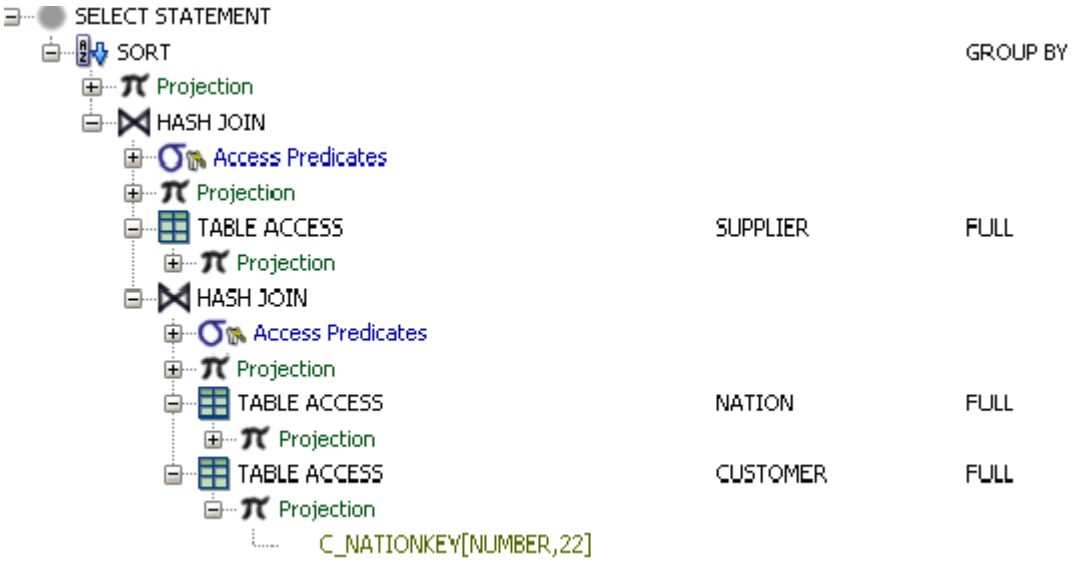
```
CREATE TABLE CATEGORIA
    (C_IDCat NUMBER(5,0),
     C_NomeCat VARCHAR2(20 BYTE),
       PRIMARY KEY (C_IDCat)
);

CREATE TABLE PROD
    (P_IDProdotto NUMBER(5,0),
     P_Nome VARCHAR2(20 BYTE),
       P_Prezzo NUMBER(5,0),
       P_Popolarita NUMBER(5,0),
       P_IDCat NUMBER(5,0),
       PRIMARY KEY (P_IDProdotto),
       FOREIGN KEY (P_IDCat) REFERENCES CATEGORIA (C_IDCat)
  );

CREATE TABLE LISTADESIDERI
    (L_IDCliente NUMBER(5,0),
     L_IDProdotto NUMBER(5,0),
     PRIMARY KEY (L_IDCliente,L_IDProdotto),
       FOREIGN KEY (L_IDProdotto) REFERENCES PROD (P_IDProdotto)
  );


create or replace
procedure PubblicizzaProdotti (vIdCliente number) is

-- Lista desideri
cursor curLista is
select L_IDProdotto
from LISTADESIDERI
where L_IDCliente=vIdCliente;

-- Prodotti simili
cursor curProdSim (prod_id IN numeric) is
select P1.P_IDProdotto, 2*(P2.P_Prezzo-P1.P_Prezzo)/P2.P_Prezzo*100 +
4*UTL_MATCH.EDIT_DISTANCE_SIMILARITY(P1.P_Nome,P2.P_Nome) + P1.P_Popolarita as C
--select P1.P_IDProdotto, P1.P_Popolarita as C
from PROD P1, PROD P2
where P1.P_IDProdotto <> P2.P_IDProdotto AND P2.P_IDProdotto = prod_id AND
P1.P_IDCat=P2.P_IDCat
order by 2 desc;

vLista curLista%ROWTYPE;
vPS curProdSim %ROWTYPE;

begin

for vLista in curLista loop
  open curProdSim(vLista.L_IDProdotto);
    fetch curProdSim into vPS;
      exit when curProdSim%NOTFOUND;
    dbms_output.put_line('Prodotto: '||vPS.P_IDProdotto||' Costo: '||vPS.C);
    fetch curProdSim into vPS;
      exit when curProdSim%NOTFOUND;
    dbms_output.put_line('Prodotto: '||vPS.P_IDProdotto||' Costo: '||vPS.C);
  close curProdSim;
  end loop;
end;
```

```
explain plan for select n_name, count(*)
from CUSTOMER,NATION,SUPPLIER
where C_NATIONKEY=S_NATIONKEY and C_NATIONKEY=N_NATIONKEY
GROUP BY N_NAME;
@?/RDBMS/ADMIN/UTLXPLS;
```



$NP_{NATION} = \lceil 25 \times 106 / (4096 \times 0{,}69) \rceil = 1$
$NP_{CUSTORMER} = \lceil 150.000 \times 159 / (4096 \times 0{,}69) \rceil = 8.386$

Costo hybrid hash join NATION – CUSTOMER $2 \times 8.386 + 8386 + 1 =$ **25.159**

ATTENZIONE: si è assunto che la relazione CUSTOMER non potesse essere mantenuta in memoria (8386 >101) e che quindi fosse necessario utilizzare per essa la tecnica dell'hybrid hash join. Si accetteranno anche soluzioni che stimano il costo utilizzando la formula dell'hash join (8.386 + 1)

$NP_{CUSTOMER+NATION} = \lceil 150.000 \times (158+105) / (4096 \times 0{,}69) \rceil = 13.959$
$NP_{SUPPLIER} = \lceil 10.000 \times 144 / (4096 \times 0{,}69) \rceil = 506$

Costo hybrid hash join NATION – CUSTOMER – SUPPLIER $3 \times (13.959 + 506) =$ **43.395**

$NP_{CUSTOMER+NATION+SUPPLIER} = \lceil 60.000.414 \times (158+105+143) / (4096 \times 0{,}69) \rceil = 8.619.286$

Il numero di tuple generate dal secondo join è così elevato perché la condizione di join è sul campo nationkey. Il numero di tuple deve essere calcolato mediante la query

```
select count(*)
from customer, supplier
where s_nationkey=c_nationkey;
```

il valore può essere poi verificato anche statisticamente: le nazioni sono 25, i fornitori (supplier) 10.000 e i clienti (customer) 150.000, quindi in media ci saranno 10.000 / 25 = 400 fornitori e 150.000 / 25 = 6.000 clienti per ogni nazione. Quindi $6.000 \times 400 \times 25 = 60.000.000$

Costo del group by $2 \times 8.619.286 \times (\lceil \log_{100} 8.619.286 \rceil + 1) = 2 \times 8.619.286 \times (4+1) = 86.192.860$

Costo Totale = 25.159 + 43.395 + 86.192.860 = 86.261.414