

```

create table PUNTORACCOLTA (
P_ID int,
P_Indirizzo varchar2(16),
P_NumBici int,
primary key (P_ID));

create table DISTANZE(
D_Punto1 int,
D_Punto2 int,
D_Dist int,
primary key(D_Punto1, D_Punto2),
foreign key (D_Punto1) references PUNTORACCOLTA(P_ID),
foreign key (D_Punto2) references PUNTORACCOLTA(P_ID));

create or replace
procedure Spostamenti IS
--cursore
cursor cPRto is
select * from PUNTORACCOLTA
where P_NumBici < 10
order by P_NumBici;

cursor cPRfrom (toPR int,nBici int) is
select P_ID from PUNTORACCOLTA, DISTANZE
--condizione per scegliere solo scaglioni "utili"
where D_Punto1=toPR and D_Punto2=P_ID and P_NumBici > nBici
order by D_Dist desc;

vPRto cPRto%ROWTYPE;
vPRfrom cPRfrom%ROWTYPE;
vNumMin int;
vTras int;

begin
open cPRto;
loop
--recupero di ogni PR
fetch cPRto into vPRto;
exit when cPRto%NOTFOUND;

--trovo posto più vicino
if (vPRto.P_NumBici<5) then
vNumMin:=25;
vTras:=10;
else
vNumMin:=20;
vTras:=5;
end if;
--calcolo del punto più vicino
open cPRfrom(vPRto.P_ID,vNumMin);
fetch cPRfrom into vPRfrom;
--scala e accresce
UPDATE PUNTORACCOLTA SET P_NumBici=P_NumBici-vTras WHERE P_ID=vPRfrom.P_ID;
UPDATE PUNTORACCOLTA SET P_NumBici=P_NumBici+vTras WHERE P_ID=vPRto.P_ID;
--stampa
DBMS_OUTPUT.PUT_LINE('Spostate ' || vTras || 'da ' || vPRfrom.P_ID || ' a '
|| vPRto.P_ID);
close cPRfrom;

end loop;
close cPRto;
end;

```

## ALTERNATIVA CON UN SOLO CURSORE:

```
create or replace
procedure Spostamenti IS
--cursore
cursor cPRto is
select * from PUNTORACCOLTA
where P_NumBici < 10
order by P_NumBici;

vPRto cPRto%ROWTYPE;
vNumMin int;
vTras int;
vP_ID int;

begin
open cPRto;
loop
  --recupero di ogni PR
  fetch cPRto into vPRto;
  exit when cPRto%NOTFOUND;

  --trovo posto più vicino
  if (vPRto.P_NumBici<5) then
    vNumMin:=25;
    vTras:=10;
  else
    vNumMin:=20;
    vTras:=5;
  end if;
  --calcolo del punto più vicino

  select P_ID into vP_ID from (
  select P_ID, D_Dist from PUNTORACCOLTA, DISTANZE
--condizione per scegliere solo scaglioni "utili"
  where D_Punto1=vPRto.P_ID and D_Punto2=P_ID and P_NumBici > vNumMin
  order by D_Dist desc)

where rownum=1;

  --scala e accresce
  UPDATE PUNTORACCOLTA SET P_NumBici=P_NumBici-vTras WHERE P_ID=vP_ID;
  UPDATE PUNTORACCOLTA SET P_NumBici=P_NumBici+vTras WHERE P_ID=vPRto.P_ID;
  --stampa
  DBMS_OUTPUT.PUT_LINE('Spostate ' || vTras || 'da ' || vP_ID || ' a ' ||
vPRto.P_ID);

end loop;
close cPRto;
end;
```

```

SELECT O_CLERK, COUNT(*)
FROM LINEITEM, ORDERS
WHERE L_ORDERKEY=O_ORDERKEY
      AND L_DISCOUNT <=0.01 AND O_ORDERPRIORITY = '3-MEDIUM'
group by O_CLERK;

```

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

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

$Sel(L\_DISCOUNT \leq 0.01) = 0,01 - 0 / 0,1 = 10\%$

$Sel(O\_ORDERPRIORITY = '3-MEDIUM') = 1/5$

$$NL_{L\_PARTKEY} = \lceil (1.500.000 \times 4 + 6.001.215 \times 4) / (4.096 \times 0,69) \rceil = 10.617$$

*Si accede all'indice in base al valore di un ordine al fine di recuperare tutte le linee d'ordine corrispondenti a quello specifico ordine. L'operazione è ripetuta per tutti gli ordini selezionati dal predicato su ORDERS*

$$\text{Costo di accesso all'indice } 3-1 + \lceil 1/1.500.000 \times 10.617 \rceil + \lceil 1/1.500.000 \times 239.944 \rceil = 2+1+1 = 4$$

$$\text{Costo(Nested Loop}_{LI-P}) = 56.259 + 1.500.000 / 5 \times 4 = \mathbf{1.256.259}$$

$$NT_{LI+O} = \lceil 6.001.215 \times 0,1 \times 1/5 \rceil = 120.025$$

$$NP_{LI+O} = \lceil 120.025 \times (113+106) / (4096 \times 0,69) \rceil = 9.301$$

$$\text{Costo (GB}_{LI+O}) = 2 \times 9.301 \times (\lceil \log_{100}(9.301) \rceil + 1) = 2 \times 9.301 \times (2+1) = \mathbf{55.806}$$

$$\mathbf{\text{Costo totale} = 1.256.259 + 55.806 = 1.312.065}$$