

```

CREATE TABLE RICHIESTE
  (R_IDRichiesta NUMBER(5,0),
  R_Titolare VARCHAR2(20 BYTE),
  R_Data DATE,
  R_Costo NUMBER(5,0),
  R_Descrizione VARCHAR2(20 BYTE),
  R_NumStudenti NUMBER(5,0),
  PRIMARY KEY (R_IDRichiesta)
);

```

```

CREATE TABLE FONDIASSEGNATI
  (F_IDFondi NUMBER(5,0),
  F_Titolare VARCHAR2(20 BYTE),
  F_Data DATE,
  F_Descrizione VARCHAR2(20 BYTE),
  F_Finanziamento NUMBER(5,0),
  PRIMARY KEY (F_IDFondi)
);

```

create or replace

```

PROCEDURE AssegnaFondi(vImporto number, vData date) is

```

```

-- Richieste

```

```

cursor curRichieste is

```

```

select count (F_IDFONDI) as P1 , R_Costo/R_NumStudenti as P2, r_idrichiesta,
R_Costo

```

```

from RICHIESTE left outer join FONDIASSEGNATI

```

```

on (R_Titolare = F_Titolare and F_data between vData-365 and vData)

```

```

group by r_idrichiesta, R_Costo, R_Costo/R_NumStudenti

```

```

order by 1 asc, 2 asc;

```

```

-- Nota: "sign(count (F_IDFONDI)) as P1" per pesare tutti uguali i titolari di
più di una richiesta

```

```

vRimanente number(5,0);

```

```

begin

```

```

vRimanente:=vImporto;

```

```

--scorro il cursore

```

```

for vRichieste in curRichieste

```

```

loop

```

```

  if vRichieste.R_Costo <= vRimanente then

```

```

    --richiesta completamente finanziata

```

```

    DBMS_OUTPUT.PUT_LINE('Richiesta: ' || vRichieste.R_IDRichiesta ||',

```

```
Finanziamento: ' || vRichieste.R_Costo);

```

```

    vRimanente:=vRimanente- vRichieste.R_Costo;

```

```

  else

```

```

    --richiesta non completamente finanziata

```

```

    DBMS_OUTPUT.PUT_LINE('Richiesta: ' || vRichieste.R_IDRichiesta ||',

```

```
Finanziamento: ' || vRimanente ||' su ' || vRichieste.R_Costo);

```

```

    vRimanente:=0;

```

```

  end if;

```

```

  if vRimanente<=0 then

```

```

    exit;

```

```

  end if;

```

```

end loop;

```

```

if vRimanente>0 then

```

```

  --fondi non assegnati

```

```

  DBMS_OUTPUT.PUT_LINE('Fondi non assegnati: ' || vRimanente );

```

```

end if;

```

```

end;

```

```

-- Con due cursori
create or replace PROCEDURE AssegnaFondi2(vImporto number, vData date) is

-- Richieste
cursor curRichiestel is
select R_Costo/R_NumStudenti as P2, r_idrichiesta, R_Costo
from RICHIESTE where R_Titolare not in
(select F_Titolare from FONDIASSEGNATI where F_data between vData-365 and vData)
order by 1 asc;

cursor curRichieste2 is
select R_Costo/R_NumStudenti as P2, r_idrichiesta, R_Costo
from RICHIESTE where R_Titolare in
(select F_Titolare from FONDIASSEGNATI where F_data between vData-365 and vData)
order by 1 asc;

vRimanente number(5,0);
begin
vRimanente:=vImporto;

--scorro il primo cursore
for vRichieste in curRichiestel
loop
if vRichieste.R_Costo <= vRimanente then
--richiesta completamente finanziata
DBMS_OUTPUT.PUT_LINE('Richiesta: ' || vRichieste.R_IDRichiesta ||',
Finanziamento: ' || vRichieste.R_Costo);
vRimanente:=vRimanente- vRichieste.R_Costo;
else --richiesta non completamente finanziata
DBMS_OUTPUT.PUT_LINE('Richiesta: ' || vRichieste.R_IDRichiesta ||',
Finanziamento: ' || vRimanente ||' su ' || vRichieste.R_Costo);
vRimanente:=0;
end if;
if vRimanente<=0 then exit;
end if;
end loop;

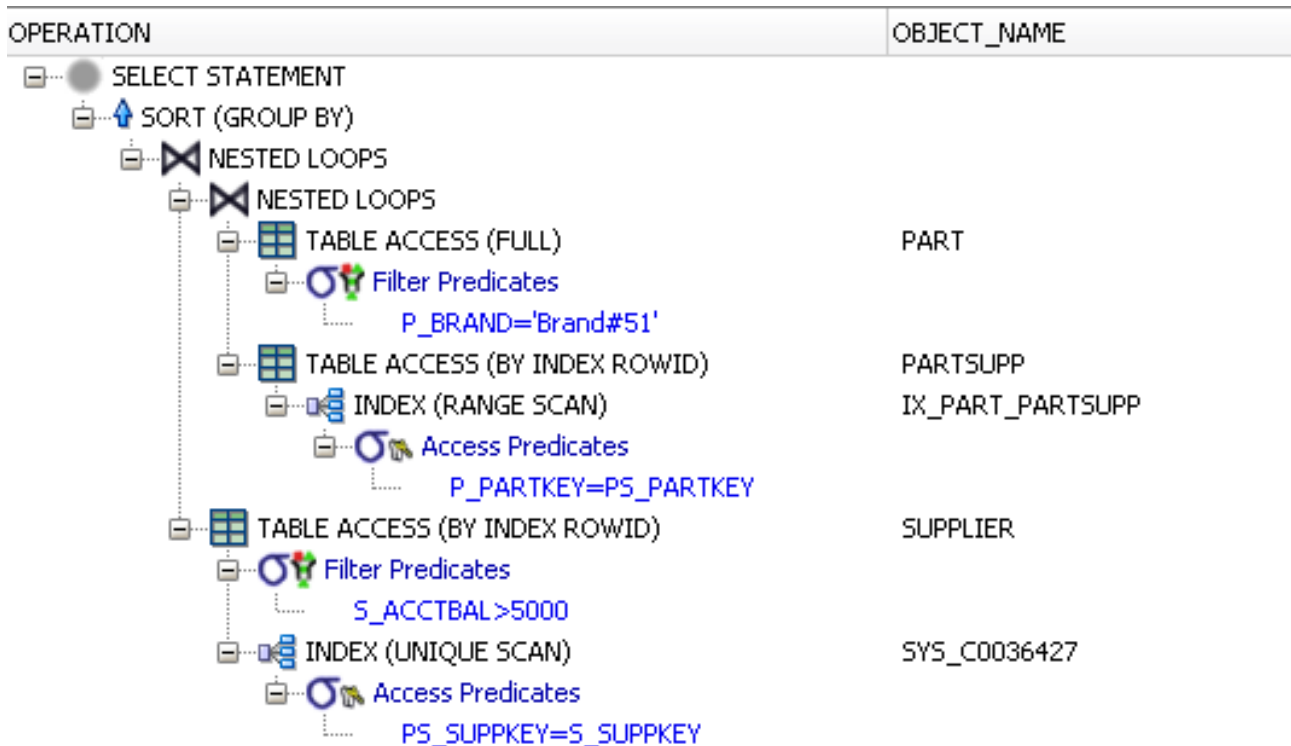
if vRimanente>0 then
--scorro il secondo cursore
for vRichieste in curRichieste2
loop
if vRichieste.R_Costo <= vRimanente then
--richiesta completamente finanziata
DBMS_OUTPUT.PUT_LINE('Richiesta: ' || vRichieste.R_IDRichiesta ||',
Finanziamento: ' || vRichieste.R_Costo);
vRimanente:=vRimanente- vRichieste.R_Costo;
else --richiesta non completamente finanziata
DBMS_OUTPUT.PUT_LINE('Richiesta: ' || vRichieste.R_IDRichiesta ||',
Finanziamento: ' || vRimanente ||' su ' || vRichieste.R_Costo);
vRimanente:=0;
end if;
if vRimanente<=0 then exit;
end if;
end loop;
end if;
if vRimanente>0 then
--fondi non assegnati
DBMS_OUTPUT.PUT_LINE('Fondi non assegnati: ' || vRimanente );
end if;
end;

```

```

select S_SUPPKEY, S_NAME, sum(S_ACCTBAL)
from TPCD.PART, TPCD.PARTSUPP, TPCD.SUPPLIER
where P_PARTKEY=PS_PARTKEY and PS_SUPPKEY=S_SUPPKEY and S_ACCTBAL
>5000 and P_BRAND = 'Brand#51'
GROUP BY S_SUPPKEY, S_NAME;

```



$$NP_P = \lceil 200.000 \times 131 / (4096 \times 0,69) \rceil = 9.271$$

$$NP_{PS} = \lceil 800.000 \times 143 / (4096 \times 0,69) \rceil = 40.478$$

$$NP_S = \lceil 10.000 \times 144 / (4096 \times 0,69) \rceil = 510$$

$$\text{Sel}(P_BRAND = 'Brand\#51') = 1/25$$

$$\text{Sel}(S_ACCTBAL > 5000) = \frac{1}{(9999,72 - 5000) \times (998,22 + 9999,72)} = \frac{1}{10997,94} = 0.456$$

$$NL_{PS_PARTKEY} = \lceil (200.000 \times 4 + 800.000 \times 4) / (4.096 \times 0,69) \rceil = 1.416$$

$$\text{Costo di accesso all'indice su } PS_PARTKEY = 2 + \lceil 1/200.000 \times 1.416 \rceil + \lceil 1/200.000 \times 40.478 \rceil = 2 + 1 + 1 = 4$$

$$\text{Costo}(\text{Nested Loop } P+PS) = 9.271 + 200.000 \times 1/25 \times 4 = \mathbf{41.271}$$

$$ET_{P+PS} = \lceil 800.000 \times 1/25 \rceil = 32.000$$

$$NP_{P+PS} = \lceil 32.000 \times (131 + 143) / (4096 \times 0,69) \rceil = 3.103$$

$$NL_{S_SUPPKEY} = \lceil (10.000 \times 4 + 10.000 \times 4) / (4.096 \times 0,69) \rceil = 29$$

$$\text{Costo di accesso all'indice su } S_SUPPKEY = 1 + \lceil 1/10.000 \times 29 \rceil + \lceil 1/10.000 \times 510 \rceil = 1 + 1 + 1 = 3$$

$$\text{Costo}(\text{Nested Loop}_{P+PS-S}) = 3.103 + 32.000 \times 3 = \mathbf{99.103}$$

$$ET_{P+PS} = \lceil 800.000 \times 1/25 \times 0.456 \rceil = 14.592$$

$$NP_{P+PS+S} = \lceil 14.592 \times (131+143+144) / (4096 \times 0,69) \rceil = 2.159$$

$$\text{Costo (GB)} = 2 \times 2.159 \times (\lceil \log_{100}(2.159) \rceil + 1) = 2 \times 2.159 \times (2+1) = \mathbf{12.954}$$

$$\mathbf{\text{Costo totale} = 41.271 + 99.103 + 12.954 = 153.328}$$