

# ΣΤΕΛΙΟΣ ΒΑΚΟΥΦΗΣ

Τέταρτη Σειρά Ασκήσεων SQL

## Ζήτημα 1

Να διατυπώσετε τρία έως πέντε ερωτήματα σε φυσική γλώσσα. Στην συνέχεια να γράψετε αντίστοιχα SQL επερωτήματα και να δημιουργήσετε κατάλληλα ευρετήρια που να επιταχύνουν την εκτέλεσή τους.

1. How many unique books of the same author does the library have that are written in Italian (ITA) for each author and for "Unknown Author"?

USE LIBRARY

```
SELECT a.author, COUNT(DISTINCT br.bibno) AS "Books", br.lang AS "Language"
      FROM bibrecs AS br
    LEFT JOIN bibauthors AS ba
          ON br.bibno = ba.bibno
    LEFT JOIN authors AS a
          ON ba.aid = a.aid
 WHERE br.lang = 'ITA'
 GROUP BY a.author, br.lang
 ORDER BY Books DESC
;
```

```
CREATE NONCLUSTERED INDEX Language_IDX
      ON bibrecs (lang);
```

```
DROP INDEX Language_IDX
      ON bibrecs;
```

2. What is the distribution between women who borrow books and men who borrow books?

USE LIBRARY

```
SELECT COUNT(bid) AS "Borrowers", b.sex AS "Gender"
      FROM borrowers AS b
 GROUP BY sex
;
```

```
CREATE INDEX Sex_IDX ON borrowers(sex);
```

```
DROP INDEX Sex_IDX ON borrowers;
```

3. For each book that is being borrowed show us the distribution of gender to authors if the number is more than 10, the year is 2006 and the book copy is from "OPA".

```
USE LIBRARY
```

```
SELECT ISNULL(a.author, 'Unkown') AS Author,
COUNT(b.sex) AS "Borrowers",
(CASE
    WHEN b.sex = 'F' THEN 'Female' ELSE 'Male'
END) AS "Gender"

FROM loanstats AS ls
INNER JOIN borrowers AS b
    ON ls.bid = b.bid
LEFT JOIN copies AS c
    ON ls.copyno = c.copyno
LEFT JOIN bibauthors AS ba
    ON c.bibno = ba.bibno
LEFT JOIN authors AS a
    ON ba.aid = a.aid
WHERE DATEPART(YEAR, ls.loandate) = '2006' AND
      c.copyloc = 'OPA'
GROUP BY a.author, b.sex
HAVING COUNT(b.sex) > 10
ORDER BY COUNT(b.sex) DESC
;

CREATE NONCLUSTERED INDEX Copyloc_IDX
    ON copies(copyloc)
;

CREATE NONCLUSTERED INDEX Bibno_Aid_IDX
    ON bibauthors(bibno, aid)
;

CREATE NONCLUSTERED INDEX Aid_Authors_IDX
    ON authors(aid, author)
;

DROP INDEX Bibno_Aid_IDX
    ON bibauthors
;

DROP INDEX Aid_Authors_IDX
    ON authors
;

DROP INDEX Copyloc_IDX
    ON copies
;
```

4. Who is the most popular author for women and for men? (Exclude "Unknown Author")

```
CREATE VIEW genders_authors AS
  SELECT a.author,
         COUNT(b.sex) AS Borrowers,
         (CASE
            WHEN b.sex = 'F' THEN 'Female' ELSE 'Male'
          END) AS "Gender"

  FROM loanstats AS ls
 INNER JOIN borrowers AS b
    ON ls.bid = b.bid
 LEFT JOIN copies AS c
    ON ls.copyno = c.copyno
 LEFT JOIN bibauthors AS ba
    ON c.bibno = ba.bibno
 LEFT JOIN authors AS a
    ON ba.aid = a.aid
 GROUP BY a.author, b.sex
;

SELECT *
  FROM genders_authors AS ga
 WHERE ga.Borrowers IN (
    SELECT MAX(Borrowers)
      FROM genders_authors
     WHERE author IS NOT NULL
    GROUP BY Gender
  )
;

CREATE NONCLUSTERED INDEX Copyno_IDX
  ON loanstats(copyno)
;

CREATE NONCLUSTERED INDEX Bibno_Aid_IDX
  ON bibauthors(bibno, aid)
;

CREATE NONCLUSTERED INDEX Aid_Authors_IDX
  ON authors(aid, author)
;

DROP INDEX Copyno_IDX
  ON loanstats
;

DROP INDEX Bibno_Aid_IDX
  ON bibauthors
;

DROP INDEX Aid_Authors_IDX
  ON authors
;DROP VIEW genders_authors
```

5. How many copies of books does the library have for publishers “ΟΠΑ” and “CEPR”?

```
SELECT p.pubname, COUNT(c.bibno) AS "Copies of Books"
FROM copies AS c
LEFT JOIN bibrecs AS b
    ON c.bibno = b.bibno
LEFT JOIN publishers AS p
    ON b.pubid = p.pubid
WHERE p.pubname IN ('ΟΠΑ', 'CEPR')
GROUP BY p.pubname
ORDER BY [Copies of Books] DESC
;
```

```
CREATE NONCLUSTERED INDEX Pubid_Pubname_IDX
ON publishers(pubid, pubname)
;
```

```
CREATE NONCLUSTERED INDEX Bibno_IDX
ON copies(bibno)
;
```

```
CREATE NONCLUSTERED INDEX Pubid_IDX
ON bibrecs(pubid)
;
```

```
DROP INDEX Pubid_Pubname_IDX
ON publishers
;
```

```
DROP INDEX Bibno_IDX
ON copies
;
```

```
DROP INDEX Pubid_IDX
ON bibrecs
;
```

## Ζήτημα 2

Να αναπτύξετε μία αποθήκη δεδομένων ώστε η διοίκηση της βιβλιοθήκης να είναι σε θέση να αναλύει τα δεδομένα και να παράγει στατιστικές αναφορές σχετικά με την πορεία των δανεισμών. Οι απαιτήσεις της διοίκησης εστιάζουν στην ανάλυση του αριθμού των δανεισμών ανά τοποθεσία (copyloc), είδος τεκμηρίου (material), τμήμα (depcode) και φύλο (sex) δανειζομένων, καθώς και οποιονδήποτε συνδυασμό αυτών. Εξυπακούεται ότι στην ανάλυση των δεδομένων θα πρέπει να ληφθεί υπόψη ο παράγοντας του χρόνου έτσι ώστε, η διοίκηση της βιβλιοθήκης να είναι σε θέση να παράγει στατιστικές αναφορές με τα στοιχεία των δανεισμών ανα ημέρα, εβδομάδα, μήνα και τετράμηνο για το διάστημα ενός ή περισσότερων ετών. Συγκεκριμένα:

1. Να δημιουργήσετε μία βάση δεδομένων με το όνομα LIBDW (Library Data Warehouse).
2. Να υλοποιήσετε το λογικό σχήμα της αποθήκης δεδομένων το οποίο θα πρέπει να έχει την μορφή αστέρα (Star Schema).
3. Να γράψετε κατάλληλες εντολές σε γλώσσα SQL, οι οποίες θα τροφοδοτούν το σχήμα της αποθήκης με τα απαραίτητα στοιχεία από την βιβλιογραφική βάση LIBRARY.

### BUILD FACTORY TABLE

```
USE LIBDW
```

```
SELECT * INTO FACTORY_TABLE
FROM
(
    SELECT ROW_NUMBER() OVER (ORDER BY borrowers.bid ASC) AS ID,
           borrowers.bid, borrowers.sex, borrowers.bname,
           departments.depname, departments.depcode,
           loanstats.lid, loanstats.copyno, loanstats.loandate,
           copies.copyloc, copies.bibno,
           bibrecs.material
    FROM borrowers
    LEFT JOIN departments
        ON borrowers.depcode = departments.depcode
    LEFT JOIN loanstats
        ON borrowers.bid = loanstats.bid
    LEFT JOIN copies
        ON loanstats.copyno = copies.copyno
    LEFT JOIN bibrecs
        ON copies.bibno = bibrecs.bibno
) AS FC_TABLE
;
```

## STAR SCHEMA CREATION AND DATA INSERTION

```
USE LIBDW
```

```
--- Calender table creation and insertion
```

```
CREATE TABLE Calender (  
    cal_key DATE NOT NULL,  
    lday INT NOT NULL,  
    lweek INT NOT NULL,  
    lmonth INT NOT NULL,  
    lquarter INT NOT NULL,  
    lyear INT NOT NULL,  
    PRIMARY KEY (cal_key)  
);
```

```
INSERT INTO Calender  
SELECT  
    DISTINCT ft.loandate,  
    DATEPART(DAYOFYEAR, ft.loandate) AS "lday",  
    DATEPART(WEEK, ft.loandate) AS "lweek",  
    DATEPART(MONTH, ft.loandate) AS "lmonth",  
    DATEPART(QUARTER, ft.loandate) AS "lquarter",  
    DATEPART(YEAR, ft.loandate) AS "lyear"  
FROM FACTORY_TABLE AS ft  
WHERE ft.loandate IS NOT NULL  
;
```

```
--- Customers table creation and insertion
```

```
CREATE TABLE Customers(  
    p_id INT PRIMARY KEY NOT NULL,  
    sex VARCHAR(1) NOT NULL,  
    pname VARCHAR(65) NOT NULL  
);
```

```
INSERT INTO Customers  
SELECT DISTINCT ft.bid, ft.sex, ft.bname  
FROM FACTORY_TABLE AS ft  
;
```

```
--- Departments table creation and insertion
```

```
CREATE TABLE Departments(  
    d_id INT PRIMARY KEY NOT NULL,  
    depname VARCHAR(65) NOT NULL  
);
```

```
INSERT INTO Departments  
SELECT DISTINCT ft.depname, ft.depname  
FROM FACTORY_TABLE AS ft  
;
```

```

--- BookMaterials table creation and insertion
CREATE TABLE BookMaterials(
    bn_id INT PRIMARY KEY NOT NULL IDENTITY(1, 1),
    material VARCHAR(25) NOT NULL
);

INSERT INTO BookMaterials
    SELECT DISTINCT ft.material
    FROM FACTORY_TABLE AS ft
    WHERE ft.bibno IS NOT NULL
;

--- CopyLocations table creation and insertion
CREATE TABLE CopyLocations(
    cl_id INT PRIMARY KEY NOT NULL IDENTITY(1, 1),
    copyloc VARCHAR(3) NOT NULL
);

INSERT INTO CopyLocations
    SELECT DISTINCT ft.copyloc
    FROM FACTORY_TABLE AS ft
    WHERE ft.copyloc IS NOT NULL
;

--- LoanedBooks table creation and insertion
CREATE TABLE LoanedBooks(
    lb_id INT PRIMARY KEY NOT NULL,
    loandate DATE NOT NULL,
    bid INT NOT NULL,
    material_id INT NOT NULL,
    copy_loc INT NOT NULL,
    deptime INT NOT NULL,
    FOREIGN KEY (loandate) REFERENCES Calender(cal_key),
    FOREIGN KEY (bid) REFERENCES Customers(p_id),
    FOREIGN KEY (material_id) REFERENCES BookMaterials(bn_id),
    FOREIGN KEY (copy_loc) REFERENCES CopyLocations(cl_id),
    FOREIGN KEY (deptime) REFERENCES Departments(d_id)
);

INSERT INTO LoanedBooks
    SELECT
        ft.ID,
        (SELECT cal_key FROM Calender WHERE cal_key = ft.loandate),
        (SELECT p_id FROM Customers WHERE p_id = ft.bid),
        (SELECT bn_id FROM BookMaterials WHERE material =
ft.material),
        (SELECT cl_id FROM CopyLocations WHERE copyloc = ft.copyloc),
        (SELECT d_id FROM Departments WHERE depname = ft.depname)
    FROM FACTORY_TABLE AS ft
    WHERE ft.copyno IS NOT NULL
;

```

## Ζήτημα 3

Χρησιμοποιώντας την αποθήκη δεδομένων που δημιουργήσατε στο προηγούμενο ζήτημα, να γράψετε και να εκτελέσετε επερωτήσεις σε γλώσσα SQL, οι οποίες να απαντούν στα ακόλουθα ερωτήματα (απαιτήσεις) της διοίκησης της βιβλιοθήκης:

1. Εμφανίστε έναν κατάλογο με τον αριθμό των δανεισμών ανα έτος και τμήμα (deptcode).

```
USE LIBDW
```

```
SELECT c.lyear, d.depname, COUNT(*) AS "Loaned Books"
FROM LoanedBooks AS lb
LEFT JOIN Calender AS c
      ON lb.loandate = c.cal_key
LEFT JOIN Departments AS d
      ON lb.depcode = d.d_id
GROUP BY c.lyear, d.depname
ORDER BY c.lyear DESC, [Loaned Books] DESC
;
```

2. Εμφανίστε έναν κατάλογο με τον αριθμό των δανεισμών ανα τοποθεσία (copyloc) και είδος τεκμηρίου (material).

```
USE LIBDW
```

```
SELECT cl.copyloc, bm.material, COUNT(*) AS "Loaned Books"
FROM LoanedBooks AS lb
LEFT JOIN CopyLocations AS cl
      ON lb.copy_loc = cl.cl_id
LEFT JOIN BookMaterials AS bm
      ON lb.material_id = bm.bn_id
GROUP BY cl.copyloc, bm.material
ORDER BY cl.copyloc DESC, [Loaned Books] DESC
;
```

3. Εμφανίστε έναν κατάλογο με τον αριθμό των δανεισμών ανα μήνα και φύλο για το έτος 2000.

```
USE LIBDW
```

```
SELECT cal.lmonth, c.sex, COUNT(*) AS "Loaned Books"
FROM LoanedBooks AS lb
LEFT JOIN Customers AS c
      ON lb.bid = c.p_id
LEFT JOIN Calender AS cal
      ON lb.loandate = cal.cal_key
WHERE cal.lyear = '2000'
GROUP BY cal.lmonth, c.sex
ORDER BY cal.lmonth
;
```



4. Η διοίκηση της βιβλιοθήκης θέλει μια αναφορά που θα περιέχει μόνο τις ακόλουθες πληροφορίες.
- Τον συνολικό αριθμό των δανεισμών
  - Το αριθμό των δανεισμών ανα έτος
  - Τον αριθμό των δανεισμών ανα έτος και τμήμα
  - Τον αριθμό των δανεισμών ανα έτος, τμήμα και φύλο.
- Γράψτε **μία μόνο** επερώτηση σε γλώσσα SQL η οποία να παράγει την παραπάνω αναφορά.

```
USE LIBDW
```

```
SELECT
```

```
    cal.loanyear, dep.depname, c.sex, COUNT(*)
FROM   LoanedBooks AS lb
LEFT JOIN Calender AS cal
      ON lb.loandate = cal.cal_key
LEFT JOIN Departments AS dep
      ON lb.depcode = dep.d_id
LEFT JOIN Customers AS c
      ON lb.bid = c.p_id
GROUP BY ROLLUP(cal.loanyear, dep.depname, c.sex)
ORDER BY cal.loanyear DESC, dep.depname DESC, c.sex DESC
```

```
;
```

5. Γράψτε μια επερώτηση σε γλώσσα SQL το αποτέλεσμα της οποίας είναι η δημιουργία ενός κύβου (data cube), κάθε κελί του οποίου περιέχει τον αριθμό των δανεισμών για έναν συγκεκριμένο συνδυασμό τιμών (έτους, τοποθεσίας, φύλου) όπου:

- έτος= έτος δανεισμού
- τοποθεσία=τοποθεσία αντιτύπου
- φύλο=φύλο δανειζομένου

```
USE LIBDW
```

```
SELECT
```

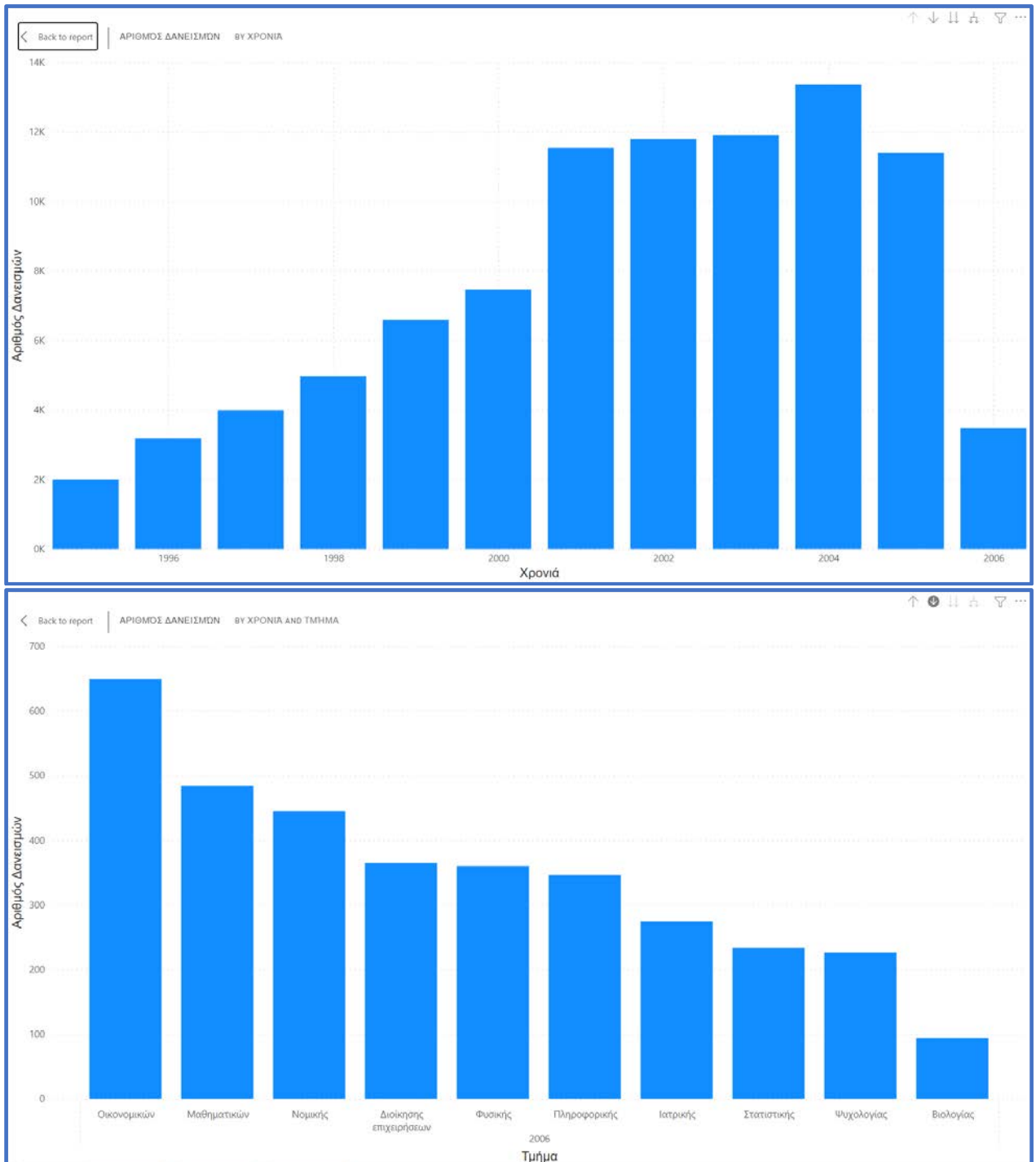
```
    cal.loanyear, cl.copyloc, c.sex, COUNT(*) AS "Loaned Books"
FROM   LoanedBooks AS lb
LEFT JOIN Calender AS cal
      ON lb.loandate = cal.cal_key
LEFT JOIN CopyLocations AS cl
      ON lb.depcode = cl.cl_id
LEFT JOIN Customers AS c
      ON lb.bid = c.p_id
GROUP BY CUBE(cal.loanyear, cl.copyloc, c.sex)
ORDER BY cal.loanyear DESC, cl.copyloc DESC, c.sex DESC
```

```
;
```

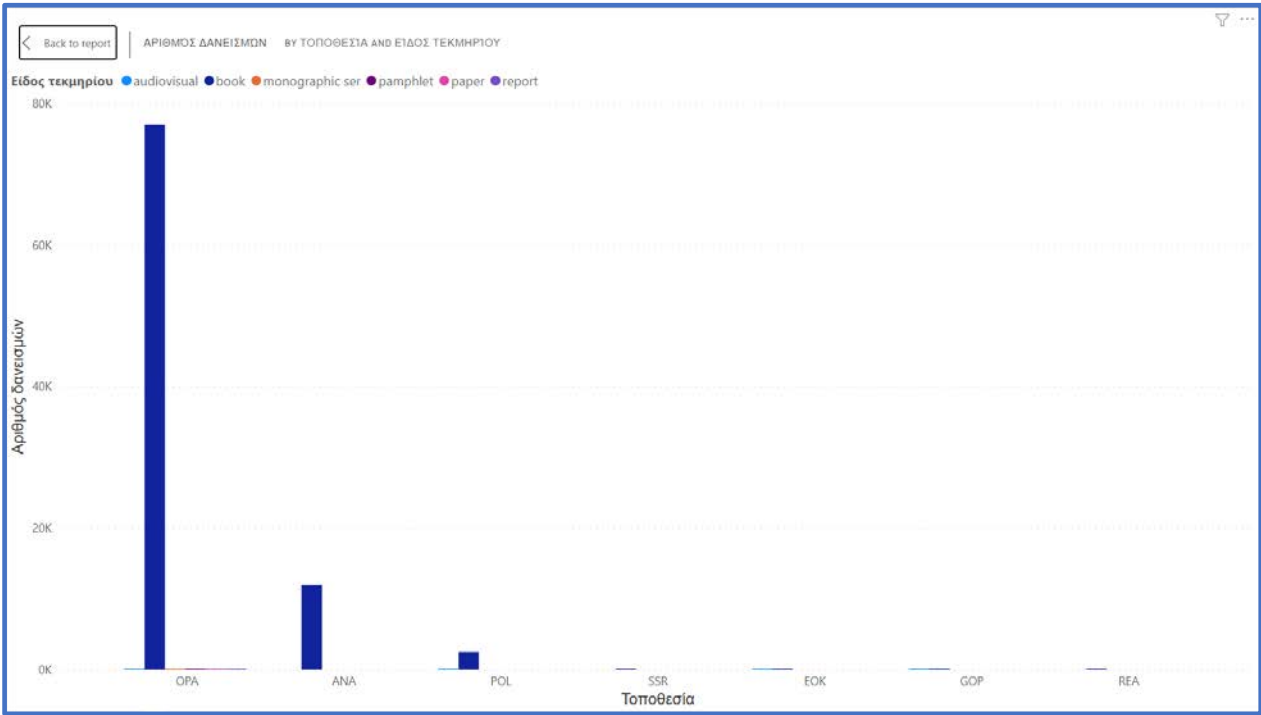
## Ζήτημα 4

Να χρησιμοποιήσετε το powerBI ώστε να συνδεθείτε στην αποθήκη δεδομένων που δημιουργήσατε. Στη συνέχεια να φτιάξετε κατάλληλα διαγράμματα ώστε να αναπαραστήσετε γραφικά τις αναφορές που ζητούνται στο προηγούμενο ζήτημα (ζήτημα 3).

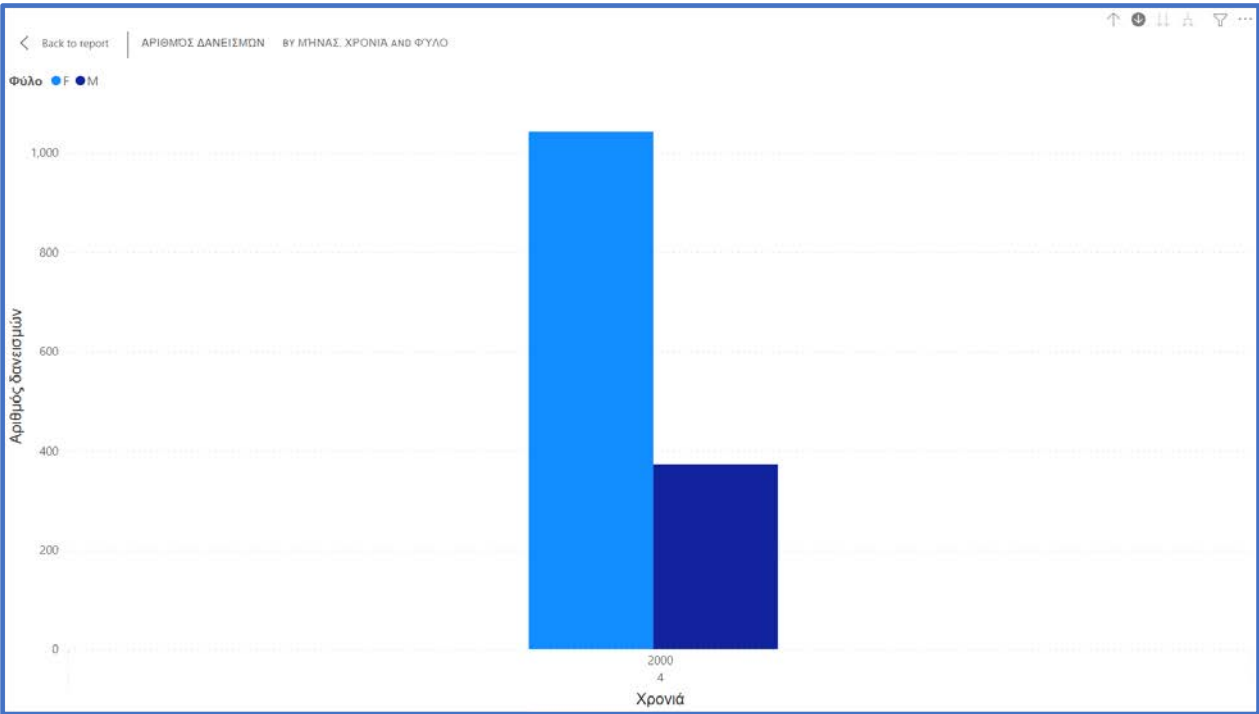
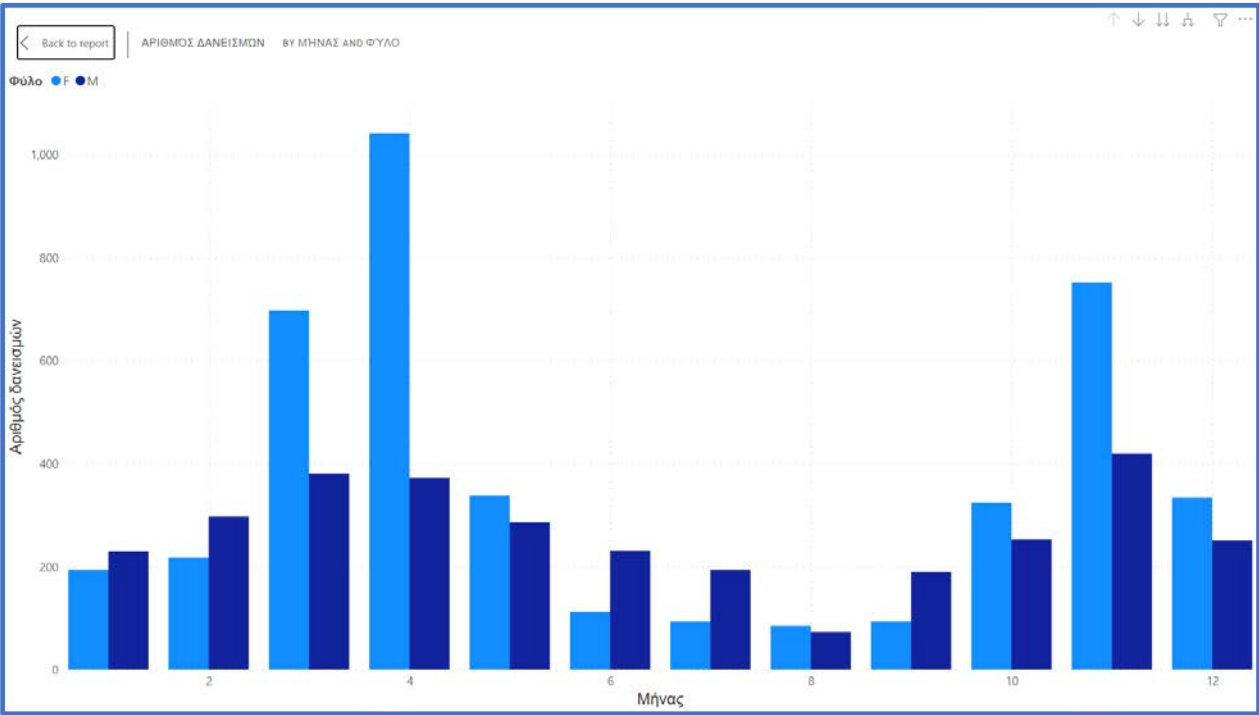
### Ερώτημα 1.



Ερώτημα 2.

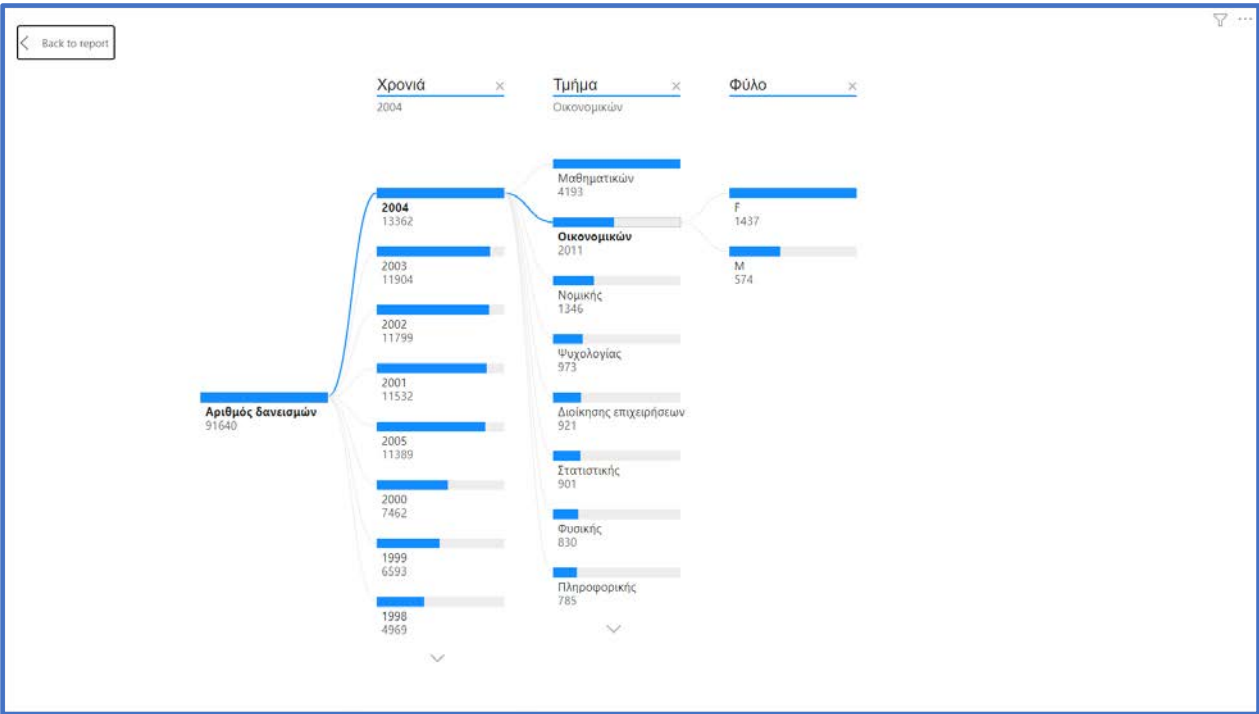


Ερώτημα 3.



Ερώτημα 4.

Χρονιά	Βιολογίας	Διοίκησης επιχειρήσεων	Ιατρικής	Μαθηματικών	Νομικής	Οικονομικών	Πληροφορικής	Στατιστικής	Φυσικής	Ψυχολογίας	Total
2004	673	921	729	4193	1346	2011	785	901	830	973	13362
2003	437	846	541	3060	1873	2620	877	485	733	432	11904
2002	587	921	606	2286	1772	2669	1098	448	907	505	11799
F	207	496	300	1471	766	1154	645	201	362	329	5931
M	380	425	306	815	1006	1515	453	247	545	176	5868
2001	277	1039	429	3371	1835	1942	1119	179	949	392	11532
2005	370	944	963	2078	1400	1919	848	840	1040	987	11389
2000	106	974	149	2079	1305	1070	754	55	866	104	7462
1999	212	816	119	1291	1116	903	843	70	1193	30	6593
1998	22	502	69	836	969	831	931	85	693	31	4969
1997	11	290	69	732	1077	748	546	24	450	32	3979
2006	94	365	274	484	445	649	346	233	360	226	3476
1996	11	178	27	795	746	783	365	16	246	13	3180
1995	5	61	38	594	416	520	155	31	144	31	1995
Total	2805	7857	4013	21799	14300	16665	8667	3367	8411	3756	91640



Ερώτημα 5.

Χρονιά	ANA	EOK	GOP	OPA	POL	REA	SSR	Total
1995	20			1750	224		1	1995
1996	43			2842	290		5	3180
1997	77			3591	304		7	3979
1998	110	2		4642	213		2	4969
1999	222	1		6130	237		3	6593
2000	487			6724	244	1	6	7462
2001	1422			9839	267	1	3	11532
2002	2088	12	1	9372	321		5	11799
F	1036		1	4743	149		2	5931
M	1052	12		4629	172		3	5868
2003	1896	4	2	9794	203		5	11904
2004	2487	3	1	10708	158		5	13362
2005	2447	1	3	8881	53		4	11389
2006	665		1	2808	2			3476
Total	11964	23	8	77081	2516	2	46	91640

