

(DataInizio date, DataFine date) is

v\_pvMedio real;

```
cursor cur is
select V_CODICE, v_data, v_prezzoVendita, v_prodotto, c_cap,c_provincia
from vendite ,clienti
where c_cf=v_cliente
and v_data between DataInizio and DataFine;
```

vr\_cur cur%rowtype;

begin

```
open cur;
loop
fetch cur into vr_cur;
exit when cur%NOTFOUND;
```

```
select avg(v_prezzoVendita) into v_pvMedio
from vendite
where v_prodotto=cur.v_prodotto
and v_data between cur.v_data-30 and cur.v_data;
```

```
if (cur.v_prezzoVendita < v_pvMedio-(v_pvMedio*0.2)) or (cur.v_prezzoVendita >
v_pvMedio+(v_pvMedio*0.2)) then
insert into warning values (cur.v_codice, 'Rule1');
end if;
```

```
SELECT C_CAP,C_PROVINCIA
FROM COMUNI
WHERE C_CAP=vr_cur.c_cap;
```

```
if (c_provincia <> vr_cur.c_provincia) then
insert into Warning values (cur.v_codice, 'Rule2');
end if;
end;
commit;
```

```

explain plan for
select SUM(L_EXTENDEDPRICE)
from LINEITEM,PART,ORDERS
where L_ORDERKEY=O_ORDERKEY and L_PARTKEY=P_PARTKEY and O_SHIPPRIORITY='5-LOW'
and P_TYPE = 'STANDARD BURNISHED NICKEL' ;
@?/RDBMS/ADMIN/UTLXPLS

```

Plan Table

Operation	Name	Rows	Bytes	Cost	Pstart	Pstop
SELECT STATEMENT		1	48	53769		
SORT AGGREGATE		1	48			
MERGE JOIN		2K	93K	53769		
SORT JOIN		40K	1M	45627		
NESTED LOOPS		40K	1M	43850		
TABLE ACCESS FULL	PART	1K	32K	1162		
TABLE ACCESS BY INDEX	LINEITEM	6M	97M	32		
INDEX RANGE SCAN	IX_PART_L	6M		2		
SORT JOIN		75K	439K	8142		
TABLE ACCESS FULL	ORDERS	75K	439K	7147		

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

$$ET_{PART} = \lceil 200.000 \times 1/150 \rceil = 1.334$$

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

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

Accesso a LINEITEM

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

$$\text{Accesso a LINEITEM} = 1.334 \times 33 = \mathbf{44.022}$$

Assumo ORDERS ordinata su O\_ORDERKEY mentre il risultato del join precedente è da ordinare

$$NR_{LINEITEM+PART} = \lceil 6.001.215 \times 1/150 \rceil = 40.009$$

$$NP_{LINEITEM+PART} = \lceil 40.009 \times (113+133) / (4.096 \times 0,69) \rceil = 3.483$$

$$NP_{ORDERS} = \lceil 1.500.000 \times 106 / (4096 \times 0,69) \rceil = \mathbf{56.259}$$

$$\text{Sort LINEITEM+PART} = 2 \times 3.483 \times (\lceil \log_{100} \lceil 3.483/101 \rceil \rceil + 1) = 6.966 \times (\lceil \log_{100} (35) \rceil + 1) = \mathbf{20.898}$$

Assumendo l'indipendenza dei predicate di selezione le righe d'ordine precedentemente recuperate apparterranno uniformemente a ordini con le diverse priorità, quindi:

$$NR_{LINEITEM+PART+ORDERS} = \lceil 40.009 \times 1/5 \rceil = 8.002$$

$$NP_{LINEITEM+PART+ORDERS} = \lceil 8.002 \times (246+106) / (4.096 \times 0,69) \rceil = 873$$

$$\text{Costo Group by} = 2 \times 873 \times (\lceil \log_{100} \lceil 873/101 \rceil \rceil + 1) = 1.746 \times (1 + 1) = \mathbf{3.492}$$

$$\text{Costo totale} = 9.412 + 44.022 + 56.259 + 20.898 + 3.492 = \mathbf{134.083}$$