```
(par_cliente ordini.o_codcliente%type, par_data ordini.o_data%type) IS

CURSOR cursore_ordini IS
select d_ordine, sum (p_prezzo*d_quantita) AS totale_ordine
from dettagli, ordini, prodotti
where d_ordine = o_codice
and d_prodotto = p_cod
and o_data >= par_data
and o_codcliente = par_cliente
group by d_ordine;

v_totale_ordini prodotti.p_prezzo%type;

BEGIN

v_totale_ordini := 0;

FOR bookmark_ordine in cursore_ordini
LOOP
      if (bookmark_ordine.totale_ordine > 1500) then
            v_totale_ordini := v_totale_ordini + (
(bookmark_ordine.totale_ordine * 0.9) * 0.85) ;
      ELSIF (bookmark_ordine.totale_ordine > 300) then
            v_totale_ordini := v_totale_ordini + (bookmark_ordine.totale_ordine
* 0.9);
      ELSE
            v_totale_ordini := v_totale_ordini + bookmark_ordine.totale_ordine;
      END IF;

END LOOP;

if (v_totale_ordini > 5000) then
      v_totale_ordini := v_totale_ordini * 0.85;
elsif (v_totale_ordini > 1500) then
      v_totale_ordini := v_totale_ordini * 0.9;
end if;


DBMS_OUTPUT.PUT_LINE('Importo totale scontato: ' || v_totale_ordini);

END;
```

```
explain plan for select sum(O_TOTALPRICE)
from ORDERS,CUSTOMER
where O_CUSTKEY=C_CUSTKEY AND C_NAME LIKE 'A%'
group by O_CLERK;
@?/RDBMS/ADMIN/UTLXPLS;

Plan Table
--------------------------------------------------------------------------------
| Operation                 |  Name    |  Rows | Bytes|  Cost  | Pstart| Pstop |
--------------------------------------------------------------------------------
| SELECT STATEMENT          |          |    30 |    1K|     38 |       |       |
|  SORT GROUP BY            |          |    30 |    1K|     38 |       |       |
|   NESTED LOOPS            |          |    30 |    1K|     36 |       |       |
|    TABLE ACCESS BY INDEX R|CUSTOMER  |     2 |   48 |      4 |       |       |
|     INDEX RANGE SCAN      |INDXNAME  |     2 |      |      3 |       |       |
|    TABLE ACCESS BY INDEX R|ORDERS    |     1M|   38M|     16 |       |       |
|     INDEX RANGE SCAN      |IX_CUST_O |     1M|      |      2 |       |       |
```

$NP_{ORDERS} = \lceil 1.500.000 \times 106 / 4.096 \rceil = 38.819$
$NP_{CUSTOMER} = \lceil 150.000 \times 158 / 4.096 \rceil = 5.787$
Sel(C_NAME)=1/26

$NL_{INDXNAME} = \lceil (150.000 \times 4 + 4 \times 150.000) / 4.096 \rceil = 293$
$NL_{IX\_CUST\_ORDERS} = \lceil (150.000 \times 4 + 4 \times 1.500.000) / 4.096 \rceil = 1.612$

Accesso unclustered a CUSTOMER con indice
$2\text{-}1 + \lceil 1/26 \times 293 \rceil + \Phi(150.000/26, 5.787) = 1 + 12 + 3.652 = 3.665$

Accesso unclustered a ORDERS con indice *per ogni* cliente recuperato
$2\text{-}1 + \lceil 1/150.000 \times 1.612 \rceil + \Phi(1.500.000/150.000, 38.819) = 1 + 1 + 10 = 12$

Costo Nested Loop = $3.655 + 12 \times 150.000 / 26 =$ **72.886**

$NT_{PARTSUPP+PART} = \lceil 1.500.000 / 26 \rceil = 57.693$

$NP_{PARTSUPP+PART} = \lceil 57.693 \times (106+158) / 4.096 \rceil = 3.719$

Costo SORT GROUP BY= $2 \times 3.719 \times (\lceil \log_{100} 3.719 \rceil + 1) = 2 \times 3.719 \times 3 = 22.314$

Costo totale = 72.886 + 22.314 = 95.200