L’esercizio poteva essere risolto utilizzando una semplice query SQL oppure un cursore abbinato a una porzione di codice usato per calcolare il numero di mesi di possesso.

```
CURSOR cDatiIMU IS
SELECT RenditaCatastale, AddizionaleComunale, DataAcquisto, Quota, Coefficiente
FROM CATASTO, PROPRIETARI, POSSESSO, DESTINAZIONIUSO
WHERE PROPRIETARI.CF=vCF AND POSSESSO.CF=POSSESSO.CF AND POSSESSO.UID=CATASTO.UID AND
Tipo=Destinazione;

vDatiIMU cDatiIMU%ROWTYPE;
number vTotIMU:=0;
number vMesi;

BEGIN
  OPEN cDatiIMU;
  LOOP FETCH cDatiIMU INTO vDatiIMU;
  EXIT WHEN vDatiIMU %NOTFOUND;

    vMesi:= CEIL(TO_DATE(’31/12/2012’,’DD/MM/YYYY’) - vDatiIMU.DataAcquisto)/ 30
    IF vMesi>12 THEN
      vMesi := 12;
    ENDIF;
  vTotIMU:=vTotIMU + vDatiIMU.RenditaCatastale * vDatiIMU.Coefficiente
                       * (0.05+vDatiIMU.AddizionaleComunale) * vDatiIMU.Quota * vMesi/12;
  END LOOP;
  CLOSE cDatiIMU;
  DBMS_OUTLINE.PUT_LINE('Il totale dell IMU risulta ' || vTotIMU);
END;
```

Si fa notare che la sottraendo a 13 il mese di acquisto non si ottiene il valore corretto. Si verifichi il calcolo supponendo che l’immobile sia stato acquistato nel Novembre del 2011. I mesi di possesso effettivi sono 14 mentre questa soluzione restituirebbe 13-11=2.

```
explain plan for
SELECT O_CLERK,SUM(L_QUANTITY)
FROM ORDERS,LINEITEM
WHERE O_ORDERKEY=L_ORDERKEY AND L_PARTKEY=200
GROUP BY O_CLERK
ORDER BY O_CLERK;
@?/RDBMS/ADMIN/UTLXPLS;

Plan Table
--------------------------------------------------------------------------------
| Operation                 |  Name    |  Rows | Bytes|  Cost  | Pstart| Pstop |
--------------------------------------------------------------------------------
| SELECT STATEMENT          |          |    31 |    1K|     99 |       |       |
|  SORT ORDER BY            |          |    31 |    1K|     99 |       |       |
|   SORT GROUP BY           |          |    31 |    1K|     99 |       |       |
|    NESTED LOOPS           |          |    31 |    1K|     96 |       |       |
|     TABLE ACCESS BY INDEX |LINEITEM  |    31 |  434 |     34 |       |       |
|      INDEX RANGE SCAN     |IX_PART_L |    31 |      |      3 |       |       |
|     TABLE ACCESS BY INDEX |ORDERS    |     1M|   28M|      2 |       |       |
|      INDEX UNIQUE SCAN    |SYS_C0096 |     1M|      |      1 |       |       |
```

$NP_O = \lceil 1.500.000 \times 109/ (4096 \times 0{,}69) \rceil = 57.851$
$NP_{LI} = \lceil 6.001.215 \times 113/ (4096 \times 0{,}69) \rceil = 239.944$
$NL_{L\_PARTKEY} = \lceil (6.001.215 \times 4 + 200.000 \times 4) / (4.096 \times 0{,}69) \rceil = 8.777$
$NL_{O\_ORDERKEY} = \lceil (1.500.000 \times 4 + 1.500.000 \times 4) / (4.096 \times 0{,}69) \rceil = 4.246$

$SEL(L\_PARTKEY=200) = 1/200.000$
$NT_{LI\text{-}filtered}(L\_PARTKEY=200) = \lceil 6.001.215/200.000 \rceil = 31$
Costo di ogni accesso a ORDERS = 2-1 + $\lceil 4.246 / 1.500.000 \rceil + \lfloor 1/1.500.000 \times 57.851 \rceil = 3$

Ricordandosi che LINEITEM non è ordinata sulla L_PARTKEY

Costo(NestedLoop$_{LI+O}$)= 2-1 + $\lceil 1/200.000 \times 8.777 \rceil$ + **Φ(**6.001.215/200.000**,** 239.944**)** +
+ 31× 3 = 1+1+93=1+1+31+93=**126**

NT_{LI+O}=31
$NP_{LI+O} = \lceil 31 \times (113+109)/ (4096 \times 0{,}69) \rceil = 3$

$GROUP\ BY(O\_CLERK) = 2 \times 3 \times (\lceil \log_{100}(3) \rceil + 1) = 12$

In realtà, vista la dimensione della tabella, non è necessario utilizzare il sort z-vie ma è sufficiente utilizzare un sort in memoria centrale e il costo si ridurrebbe a 6 pagine. Sono accettate entrambe le soluzioni.

$NT_{GB(O\_CLERK)}$**= Φ(**31**,** 1.000**)=31**

Il group-by non riduce la cardinalità della tabella quindi il costo del successivo ordinamento è ancora pari a 6 oppure 12 pagine a seconda che si opti per ordinamento interno/esterno. In realtà il secondo ordinamento non è necessario perché l'operazione di group by fornisce lo stesso ordinamento richiesto in output. Sono accettate entrambe le soluzioni.

Costo Totale$_{Opzione1}$: 126+6=132 Costo Totale$_{Opzione2}$: 126+6+6=138
Costo Totale$_{Opzione3}$: 126+12=138 Costo Totale$_{Opzione4}$: 126+12+12=150