```
(v_maxcal number, v_alimentonocivo alimenti.a_id%type,
 v_utente utenti.u_id%type) is

cursor p_cursor (curs_pasto in pietanze.p_pasto%type) is
    select p_id, p_maxrip, sum(a_calgr*c_qtagr) as CAL_P
    from pietanze, alimenti, composizione
    where p_id = c_id2
    and a_id = c_id1
    and p_pasto = curs_pasto
    group by p_id
    order by CAL_P desc;

v_max number := 0;
v_Nocivo number := 0;
v_rip number;
v_calorieraggiunte number;
p_cursor_row p_cursor%rowtype;

begin

for i in 1..7
  loop
  end if;
    for j in 1..3
    loop
      if j = 1 then
            v_max := (v_maxcal*0.30);
      elsif j = 2 then
            v_max := (v_maxcal*0.50);
      else  v_max := (v_maxcal*0.20);
      end if;

      v_calorieraggiunte := 0;
      open p_cursor(j);
      loop
      fetch p_cursor into p_cursor_row;

        select count(d_pietanza) into v_rip
        from dieta
        where d_pietanza = p_cursor_row.p_id
        and d_utente = v_utente;

        v_Nocivo :=0;
        select count(*) into v_Nocivo
        from alimenti, composizione
        where a_id = c_id1 and a_Nome= v_alimentonocivo
        and c_id2= p_cursor_row.p_id;


        if (v_calorieraggiunte + p_cursor_row.CAL_P) < v_max
            and v_rip < p_cursor_row.p_maxrip and v_Nocivo<1 then

            insert into dieta values(v_utente, p_cursor_row.p_id, i, j);
            v_calorieraggiunte := v_calorieraggiunte + p_cursor_row.CAL_P;
        end if;
      exit when p_cursor%notfound ;
      end loop;
      close p_cursor;
    end loop;
  end loop;
  commit;
end;
```

Oppure

```
(v_maxcal number, v_alimentonocivo alimenti.a_id%type,
 v_utente utenti.u_id%type) is

cursor p_cursor (curs_pasto in pietanze.p_pasto%type) is
    select p_id, p_maxrip, sum(a_calgr*c_qtagr) as CAL_P
    from pietanze, alimenti, composizione
    where p_id = c_id2
    and a_id = c_id1
    and p_pasto = curs_pasto
    and p_id not in (select distinct cc.id2
                     from alimenti aa, composizione cc
                     where aa.a_id = cc.c_id1 and aa.a_Nome = v_alimentonocivo)
    group by p_id
    order by CAL_P desc;

v_max number := 0;
v_rip number;
v_calorieraggiunte number;
p_cursor_row p_cursor%rowtype;

begin
for i in 1..7
  loop
  end if;
    for j in 1..3
    loop
      if j = 1 then
            v_max := (v_maxcal*0.30);
      elsif j = 2 then
            v_max := (v_maxcal*0.50);
      else  v_max := (v_maxcal*0.20);
      end if;

      v_calorieraggiunte := 0;
      open p_cursor(j);
      loop
      fetch p_cursor into p_cursor_row;

        select count(d_pietanza) into v_rip
        from dieta
        where d_pietanza = p_cursor_row.p_id
        and d_utente = v_utente;

        if (v_calorieraggiunte + p_cursor_row.CAL_P) < v_max
            and v_rip < p_cursor_row.p_maxrip then

            insert into dieta values(v_utente, p_cursor_row.p_id, i, j);
            v_calorieraggiunte := v_calorieraggiunte + p_cursor_row.CAL_P;
        end if;
      exit when p_cursor%notfound ;
      end loop;
      close p_cursor;
    end loop;
  end loop;
  commit;
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;

Plan Table
--------------------------------------------------------------------------------
| Operation                 |  Name    |  Rows | Bytes|  Cost  | Pstart| Pstop |
--------------------------------------------------------------------------------
| SELECT STATEMENT          |          |    25 |  875 |5659762 |       |       |
|  SORT GROUP BY            |          |    25 |  875 |5659762 |       |       |
|   HASH JOIN               |          |    60M|    1G|   4064 |       |       |
|    TABLE ACCESS FULL      |SUPPLIER  |    10K|   29K|     65 |       |       |
|    HASH JOIN              |          |   150K|    4M|   1092 |       |       |
|     TABLE ACCESS FULL     |NATION    |    25 |  725 |      1 |       |       |
|     TABLE ACCESS FULL     |CUSTOMER  |   150K|  439K|   1059 |       |       |
--------------------------------------------------------------------------------
```

$NP_{NATION} = \lceil 25 \times 105 / (4096 \times 0{,}69) \rceil = 1$
$NP_{CUSTORMER} = \lceil 150.000 \times 158 / (4096 \times 0{,}69) \rceil = 8.386$

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)

$NP_{CUSTOMER+NATION} = \lceil 150.000 \times (158+105) / (4096 \times 0{,}69) \rceil = 13.959$
$NP_{SUPPLIER} = \lceil 10.000 \times 143 / (4096 \times 0{,}69) \rceil = 506$

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