

10/1/2018

PROJECT ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Το παρόν αποτελεί μια ελάχιστα τροποποιημένη έκδοση της εξαμηνιαίας εργασίας στο μάθημα «Βάσεις Δεδομένων II» και έχει ως στόχο την εκπαίδευση σε ORACLE SQL και PL/SQL. Οι πλατφόρμες που χρησιμοποιήθηκαν είναι οι oracle apex και Oracle 11g XE. Συμπεριλαμβάνονται και τα scripts για τη δημιουργία των αρχικών πινάκων και την εκχώρηση των τιμών τους.

Η εργασία έγινε σε ατομικό επίπεδο και βαθμολογήθηκε με άριστα. Ωστόσο, αυτό δεν αναιρεί την πιθανή ύπαρξη λαθών. Κάθε προτάσεις διόρθωσης αλλά και βελτίωσης, είναι πλήρως ευπρόσδεκτες. Ακόμα, ενθαρρύνω τη συγγραφή νέων ερωτημάτων σε SQL, αλλά ακόμα περισσότερο σε PL/SQL.

Με εκτίμηση

Τασούλας Θεοφάνης

1. Ποιες είναι οι δομές κατάρτισης (τίτλοι στηλών: Κωδικός Δομής, Επωνυμία, Πόλη) που είναι Ανώνυμες Εταιρίες (Α.Ε.) και έχουν έδρα την «Πάτρα»;

SELECT Center_ID, Title, City

FROM TRAINING_CENTER

WHERE City='Πάτρα' AND Title LIKE '%Α.Ε.%'

The screenshot shows the Oracle SQL Workshop interface. The SQL command entered is: `SELECT Center_ID, Title, City FROM TRAINING_CENTER WHERE City='Πάτρα' AND Title LIKE '%Α.Ε.%'`. The query has been executed, returning 1 row. The results table has three columns: CENTER_ID, TITLE, and CITY. The single row returned is: 512, Οριζοντες Α.Ε., Πάτρα.

CENTER_ID	TITLE	CITY
512	Οριζοντες Α.Ε.	Πάτρα

1 rows returned in 0.01 seconds

2. Να βρεθεί η μέση αξία (μέσος όρος) των επιταγών κατάρτισης όλων των προσκλήσεων (τίτλοι στηλών: Μέση Αξία Επιταγών Κατάρτισης).

SELECT AVG(Voucher_Value) FROM VOUCHER_CALL

The screenshot shows the Oracle SQL Workshop interface. The SQL command entered is: `SELECT AVG(Voucher_Value) FROM VOUCHER_CALL`. The query has been executed, returning 1 row. The results table has one column: AVG(VOUCHER_VALUE). The single row returned is: 4856.

AVG(VOUCHER_VALUE)
4856

1 rows returned in 0.01 seconds

3. Πόσα τμήματα με ωράριο «Πρωινό» βρίσκονται σε κατάσταση «Ολοκληρωμένο» (τίτλοι στηλών: Αριθμός Πρωϊνών Τμημάτων που Ολοκληρώθηκαν);

```
SELECT COUNT(Class_ID)
```

```
FROM VOUCHER_CLASS
```

```
WHERE Time_Zone = 'Πρωινό' AND Status = 'Ολοκληρωμένο'
```

The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the query: `SELECT COUNT(Class_ID) FROM VOUCHER_CLASS WHERE Time_Zone = 'Πρωινό' AND Status = 'Ολοκληρωμένο'`. The Results tab shows a single row with the value 2 under the column heading COUNT(CLASS_ID). The interface includes a top navigation bar with tabs like Application Express, App Builder, SQL Workshop, Team Development, and Packaged Apps. A schema dropdown is set to BASEIS2. The bottom status bar indicates 1 row returned in 0.02 seconds.

COUNT(CLASS_ID)
2

4. Να βρεθούν οι ωφελούμενοι (τίτλοι στηλών: ΑΦΜ, Ονοματεπώνυμο, Φύλο, Ηλικία) που έχουν ηλικία μεταξύ 20 και 29 ετών.

```
SELECT TAX_ID, SEX, TO_NUMBER(TO_CHAR(sysdate, 'YYYY')) -  
BIRTH_YEAR AS AGE
```

```
FROM BENEFICIARY
```

```
WHERE TO_NUMBER(TO_CHAR(sysdate, 'YYYY')) - BIRTH_YEAR >= 20
```

```
AND TO_NUMBER(TO_CHAR(sysdate, 'YYYY')) - BIRTH_YEAR <= 29
```

The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the query: `SELECT TAX_ID, SEX, TO_NUMBER(TO_CHAR(sysdate, 'YYYY')) - BIRTH_YEAR AS AGE FROM BENEFICIARY WHERE TO_NUMBER(TO_CHAR(sysdate, 'YYYY')) - BIRTH_YEAR >= 20 AND TO_NUMBER(TO_CHAR(sysdate, 'YYYY')) - BIRTH_YEAR <= 29`. The Results tab shows a table with 5 rows. The columns are TAX_ID, SEX, and AGE. The interface includes a top navigation bar with tabs like Application Express, App Builder, SQL Workshop, Team Development, and Packaged Apps. A schema dropdown is set to BASEIS2. The bottom status bar indicates 5 rows returned in 0.00 seconds.

TAX_ID	SEX	AGE
58245987	A	24
56047895	Θ	27
48046418	A	27
59056316	Θ	26
35678456	Θ	29

5. Πόσοι ωφελούμενοι έχουν εγγραφεί σε τμήμα(τα) με κατάσταση «Σε Προγραμματισμό» (τίτλοι στηλών: Αριθμός Ωφελούμενων σε Αναμονή);

```
SELECT COUNT(BENEFICIARY.Tax_ID)
```

```
FROM BENEFICIARY JOIN REGISTERED ON BENEFICIARY.Tax_ID =  
REGISTERED.Tax_ID JOIN VOUCHER_CLASS ON REGISTERED.Class_ID =  
VOUCHER_CLASS.Class_ID
```

```
WHERE Status='Σε Προγραμματισμό'
```

The screenshot shows the Oracle SQL Workshop interface. The SQL Command area contains the following query:

```
SELECT COUNT(BENEFICIARY.Tax_ID)  
FROM BENEFICIARY JOIN REGISTERED ON BENEFICIARY.Tax_ID = REGISTERED.Tax_ID JOIN VOUCHER_CLASS ON REGISTERED.Class_ID = VOUCHER_CLASS.Class_ID  
WHERE Status='Σε Προγραμματισμό'
```

The Results tab is selected, showing a single row with the value 8. The text "COUNT(BENEFICIARY.TAX_ID)" is displayed above the result. Below the result, it says "1 rows returned in 0.05 seconds" and provides a "Download" link.

6. Σε ποιες προσκλήσεις (τίτλοι στηλών: Κωδικός πρόσκλησης, Περιγραφή) και με ποια ημερο- μηνία εγγραφής συμμετέχει η δομή κατάρτισης με επωνυμία «Ορίζοντες Α.Ε.»; Τα αποτελέσματα να εμφανιστούν ταξινομημένα με αύξουσα κατάταξη ως προς την ημερομηνία εγγραφής.

```
SELECT VOUCHER_CALL.Call_ID, Description
```

```
FROM VOUCHER_CALL JOIN PARTICIPATES ON VOUCHER_CALL.Call_ID =  
PARTICIPATES.Call_ID JOIN TRAINING_CENTER ON  
PARTICIPATES.Center_ID = TRAINING_CENTER.Center_ID
```

```
WHERE Title = 'Ορίζοντες Α.Ε.' ORDER BY Registration_Date
```

The screenshot shows the Oracle SQL Workshop interface. The SQL Command area contains the following query:

```
SELECT VOUCHER_CALL.Call_ID, Description FROM VOUCHER_CALL JOIN PARTICIPATES ON VOUCHER_CALL.Call_ID = PARTICIPATES.Call_ID JOIN TRAINING_CENTER ON  
PARTICIPATES.Center_ID = TRAINING_CENTER.Center_ID  
WHERE Title = 'Ορίζοντες Α.Ε.'  
ORDER BY Registration_Date
```

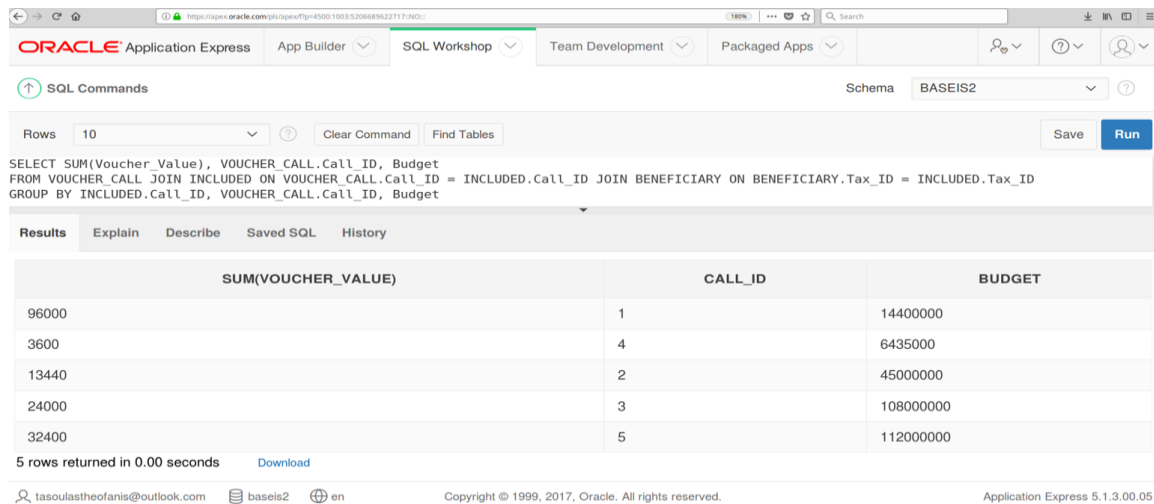
The Results tab is selected, showing a table with 4 rows. The table has two columns: CALL_ID and DESCRIPTION. The data is as follows:

CALL_ID	DESCRIPTION
3	Επιταγή εισόδου για νέους ηλικίας από 25 έως 29 ετών σε ιδιωτικές επιχειρήσεις για απόκτηση εργασιακής εμπειρίας
5	Επιταγή κατάρτισης ανέργων σε τομείς οικονομικής δραστηριότητας με υποχρεωτική εγγυημένη απασχόληση
4	Επιταγή επαγγελματικής κατάρτισης φορτοεκφορτωτών ξηράς - λιμένος
1	Κατάρτιση ανέργων σε θέματα μεταρρύθμισης για το πρωτοβάθμιο εθνικό δίκτυο υγείας

Below the table, it says "4 rows returned in 0.01 seconds" and provides a "Download" link.

7. Να υπολογιστεί για κάθε πρόσκληση η συνολική αξία των επιταγών κατάρτισης που έχουν διατεθεί σε ωφελούμενους μέχρι σήμερα. Στην απάντηση να συμπεριληφθεί ο κωδικός και ο προϋπολογισμός της πρόσκλησης, καθώς και η αξία που υπολογίστηκε από το ερώτημα (τίτλοι στηλών: Κωδικός Πρόσκλησης, Προϋπολογισμός, Συνολική Αξία Επιταγών).

```
SELECT SUM(Voucher_Value), VOUCHER_CALL.Call_ID, Budget
FROM VOUCHER_CALL JOIN INCLUDED ON VOUCHER_CALL.Call_ID =
INCLUDED.Call_ID JOIN BENEFICIARY ON BENEFICIARY.Tax_ID =
INCLUDED.Tax_ID
GROUP BY INCLUDED.Call_ID, VOUCHER_CALL.Call_ID, Budget
```



The screenshot shows the Oracle SQL Workshop interface. The SQL command entered is:

```
SELECT SUM(Voucher_Value), VOUCHER_CALL.Call_ID, Budget
FROM VOUCHER_CALL JOIN INCLUDED ON VOUCHER_CALL.Call_ID = INCLUDED.Call_ID JOIN BENEFICIARY ON BENEFICIARY.Tax_ID = INCLUDED.Tax_ID
GROUP BY INCLUDED.Call_ID, VOUCHER_CALL.Call_ID, Budget
```

The results are displayed in a table with 5 rows:

SUM(VOUCHER_VALUE)	CALL_ID	BUDGET
96000	1	14400000
3600	4	6435000
13440	2	45000000
24000	3	108000000
32400	5	112000000

5 rows returned in 0.00 seconds

8. Ποιοι ωφελούμενοι (τίτλοι στηλών: ΑΦΜ, Ονοματεπώνυμο) συμμετέχουν σε περισσότερα από ένα τμήματα, τα οποία βρίσκονται σε στάδιο υλοποίησης;

```
SELECT BENEFICIARY.Tax_ID, FULL_NAME
FROM VOUCHER_CALL JOIN VOUCHER_CLASS ON
VOUCHER_CALL.Call_ID = VOUCHER_CLASS.Call_ID JOIN REGISTERED
ON VOUCHER_CLASS.Class_ID = REGISTERED.Class_ID JOIN
BENEFICIARY ON BENEFICIARY.Tax_ID = REGISTERED.Tax_ID
WHERE STATUS='Σε Υλοποίηση' GROUP BY BENEFICIARY.Tax_ID,
FULL_NAME HAVING COUNT(VOUCHER_CLASS.Class_ID)>1;
```

Oracle Application Express SQL Workshop

Schema: BASEIS2

Rows: 10

Clear Command Find Tables

Save Run

```
SELECT BENEFICIARY.Tax_ID, FULL_NAME
FROM VOUCHER_CALL JOIN VOUCHER_CLASS ON VOUCHER_CALL.Call_ID = VOUCHER_CLASS.Call_ID JOIN REGISTERED ON VOUCHER_CLASS.Class_ID =
REGISTERED.Class_ID JOIN BENEFICIARY ON BENEFICIARY.Tax_ID = REGISTERED.Tax_ID
WHERE STATUS='Έξοδος' GROUP BY BENEFICIARY.Tax_ID, FULL_NAME HAVING COUNT(VOUCHER_CLASS.Class_ID)>1;
```

Results Explain Describe Saved SQL History

TAX_ID	FULL_NAME
35678456	Ελένη Αυγερίου
58245987	Λάζρος Χριστιδής
59056316	Μαρία Χριστοδούλου
56047895	Βασιλική Καντόρου

4 rows returned in 0.02 seconds Download

tasoulastheoanis@outlook.com baseis2 en Copyright © 1999, 2017, Oracle. All rights reserved. Application Express 5.1.3.00.05

9. Ποιοι ωφελούμενοι (τίτλοι στηλών: ΑΦΜ, Ονοματεπώνυμο, Έτος Γέννησης) δεν επιλέχθηκαν στην πρόσκληση με κωδικό πρόσκλησης «4»;

```
SELECT DISTINCT BENEFICIARY.Tax_ID, FULL_NAME, BIRTH_YEAR
FROM SEL JOIN BENEFICIARY ON BENEFICIARY.Tax_ID = SEL.Tax_ID
WHERE BENEFICIARY.TAX_ID NOT IN
(SELECT BENEFICIARY.Tax_ID
FROM SEL JOIN BENEFICIARY ON BENEFICIARY.Tax_ID = SEL.Tax_ID
WHERE CALL_ID=4)
```

Oracle Application Express SQL Workshop

Schema: BASEIS2

Rows: 50

Clear Command Find Tables

Save Run

```
SELECT DISTINCT BENEFICIARY.Tax_ID, FULL_NAME, BIRTH_YEAR
FROM SEL JOIN BENEFICIARY ON BENEFICIARY.Tax_ID = SEL.Tax_ID
WHERE BENEFICIARY.TAX_ID NOT IN (SELECT BENEFICIARY.Tax_ID FROM SEL JOIN BENEFICIARY ON BENEFICIARY.Tax_ID = SEL.Tax_ID WHERE CALL_ID=4)
```

Results Explain Describe Saved SQL History

TAX_ID	FULL_NAME	BIRTH_YEAR
51045617	Αγγελική Ευθυμιπούλου	1987
48046418	Αριστείδης Τρικόπουλος	1990
56047895	Βασιλική Καντόρου	1990

3 rows returned in 0.01 seconds Download

tasoulastheoanis@outlook.com baseis2 en Copyright © 1999, 2017, Oracle. All rights reserved. Application Express 5.1.3.00.05

10. Ποιοι ωφελούμενοι (τίτλοι στηλών: ΑΦΜ, Ονοματεπώνυμο) έχουν επιλεγεί στην πρόσκληση με κωδικό πρόσκλησης «1» και επίσης έχουν εγγραφεί (για οποιαδήποτε πρόσκληση) σε τμήμα της δομής κατάρτισης με επωνυμία «Κατάρτιση Α.Ε.»;

```
SELECT BENEFICIARY.TAX_ID, FULL_NAME
FROM TRAINING_CENTER JOIN INCLUDED ON
TRAINING_CENTER.CENTER_ID = INCLUDED.CENTER_ID JOIN
BENEFICIARY ON INCLUDED.TAX_ID = BENEFICIARY.TAX_ID
WHERE CALL_ID=1 AND TITLE = 'Κατάρτιση Α.Ε.'
```

Oracle SQL Workshop interface showing the execution of an SQL query. The query is:

```
SELECT BENEFICIARY.TAX_ID, FULL_NAME
FROM TRAINING_CENTER JOIN INCLUDED ON TRAINING_CENTER.CENTER_ID = INCLUDED.CENTER_ID JOIN BENEFICIARY ON INCLUDED.TAX_ID = BENEFICIARY.TAX_ID
WHERE CALL_ID=1 AND TITLE = 'Κατάρτιση Α.Ε.'
```

The results show 2 rows returned in 0.03 seconds:

TAX_ID	FULL_NAME
51045617	Αγγελική Ευθυμιοπούλου
56047895	Βασιλική Καντόρου

11. Να βρεθεί η πρόσκληση με τη μεγαλύτερη συνολική αξία επιταγών κατάρτισης που έχουν διατεθεί για αυτήν σε ωφελούμενους μέχρι σήμερα. Στην απάντηση να περιληφθεί ο κωδικός και ο προϋπολογισμός της πρόσκλησης, καθώς και η συνολική αξία επιταγών.

Υπόδειξη: Αξιοποιήστε κατάλληλα ως όψη την απάντηση που δώσατε στο Ερώτημα 7

```
CREATE OR REPLACE VIEW EROTIMA_7 AS
SELECT SUM(Voucher_Value) AS SUM_OF_VOUCHER_VALUE,
VOUCHER_CALL.Call_ID, Budget
FROM VOUCHER_CALL JOIN INCLUDED ON VOUCHER_CALL.Call_ID =
INCLUDED.Call_ID JOIN BENEFICIARY ON BENEFICIARY.Tax_ID =
INCLUDED.Tax_ID
GROUP BY INCLUDED.Call_ID, VOUCHER_CALL.Call_ID, Budget;
```

```
SELECT CALL_ID, BUDGET, SUM_OF_VOUCHER_VALUE
FROM EROTIMA_7 WHERE SUM_OF_VOUCHER_VALUE IN
(SELECT MAX(SUM_OF_VOUCHER_VALUE)
FROM EROTIMA_7);
```

The screenshot shows the Oracle SQL Workshop interface. The SQL Commands tab is active, displaying the following SQL script:

```
CREATE VIEW EROTIMA_7 AS
SELECT SUM(Voucher.Value) AS SUM_OF_VOUCHER_VALUE, VOUCHER_CALL.Call_ID, Budget
FROM VOUCHER_CALL JOIN INCLUDED ON VOUCHER_CALL.Call_ID = INCLUDED.Call_ID JOIN BENEFICIARY ON BENEFICIARY.Tax_ID = INCLUDED.Tax_ID
GROUP BY INCLUDED.Call_ID, VOUCHER_CALL.Call_ID, Budget

SELECT CALL_ID, BUDGET, SUM_OF_VOUCHER_VALUE
FROM EROTIMA_7 WHERE SUM_OF_VOUCHER_VALUE IN (SELECT MAX(SUM_OF_VOUCHER_VALUE)
FROM EROTIMA_7)
```

The Results tab shows the output of the query:

CALL_ID	BUDGET	SUM_OF_VOUCHER_VALUE
1	14400000	96000

1 rows returned in 0.01 seconds

12. Ποιοι είναι οι ωφελούμενοι (τίτλοι στηλών: ΑΦΜ, Ονοματεπώνυμο, Αριθμός Προσκήσεων) που έχουν επιλεγεί σε περισσότερες από δύο προσκλήσεις και η κατάρτισή τους είτε έχει ολοκληρωθεί, είτε είναι σε φάση υλοποίησης; Τα αποτελέσματα να εμφανίζονται ταξινομημένα σε φθίνουσα κατάταξη ως προς το Ονοματεπώνυμο.

Υπόδειξη: Δημιουργήστε αρχικά μία όψη με τα στοιχεία των ωφελούμενων που βρίσκονται ενταγμένοι σε τμήματα τα οποία είτε έχουν ολοκληρωθεί είτε είναι σε φάση υλοποίησης.

```
CREATE OR REPLACE VIEW COMPLETED_OR_IN_IMPLEMENTATION AS
SELECT BENEFICIARY.TAX_ID, UNEMPLOYMENT_CARD, FULL_NAME,
BIRTH_YEAR, SEX, HOME_ADDRESS
FROM BENEFICIARY JOIN REGISTERED ON BENEFICIARY.TAX_ID =
REGISTERED.TAX_ID JOIN VOUCHER_CLASS ON REGISTERED.CLASS_ID =
VOUCHER_CLASS.CLASS_ID
WHERE STATUS='Ολοκληρωμένο' OR STATUS='Σε Υλοποίηση'
```



```

SELECT DISTINCT INCLUDED.TAX_ID, FULL_NAME, COUNT(CALL_ID)
FROM COMPLETED_OR_IN_IMPLEMENTATION JOIN INCLUDED ON
COMPLETED_OR_IN_IMPLEMENTATION.TAX_ID = INCLUDED.TAX_ID
GROUP BY INCLUDED.TAX_ID, FULL_NAME, CALL_ID
HAVING COUNT(CALL_ID)>1
ORDER BY FULL_NAME DESC

```

```

CREATE OR REPLACE VIEW COMPLETED_OR_IN_IMPLEMENTATION AS
SELECT BENEFICIARY.TAX_ID, UNEMPLOYMENT_CARD, FULL_NAME, BIRTH_YEAR, SEX, HOME_ADDRESS
FROM BENEFICIARY JOIN REGISTERED ON BENEFICIARY.TAX_ID = REGISTERED.TAX_ID JOIN VOUCHER_CLASS ON REGISTERED.CLASS_ID = VOUCHER_CLASS.CLASS_ID
WHERE STATUS='Ολοκληρωμένο' OR STATUS='Σε Υλοποίηση'

SELECT DISTINCT INCLUDED.TAX_ID, FULL_NAME, COUNT(CALL_ID)
FROM COMPLETED_OR_IN_IMPLEMENTATION JOIN INCLUDED ON COMPLETED_OR_IN_IMPLEMENTATION.TAX_ID = INCLUDED.TAX_ID
GROUP BY INCLUDED.TAX_ID, FULL_NAME, CALL_ID
HAVING COUNT(CALL_ID)>1
ORDER BY FULL_NAME DESC

```

Results	Explain	Describe	Saved SQL	History
TAX_ID	FULL_NAME		COUNT(CALL_ID)	
47096457	Τρύφωνας Μπαλτέας		3	
59056316	Μαρία Χριστοδούλου		4	
58245987	Λάζαρος Χριστίδης		4	
49036418	Ιωάννης Τέλογλου		3	
35678456	Ελένη Αυγερίου		4	
56047895	Βασιλική Καντόρου		3	

6 rows returned in 0.02 seconds [Download](#)

tasoulastheoanis@outlook.com baseis2 en Copyright © 1999, 2017, Oracle. All rights reserved.

13. Ποιες είναι οι επιταγές κατάρτισης που δεν εξαργυρώθηκαν (Κωδικός Επιταγής, ΑΦΜ Ωφελούμενου, Κωδικός Πρόσκλησης, Αξία Επιταγής) και ποια είναι η αξία κάθε μίας από αυτές; Υπόδειξη: Δημιουργήστε αρχικά δύο όψεις:

(α) μία με τα στοιχεία όλων των επιταγών κατάρτισης που διατέθηκαν (για όλες τις προσκλήσεις), και (β) μία με τα στοιχεία των επιταγών κατάρτισης που προσκομίστηκαν σε οποιαδήποτε Δομή Κατάρτισης (αυτών δηλαδή που εξαργυρώθηκαν).

```

CREATE OR REPLACE VIEW erwtima13A AS
SELECT VOUCHER_ID, Tax_ID, SEL.Call_ID, Voucher_Value
FROM SEL JOIN VOUCHER_CALL ON SEL.CALL_ID =
VOUCHER_CALL.CALL_ID

```

```

CREATE OR REPLACE VIEW erwtima13B AS
SELECT DISTINCT VOUCHER_ID, SEL.Tax_ID, SEL.Call_ID AS CALL_ID2,
Voucher_Value
FROM SEL
JOIN VOUCHER_CALL ON SEL.CALL_ID = VOUCHER_CALL.CALL_ID
JOIN VOUCHER_CLASS ON VOUCHER_CLASS.CALL_ID =
VOUCHER_CALL.CALL_ID
WHERE STATUS = 'Ολοκληρωμένο'

```

```

SELECT VOUCHER_ID, Tax_ID, Call_ID, Voucher_Value
FROM erwtima13A WHERE Call_ID NOT IN (SELECT CALL_ID2 FROM
erwtima13B)

```

SELECT VOUCHER_ID, Tax_ID, Call_ID, Voucher_Value FROM erwtima13A WHERE Call_ID NOT IN (SELECT CALL_ID2 FROM erwtima13B)			
Results Explain Describe Saved SQL History			
VOUCHER_ID	TAX_ID	CALL_ID	VOUCHER_VALUE
55895	049036418	4	1200
66589	049036418	5	5400
67894	047096457	4	1200
69999	059056316	4	1200
56412	059056316	5	5400
72456	035678456	4	1200
55456	035678456	5	5400
69811	058245987	4	1200
73125	058245987	5	5400
69396	056047895	5	5400
69879	051045617	5	5400
11 rows returned in 0.00 seconds Download			

14. α) Να αλλάξετε το ονοματεπώνυμο του ωφελούμενου με αριθμό κάρτας ανεργίας 7852469785214563 σε «Μαρία Χριστοδουλοπούλου».

```

UPDATE BENEFICIARY
SET FULL_NAME = 'Μαρία Χριστοδουλοπούλου'
WHERE UNEMPLOYMENT_CARD = 7852469785214563

```

Oracle Application Express SQL Workshop Team Development Packaged Apps

Schema: BASEIS2

Rows: 10 Clear Command Find Tables Save Run

```
UPDATE BENEFICIARY
SET FULL_NAME = 'Μαρία Χριστοδουλοπούλου'
WHERE UNEMPLOYMENT_CARD = 7852469785214563
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.02 seconds

14. β) Επίσης, να προσθέσετε τα στοιχεία σας ως ωφελούμενου στο συγκεκριμένο πίνακα. Στη συνέχεια, να εισάγετε στον κατάλληλο πίνακα τα απαραίτητα στοιχεία ώστε να φαίνεται ότι έχετε επιλεγεί στην πρόσκληση με κωδικό 5, κωδικό επιταγής 69788, ημερομηνία έκδοσης της επιταγής την 01/02/2017 και ημερομηνία λήξης της επιταγής την 31/03/2017.

```
INSERT INTO BENEFICIARY (TAX_ID, UNEMPLOYMENT_CARD,
FULL_NAME, BIRTH_YEAR, SEX, HOME_ADDRESS)
VALUES (00000000, 1000000000000000, 'Τασούλας Θεοφάνης', 1997, 'Α',
'Νικολάου Σοφιανού 22')
```

127.0.0.1:8080/apex/f?p=4500:1003:834652965459848::NO::

Home SQL Workshop SQL Commands

☒ Autocommit Rows: 10 Save Run

```
INSERT INTO BENEFICIARY (TAX_ID, UNEMPLOYMENT_CARD, FULL_NAME, BIRTH_YEAR, SEX, HOME_ADDRESS)
VALUES (00000000, 1000000000000000, 'Τασούλας Θεοφάνης', 1997, 'Α', 'Νικολάου Σοφιανού 22')
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.01 seconds

```
INSERT INTO SEL(TAX_ID, CALL_ID, VOUCHER_ID, DATE_OF_ISSUE,
END_DATE)
VALUES ('00000000', 5, 69788, TO_DATE('01/02/2017', 'DD/MM/YYYY'),
TO_DATE('31/03/2017', 'DD/MM/YYYY'))
```

Secure | <https://apex.oracle.com/pls/apex/f?p=4500:1003:112260653043062::NO::>

ORACLE Application Express App Builder SQL Workshop Team Development Packaged Apps

SQL Commands

Rows 10 Clear Command Find Tables

```
INSERT INTO SEL(TAX_ID, CALL_ID, VOUCHER_ID, DATE_OF_ISSUE, END_DATE)
VALUES ('00000000', 5, 69788, TO_DATE('01/02/2017', 'DD/MM/YYYY'), TO_DATE('31/03/2017', 'DD/MM/YYYY'))
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.02 seconds

15. Στον πίνακα VOUCHER_CLASS να μετατρέψετε την κατάσταση από «Σε Προγραμματισμό» σε «Σε Υλοποίηση» για τα τμήματα που η ημερομηνία έναρξης είναι

προγενέστερη της τρέχουσας ημερομηνίας.

UPDATE VOUCHER_CLASS

SET STATUS = 'Σε Υλοποίηση'

WHERE STATUS = 'Σε Προγραμματισμό' AND START_DATE < CURRENT_DATE

ORACLE App Builder SQL Workshop Team Development Packaged Apps

SQL Commands Schema BASEIS2

Rows 20 Clear Command Find Tables Save Run

```
UPDATE VOUCHER_CLASS
SET STATUS = 'Σε Υλοποίηση'
WHERE STATUS = 'Σε Προγραμματισμό' AND START_DATE < CURRENT_DATE
```

Results Explain Describe Saved SQL History

4 row(s) updated.

0.02 seconds

16. Στον πίνακα BENEFICIARY να προσθέσετε μία νέα στήλη με την ονομασία Num_Participations στην οποία να αποθηκεύσετε το συνολικό πλήθος προσκλήσεων στις οποίες έχει επιλεγεί ο ωφελούμενος.

```
ALTER TABLE BENEFICIARY  
ADD NUM_PARTICIPATIONS NUMBER
```

```
UPDATE BENEFICIARY  
SET NUM_PARTICIPATIONS=(  
    SELECT COUNT(CALL_ID)  
    FROM SEL  
    WHERE BENEFICIARY.TAX_ID=SEL.TAX_ID  
    GROUP BY TAX_ID)
```

The screenshot shows the Oracle Application Express SQL Workshop interface. At the top, there are navigation tabs: "ORACLE Application Express", "App Builder", "SQL Workshop", "Team Development", and "Packaged Apps". Below these is a section for "SQL Commands" with a search bar and buttons for "Clear Command" and "Find Tables". The main area displays the SQL command:

```
ALTER TABLE BENEFICIARY  
ADD NUM_PARTICIPATIONS NUMBER  
  
UPDATE BENEFICIARY  
SET NUM_PARTICIPATIONS=(  
    SELECT COUNT(CALL_ID)  
    FROM SEL  
    WHERE BENEFICIARY.TAX_ID=SEL.TAX_ID  
    GROUP BY TAX_ID)
```

 Below the command, there is a tabbed interface with "Results", "Explain", "Describe", "Saved SQL", and "History". The "Results" tab is active, showing the message "8 row(s) updated." and the execution time "0.02 seconds".

17. Δημιουργήστε ένα νέο πίνακα με όνομα CALL_STATISTICS (στον οποίο απαιτείται να αποθηκεύονται στατιστικά στοιχεία για τις προσκλήσεις), που θα περιέχει τις εξής πληροφορίες: τον κωδικό της πρόσκλησης (Call_ID), το πλήθος των ωφελούμενων της πρόσκλησης (Beneficiaries_Num) και το πλήθος των ωφελούμενων της πρόσκλησης που έχουν προσκομίσει την επιταγή τους σε κάποια δομή κατάρτισης (Beneficiaries_Incl_Num).

```
CREATE TABLE CALL_STATISTICS(  
    CALL_ID NUMBER NOT NULL,  
    Beneficiaries_Num NUMBER NOT NULL,
```

```
Beneficiaries_Incl_Num NUMBER NOT NULL  
);
```

```
ALTER TABLE CALL_STATISTICS  
ADD CONSTRAINT CALL_ID_PK PRIMARY KEY(CALL_ID);
```

```
ALTER TABLE CALL_STATISTICS  
ADD CONSTRAINT CALL_STATISTICS_FK  
FOREIGN KEY (CALL_ID)  
REFERENCES VOUCHER_CALL(CALL_ID);
```

```
CREATE VIEW Benef_Num AS  
SELECT CALL_ID, COUNT(CALL_ID) AS "BENEF_NUM"  
FROM INCLUDED  
GROUP BY CALL_ID;
```

```
CREATE VIEW Benef_Included_Num AS  
SELECT CALL_ID, COUNT(CALL_ID) AS "BENEF_IN_COLNUMN"  
FROM SEL  
GROUP BY CALL_ID;
```

```
INSERT INTO CALL_STATISTICS  
SELECT Benef_Included_Num.CALL_ID, BENEF_NUM, BENEF_IN_COLNUMN  
FROM Benef_Included_Num  
LEFT OUTER JOIN Benef_Num  
ON Benef_Num.CALL_ID = Benef_Included_Num.CALL_ID  
ORDER BY Benef_Num.CALL_ID;
```

```
CREATE VIEW Benef_Included_Num AS
SELECT CALL_ID, COUNT(CALL_ID) AS "BENEF_IN_COLNUMM"
FROM SEL
GROUP BY CALL_ID;
```

```
INSERT INTO CALL_STATISTICS
SELECT Benef_Included_Num.CALL_ID, BENEF_NUM, BENEF_IN_COLNUMM
FROM Benef_Included_Num
LEFT OUTER JOIN Benef_Num
ON Benef_Num.CALL_ID = Benef_Included_Num.CALL_ID
ORDER BY Benef_Num.CALL_ID;
```

Results Explain Describe Saved SQL History

5 row(s) inserted.

0.07 seconds

tasoulastheofanis@outlook.com baseis2 en

Copyright © 1999, 2017, Oracle.

18. Στον πίνακα VOUCHER_CLASS να διαγράψετε τα τμήματα που είναι σε κατάσταση «προγραμματισμού» και δεν έχουν καταχωρημένο κάποιον ωφελούμενο.

```
DELETE FROM VOUCHER_CLASS
```

```
WHERE STATUS = 'Σε Προγραμματισμό' AND CLASS_ID IN (
```

```
SELECT CLASS_ID
```

```
FROM REGISTERED
```

```
GROUP BY CLASS_ID
```

```
HAVING COUNT(TAX_ID)<1 )
```

Secure | https://apex.oracle.com/pls/apex/f?p=4500:1003:9370566153531::NQ::

ORACLE Application Express App Builder SQL Workshop Team Development Packaged Apps

SQL Commands

Rows 50 Clear Command Find Tables

```
DELETE FROM VOUCHER_CLASS
WHERE STATUS = 'Σε Προγραμματισμό' AND CLASS_ID IN (
SELECT CLASS_ID
FROM REGISTERED
GROUP BY CLASS_ID
HAVING COUNT(TAX_ID)<1
)
```

Results Explain Describe Saved SQL History

0 row(s) deleted.

0.03 seconds

ΠΑΡΑΤΗΡΗΣΗ

Κάποια ερωτήματα (όπως το 18) έχουν εκτελεσθεί με τους πλέον τροποποιημένους (από τα προηγούμενα ερωτήματα) πίνακες και όχι από τους αρχικούς πίνακες με τις αρχικές τους τιμές. Κατά συνέπεια, τα αποτελέσματα μπορεί να διαφέρουν.

PROCEDURES

- 1) Μπορεί να θεωρείται περιττό η βάση δεδομένων να αποθηκεύει πολύ παλιές πληροφορίες. Διέγραψε όλες τις στήλες από τους πίνακες με καταγραφές παλαιότερες από την ημερομηνία που θα δοθεί από το χρήστη.

```
CREATE OR REPLACE PROCEDURE DELETE_SINCE(V_DATE IN DATE) AS  
BEGIN
```

```
    DELETE FROM INCLUDED WHERE INCLUDED.Voucher_Submission_Date  
< V_DATE;
```

```
    DELETE FROM REGISTERED WHERE REGISTERED.Registration_Date <  
V_DATE;
```

```
    DELETE FROM SEL WHERE SEL.Date_of_Issue < V_DATE;
```

```
    DELETE FROM PARTICIPATES WHERE PARTICIPATES.Registration_Date <  
V_DATE;
```

```
    DBMS_OUTPUT.PUT_LINE('DATES DELETED');
```

```
END;
```

-- Δοκιμή με τον παρακάτω κώδικα --

```
BEGIN
```

```
DELETE_SINCE(TO_DATE ('1/4/2016', 'DD/MM/YYYY'));
```

```
END;
```


Oracle Application Express SQL Workshop interface. The SQL Commands tab is active, showing a PL/SQL procedure named DELETE_SINCE. The procedure is designed to delete records from the INCLUDED, REGISTERED, and PARTICIPATES tables based on a date parameter V_DATE. The results section shows the statement was processed successfully in 0.07 seconds.

```
CREATE OR REPLACE PROCEDURE DELETE_SINCE(V_DATE IN DATE) AS
BEGIN
DELETE FROM INCLUDED WHERE INCLUDED.Voucher_Submission_Date < V_DATE;
DELETE FROM REGISTERED WHERE REGISTERED.Registration_Date < V_DATE;
DELETE FROM SEL WHERE SEL.Date_of_Issue < V_DATE;
DELETE FROM PARTICIPATES WHERE PARTICIPATES.Registration_Date < V_DATE;
DBMS_OUTPUT.PUT_LINE('DATES DELETED');
END;

BEGIN
DELETE_SINCE(TO_DATE('1/4/2016', 'DD/MM/YYYY'));
END;
```

Results: DATES DELETED
statement processed.
0.07 seconds

2) Πιθανώς να είναι χρήσιμο να γνωρίζουμε το μέσο όρο της «ηλικίας των ωφελούμενων ανά δεκαετία»

CREATE OR REPLACE PROCEDURE AGE_MEAN IS

BORN_IN_2000 NUMBER(6,3); -- Πόσοι ωφελούμενοι γεννήθηκαν τη δεκαετία του 2000 --

BORN_IN_1990 NUMBER(6,3); -- Πόσοι ωφελούμενοι γεννήθηκαν τη δεκαετία του 1990 --

BORN_IN_1980 NUMBER(6,3);

BORN_IN_1970 NUMBER(6,3);

BORN_IN_1960 NUMBER(6,3);

BORN_IN_1950 NUMBER(6,3);

COUNT_BENEFICIARY NUMBER; -- Πόσοι ωφελούμενοι υπάρχουν συνολικά (Το έτος γέννησης είναι δεν είναι κενό, οπότε δεν ελέγχεται αν είναι NULL) --

BEGIN

-- Υπολόγισε το πλήθος των ωφελούμενων --

SELECT COUNT(BIRTH_YEAR) INTO COUNT_BENEFICIARY FROM BENEFICIARY;

SELECT COUNT(BIRTH_YEAR) INTO BORN_IN_2000 FROM BENEFICIARY WHERE BIRTH_YEAR >= 2000 AND BIRTH_YEAR < 2010;

```
SELECT COUNT(BIRTH_YEAR) INTO BORN_IN_1990 FROM BENEFICIARY  
WHERE BIRTH_YEAR >= 1990 AND BIRTH_YEAR < 2000;
```

```
SELECT COUNT(BIRTH_YEAR) INTO BORN_IN_1980 FROM BENEFICIARY  
WHERE BIRTH_YEAR >= 1980 AND BIRTH_YEAR < 1990;
```

```
SELECT COUNT(BIRTH_YEAR) INTO BORN_IN_1970 FROM BENEFICIARY  
WHERE BIRTH_YEAR >= 1970 AND BIRTH_YEAR < 1980;
```

```
SELECT COUNT(BIRTH_YEAR) INTO BORN_IN_1960 FROM BENEFICIARY  
WHERE BIRTH_YEAR >= 1960 AND BIRTH_YEAR < 1970;
```

```
SELECT COUNT(BIRTH_YEAR) INTO BORN_IN_1950 FROM BENEFICIARY  
WHERE BIRTH_YEAR >= 1950 AND BIRTH_YEAR < 1960;
```

```
-- Μετέτρεψε το πλήθος των ωφελουμένων σε μέσο όρο--
```

```
BORN_IN_2000 := BORN_IN_2000 / COUNT_BENEFICIARY * 100;
```

```
BORN_IN_1990 := BORN_IN_1990 / COUNT_BENEFICIARY * 100;
```

```
BORN_IN_1980 := BORN_IN_1980 / COUNT_BENEFICIARY * 100;
```

```
BORN_IN_1970 := BORN_IN_1970 / COUNT_BENEFICIARY * 100;
```

```
BORN_IN_1960 := BORN_IN_1960 / COUNT_BENEFICIARY * 100;
```

```
BORN_IN_1950 := BORN_IN_1950 / COUNT_BENEFICIARY * 100;
```

```
-- Εμφάνισε τα αποτελέσματα --
```

```
DBMS_OUTPUT.PUT_LINE('Mean of those born in 2000s is ' ||  
BORN_IN_2000 || '%');
```

```
DBMS_OUTPUT.PUT_LINE('Mean of those born in 1990s is ' ||  
BORN_IN_1990 || '%');
```

```
DBMS_OUTPUT.PUT_LINE('Mean of those born in 1980s is ' ||  
BORN_IN_1980 || '%');
```

```
DBMS_OUTPUT.PUT_LINE('Mean of those born in 1970s is ' ||  
BORN_IN_1970 || '%');
```

```
DBMS_OUTPUT.PUT_LINE('Mean of those born in 1960s is ' ||  
BORN_IN_1960 || '%');
```

```
DBMS_OUTPUT.PUT_LINE('Mean of those born in 1950s is ' ||  
BORN_IN_1950 || '%');
```

```
END;
```

```
-- Δοκιμή με τον παρακάτω κώδικα --
```

```
BEGIN
```

```
AGE_MEAN;

END;
```

The screenshot shows the Oracle Application Express interface. At the top, there's a navigation bar with the Oracle logo and 'Application Express' text. Below that, a 'SQL Commands' section is visible with a 'Schema' dropdown set to 'BASEIS2'. The 'Rows' dropdown is set to '10'. The SQL command area contains the following code:

```
DBMS_OUTPUT.PUT_LINE('Mean of those born in 1970s is ' || BORN_IN_1970 || '%');
DBMS_OUTPUT.PUT_LINE('Mean of those born in 1960s is ' || BORN_IN_1960 || '%');
DBMS_OUTPUT.PUT_LINE('Mean of those born in 1950s is ' || BORN_IN_1950 || '%');

END;

BEGIN
AGE_MEAN;
END;
```

Below the command area, there's a 'Results' tab selected, showing the output of the SQL command:

```
Mean of those born in 2000s is 0%
Mean of those born in 1990s is 62.5%
Mean of those born in 1980s is 25%
Mean of those born in 1970s is 12.5%
Mean of those born in 1960s is 0%
Mean of those born in 1950s is 0%

Statement processed.
```

At the bottom, there's a footer with the user 'tasoulastheoanis@outlook.com', the schema 'baseis2', and the version 'Application Express 5.1.3.00.05'.

TRIGGERS

- 1) Σε περίπτωση που κάποιος θα ήθελε να διαγράψει κάποιον ωφελούμενο, απλώς χρησιμοποιώντας την εντολή DELETE BENEFICIARY WHERE ΣΥΝΘΗΚΗ δε θα μπορούσε, διότι πρώτα θα έπρεπε να διαγραφούν όλες οι στήλες από τους πίνακες INCLUDED, REGISTERED και SEL, αφού δεσμεύουν το Tax_ID του ωφελούμενου ως FOREIGN KEY. Τώρα η διαδικασία αυτή γίνεται εφικτή.

```
CREATE OR REPLACE TRIGGER EASY_DELETE_BENEFICIARY
BEFORE DELETE ON BENEFICIARY
```

```
FOR EACH ROW
```

```
WHEN (old.Tax_ID IS NOT NULL)
```

```
DECLARE
```

```
DELETED_TAX_ID beneficiary.tax_id%type := :old.Tax_ID;
```

```
BEGIN
```

```
-- Διαγράφουμε τις στήλες όπου ο ΑΦΜ είναι ίδιος με τον ΑΦΜ της στήλης που
διαγράφηκε --
```

```

DELETE INCLUDED    WHERE Tax_ID = DELETED_TAX_ID;
DELETE REGISTERED WHERE Tax_ID = DELETED_TAX_ID;
DELETE SEL        WHERE Tax_ID = DELETED_TAX_ID;
DBMS_OUTPUT.PUT_LINE('BENEFICIARY DELETED');
END;

```

--Δοκιμή με τον παρακάτω κώδικα--

DELETE BENEFICIARY WHERE Tax_ID = 47096457

The screenshot shows the Oracle Application Express interface. The SQL Commands section contains the following code:

```

CREATE OR REPLACE TRIGGER EASY_DELETE_BENEFICIARY
BEFORE DELETE ON BENEFICIARY
FOR EACH ROW
WHEN (old.Tax_ID IS NOT NULL)
DECLARE
    DELETED_TAX_ID beneficiary.tax_id%type := :old.Tax_ID;
BEGIN
DELETE INCLUDED    WHERE Tax_ID = DELETED_TAX_ID;
DELETE REGISTERED WHERE Tax_ID = DELETED_TAX_ID;
DELETE SEL        WHERE Tax_ID = DELETED_TAX_ID;
DBMS_OUTPUT.PUT_LINE('BENEFICIARY DELETED');
END;

```

The command `DELETE BENEFICIARY WHERE Tax_ID = 47096457` is highlighted in blue. Below the code, the 'Results' tab is selected, showing the output:

```

BENEFICIARY DELETED
1 row(s) deleted.
0.04 seconds

```

- 2) Με τη βοήθεια της επόμενης σκανδάλης, όταν έχει επιλεγεί κάποιος ακόμη ωφελούμενος και θα προστεθεί στον πίνακα SEL, τότε αυτόματα θα εκτελεσθεί το ερώτημα 16. Δηλαδή θα αναβαθμισθεί οι αριθμός των συμμετοχών του, αλλά και των υπόλοιπων ωφελούμενων.

```

CREATE OR REPLACE TRIGGER AFTER_UPDATE_SEL
AFTER INSERT ON SEL
BEGIN

```

--Όταν εκχωρηθούν νέες τιμές στους επιλεγόμενους, τότε θα ανανεωθούν οι αριθμοί συμμετοχών των ωφελουμένων--

```

UPDATE BENEFICIARY
SET NUM_PARTICIPATIONS=(

```

```

SELECT COUNT(CALL_ID)
FROM SEL
WHERE BENEFICIARY.TAX_ID=SEL.TAX_ID
GROUP BY TAX_ID;

END;

```

```



--Δοκιμή με τον παρακάτω κώδικα, όπου Tax_ID ο ΑΦΜ του Τασούλα Θεοφάνη--
INSERT INTO SEL (Tax_ID, Call_ID, Voucher_ID, Date_of_Issue, End_Date)
VALUES(0, 5, 98765, to_date('1/11/2016', 'dd/mm/yyyy'), to_date('31/12/2016',
'dd/mm/yyyy'));

```

ORACLE Application Express

Home	Application Builder ▼	SQL Workshop ▼	Team Development ▼	Administration ▼
------	-----------------------	----------------	--------------------	------------------

Home > SQL Workshop > SQL Commands

☒ Autocommit Rows: 10  

```

CREATE OR REPLACE TRIGGER AFTER_UPDATE_SEL
AFTER INSERT ON SEL

BEGIN
--Όταν εκχωρηθούν νέες τιμές στους επιλεγόμενους, τότε θα ανανεωθούν οι αριθμοί συμμετοχών των ωφελουμένων--

UPDATE BENEFICIARY
SET NUM_PARTICIPATIONS=(
    SELECT COUNT(CALL_ID)
    FROM SEL
    WHERE BENEFICIARY.TAX_ID=SEL.TAX_ID
    GROUP BY TAX_ID);

END;

--Δοκιμή με τον παρακάτω κώδικα, όπου Tax_ID ο ΑΦΜ του Τασούλα Θεοφάνη--
INSERT INTO SEL (Tax_ID, Call_ID, Voucher_ID, Date_of_Issue, End_Date)
VALUES(0, 5, 98765, to_date('1/11/2016', 'dd/mm/yyyy'), to_date('31/12/2016', 'dd/mm/yyyy'));

```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.01 seconds

FUNCTIONS

- 1) Πιθανώς να είναι ωφέλιμο να γνωρίζουμε πόσες εγγραφές έγιναν μια συγκεκριμένη χρονιά. Για αυτό δημιουργήθηκε η παρακάτω συνάρτηση, που δέχεται ως παράμετρο το έτος από το οποίο θα υπολογισθεί το άθροισμα των ημερομηνιών.

```
CREATE OR REPLACE FUNCTION totalRegistersAtYear (Year_Variable IN
number)
```

```
RETURN number IS
```

```
total number(4) := 0;
```

```
BEGIN
```

```
SELECT COUNT(REGISTRATION_DATE) into total
```

```
FROM REGISTERED
```

```
WHERE Year_Variable = EXTRACT(YEAR FROM
REGISTRATION_DATE);
```

```
--Επέλεξε το άθροισμα από τις ημερομηνίες, όπου η χρονιά που δόθηκε
είναι ίση με τη χρονιά της ημερομηνίας εγγραφής--
```

```
RETURN total;
```

```
END;
```

```
-- Δοκιμή με τους παρακάτω κώδικες --
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE (totalRegistersAtYear(2015));
```

```
END;
```

```
SELECT DISTINCT totalRegistersAtYear(2015), totalRegistersAtYear(2016)
```

```
FROM REGISTERED
```

← → 127.0.0.1:8080/apex/f?p=4500:1001:3553597483654663::NO::

ORACLE Application Express

Home Application Builder SQL Workshop Team Development Administration

Home > SQL Workshop > SQL Commands

☒ Autocommit Rows 10 Save Run

```

CREATE OR REPLACE FUNCTION totalRegistersAtYear (Year_Variable IN number)
RETURN number IS
    total number(4) := 0;
BEGIN
    SELECT COUNT(REGISTRATION_DATE) into total
    FROM REGISTERED
    WHERE Year_Variable = EXTRACT(YEAR FROM REGISTRATION_DATE);
    --Επέλεξε το άθροισμα από τις ημερομηνίες, όπου η χρονιά που δόθηκε είναι ίση με τη χρονιά της ημερομηνίας εγγραφής--
    RETURN total;
END;

-- Δοκιμή με τους παρακάτω κώδικες --
BEGIN
    DBMS_OUTPUT.PUT_LINE (totalRegistersAtYear(2015));
END;

SELECT DISTINCT totalRegistersAtYear(2015), totalRegistersAtYear(2016)
FROM REGISTERED

```

Results Explain Describe Saved SQL History

TOTALREGISTERSATYEAR(2015)	TOTALREGISTERSATYEAR(2016)
5	11

1 rows returned in 0.01 seconds [Download](#)

- 2) _Με τη βοήθεια της συνάρτησης AverageOfStatus, που δέχεται ως παράμετρο μια κατάσταση μιας VOUCHER_CLASS (π.χ. Ολοκληρωμένο, Σε υλοποίηση, κτλ.), μπορεί κάποιος να βρει το μέσο όρο των καταστάσεων των τμημάτων, όπου έχουν μια συγκεκριμένη κατάσταση.

```

CREATE OR REPLACE FUNCTION AverageOfStatus (GIVEN_STATUS IN
VARCHAR2)

```

```

RETURN number IS

```

```

    average number(6,2);

```

```

    total_status number (6);

```

```

BEGIN

```

```

    --Επέλεξε το άθροισμα των καταστάσεων, όπου η κατάσταση είναι ίδια με
    την δοθείσα--

```

```

    SELECT DISTINCT COUNT(STATUS) INTO average

```

```

    FROM VOUCHER_CLASS

```

```

    WHERE STATUS = GIVEN_STATUS;

```

--Επέλεξε το άθροισμα των καταστάσεων--

```
SELECT DISTINCT COUNT(STATUS) INTO total_status  
FROM VOUCHER_CLASS;
```

--Επέστρεψε το μέσο όρο--

```
average := average / total_status * 100;  
RETURN average ;
```

END;

-- Δοκιμή με τους παρακάτω κώδικες --

BEGIN

DBMS_OUTPUT.PUT_LINE (AverageOfStatus('Σε Υλοποίηση'));

END;

```
SELECT DISTINCT AverageOfStatus('Ολοκληρωμένο')  
FROM VOUCHER_CLASS
```

127.0.0.1:8080/apex/f?p=4500:1001:3553597483654663::NO::

Application Express

Application Builder ▼ SQL Workshop ▼ Team Development ▼ Administration ▼

SQL Workshop > Object Browser

ORACLE Application Express

Home Application Builder ▼ SQL Workshop ▼ Team Development ▼ Administration ▼

Home > SQL Workshop > SQL Commands

Autocommit Rows 10 Save Run

```
CREATE OR REPLACE FUNCTION AverageOfStatus (GIVEN_STATUS IN VARCHAR2)  
RETURN number IS  
    average number(6,2);  
    total_status number(6);  
BEGIN  
    --Επέλεξε το άθροισμα των καταστάσεων, όπου η κατάσταση είναι ίδια με την δοθείσα--  
    SELECT DISTINCT COUNT(STATUS) INTO average  
    FROM VOUCHER_CLASS  
    WHERE STATUS = GIVEN_STATUS;  
    --Επέλεξε το άθροισμα των καταστάσεων--  
    SELECT DISTINCT COUNT(STATUS) INTO total_status  
    FROM VOUCHER_CLASS;  
    --Επέστρεψε το μέσο όρο--  
    average := average / total_status * 100;  
    RETURN average ;  
END;  
  
-- Δοκιμή με τους παρακάτω κώδικες --  
BEGIN  
    DBMS_OUTPUT.PUT_LINE (AverageOfStatus('Σε Υλοποίηση'));  
END;  
  
SELECT DISTINCT AverageOfStatus('Ολοκληρωμένο')  
FROM VOUCHER_CLASS
```

Results Explain Describe Saved SQL History

66.67

Statement processed.

0.08 seconds