fatt_parziale <= (v_fatt_totale*0.1)) then
      update CLIENTI
      set C_classe = 'Gold'
      where C_cod = v_FattXcli.C_cod;
    elsif (v_fatt_parziale <= (v_fatt_totale*0.2)) then
      update CLIENTI
      set C_classe = 'Silver'
      where C_cod = v_FattXcli.C_cod;
    elsif (v_fatt_parziale <= (v_fatt_totale*0.3)) then
      update CLIENTI
      set C_classe = 'Bronze'
      where C_cod = v_FattXcli.C_cod;
    else
      update CLIENTI
      set C_classe = 'Regular'
      where C_cod = v_FattXcli.C_cod;
    end if;

  end loop;
  close csr_fatturatocliente;
end Classifica;

```

```

explain plan for select avg(PS_AVAILQTY)
from PART,SUPPLIER,PARTSUPP
where PS_PARTKEY=P_PARTKEY AND S_SUPPKEY=PS_SUPPKEY AND
S_NAME='Supplier000000009' AND P_SIZE>10;
@?/RDBMS/ADMIN/UTLXPLS

```

Plan Table

Operation	Name	Rows	Bytes	Cost	Pstart	Pstop
SELECT STATEMENT		1	49	307		
SORT AGGREGATE		1	49			
NESTED LOOPS		65	3K	307		
NESTED LOOPS		80	3K	147		
TABLE ACCESS FULL	SUPPLIER	1	30	65		
TABLE ACCESS BY INDEX	PARTSUPP	800K	9M	82		
INDEX RANGE SCAN	IX_SUPP_PA	800K		2		
TABLE ACCESS BY INDEX R	PART	163K	956K	2		
INDEX UNIQUE SCAN	SYS_C00958	163K		1		

$$NP_{PS} = \lceil 800.000 \times 142 / (4096 \times 0,69) \rceil = 40.195$$

$$NP_{PART} = \lceil 200.000 \times 130 / (4096 \times 0,69) \rceil = 9.412$$

$$NP_{SUPP} = \lceil 10.000 \times 143 / (4096 \times 0,69) \rceil = \mathbf{506}$$

$$Sel(S_NAME='Supplier000000009') = 1/10.000$$

$$Sel(P_SIZE>10) = (50-10)/(50-1) = 0,82$$

$$EK_{SUPP} = 10.000 \times 1/10.000 = 1$$

$$NL_{PS_SUPPKEY} = \lceil (10.000 \times 4 + 800.000 \times 4) / (4096 \times 0,69) \rceil = 1.147$$

$$\begin{aligned} \text{Costo di accesso a PARTSUPP} &= 1 + \lceil (1/10.000 \times 1.147) \rceil + 1 \times \Phi(800.000 / 10.000, 40.195) \\ &= 1 + 1 + 80 = \mathbf{82} \end{aligned}$$

$$NP_{PS+SUPP} = \lceil 80 \times (142 + 143) / (4096 \times 0,69) \rceil = 9$$

$$NL_{P_PARTKEY} = \lceil (200.000 \times 4 + 200.000 \times 4) / (4096 \times 0,69) \rceil = 567$$

$$\begin{aligned} \text{Costo di accesso a una tupla di PART} &= 1 + \lceil (1/200.000 \times 567) \rceil + 1 \times \Phi(200.000 / 200.000, 9.412) \\ &= 1 + 1 + 1 = \mathbf{3} \end{aligned}$$

$$\text{Costo di accesso a PART} = 80 \times 3 = \mathbf{240}$$

Le tuple a cui accedere non sono in fila quindi dovrò discendere l'albero ogni volta e accedere a foglie diverse INDEX_UNIQUE_SCAN

$$EK_{PS+SUPP+PART} = 80 \times 0,82 = 66$$

$$NP_{PS+SUPP+PART} = \lceil 66 \times (142 + 143 + 133) / (4096 \times 0,69) \rceil = 10$$

Group by (PS+SUPP+PART) = 2 × 10 × (⌈log₁₀₀ 1⌉ + 1) = 20 (l'ordinamento per il group by viene fatto in memoria quindi è sufficiente leggere e scrivere i dati, era anche possibile supporre esplicitamente che i dati fossero già in memoria, in questo caso il costo sarebbe stato 0)

$$\text{Costo finale} = 506 + 82 + 240 + 20 = \mathbf{848}$$