

```

create table DOMANDE (
D_ID int,
D_Testo varchar2(100),
D_RispostaCorretta int,
D_LivelloComplessità int,
D_AreaDomanda int,
primary key (D_ID));

create table RISPOSTE (
R_IDDomanda int,
R_IDRisposta int,
R_TestoRisposta varchar2(100),
primary key
(R_IDDomanda,R_IDRisposta),
foreign key (R_IDDomanda)
references DOMANDE(D_ID));

create table RISPOSTESTUDENTE(
RS_IDStudente int,
RS_IDCompito int,
RS_IDDomanda int,
RS_Risposta int,
primary key (RS_IDStudente, RS_IDCompito, RS_IDDomanda),
foreign key (RS_IDDomanda) references references DOMANDE(D_ID));

create or replace
procedure CorreggiCompito(IDStudente int, IDCompito int) IS
--cursore
cursor cDom is
select * from RISPOSTESTUDENTE, DOMANDE
where RS_IDDomanda=D_ID and RS_IDStudente=IDStudente AND RS_IDCompito=IDCompito
order by D_AreaDomanda;

vParzArea float;
vArea int;
vNumAree int;
vNum int;
vTot float;

begin
vArea:=1;
vParzArea:=0;
vTot:=0;
vNum:=0;
vNumAree:=0;
FOR vDom IN cDom
LOOP
if ((vDom.D_AreaDomanda>vArea) and (vNum>0)) then
DBMS_OUTPUT.PUT_LINE('Area: ' || vArea || ' Parziale: ' || vParzArea/vNum);
vTot := vTot + vArea*vParzArea/vNum;
vNumAree:=vNumAree+1;
vParzArea:=0;
vArea:=vDom.D_AreaDomanda;
vNum:=0;
end if;

--aggiorna conteggi parziali
vNum:=vNum+1;
if (vDom.D_RispostaCorretta=vDom.RS_Risposta) then
vParzArea:=vParzArea+vDom.D_LivelloComplessità;
else
vParzArea:=vParzArea-0.5;
end if;
END LOOP;
--stampe finali
DBMS_OUTPUT.PUT_LINE('Area: ' || vArea || ' Parziale: ' || vParzArea);
DBMS_OUTPUT.PUT_LINE('Totale: ' || vTot/vNumAree);

end;

```

```

select P_BRAND, sum(L_QUANTITY)
from PART,LINEITEM,ORDERS
where P_PARTKEY=L_PARTKEY and L_ORDERKEY=O_ORDERKEY
and P_TYPE='ECONOMY ANODIZED NICKEL'
and O_ORDERPRIORITY='3-MEDIUM'
group by P_BRAND;

```

OPERATION	OBJECT_NAME	CARDINALITY	C
SELECT STATEMENT			25
SORT (GROUP BY)			25
NESTED LOOPS			8142
NESTED LOOPS			39998
TABLE ACCESS (FULL)	PART		1333
TABLE ACCESS (BY INDEX ROWID)	LINEITEM		30
INDEX (RANGE SCAN)	IX_PART_LI		30
TABLE ACCESS (BY INDEX ROWID)	ORDERS		1
INDEX (UNIQUE SCAN)	SYS_C0010882		1

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

$$NP_O = \lceil 1.500.000 \times 106 / (4096 \times 0,69) \rceil = 56.259$$

$$NP_P = \lceil 200.000 \times 133 / (4096 \times 0,69) \rceil = 9.412$$

$Sel(P_TYPE='ECONOMY ANODIZED NICKEL') = 1/150$

$Sel(O_ORDERPRIORITY='3-MEDIUM') = 1/5$

$$NL_{L_PARTKEY} = \lceil (200.000 \times 4 + 6.001.215 \times 4) / (4.096 \times 0,69) \rceil = 8.777$$

Si accede all'indice in base al valore di una parte al fine di recuperare tutte le linee d'ordine corrispondenti a quella specifica parte. L'operazione è ripetuta per tutte le parti selezionate dal predicato su P_TYPE

$$\text{Costo di accesso all'indice 2} + \lceil 1/200.000 \times 8.777 \rceil + \Phi(6.001.215/200.000, 239.944) = 2+1+31 = 34$$

$$\text{Costo(Nested Loop}_{LI+P}) = 9.412 + \lceil 200.000 / 150 \times 34 \rceil = \mathbf{54.746}$$

$$NT_{LI+P} = \lceil 6.001.215 / 150 \rceil = 40.009$$

$$NP_{LI+P} = \lceil 40.009 \times (113+133) / (4096 \times 0,69) \rceil = 3.483$$

$$NL_{O_ORDERKEY} = \lceil (1.500.000 \times 4 + 1.500.000 \times 4) / (4.096 \times 0,69) \rceil = 4.246$$

$$\text{Costo di accesso all'indice 3-1} + \lceil 1/1.500.000 \times 4.246 \rceil + 1 = 2+1+1 = 4$$

Si accede all'indice su ORDERS per recuperare l'ordine relativo a un certo lineitem. L'accesso quindi recupera una sola tupla e accede a una sola pagina dati

$$\text{Costo(Nested Loop}_{(LI+P)-O}) = 3.483 + 40.009 \times 4 = \mathbf{163.519}$$

$$NT_{(LI+P)-O} = \lceil 40.009 / 5 \rceil = \lceil 6.001.215 / (150 \times 5) \rceil = 8.002$$

Il risultato del secondo join è pari alla cardinalità di LINEITEM a cui sono applicate le due selezioni oppure alla cardinalità del risultato del primo join a cui viene applicata la seconda selezione

$$NP_{(LI+P)-O} = \lceil 8.002 \times (113+133+106) / (4096 \times 0,69) \rceil = 997$$

$$\text{Costo (GB}_{LI+O}) = 2 \times 991 \times (\lceil \log_{100}(997) \rceil + 1) = 2 \times 997 \times (2+1) = \mathbf{5.982}$$

$$\mathbf{\text{Costo totale} = 54.746 + 163.519 + 5.982 = 224.247}$$