ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

Ομάδα 17

Δαδιώτης Δημήτριος - 216108

Ιωάννης Σκουλούδης - 2021091

Περιεχόμενα

1ο ερώτημα – Διαχείριση παραγγελιών **3**
i) Προσθήκη στήλης Address 3
ii) Προσθήκη στήλης ProductTypes 5
iii) Συνάρτηση mapToProductTypes 5
iv) Οντότητα order 7
v) Συνάρτηση υπολογισμού τελικού κέρδους παραγγελίας 8
vi)Συνάρτηση συγχώνευσης παραγγελιών 9
Παράδειγμα χρήσης της v) και vii) 10
vii)Procedure εντοπισμού πελάτη μέσω διεύθυνσης 12
Παράδειγμα χρήσης της fill_details 13
Παράδειγμα χρήσης της fill_details2 14
2ο Ερώτημα – Εξαγωγή σε XML **15**
3ο Ερώτημα – Ερωτήσεις XPath **16**
1) 16
2) 18
3) 19

1ο ερώτημα – Διαχείριση παραγγελιών

Ανοίξτε το αρχείο **2h_ergasia.sql**

i) Προσθήκη στήλης Address

```oracle_plsql
CREATE OR REPLACE TYPE ADDRESS AS OBJECT
(
city VARCHAR2(30),
street VARCHAR2(40),
num number(3)
);

alter table customers add address ADDRESS;

update customers set address=null;


create or replace procedure fill_address_column as

cursor c1 is
select customer_id from customers;

type cities_type is table of varchar(15);
type streets_type is table of varchar(30);
type code_type is table of number;


cities cities_type := cities_type('Athens', 'Thessaloniki', 'Patras', 'Heraklion', 'Larissa', 'Volos', 'Ioannina');

streets streets_type  := streets_type ('Adrianou Street', 'Athinas Street', 'Ermou Street', 'Panepistimiou Avenue',
'Patision Street', 'Vasilissis Sofias Avenue', 'Kifisias Avenue', 'Syngrou Avenue', 'Vouliagmenis
Avenue', 'Stadiou Street', 'Leoforos Alexandras', 'Akadimias Street', 'Solonos Street', 'Ploutarchou
Street', 'Ermou Street', 'Miaouli Street', 'Asklipiou Street','Filellinon Street', 'Tritis Septemvriou Street', 'Aiolou Street');

codes code_type := code_type();

i NUMBER := 1;

rand_city number;
rand_street number;
rand_num number;

new_address ADDRESS;
```

```oracle_plsql
begin

    while i<=100 loop

        codes.extend;
        codes(i) := i;
        i := i + 1;

    end loop;


    for cursor_row in c1 loop

        rand_city := trunc(dbms_random.value(1,cities.count)) ;
        rand_street := trunc(dbms_random.value(1,streets.count)) ;
        rand_num := trunc(dbms_random.value(1,codes.count)) ;


        new_address := ADDRESS(cities(rand_city), streets(rand_street), codes(rand_num));

        UPDATE Customers c
        SET   c.address = new_address
        WHERE c.customer_id = cursor_row.customer_id;

    end loop;

end;

execute  fill_address_column;


select c.customer_id,c.address.city,c.address.street,c.address.num
from customers c;
```

[Link με μεγαλύτερη εικόνα]

Δημιουργούμε το Object Address και το προσθέτουμε στον πίνακα customers, αρχικοποιούμε τις τιμές του Object σε null και στην συνέχεια με το procedure fill_address_column δηλώνουμε τυχαίες τιμές για City, Street και Number και γεμίζουμε το Address με αυτές για καθε row του πίνακα customers.Ύστερα τρέξτε το υπογραμμισμένο select query που φαίνεται στην εικόνα για να δείτε το address(city,street,number)ανα customer_id.

```
select c.customer_id,c.address.city,c.address.street,c.address.num
from customers c;
```

t Output × Query Result 3 ×

SQL | Fetched 50 rows in 0.026 seconds

CUSTOMER_ID	ADDRESS.CITY	ADDRESS.STREET	ADDRESS.NUM
21916	Larissa	PloutarchouStreet	96
37917	Larissa	Asklipiou Street	33
19253	Larissa	PloutarchouStreet	31
32637	Patras	Filellinon Street	46
46193	Athens	PloutarchouStreet	9
3305	Athens	Filellinon Street	4
50606	Patras	Akadimias Street	58
7718	Larissa	Miaouli Street	44
25496	Volos	Adrianou Street	91
42384	Thessaloniki	Ermou Street	83
13934	Heraklion	Ermou Street	96
21045	Patras	Stadiou Street	41
35267	Thessaloniki	PloutarchouStreet	8
41490	Heraklion	Leoforos Alexandras	17
8102	Patras	Athinas Street	82
50683	Heraklion	PloutarchouStreet	53
7795	Thessaloniki	Adrianou Street	46
2336	Volos	Leoforos Alexandras	50
25573	Athens	Solonos Street	90
42461	Volos	Ermou Street	58
19384	Larissa	Adrianou Street	11
49247	Athens	Syngrou Avenue	23
39716	Heraklion	Panepistimiou Avenue	11
25914	Volos	Adrianou Street	73

ii) Προσθήκη στήλης ProductTypes

```
create index categoryname_idx ON products(categoryname);
```

```
create type ProductTypesList as table of varchar(10);

alter table products
add producttypes ProductTypesList
nested table producttypes store as producttypes_table;
```

iii) Συνάρτηση mapToProductTypes

```
create or replace function  mapToProductTypes( categoryname VARCHAR2) return ProductTypesList is

producttypes ProductTypesList := ProductTypesList();

begin

if categoryname in('Recordable DVD Discs','Camcorders','Camera Batteries','Camera Media','Cameras') then
producttypes.extend;
producttypes(producttypes.LAST) := 'video';
end if;

if categoryname in('CD-ROM','Home Audio') then
producttypes.extend;
producttypes(producttypes.LAST) := 'audio';
end if;

if categoryname in('CD-ROM','Recordable DVD Discs','Bulk Pack Diskettes','Recordable CDs','Memory') then
producttypes.extend;
producttypes(producttypes.LAST) := 'storage';
end if;

if categoryname in('CD-ROM','Recordable DVD Discs','Game Consoles','Y Box Games','Y Box Accessories') then
producttypes.extend;
producttypes(producttypes.LAST) := 'games';
end if;

if categoryname in('Modems/Fax','Accessories','Desktop PCs','Memory','Operating Systems','Monitors','Portable PCs') then
producttypes.extend;
producttypes(producttypes.LAST) := 'computer';
end if;

if categoryname in('Documentation','Modems/Fax','Printer Supplies','Accessories') then
producttypes.extend;
producttypes(producttypes.LAST) := 'other';
end if;

return producttypes;

end mapToProductTypes;


DECLARE
  result ProductTypesList;

BEGIN

  result := mapToProductTypes('CD-ROM');


  FOR i IN 1..result.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE(result(i));
  END LOOP;

END;

UPDATE products p
SET   p.producttypes = mapToProductTypes( p.categoryname );

select p.product_id,p.categoryname, p.producttypes
from products p;
```

[Link με μεγαλύτερη εικόνα]

Δημιουργούμε το function mapToProductTypes το οποίο αντιστοιχεί τις κατηγορίες video, audio, storage, games, computer και other αναλογα με το categoryname του προϊόντος και επιστρέφει την λιστα με τις κατηγορίες στις οποίες ανήκει.Δεξιά της εικόνας υπάρχει το παράδειγμα δίνοντας ως παραμετρο το ονομα CD-ROM και στη συνέχεια τυπώνουμε κάθε στοιχείο της λίστας που επιστρέφει η συνάρτηση.Το αποτέλεσμα είναι το εξής:

```
DECLARE
  result ProductTypesList;

BEGIN

  result := mapToProductTypes('CD-ROM');


  FOR i IN 1..result.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE(result(i));
  END LOOP;

END;
```

Script Output

Task completed in 1,209 seconds

```
audio
storage
games


PL/SQL procedure successfully completed.
```

Στο τέλος αφου βεβαιωθείτε ότι η συνάρτηση λειτουργεί,κάνουμε update την στήλη **p.producttypes** του πίνακα products p για κάθε προϊόν

```
select p.product_id,p.categoryname, p.producttypes
from products p;
```

t Output × | Query Result ×

SQL | Fetched 50 rows in 0.028 seconds

PRODUCT_ID	CATEGORYNAME	PRODUCTTYPES
15	Desktop PCs	IT216108.PRODUCTTYPESLIST('computer')
28	Operating Systems	IT216108.PRODUCTTYPESLIST('computer')
113	Recordable CDs	IT216108.PRODUCTTYPESLIST('storage')
114	Recordable CDs	IT216108.PRODUCTTYPESLIST('storage')
115	Recordable CDs	IT216108.PRODUCTTYPESLIST('storage')
116	Recordable CDs	IT216108.PRODUCTTYPESLIST('storage')
117	Recordable CDs	IT216108.PRODUCTTYPESLIST('storage')
118	Recordable CDs	IT216108.PRODUCTTYPESLIST('storage')
119	Recordable CDs	IT216108.PRODUCTTYPESLIST('storage')
20	Home Audio	IT216108.PRODUCTTYPESLIST('audio')
22	Accessories	IT216108.PRODUCTTYPESLIST('computer', 'other')
24	Modems/Fax	IT216108.PRODUCTTYPESLIST('computer', 'other')
27	Accessories	IT216108.PRODUCTTYPESLIST('computer', 'other')
29	Home Audio	IT216108.PRODUCTTYPESLIST('audio')
32	Accessories	IT216108.PRODUCTTYPESLIST('computer', 'other')
33	Modems/Fax	IT216108.PRODUCTTYPESLIST('computer', 'other')
46	Accessories	IT216108.PRODUCTTYPESLIST('computer', 'other')
47	Accessories	IT216108.PRODUCTTYPESLIST('computer', 'other')
147	Y Box Accessories	IT216108.PRODUCTTYPESLIST('games')
148	Y Box Accessories	IT216108.PRODUCTTYPESLIST('games')
18	Portable PCs	IT216108.PRODUCTTYPESLIST('computer')
19	Accessories	IT216108.PRODUCTTYPESLIST('computer', 'other')
23	Accessories	IT216108.PRODUCTTYPESLIST('computer', 'other')
30	Accessories	IT216108.PRODUCTTYPESLIST('computer', 'other')
31	Accessories	IT216108.PRODUCTTYPESLIST('computer', 'other')
34	CD-ROM	IT216108.PRODUCTTYPESLIST('audio', 'storage', 'games')
35	CD-ROM	IT216108.PRODUCTTYPESLIST('audio', 'storage', 'games')
36	CD-ROM	IT216108.PRODUCTTYPESLIST('audio', 'storage', 'games')
37	CD-ROM	IT216108.PRODUCTTYPESLIST('audio', 'storage', 'games')
38	CD-ROM	IT216108.PRODUCTTYPESLIST('audio', 'storage', 'games')

iv) Οντότητα order

Αρχικά δημιουργούμε τα κατάλληλα types **product_type**, **order_item**, **order_item_list**, **order_type** και στη συνέχεια δημιουργούμε τον πίνακα **order_object_type**

```
create type arrayproducttypes as varray(6) of varchar(10);

create or replace type product_type as object
(
product_id number,
productname VARCHAR2(46),
categoryname VARCHAR2(50),
listprice VARCHAR2(10),
producttypes arrayproducttypes
);

create or replace type order_item as object
(
days_to_process NUMBER,
price NUMBER(10,2),
cost NUMBER(10,2),
channel VARCHAR2(20),
product product_type
);

create type order_item_list as table of order_item;

create or replace type order_type as object (
    order_id number,
    customer_id number,
    items order_item_list
);

create table order_objects_table of order_type
nested table items store as items_table;
```

v) Συνάρτηση υπολογισμού τελικού κέρδους παραγγελίας

Πριν τρέξετε τις συναρτήσεις ανοίξτε το αρχείο **order_object_inserts** στο οποίο υπάρχουν ενδεικτικά inserts στον πίνακα **order_objects_table** και τρέξτε το.

Δημιουργούμε το πακέτο **order_package** και μέσα σε αυτό ορίζουμε δύο συναρτήσεις,όπου η μια από αυτές είναι η **object_order_profit** παίρνει ως παράμετρο ένα order_id απο τον πίνακα **order_objects_table** και υπολογίζει το τελικο κερδος της παραγγελίας αφαιρώντας από το **price** της κάθε order_item που υπάρχει στην παραγγελία το **cost**.

```
create or replace package order_package as
function object_order_profit(v_order_id number)return number;
function merge_orders(a number,b number) return number;
end order_package;

create or replace package body order_package as

    function object_order_profit(v_order_id number)return number is

        profit number:=0;

        cursor cl is
        select t.price,t.cost
        from order_objects_table oo,table(oo.items) t
        where oo.order_id=v_order_id;

        begin

            for cursor_row in cl loop

                profit:= profit +(cursor_row.price - cursor_row.cost);

            end loop;

        return profit;
end object_order_profit;

    function merge_orders(a number,b number) return number is

        new_profit number;

        a_order order_type;
        b_order order_type;
        begin

            select value(o) into a_order
            from order_objects_table o
            where o.order_id =a;

            select value(o) into b_order
            from order_objects_table o
            where o.order_id = b;

            if a_order is null or b_order  is null then
                new_profit := 0;
                return new_profit;
            end if;

```

Παρακάτω υπάρχει παράδειγμα για την v) και vi)

vi)Συνάρτηση συγχώνευσης παραγγελιών

Δημιουργούμε την συνάρτηση **merge_orders(a,b)** μεσα στο πακέτο **order_package** η οποία παίρνει δυο order_id a και b βρίσκει τα order_item που βρίσκονται στην στήλη items κάθε παραγγελίας και προσθέτει τα order_item της a στην b και επιστρέφει το καινούργιο συνολικό κέρδος της b.

```
                profit:= profit +(cursor_row.price - cursor_row.cost);

            end loop;

        return profit;
end object_order_profit;

    function merge_orders(a number,b number) return number is

        new_profit number;

        a_order order_type;
        b_order order_type;
        begin

            select value(o) into a_order
            from order_objects_table o
            where o.order_id =a;

            select value(o) into b_order
            from order_objects_table o
            where o.order_id = b;

            if a_order is null or b_order  is null then
                new_profit := 0;
                return new_profit;
            end if;

            for i in 1..a_order.items.count loop
                b_order.items.extend;
                b_order.items(b_order.items.last) := a_order.items(i);

            end loop;

            UPDATE order_objects_table o
            SET o.items = b_order.items
            WHERE o.order_id = b;

            new_profit :=order_package.object_order_profit(b);


    return new_profit;
end merge_orders;

end order_package;
```

Παράδειγμα χρήσης της v) και vii)

Σε αυτό το παράδειγμα καλούμε την **object_order_profit** για τις παραγγελιες 147,169,5592,5593 του order_objects_table και στην συνέχεια συγχωνεύουμε την 5592 στην 5593 με την **merge_orders**.Τέλος καλούμε ξανά την **object_order_profit** για να βεβαιωθούμε ότι το νέο τελικό κέρδος που επιστρέφει η **merge_orders** είναι σωστό **.**

```
SET SERVEROUTPUT ON;

begin
DBMS_OUTPUT.PUT_LINE('order 147  profit:' || order_package.object_order_profit(147));
DBMS_OUTPUT.PUT_LINE('order 169  profit:' || order_package.object_order_profit(169));
DBMS_OUTPUT.PUT_LINE('order 5592 profit:' || order_package.object_order_profit(5592));
DBMS_OUTPUT.PUT_LINE('order 5593 profit:' ||order_package.object_order_profit(5593));
DBMS_OUTPUT.PUT_LINE('new profit for order 5592 merged into order 5593 items is: '|| order_package.merge_orders(5592,5593));
DBMS_OUTPUT.PUT_LINE('order 5593 new profit:' || order_package.object_order_profit(5593));
end;


--vii)
```

Script Output | Task completed in 0,062 seconds

```
order 147  profit:65
order 169  profit:77
order 5592 profit:110
order 5593 profit:80
new profit for order 5592 merged into order 5593 items is: 190
order 5593 new profit:190


PL/SQL procedure successfully completed.
```

Η παραγγελία 5593 πριν καλέσουμε την **merge_orders(5592,5593)**:

2h_ergasia.sql | order_object_inserts.sql | ORDER_OBJECTS_TABLE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	ORDER_ID	CUSTOMER_ID	ITEMS
1	133	72	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
2	134	72	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
3	135	72	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
4	136	72	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT202...
5	147	80	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
6	166	92	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
7	167	92	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
8	168	92	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
9	169	92	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT202...
10	170	95	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
11	171	95	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
12	5592	2519	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
13	5593	2519	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])

```
SELECT o.order_id,o.customer_id,t.days_to_process,t.price,t.cost,t.channel,t.product.product_id,t.product.productname,t.product.categoryname,t.product.listprice,t.product.producttypes
from order_objects_table o,table(o.items) t
where o.order_id=5593;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0,013 seconds

	ORDER_ID	CUSTOMER_ID	DAYS_TO_PROCESS	PRICE	COST	CHANNEL	PRODUCT.PRODUCT_ID	PRODUCT.PRODUCTNAME	PRODUCT.CATEGORYNAME	PRODUCT.LISTPRICE	PRODUCT.PRODUCTTYPES
1	5593	2519	0	629	549	Direct Sales	20	Home Theatre Package with DVD-Audio/Video Play	Home Audio	599.99	IT2021091.ARRAYPRODUCTTYPES('audio')

Η παραγγελία 5593 μετά την **merge_orders(5592,5593):**

2h_ergasia.sql | order_object_inserts.sql | ORDER_OBJECTS_TABLE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	ORDER_ID	CUSTOMER_ID	ITEMS
1	133	72	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
2	134	72	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
3	135	72	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
4	136	72	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT202...
5	147	80	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
6	166	92	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
7	167	92	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
8	168	92	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
9	169	92	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT202...
10	170	95	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
11	171	95	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])
12	5592	2519	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM])
13	5593	2519	IT2021091.ORDER_ITEM_LIST([IT2021091.ORDER_ITEM], [IT2021091.ORDER_ITEM])

```
SELECT o.order_id,o.customer_id,t.days_to_process,t.price,t.cost,t.channel,t.product.product_id,t.product.productname,t.product.categoryname,t.product.listprice,t.product.producttypes
from order_objects_table o,table(o.items) t
where o.order_id=5593;
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0,011 seconds

	ORDER_ID	CUSTOMER_ID	DAYS_TO_PROCESS	PRICE	COST	CHANNEL	PRODUCT.PRODUCT_ID	PRODUCT.PRODUCTNAME	PRODUCT.CATEGORYNAME	PRODUCT.LISTPRICE	PRODUCT.PRODUCTTYPES
1	5593	2519	0	629	549	Direct Sales	20	Home Theatre Package with DVD-Audio/Video Play	Home Audio	599.99	IT2021091.ARRAYPRODUCTTYPES('audio')
2	5593	2519	0	609	499	Partners	20	Home Theatre Package with DVD-Audio/Video Play	Home Audio	599.99	IT2021091.ARRAYPRODUCTTYPES('audio')

Βλέπουμε πως το orde_item της 5592 έχει προστεθεί στην order_item_list της **5593**.

vii)Procedure εντοπισμού πελάτη μέσω διεύθυνσης

Αρχικά δημιουργούμε τον πίνακα **customer_details** που αποθηκεύονται τα αποτελέσματα

```
--vii)
drop table customer_order_details;

create table customer_order_details
(
customer_id number,
address varchar2(80),
product_count number,
orders_profit number
);

delete from customer_order_details;
```

Στη συνέχεια δημιουργούμε την procedure **fill_details(v_city varchar,v_street varchar,v_code number)** όπου δίνουμε την πόλη,την οδό και τον αριθμό ως παραμέτρους και τυπώνει για κάθε πελάτη που αντιστοιχεί στην διεύθυνση την διεύθυνση ως ενα string,το πλήθος των product που

παρήγγειλε και το συνολικό κέρδος από τις παραγγελίες που έκανε και στο τέλος καταχωρεί τον κάθε πελάτη που βρήκε στον c**ustomer_details** με τις αντίστοιχες πληροφορίες και το customer_id του.Έχουμε υλοποιήσει αυτη την procedure με δύο τρόπους.Η **fill_details** λειτουργεί με τον πίνακα **order_objects_table**,ενώ η **fill_details2** λειτουργεί με τον πίνακα orders.

Παράδειγμα χρήσης της fill_details

```
select c.address.city,c.address.street,c.address.num
from customers c
where c.customer_id=72;


EXECUTE fill_details('Volos','Solonos Street', 8);
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 1 in 0,01 seconds

	ADDRESS.CITY	ADDRESS.STREET	ADDRESS.NUM
1	Volos	Solonos Street	8

Κάνοντας execute την fill_details μας τυπώνει αυτό:

```
EXECUTE fill_details('Volos','Solonos Street', 8);
```

Script Output × | Query Result ×

Task completed in 0,158 seconds

```
address is:Volos,Solonos Street,8
profit is:
count of products is:
address is:Volos,Solonos Street,8
profit is:219
count of products is:13
address is:Volos,Solonos Street,8
profit is:
count of products is:
address is:Volos,Solonos Street,8
profit is:
count of products is:
address is:Volos,Solonos Street,8
profit is:
count of products is:
address is:Volos,Solonos Street,8
profit is:
count of products is:


PL/SQL procedure successfully completed.
```

2h_ergasia.sql | order_object_inserts.sql | CUSTOMER_ORDER_DETAILS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes

Sort.. | Filter:

	CUSTOMER_ID	ADDRESS	PRODUCT_COUNT	ORDERS_PROFIT
1	7046	Volos,Solonos Street,8	(null)	(null)
2	72	Volos,Solonos Street,8	13	219
3	50987	Volos,Solonos Street,8	(null)	(null)
4	16280	Volos,Solonos Street,8	(null)	(null)
5	48918	Volos,Solonos Street,8	(null)	(null)
6	101627	Volos,Solonos Street,8	(null)	(null)

Ο λόγος που εμφανίζει κι άλλους χρήστες με nulls είναι διότι βρήκε κι άλλους εκτός από τον customer με id=**72** στον πίνακα **customers** με την συγκεκριμένη διεύθυνση.Παρόλα αυτά μόνο ο customer 72 είναι καταχωρημένος στον **order_objects_table** οπότε αντλεί δεδομένα μόνο για αυτον τον πελάτη.

Παράδειγμα χρήσης της fill_details2

Αν τρέξουμε την fill_details2 όπως στο προηγούμενο παράδειγμα βλέπουμε ότι αφου πρώτα αδειάσουμε τον πίνακα customer_details:

```
387
388 EXECUTE fill_details2('Volos','Solonos Street', 8);
389
```

Script Output | Task completed in 0,534 seconds

```
address is:Volos,Solonos Street,8
profit is:
count of products is:0
address is:Volos,Solonos Street,8
profit is:227,87
count of products is:13
address is:Volos,Solonos Street,8
profit is:
count of products is:0
address is:Volos,Solonos Street,8
profit is:3654,38
count of products is:85
address is:Volos,Solonos Street,8
profit is:
count of products is:0
address is:Volos,Solonos Street,8
profit is:
count of products is:0
```

2h_ergasia.sql | order_object_inserts.sql | CUSTOMER_ORDER_DETAILS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	CUSTOMER_ID	ADDRESS	PRODUCT_COUNT	ORDERS_PROFIT
1	7046	Volos,Solonos Street,8	0	(null)
2	72	Volos,Solonos Street,8	13	227,87
3	50987	Volos,Solonos Street,8	0	(null)
4	16280	Volos,Solonos Street,8	85	3654,38
5	48918	Volos,Solonos Street,8	0	(null)
6	101627	Volos,Solonos Street,8	0	(null)

Τώρα βλέπουμε ότι καποιοι πελάτες έχουν ακόμα null και 0 στα orders_profit και product_count αντίστοιχα.Αυτό γίνεται γιατί οι συγκεκριμένοι πελάτες δεν έχουν κάνει κάποια παραγγελία για να έχουν product_count και orders_profit,δηλαδή δεν υπάρχουν στον πίνακα **orders**.(Υπενθυμίζω πως οι αναθέσεις στις διευθύνσης είναι τυχαίες επειδή έτσι ζητάει το ερώτημα οπότε μην περιμένετε ίδια αποτελέσματα με τα ενδεικτικά screenshots αν το τρέξετε μόνοι σας.)

2ο Ερώτημα – Εξαγωγή σε XML

Ανοίξτε το αρχείο **xml.sql**

Αρχικά δημιουργούμε τον πίνακα **customer_orders** που περιέχει τα διακριτά προϊόντα κάθε πελάτη που έχει αγοράσει και του έχουν παραδοθεί εντός 20 ημερών το πολύ.Στη συνέχεια τρέχουμε το script για να δημιουργήσουμε το xml αρχείο που φέρνει τους 30 πρώτους πελάτες(το αρχείο έβγαινε πολύ μεγάλο).

Worksheet | Query Builder

```
create table customer_orders as
select distinct p.product_id,o.customer_id
from orders o join products p on o.product_id=p.product_id
where o.days_to_process<=20
order by customer_id;


create table test1 as
SELECT    distinct XMLElement("customers",
                    XMLAGG(
                        XMLElement("customer",XMLAttributes(c.customer_id as "id",c.marital_status as "maritalstatus",c.gender as "gender"
                        ), XMLFOREST(
                               c.age_group as "agegroup",
                               c.income_level as "incomelevel"


                        ) ,XMLELEMENT("address",XMLForest(
                                c.address.city AS "city",
                                c.address.street AS "street",
                                c.address.num AS "number"
                            )) ,XMLElement("products",
                                    XMLAgg(
                                   XMLElement("product",
                                       XMLAttributes(p.product_id as "id"),
                                       XMLForest(
                                           p.productname AS "productname",
                                           p.categoryname AS "productcategory"),
                                       XMLELEMENT("producttypes",(SELECT XMLAGG(XMLELEMENT("producttype",t.column_value))from products po,table(po.producttypes)t where po.product_id=p.product_id  )
                                               )



                                               )
                                           )

                                       )



                     )
                    )
                   ).getClobVal() AS xml_output
from customers c join customer_orders co on c.customer_id=co.customer_id join products p on co.product_id=p.product_id
where c.customer_id<=30
group by c.customer_id,c.marital_status ,c.gender,c.age_group,c.income_level,c.address.city ,c.address.street ,c.address.num;
```

3ο Ερώτημα – Ερωτήσεις XPath

Αν δεν θέλετε να τρέξετε το script και για να συμβαδίζουν τα αποτελέσματα σας απο τις ερωτήσεις xpath με αυτα απο τα screenshots(γιατι αν αποφασίσετε να τρέξετε το script αφού χρησιμοποιήσετε και την procedure για να γεμίσετε τον πίνακα customers με addresses θα έχουν άλλες τυχαίες τιμές) καντε copy paste το έτοιμο xml στο αρχειο **xml_output** που έχει δημιουργηθεί τρέχοντας το script και ύστερα κάνοντας το format απο αυτο το site:

https://www.freeformatter.com/xml-formatter.html

Για τα ερωτήματα xpath χρησιμοποιήθηκε αυτό το site:

http://xpather.com/

1)

Αρχικά τρέχουμε **//customer[agegroup='Above 70' and @gender='Male']/@id** και βλέπουμε ότι υπάρχουν δύο άντρες με agegroup πανω απο 70 σε αυτο το xml αρχείο.

```
//customer[agegroup='Above 70' and @gender='Male']/@id
```

Elements found: 2

XML mode | Format | Save

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer id="2" maritalstatus="Unknown" gender="Female">
    <agegroup>60-70</agegroup>
    <incomelevel>medium</incomelevel>
    <address>
      <city>Heraklion</city>
      <street>Tritis Septemvriou Street</street>
      <number>29</number>
    </address>
    <products>
      <product id="13">
        <productname>5MP Telephoto Digital Camera</productname>
        <productcategory>Cameras</productcategory>
        <producttypes>
```

1. 3
2. 4

Ύστερα τρέχουμε το **//customer[@gender='Male'and agegroup='Above 70']/products/product[productcategory='Monitors']/../../@id**

```
//customer[ @gender='Male'and agegroup='Above 70']/products/product[productcategory='Monitors']/../../@id
```

No elements found.

XML mode | Format | Save

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer id="2" maritalstatus="Unknown" gender="Female">
    <agegroup>60-70</agegroup>
    <incomelevel>medium</incomelevel>
    <address>
      <city>Heraklion</city>
      <street>Tritis Septemvriou Street</street>
      <number>29</number>
    </address>
    <products>
      <product id="13">
        <productname>5MP Telephoto Digital Camera</productname>
```

Δεν εμφανίζει τίποτα διότι δεν υπάρχουν αντρες πανω απο 70 που να αγόρασαν Monitors στο συγκεκριμένο xml.Αν τρέξουμε την ίδια ερώτηση με διαφορετικές τιμές θα δούμε ότι δίνει αποτέλεσμα.Για παράδειγμα

- **//customer[@gender='Female'and agegroup='60-70']/products/product[productcategory='Cameras']/../../@id**

```
//customer[ @gender='Female'and agegroup='60-70']/products/product[productcategory='Cameras']/../../@id
```

Elements found: 2

XML mode | Format | Save

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer id="2" maritalstatus="Unknown" gender="Female">
    <agegroup>60-70</agegroup>
    <incomelevel>medium</incomelevel>
    <address>
      <city>Heraklion</city>
      <street>Tritis Septemvriou Street</street>
      <number>29</number>
    </address>
    <products>
      <product id="13">
        <productname>5MP Telephoto Digital Camera</productname>
        <productcategory>Cameras</productcategory>
```

1. 2
2. 11

- **//customer[@gender='Male'and agegroup='Above**

70']/products/product[productcategory='Accessories']/../../@id

```
//customer[ @gender='Male'and agegroup='Above 70']/products/product[productcategory='Accessories']/../../@id
```

Elements found: 2

XML mode | Format | Save

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer id="2" maritalstatus="Unknown" gender="Female">
    <agegroup>60-70</agegroup>
    <incomelevel>medium</incomelevel>
    <address>
      <city>Heraklion</city>
      <street>Tritis Septemvriou Street</street>
      <number>29</number>
    </address>
    <products>
      <product id="13">
        <productname>5MP Telephoto Digital Camera</productname>
```

1. 3
2. 4

2)

Για αυτό το ερώτημα τρέχουμε το

//customer[@gender='Female']//products/product[producttypes/producttype='games']/productcategory

```
//customer[@gender='Female']//products/product[producttypes/producttype='games']/productcategory
```

Elements found: 117

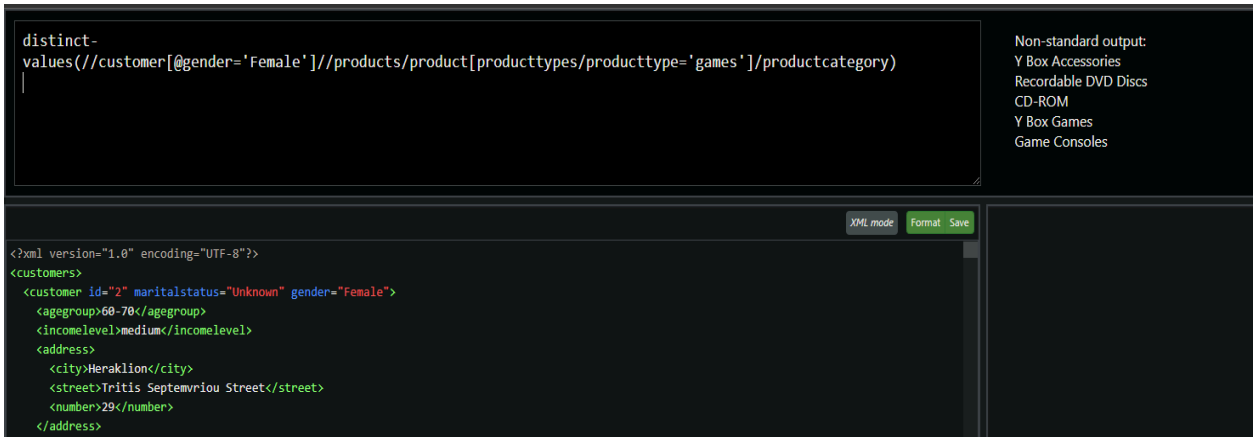
XML mode | Format | Save

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer id="2" maritalstatus="Unknown" gender="Female">
    <agegroup>60-70</agegroup>
    <incomelevel>medium</incomelevel>
    <address>
      <city>Heraklion</city>
      <street>Tritis Septemvriou Street</street>
      <number>29</number>
    </address>
    <products>
      <product id="13">
        <productname>5MP Telephoto Digital Camera</productname>
        <productcategory>Cameras</productcategory>
        <producttypes>
          <producttype>video</producttype>
        </producttypes>
      </product>
      <product id="148">
        <productname>Xtend Memory</productname>
        <productcategory>Y Box Accessories</productcategory>
        <producttypes>
          <producttype>games</producttype>
        </producttypes>
      </product>
      <product id="133">
        <productname>Model K8822S Cordless Phone Battery</productname>
        <productcategory>Camera Batteries</productcategory>
        <producttypes>
          <producttype>video</producttype>
        </producttypes>
      </product>
      <product id="132">
        <productname>Model C9827B Cordless Phone Battery</productname>
        <productcategory>Camera Batteries</productcategory>
        <producttypes>
          <producttype>video</producttype>
        </producttypes>
      </product>
      <product id="131">
```

1. Y Box Accessories
2. Recordable DVD Discs
3. Recordable DVD Discs
4. Recordable DVD Discs
5. Recordable DVD Discs
6. Recordable DVD Discs
7. Recordable DVD Discs
8. CD-ROM
9. Y Box Accessories
10. Y Box Accessories
11. Y Box Games
12. Y Box Games
13. Y Box Games
14. Y Box Games
15. Y Box Games
16. Y Box Games
17. Y Box Games
18. Recordable DVD Discs
19. Recordable DVD Discs
20. Recordable DVD Discs
21. Recordable DVD Discs
22. Recordable DVD Discs
23. CD-ROM
24. CD-ROM
25. CD-ROM
26. Game Consoles
27. Y Box Accessories
28. Y Box Accessories
29. Y Box Games
30. Y Box Games
31. Y Box Games
32. Y Box Games
33. Y Box Games
34. Y Box Games
35. Recordable DVD Discs
36. Recordable DVD Discs
37. CD-ROM
38. Y Box Accessories

Κι αν δεν θέλετε να δείτε διπλότυπα τρέξτε:

distinct-values(//customer[@gender='Female']//products/product[producttypes/producttype='games']/productcategory)



3)

Για αυτό το ερώτημα τρέξτε: **//customer[incomelevel='high' and address/city='Volos' and address/street='Ermou Street']//products/product/producttypes/producttype**

Σε αυτό το αρχείο δεν υπάρχει καποιος customer που να έχει income level=high και να μένει στο Βόλο στην οδό Ερμού και γι'αυτό δεν επιστρέφει κάποιο αποτέλεσμα.Αντιθέτως αν αλλάξουμε την πολη και την οδό θα έχουμε: **//customer[incomelevel='high' and address/city='Larissa' and address/street='Filellinon Street']//products/product/producttypes/producttype**

```
//customer[incomelevel='high' and address/city='Larissa' and address/street='Filellinon
Street']//products/product/producttypes/producttype
```

Elements found: 47

XML mode | Format | Save

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer id="2" maritalstatus="Unknown" gender="Female">
    <agegroup>60-70</agegroup>
    <incomelevel>medium</incomelevel>
    <address>
      <city>Heraklion</city>
      <street>Tritis Septemvriou Street</street>
      <number>29</number>
    </address>
    <products>
      <product id="13">
        <productname>5MP Telephoto Digital Camera</productname>
        <productcategory>Cameras</productcategory>
        <producttypes>
          <producttype>video</producttype>
        </producttypes>
      </product>
      <product id="148">
        <productname>Xtend Memory</productname>
        <productcategory>Y Box Accessories</productcategory>
        <producttypes>
          <producttype>games</producttype>
        </producttypes>
      </product>
      <product id="133">
        <productname>Model K88225 Cordless Phone Battery</productname>
        <productcategory>Camera Batteries</productcategory>
        <producttypes>
          <producttype>video</producttype>
        </producttypes>
      </product>
      <product id="132">
        <productname>Model C9827B Cordless Phone Battery</productname>
        <productcategory>Camera Batteries</productcategory>
```

1. video
2. games
3. games
4. games
5. games
6. games
7. games
8. games
9. games
10. games
11. video
12. video
13. video
14. video
15. video
16. video
17. other
18. video
19. storage
20. games
21. video
22. storage
23. games
24. video
25. storage
26. games
27. video
28. storage
29. games
30. computer
31. other
32. audio
33. storage

Για αποτέλεσμα χωρίς διπλότυπα τρέξτε:**distinct-values(//customer[incomelevel='high' and address/city='Larissa' and address/street='Filellinon Street']//products/product/producttypes/producttype)**

```
/distinct-values(//customer[incomelevel='high' and address/city='Larissa' and address/street='Filellinon Street']//products/product/producttypes/producttype)
```

Non-standard output:
video
games
other
storage
computer
audio

XML mode Format Save

```
  </products>
</customer>
<customer id="27" maritalstatus="Single" gender="Female">
  <agegroup>Above 70</agegroup>
  <incomelevel>high</incomelevel>
  <address>
    <city>Larissa</city>
    <street>Filellinon Street</street>
    <number>36</number>
  </address>
  <products>
    <product id="13">
      <productname>5MP Telephoto Digital Camera</productname>
      <productcategory>Cameras</productcategory>
      <producttypes>
        <producttype>video</producttype>
      </producttypes>
    </product>
    <product id="148">
      <productname>Xtend Memory</productname>
      <productcategory>Y Box Accessories</productcategory>
      <producttypes>
        <producttype>games</producttype>
```