



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Εισαγωγή στις Βάσεις Δεδομένων II

Ενότητα: Η γλώσσα SQL

Διδάσκων: Πηγουνάκης Κωστής
ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΣΧΟΛΗ ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΣΤΗΜΩΝ



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται στην άδεια χρήσης **Creative Commons** και ειδικότερα ***Αναφορά – Μη εμπορική Χρήση – Όχι Παράγωγο Έργο 3.0 Ελλάδα*** (***Attribution – Non Commercial – Non-derivatives 3.0 Greece***)



[ή επιλογή ενός άλλου από τους έξι συνδυασμούς]

[και αντικατάσταση λογότυπου άδειας όπου αυτό έχει μπει (σελ. 1, σελ. 2 και τελευταία)]

- Εξαιρείται από την ως άνω άδεια υλικό που περιλαμβάνεται στις διαφάνειες του μαθήματος, και υπόκειται σε άλλου τύπου άδεια χρήσης. Η άδεια χρήσης στην οποία υπόκειται το υλικό αυτό αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Structured Query Language (SQL)

Εισαγωγή στις Σχεσιακές Βάσεις Δεδομένων (ΟΙΚ3501)

Κωστής Πηγουνάκης

Τελευταία ενημέρωση : 18/5/2015

Structured Query Language

- Κοινή γλώσσα για όλα τα σχεσιακά ΣΔΒΔ: η γλώσσα SQL.
- Κάθε σχεσιακό ΣΔΒΔ έχει τη δυνατότητα να δεχθεί και να εκτελέσει εντολές της γλώσσας SQL.
- Βασίζεται στη σχεσιακή άλγεβρα, αν και δεν είναι αυστηρή υλοποίηση των πράξεών της
- Είναι μια δηλωτική γλώσσα (μη-διαδικαστική) στην οποία περιγράφουμε τι θέλουμε κι όχι πώς θα το αποκτήσουμε
- Μπορεί να ενσωματωθεί σε γλώσσες προγραμματισμού
- Το λογισμικό του σχεσιακού ΣΔΒΔ αναλαμβάνει την εύρεση του τρόπου ανάκτησης των δεδομένων από τη βάση

Σύντομο ιστορικό της SQL

- Η SQL αναπτύχθηκε το 1974 από τον Donald D. Chamberlin Raymond F. Boyce, που εργαζόταν στην **IBM** και αρχικά χρησιμοποιήθηκε στα δικά της ΣΔΒΔ.
- Στα τέλη της δεκαετίας του '70, η **ORACLE**[®] δημιούργησε το δικό της ΣΔΒΔ, αλλά υιοθέτησε την SQL.
- Σύντομα ακολούθησαν και άλλες εταιρείες λογισμικού που ανέπτυξαν τα δικά τους συστήματα τα οποία υιοθέτησαν την SQL.
- Η ευρεία διάδοση της SQL οφείλεται στο γεγονός ότι τόσο το Αμερικανικό Ινστιτούτο Τυποποίησης (American National Standards Institute — ANSI) όσο και ο Παγκόσμιος Οργανισμός Τυποποίησης (International Organization for Standardization — ISO) όρισαν την SQL ως την κοινή γλώσσα των ΣΔΒΔ.

Πρότυπα της SQL

- Παρά τις σημαντικές προσπάθειες που καταβλήθηκαν από τους διεθνείς οργανισμούς, ο στόχος των προτύπων της SQL δεν επιτεύχθηκε απόλυτα:
- Τα σχεσιακά ΣΔΒΔ υποστηρίζουν εκδοχές της SQL με κεντρικό τους πυρήνα το πρότυπο, παρέχοντας όμως πολλές πρόσθετες δυνατότητες (επεκτάσεις) που δεν συμπεριλαμβάνονται στο πρότυπο.

Εντολές SQL

- Οι εντολές της SQL μπορεί να πληκτρολογηθούν από το χρήστη και να υποβληθούν προς άμεση εκτέλεση στο σχεσιακό ΣΔΒΔ και τα αποτελέσματα των εντολών SQL εμφανίζονται στην οθόνη του χρήστη.
- Η SQL είναι μία γλώσσα για τη διαχείριση δεδομένων – δεν μία πλήρης γλώσσα προγραμματισμού όπως η C++ ή η Java
- Πολλές φορές είναι απαραίτητο η SQL να ενσωματωθεί μέσα στον κώδικα μίας άλλης γλώσσας προγραμματισμού. Τότε, η SQL αναφέρεται ως ενσωματωμένη SQL (embedded SQL).
- Στην περίπτωση αυτή, τα αποτελέσματα των SQL εντολών γνωστοποιούνται στο πρόγραμμα που είναι γραμμένο στη γενική γλώσσα προγραμματισμού, το οποίο και τα χειρίζεται.

Εισαγωγικά για τις εντολές SQL

- Οι εντολές της SQL θυμίζουν έντονα απλές προτάσεις της Αγγλικής γλώσσας — άλλωστε έγινε προσπάθεια γι' αυτό, ώστε να είναι εύκολη η εκμάθησή της.
- Κάθε εντολή περιέχει ένα σύνολο **δεσμευμένων λέξεων**, δηλαδή λέξεων που έχουν συγκεκριμένο νόημα που δεν μπορεί να αλλάξει, και ένα σύνολο λέξεων που ορίζονται από το χρήστη.
- Ο τρόπος γραφής μίας εντολής SQL είναι αυστηρά καθορισμένος και ακολουθεί συγκεκριμένους συντακτικούς κανόνες.
- Στην περίπτωση που οι κανόνες αυτοί παραβιάζονται, η εντολή δεν μπορεί να γίνει κατανοητή από το ΣΔΒΔ.

Δομικές μονάδες SQL I

- Δομικές μονάδες μιας τυπικής εκδοχής της SQL :
 - **Γλώσσα ορισμού δεδομένων (Data Definition Language, DDL):** εντολές που μας επιτρέπουν να υλοποιήσουμε σχέσεις και γενικά όλη τη δομή μιας βάσης δεδομένων.
 - **Γλώσσα χειρισμού δεδομένων (Data Manipulation Language, DML):** διαχείριση των δεδομένων της εφαρμογής, όπως την εισαγωγή, διαγραφή, ανάκτηση και τροποποίηση δεδομένων.

Δομικές μονάδες SQL II

- **Ορισμός όψεων της βάσης (View Definition):** δημιουργία εικονικών πινάκων (views -virtual tables) που περιέχουν δεδομένα από έναν ή περισσότερους πίνακες της βάσης.
- **Ορισμός εξουσιοδοτήσεων (Authorization):** δημιουργία ομάδων χρηστών και απόδοση διαφορετικών δικαιωμάτων πρόσβασης σε καθένα
- **Διαχείριση ακεραιότητας (Integrity):** επιτρέπει το λεπτομερή έλεγχο των δεδομένων που καταχωρούνται ώστε να μην παραβιάζονται οι **κανόνες** ακεραιότητας οι οποίοι, όταν τηρούνται, απομακρύνουν τον κίνδυνο **ασυνεπών δεδομένων**

Γλώσσα ορισμού δεδομένων

Η διαχείριση μέσω της DDL περιλαμβάνει τον **ορισμό** τη **μεταβολή της δομής** και τη **διαγραφή** των παρακάτω αντικειμένων:

- **Πίνακες (Tables):** το δομικό χαρακτηριστικό μιας σχεσιακής βάσης δεδομένων, καθώς περιέχουν τα δεδομένα που καταχωρούνται σε αυτή
- **Όψεις (Views):** εικονικοί πίνακες (virtual tables) οι οποίοι περιέχουν δεδομένα από έναν ή περισσότερους πίνακες της βάσης.
- **Δείκτες (Indices):** ειδικές δομές δεδομένων που επιταχύνουν τη διαδικασία της αναζήτησης πληροφοριών από τη βάση.

Εντολές ορισμού δεδομένων

- CREATE TABLE (δημιουργία πίνακα)
- CREATE INDEX (δημιουργία ευρετηρίου)
- CREATE VIEW (δημιουργία όψης)

- DROP TABLE (διαγραφή πίνακα)
- DROP INDEX (διαγραφή ευρετηρίου)
- DROP VIEW (διαγραφή όψης)

- ALTER (τροποποίηση της δομής των αντικειμένων της βάσης, δηλ, πινάκων, ευρετηρίων, όψεων)

CREATE TABLE

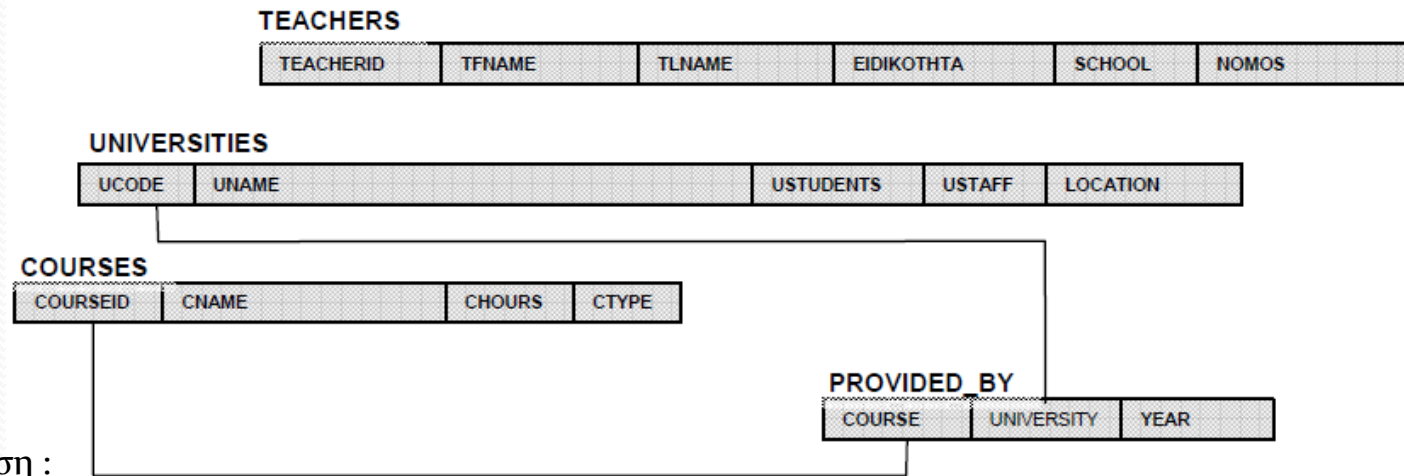
- CREATE TABLE πίνακας (όνομα_στήλης τύπος_στήλης, ...);

```
CREATE TABLE EMPLOYEE (  
  FNAME      VARCHAR (15)  NOT NULL,  
  MINIT      CHAR (1),  
  LNAME      VARCHAR (15)  NOT NULL,  
  SSN        CHAR (9)      NOT NULL,  
  BDATE      DATE,  
  ADDRESS    VARCHAR (30),  
  SEX        VARCHAR (30),  
  SALARY     INTEGER,  
  SUPERSSN   CHAR (9),  
  DNO        INTEGER ,  
  PRIMARY KEY (`SSN`) );
```

Παρατηρήσεις για CREATE

- Ο ορισμός του πρωτεύοντος κλειδιού αλλάζει ανάμεσα στις παραλλαγές της SQL
- Οι πίνακες που δημιουργούνται με την CREATE TABLE, είναι οι **βασικοί (base tables)**, διότι τα δεδομένα τους αποθηκεύονται σε αρχεία του συστήματος.
- Οι όψεις είναι **εικονικοί πίνακες (virtual tables)**, που μπορεί να αποθηκεύονται και αυτοί σε αρχείο, χωρίς όμως κάτι τέτοιο να είναι υποχρεωτικό.
- Η σειρά που αποθηκεύονται οι τιμές των πεδίων στους πίνακες, είναι ίδια με τη σειρά που έχουν δηλωθεί τα πεδία των πινάκων κατά την κλήση της εντολής.

Παράδειγμα CREATE για ΒΔ



Υλοποίηση :

```
CREATE TABLE TEACHERS(TEACHERID INT, TFNAME TEXT(50), TLNAME TEXT(50), EIDIKOTHTA  
TEXT(30), SCHOOL TEXT(10), NOMOS TEXT(50));
```

```
CREATE TABLE UNIVERSITIES(UCODE TEXT(10), UNAME TEXT(50), USTUDENTS INT, USTAFF INT,  
LOCATION TEXT(50));
```

```
CREATE TABLE COURSES(COURSEID INT, CNAME TEXT(50), CHOURS INT, CTYPE TEXT(1));
```

```
CREATE TABLE PROVIDED_BY(COURSE INT, UNIVERSITY TEXT(10), YEAR TEXT(10));
```


Παραδείγματα DROP – ALTER

DROP TABLE DEPENDENTS

Διαγράφει τον πίνακα DEPENDENTS και όλα τα δεδομένα

ALTER TABLE table_name **ADD** column_name datatype

ALTER TABLE EMPLOYEE **ADD** JOB VARCHAR(12)

προσθέτει στον πίνακα EMPLOYEE το πεδίο JOB που είναι συμβολοσειρά με μέγιστο μήκος ίσο με 12 χαρακτήρες.

ALTER TABLE table_name **DROP COLUMN** column_name

ALTER TABLE table_name **MODIFY** column_name datatype

ALTER TABLE table_name **RENAME COLUMN** old_name to new_name

Διαχείριση όψεων (Views)

- **Όψη (view):** ένας απλός πίνακας που προκύπτει από το συνδυασμό των πεδίων ενός ή περισσότερων πινάκων, οι οποίοι μπορεί να είναι είτε βασικοί πίνακες ή άλλες όψεις (εικονικοί πίνακες)
- Οι βασικοί πίνακες από τους οποίους δημιουργείται μια όψη, αναφέρονται και ως **πίνακες ορισμού της όψης (defining tables)**.
- Για να δημιουργήσουμε μια όψη χρησιμοποιούμε την εντολή **CREATE VIEW**
- Για να διαγράψουμε μια όψη χρησιμοποιούμε την εντολή **DROP VIEW**

Παραδείγματα VIEW

```
CREATE VIEW WORKS_ON1
  AS SELECT FNAME, LNAME, PNAME, HOURS
  FROM EMPLOYEE, PROJECT, WORKS_ON
  WHERE SSN = ESSN
  AND PNO = PNUMBER
```

```
CREATE VIEW DEPT_INFO (DNAME, EMP_NO, AVG_SAL)
  AS SELECT DNAME, COUNT (*), AVERAGE (SALARY)
  FROM DEPARTMENT, EMPLOYEE
  WHERE DNUMBER=DNO
  GROUP BY DNAME
```

Διαχείριση δεικτών I

- **Δείκτης (index):** μια δομή δεδομένων που αποθηκεύεται σε ειδικά αρχεία της βάσης (index files) και έχει στόχο να επιταχύνει τη διαδικασία αναζήτησης πληροφορίας από τους πίνακες της ΒΔ
- Οι δείκτες συνήθως ορίζονται για συγκεκριμένα πεδία αυτών των πινάκων (**indexing fields**) και για κάθε τιμή των εν λόγω πεδίων, αποθηκεύουν ένα σύνολο από pointers που δείχνουν τα δεδομένα της κάθε εγγραφής.

Διαχείριση δεικτών II

- Τα αρχεία δεικτών είναι αρκετά μικρότερα σε μέγεθος σε σχέση με τα αρχεία δεδομένων και η διαδικασία αναζήτησης της πληροφορίας γίνεται μέσω γνωστών τεχνικών αναζήτησης, όπως η **δυναμική αναζήτηση**

http://en.wikipedia.org/wiki/Binary_search_algorithm

- Η δημιουργία ενός **δείκτη** γίνεται χρησιμοποιώντας την εντολή **CREATE INDEX**
- Η διαγραφή ενός δείκτη γίνεται με την εντολή **DROP INDEX**

Παραδείγματα index

```
CREATE INDEX LNAME_INDEX  
ON EMPLOYEE (LNAME)
```

```
CREATE INDEX NAMES_INDEX  
ON EMPLOYEE (LNAME ASC, FNAME DESC, MINIT)
```

```
CREATE UNIQUE INDEX SSN_INDEX  
ON EMPLOYEE (SSN)
```

```
DROP INDEX LNAME_INDEX
```

Γλώσσα χειρισμού δεδομένων

- Η **γλώσσα χειρισμού δεδομένων (Data Manipulation Language, DML)**, επιτρέπει τη διαχείριση των δεδομένων των πινάκων της ΒΔ, και πιο συγκεκριμένα, την **εισαγωγή, διαγραφή, και τροποποίηση των εγγραφών** των πινάκων.
- Μέσω της DML έχουμε τη δυνατότητα να ανακτήσουμε από τους πίνακες, δεδομένα, τα οποία πληρούν κάποια συγκεκριμένα κριτήρια.
- Η DML **δεν ελέγχει** αν οι εγγραφές που επηρεάζονται ικανοποιούν τους κανόνες ακεραιότητας του λογικού σχεδιασμού που αποτυπώνονται στο σχήμα της ΒΔ

SELECT

Τελεστής Επιλογής και Προβολής:

SELECT *<attribute list>*

FROM *<table list>*

WHERE *<condition>*

συνθήκη σύζευξης (join condition) / AND, OR, NOT

```
SELECT BDATE, ADDRESS  
FROM EMPLOYEE  
WHERE FNAME='John' AND MINIT='B' AND LNAME='Smith'
```

```
SELECT FNAME, LNAME, ADDRESS  
FROM EMPLOYEE , DEPARTMENT  
WHERE DNAME = 'Research' AND  
DEPARTMENT.DNUMBER = EMPLOYEE.DNO
```


Λέξεις-κλειδιά (keywords)

- **DISTINCT**

SELECT **DISTINCT** SALARY FROM EMPLOYEE

- **UNION**

SELECT PNUMBER FROM PROJECT, DEPARTMENT, EMPLOYEE
WHERE DNUM = DNUMBER AND MGRSSN = SSN AND LNAME = 'Smith'

UNION

SELECT PNUMBER FROM PROJECT, WORKS_ON, EMPLOYEE
WHERE PNUMBER = PNO AND ESSN = SSN AND LNAME = 'Smith'

- **IN (NESTED QUERIES)**

SELECT DISTINCT PNAME FROM PROJECT
WHERE PNUMBER **IN**

(SELECT PNUMBER FROM PROJECT, DEPARTMENT, EMPLOYEE
WHERE DNUM = DNUMBER AND MGRSSN = SSN AND LNAME = 'Smith')

OR PNUMBER **IN**

(SELECT PNO FROM WORKS_ON, EMPLOYEE WHERE ESSN = SSN AND LNAME = 'Smith')

INSERT

- **INSERT INTO** *<TABLE NAME>*

VALUES (*<v₁>*, *<v₂>*, , *<v_n>*)

INSERT INTO **EMPLOYEE**

```
VALUES ( 'Richard',  
        'K', 'Marini',  
        '653298653',  
        '30-Dec-52',  
        '98 Oak Forest, Katy, TX',  
        'M',  
        37000,  
        '987654321',  
        4)
```

INSERT INTO **EMPLOYEE** (FNAME,LNAME, SSN)

```
VALUES ( 'Richard', 'Marini', '653298653') NULLS!!!
```

INSERT INTO **DEPTS_INFO** (DNAME, EMP_NO, TOTAL_SAL)

```
SELECT DNAME, COUNT (*), SUM (SALARY)  
FROM DEPARTMENT, EMPLOYEE  
WHERE DNUMBER = DNO  
GROUP BY DNAME
```

DELETE

- Χρησιμοποιείται για τη **διαγραφή εγγραφών** από **ένα πίνακα** κάθε φορά.
- DELETE FROM EMPLOYEE
WHERE LNAME='Brown'
- DELETE FROM EMPLOYEE
WHERE SSN = '123456789'
- DELETE FROM EMPLOYEE
WHERE DNO
IN (SELECT DNUMBER FROM DEPARTMENT WHERE DNAME='Research')

UPDATE

- **UPDATE** *<TABLE_NAME>* **SET** *<Field_Name₁>* = '*<value₁>*', *<Field_Name₂>* = '*<value₂>*', ..., *<Field_Name_n>* = '*<value_n>*'

```
UPDATE PROJECT SET PLOCATION = 'Bellaire', DNUM = 5
WHERE PNUMBER = 10
```

```
UPDATE EMPLOYEE SET SALARY = SALARY * 1.1
WHERE DNO
IN ( SELECT DNUMBER FROM DEPARTMENT WHERE DNAME = 'Research' )
```

Περισσότερα για εξάσκηση:

- http://w3schools.com/sql/sql_tryit.asp

Οι επόμενες διαφάνειες βασίζονται
στο υλικό που συνοδεύει το βιβλίο
του Ε. Κεχρή «Σχεσιακές Βάσεις
Δεδομένων» -2^η έκδοση

Συγκεντρωτικές συναρτήσεις

- **Συγκεντρωτικές συναρτήσεις** (aggregate functions): Ειδικές συναρτήσεις της SQL για σύνθετες αριθμητικές πράξεις
- Οι συγκεντρωτικές συναρτήσεις είναι οι εξής:
 - **COUNT** : απαριθμεί το πλήθος των εγγραφών
 - **SUM** : υπολογίζει το άθροισμα μίας στήλης ή μίας αριθμητικής παράστασης μεταξύ στηλών
 - **MIN** : υπολογίζει το ελάχιστο μίας αριθμητικής στήλης
 - **MAX** : υπολογίζει το μέγιστο μίας αριθμητικής στήλης
 - **AVG** : υπολογίζει τη μέση τιμή μίας αριθμητικής στήλης

Παράδειγμα συγκεντρωτικών συναρτήσεων

ΣΥΝΤΗΡΗΣΗ

<u>ΑριθμόςΚυκλοφορίας</u>	<u>ΕτήσιοΚόστος</u>
EPH 1234	4500
YAA 2345	1200
EPH 3455	900
KZA 4567	3510
PHZ 6789	2700

```
SELECT AVG(ΕτήσιοΚόστος)  
FROM ΣΥΝΤΗΡΗΣΗ
```

AVG(ΕτήσιοΚόστος)
2562

Ο μέσος όρος το κόστους της συντήρησης

```
SELECT MIN(ΕτήσιοΚόστος)  
FROM ΣΥΝΤΗΡΗΣΗ
```

MIN(ΕτήσιοΚόστος)

900

Το ελάχιστο κόστους της συντήρησης

```
SELECT SUM(ΕτήσιοΚόστος)  
FROM ΣΥΝΤΗΡΗΣΗ
```

SUM(ΕτήσιοΚόστος)

12810

Το συνολικό κόστος συντήρησης

```
SELECT COUNT (ΑριθμόςΚυκλοφορίας)  
FROM ΣΥΝΤΗΡΗΣΗ  
WHERE ΕτήσιοΚόστος > 3000
```

COUNT(ΑριθμόςΚυκλοφορίας)

2

Το πλήθος των αυτοκινήτων με κόστος συντήρησης πάνω από 3000 €

Ομαδοποίηση εγγραφών

- Πολλές φορές είναι αντιμετωπίζουμε έναν πίνακα ως αποτελούμενο από ομάδες εγγραφών (γραμμών).
- Η ομαδοποίηση των εγγραφών επιτρέπει την εξαγωγή αποτελεσμάτων ανά ομάδα.
- Η ομαδοποίηση γίνεται εφικτή μέσω της εντολής **SELECT** με προσθήκη της λέξης-κλειδί **GROUP BY**

Παράδειγμα ομαδοποίησης I

ΕΠΙΔΟΣΕΙΣ_ΠΩΛΗΤΩΝ

<u>Κωδικός Πωλητή</u>	<u>Κωδικός Προϊόντος</u>	<u>Πωλήσεις</u>
1	1001	55
1	1002	60
1	1003	25
2	1001	70
2	1002	15

- Ο πίνακας **ΕΠΙΔΟΣΕΙΣ_ΠΩΛΗΤΩΝ** δείχνει τις πωλήσεις σε κάθε προϊόν που έκαναν οι πωλητές μίας εταιρείας στη διάρκεια του προηγούμενου μήνα.
- Θεωρούμε ότι ο πίνακας αυτός αποτελείται από δύο ομάδες εγγραφών:
 - α' ομάδα: πωλήσεις του πωλητή με κωδικό 1 (**ομάδα A**), ενώ
 - β' ομάδα: πωλήσεις του πωλητή με κωδικό 2 (**ομάδα B**).
- Στην περίπτωση αυτή ο χωρισμός σε ομάδες γίνεται ανάλογα με την τιμή που έχει η στήλη **Κωδικός Πωλητή**.

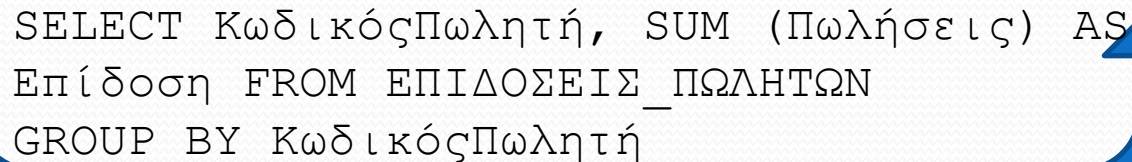
Ομαδοποίηση με το Group by

ΕΠΙΔΟΣΕΙΣ_ΠΩΛΗΤΩΝ

<u>Κωδικός Πωλητή</u>	<u>Κωδικός Προϊόντος</u>	<u>Πωλήσεις</u>
1	1001	55
1	1002	60
1	1003	25
2	1001	70
2	1002	15

<u>Κωδικός Πωλητή</u>	<u>Επίδοση</u>
1	130
2	85

```
SELECT ΚωδικόςΠωλητή, SUM (Πωλήσεις) AS  
Επίδοση FROM ΕΠΙΔΟΣΕΙΣ_ΠΩΛΗΤΩΝ  
GROUP BY ΚωδικόςΠωλητή
```



- Στην SQL είναι δυνατό οι συγκεντρωτικές συναρτήσεις να υπολογιστούν για κάθε ομάδα χωριστά.
- Αυτό γίνεται με χρήση των δεσμευμένων λέξεων **GROUP BY** με τη βοήθεια των οποίων δηλώνονται οι διάφορες ομάδες του πίνακα.
- Έτσι, το σύνολο των πωλήσεων για κάθε πωλητή χωριστά υπολογίζεται από την παραπάνω εντολή

Παράδειγμα ομαδοποίησης II

ΕΠΙΔΟΣΕΙΣ_ΠΩΛΗΤΩΝ

<u>Κωδικός Πωλητή</u>	<u>Κωδικός Προϊόντος</u>	<u>Πωλήσεις</u>
1	1001	55
1	1002	60
1	1003	25
2	1001	70
2	1002	15

- Οι εγγραφές ενός πίνακα μπορούν να ομαδοποιηθούν με διάφορους τρόπους, ανάλογα με τον υπολογισμό που απαιτείται.
- Έτσι, πίνακας **ΕΠΙΔΟΣΕΙΣ_ΠΩΛΗΤΩΝ** μπορεί να θεωρηθεί ότι αποτελείται από τρεις ομάδες εγγραφών:
 - η α' ομάδα αφορά τις πωλήσεις του προϊόντος 1001 (**ομάδα 1**),
 - η β' ομάδα αφορά τις πωλήσεις του προϊόντος 1002 (**ομάδα 2**) και
 - η γ' ομάδα αφορά τις πωλήσεις του προϊόντος 1003 (**ομάδα 3**).


Παράδειγμα ομαδοποίησης II

ΕΠΙΔΟΣΕΙΣ_ΠΩΛΗΤΩΝ

<u>Κωδικός Πωλητή</u>	Κωδικός Προϊόντος	Πωλήσεις
1	1001	55
1	1002	60
1	1003	25
2	1001	70
2	1002	15

Κωδικός Προϊόντος	ΠωλήσειςΠΡ
1001	125
1002	75
1003	25

```
SELECT ΚωδικόςΠροϊόντος, SUM (Πωλήσεις) AS  
ΠωλήσειςΠΡ FROM ΕΠΙΔΟΣΕΙΣ_ΠΩΛΗΤΩΝ  
GROUP BY ΚωδικόςΠροϊόντος
```



- Τώρα, η ομαδοποίηση γίνεται ανάλογα με την τιμή που έχει η στήλη **Κωδικός Προϊόντος**.
- Αυτή η ομαδοποίηση είναι απαραίτητη όταν θέλουμε να υπολογίσουμε το σύνολο των πωλήσεων για κάθε προϊόν.
- Η εντολή SQL που δίνει τον υπολογισμό αυτόν φαίνεται παραπάνω

Επιλογή με περισσότερους πίνακες

- Μέχρι τώρα είδαμε πώς μπορούμε να αναζητούμε με τη βοήθεια της SQL δεδομένα από ένα μόνο πίνακα.
- Ορισμένες φορές, ωστόσο, είναι απαραίτητο να αναζητήσουμε δεδομένα από δύο ή περισσότερους πίνακες, όπως εξηγείται στο επόμενο παράδειγμα.

Συνδυασμός δύο πινάκων

ΠΡΟΪΟΝΤΑ

Εγγραφή	<u>Κωδικός Προϊόντος</u>	<u>Περιγραφή</u>	<u>Τιμή Αγοράς</u>	<u>Κωδικός Προμηθευτή</u>	<u>Κατηγορία</u>
π1	1001	Στυλό	0.75	Π200	1
π2	1002	Μολύβι	1	Π250	1
π3	1003	Μαρκαδόρος	1.3	Π130	5
π4	1004	Δισκέτα 3,5"	0.8	Π120	2

ΚΑΤΗΓΟΡΙΕΣ

Εγγραφή	<u>Κωδικός Κατηγορίας</u>	<u>Ονομασία</u>
κ1	1	Είδη γραφείου
κ2	2	Είδη υπολογιστή
κ3	3	Υπολογιστής

Μπορούμε να συνδυάσουμε την εγγραφή π1 με την κ1 και να δημιουργήσουμε την εγγραφή:

εγγραφή	<u>Κωδικός Προϊόντος</u>	<u>Περιγραφή</u>	<u>Τιμή Αγοράς</u>	<u>Κωδικός Προμηθευτή</u>	<u>Κατηγορία</u>	<u>Κωδικός Κατηγορίας</u>	<u>Ονομασία</u>
πκ1: π1 με κ1	1001	Στυλό	0.75	Π200	1	1	Είδη γραφείου

Με την ίδια λογική μπορούμε να συνδυάσουμε την εγγραφή π1 με την κ2 και να δημιουργήσουμε την εγγραφή:

εγγραφή	<u>Κωδικός Προϊόντος</u>	<u>Περιγραφή</u>	<u>Τιμή Αγοράς</u>	<u>Κωδικός Προμηθευτή</u>	<u>Κατηγορία</u>	<u>Κωδικός Κατηγορίας</u>	<u>Ονομασία</u>
πκ2: π1 με κ2	1001	Στυλό	0.75	Π200	1	2	Είδη υπολογιστή

Καρτεσιανό γινόμενο

Συνολικά μπορούμε να συνδυάσουμε κάθε μία από τις 4 εγγραφές του α' πίνακα με κάθε μία από τις 3 εγγραφές του β' πίνακα και έτσι να δημιουργήσουμε τις παρακάτω 12 εγγραφές:

εγγραφή	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία	Κωδικός Κατηγορίας	Ονομασία
πκ1: π1 με κ1	1001	Στυλό	0.75	Π200	1	1	Είδη γραφείου
πκ2: π1 με κ2	1001	Στυλό	0.75	Π200	1	2	Είδη υπολογιστή
πκ3: π1 με κ3	1001	Στυλό	0.75	Π200	1	3	Υπολογιστής
πκ4: π2 με κ1	1002	Μολύβι	1	Π250	1	1	Είδη γραφείου
πκ5: π2 με κ2	1002	Μολύβι	1	Π250	1	2	Είδη υπολογιστή
πκ6: π2 με κ3	1002	Μολύβι	1	Π250	1	3	Υπολογιστής
πκ7: π3 με κ1	1003	Μαρκαδόρος	1.3	Π130	5	1	Είδη γραφείου
πκ8: π3 με κ2	1003	Μαρκαδόρος	1.3	Π130	5	2	Είδη υπολογιστή
πκ9: π3 με κ3	1003	Μαρκαδόρος	1.3	Π130	5	3	Υπολογιστής
πκ10: π4 με κ1	1004	Δισκέτα 3,5"	0.8	Π120	2	1	Είδη γραφείου
πκ11: π4 με κ2	1004	Δισκέτα 3,5"	0.8	Π120	2	2	Είδη υπολογιστή
πκ12: π4 με κ3	1004	Δισκέτα 3,5"	0.8	Π120	2	3	Υπολογιστής

Καρτεσιανό γινόμενο στην SQL

Με τη βοήθεια της SQL μπορούμε να εμφανίσουμε το καρτεσιανό γινόμενο των πινάκων **ΠΡΟΪΟΝΤΑ**, **ΚΑΤΗΓΟΡΙΕΣ** με την εντολή:

```
SELECT ΠΡΟΪΟΝΤΑ.*, ΚΑΤΗΓΟΡΙΕΣ.*  
FROM ΠΡΟΪΟΝΤΑ, ΚΑΤΗΓΟΡΙΕΣ
```

Καρτεσιανό γινόμενο και μη αντιστοιχισμένες εγγραφές

- Η εγγραφή πκ2 συνδυάζει το προϊόν 1001 που είναι στην κατηγορία 1 με τα είδη υπολογιστή που έχουν κωδικό κατηγορίας 2. Επομένως για τη εγγραφή πκ2 δεν υπάρχει αντιστοιχία (μη αντιστοιχισμένη εγγραφή).
- Για τον ίδιο λόγο μη αντιστοιχισμένες είναι οι εγγραφές που εμφανίζονται με κόκκινα γράμματα.

εγγραφή	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία	Κωδικός Κατηγορίας	Ονομασία
πκ1: π1 με κ1	1001	Στυλό	0.75	Π200	1	1	Είδη γραφείου
πκ2: π1 με κ2	1001	Στυλό	0.75	Π200	1	2	Είδη υπολογιστή
πκ3: π1 με κ3	1001	Στυλό	0.75	Π200	1	3	Υπολογιστής
πκ4: π2 με κ1	1002	Μολύβι	1	Π250	1	1	Είδη γραφείου
πκ5: π2 με κ2	1002	Μολύβι	1	Π250	1	2	Είδη υπολογιστή
πκ6: π2 με κ3	1002	Μολύβι	1	Π250	1	3	Υπολογιστής
πκ7: π3 με κ1	1003	Μαρκαδόρος	1.3	Π130	5	1	Είδη γραφείου
πκ8: π3 με κ2	1003	Μαρκαδόρος	1.3	Π130	5	2	Είδη υπολογιστή
πκ9: π3 με κ3	1003	Μαρκαδόρος	1.3	Π130	5	3	Υπολογιστής
πκ10: π4 με κ1	1004	Δισκέτα 3,5"	0.8	Π120	2	1	Είδη γραφείου
πκ11: π4 με κ2	1004	Δισκέτα 3,5"	0.8	Π120	2	2	Είδη υπολογιστή
πκ12: π4 με κ3	1004	Δισκέτα 3,5"	0.8	Π120	2	3	Υπολογιστής

Αντιστοιχισμένες εγγραφές

- **Αντιστοιχισμένες εγγραφές** : αυτές που έχουν κάποιο κοινό πεδίο, βάση του οποίου μπορεί να γίνει αντιστοίχιση.
- Για να κρατήσουμε τις αντιστοιχισμένες εγγραφές του καρτεσιανού γινομένου χρησιμοποιούμε τον όρο **WHERE** και περιγράφουμε τη συνθήκη που θέλουμε να ικανοποιούν τα δεδομένα που εμφανίζονται (ΚΑΤΗΓΟΡΗΜΑ ΣΥΖΕΥΣΗΣ).
- Στο συγκεκριμένο παράδειγμα, οι γραμμές που θέλουμε να εμφανίζονται είναι αυτές στις οποίες κάθε προϊόν συνδυάζεται μόνο με την δική του κατηγορία:

```
SELECT ΠΡΟΪΟΝΤΑ.*, ΚΑΤΗΓΟΡΙΕΣ.*  
FROM ΠΡΟΪΟΝΤΑ, ΚΑΤΗΓΟΡΙΕΣ  
WHERE Κατηγορία = ΚωδικόςΚατηγορίας
```

Παράδειγμα αντιστοιχισμένων εγγραφών

Η εντολή:

```
SELECT ΠΡΟΪΟΝΤΑ.*, ΚΑΤΗΓΟΡΙΕΣ.*  
FROM ΠΡΟΪΟΝΤΑ, ΚΑΤΗΓΟΡΙΕΣ  
WHERE Κατηγορία = ΚωδικόςΚατηγορίας
```

Εμφανίζει τα παρακάτω δεδομένα:

εγγραφή	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία	Κωδικός Κατηγορίας	Ονομασία
πκ1: π1 με κ1	1001	Στυλό	0.75	Π200	1	1	Είδη γραφείου
πκ4: π2 με κ1	1002	Μολύβι	1	Π250	1	1	Είδη γραφείου
πκ11: π4 με κ2	1004	Δισκέτα 3,5"	0.8	Π120	2	2	Είδη υπολογιστή

- Στον προηγούμενο πίνακα η κατηγορία ενός προϊόντος εμφανίζεται δύο φορές: μία φορά στη στήλη Κατηγορία και μία φορά στη στήλη ΚωδικόςΚατηγορίας.
- Μπορούμε να εμφανίσουμε κάθε στήλη μία μόνο φορά χρησιμοποιώντας για παράδειγμα την παρακάτω εντολή:

```
SELECT ΠΡΟΪΟΝΤΑ.*, ΚΑΤΗΓΟΡΙΕΣ.Ονομασία  
FROM ΠΡΟΪΟΝΤΑ, ΚΑΤΗΓΟΡΙΕΣ  
WHERE Κατηγορία = ΚωδικόςΚατηγορίας
```

Πρακτική οδηγία

Όταν θέλουμε να συνδυάσουμε δύο ή περισσότερους πίνακες:

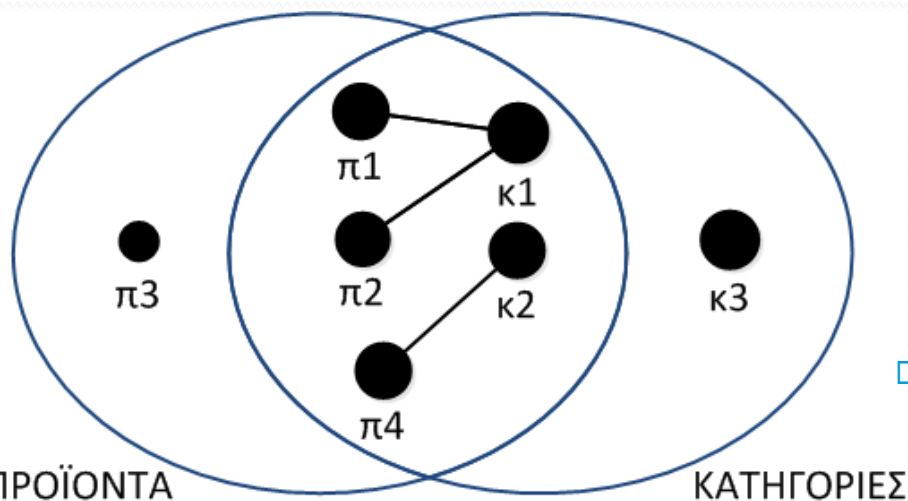
- στον όρο **SELECT** γράφουμε τα ονόματα των στηλών που θέλουμε να εμφανίζονται στο αποτέλεσμα
- στον όρο **FROM** αναφέρουμε τα ονόματα των πινάκων από τους οποίους θέλουμε να αντλήσουμε δεδομένα
- στον όρο **WHERE** περιγράφουμε με ποιόν τρόπο συνδέονται οι πίνακες από τους οποίους αντλούμε δεδομένα.

ΠΡΟΪΟΝΤΑ

Εγγραφή	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία
π1	1001	Στυλό	0.75	Π200	1
π2	1002	Μολύβι	1	Π250	1
π3	1003	Μαρκαδόρος	1.3	Π130	5
π4	1004	Δισκέτα 3,5"	0.8	Π120	2

ΚΑΤΗΓΟΡΙΕΣ

Εγγραφή	Κωδικός Κατηγορίας	Ονομασία
κ1	1	Είδη γραφείου
κ2	2	Είδη υπολογιστή
κ3	3	Υπολογιστής



- Στη γραφική παράσταση Venn, οι εγγραφές των δύο πινάκων που συσχετίζονται αναπαριστούνται ως στοιχεία τα οποία
 - ▣ συνδέονται άμεσα με μία γραμμή
 - ▣ τοποθετούνται κεντρικά ώστε να ανήκουν και στα δύο σύνολα
- Οι εγγραφές πίνακα που δεν συσχετίζονται με άλλη εγγραφή, αναπαριστούνται ως στοιχεία τα οποία τοποθετούνται περιφερειακά στο σύνολο

Ο όρος JOIN της SQL

- Η SQL προσφέρει την δεσμευμένη λέξη **JOIN** η οποία μπορεί να χρησιμοποιηθεί για να συνδυάσουμε δεδομένα δύο ή περισσότερων πινάκων.
- Στην απλούστερη μορφή του ο όρος **JOIN** χρησιμοποιείται σε μία εντολή **SELECT** για να βρούμε τις εγγραφές δύο ή περισσότερων πινάκων που σε μία στήλη τους έχουν την ίδια τιμή.

Εσωτερική σύζευξη

- Η εσωτερική σύζευξη (inner join) ανάμεσα σε δύο πίνακες που έχουν μία κοινή στήλη είναι η πράξη με την οποία εμφανίζονται οι εγγραφές που στην κοινή τους στήλη έχουν την ίδια τιμή.

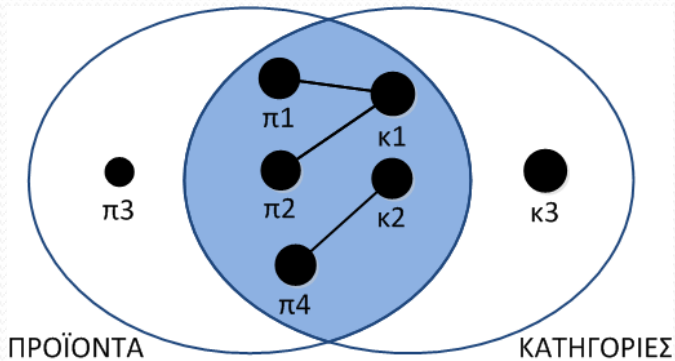
Παράδειγμα εσωτερικής σύζευξης πινάκων με τον όρο JOIN

ΠΡΟΪΟΝΤΑ

Εγγραφή	<u>Κωδικός Προϊόντος</u>	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία
π1	1001	Στυλό	0.75	Π200	1
π2	1002	Μολύβι	1	Π250	1
π3	1003	Μαρκαδόρος	1.3	Π130	5
π4	1004	Δισκέτα 3,5"	0.8	Π120	2

ΚΑΤΗΓΟΡΙΕΣ

Εγγραφή	<u>Κωδικός Κατηγορίας</u>	Όνομασία
κ1	1	Είδη γραφείου
κ2	2	Είδη υπολογιστή
κ3	3	Υπολογιστής



SELECT *
FROM ΠΡΟΪΟΝΤΑ JOIN ΚΑΤΗΓΟΡΙΕΣ
ON Κατηγορία =
ΚωδικόςΚατηγορίας

	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία	Κωδικός Κατηγορίας	Όνομασία
πκ1: π1 με κ1	1001	Στυλό	0.75	Π200	1	1	Είδη γραφείου
πκ4: π2 με κ1	1002	Μολύβι	1	Π250	1	1	Είδη γραφείου
πκ11: π4 με κ2	1004	Δισκέτα 3,5"	0.8	Π120	2	2	Είδη υπολογιστή

Αριστερή σύζευξη

- Η αριστερή σύζευξη (left join) ανάμεσα σε δύο πίνακες που έχουν μία κοινή στήλη είναι η πράξη με την οποία εμφανίζονται **όλες** τις εγγραφές του αριστερού (δηλαδή του πρώτου) πίνακα.
- Δηλαδή η αριστερή σύζευξη περιλαμβάνει:
 - τις εγγραφές των δύο πινάκων που στη κοινή στήλη τους έχουν την ίδια τιμή καθώς και
 - τις εγγραφές του αριστερού (δηλαδή του πρώτου στη σειρά) πίνακα τα οποία στην κοινή στήλη έχουν τιμή που δεν υπάρχει στον δεξιό (δεύτερο) πίνακα.

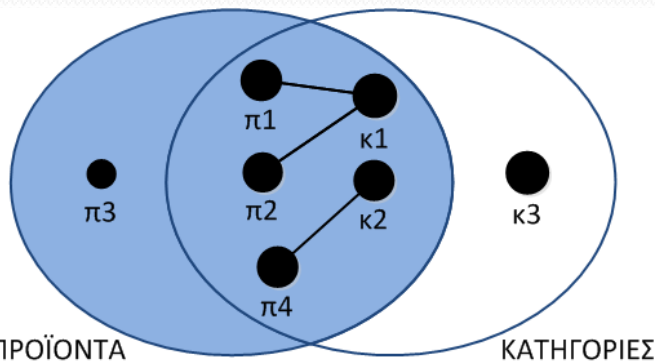
Παράδειγμα αριστερής σύζευξης πινάκων με τον όρο LEFT JOIN

ΠΡΟΪΟΝΤΑ

Εγγραφή	<u>Κωδικός Προϊόντος</u>	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία
π1	1001	Στυλό	0.75	Π200	1
π2	1002	Μολύβι	1	Π250	1
π3	1003	Μαρκαδόρος	1.3	Π130	5
π4	1004	Δισκέτα 3,5"	0.8	Π120	2

ΚΑΤΗΓΟΡΙΕΣ

Εγγραφή	<u>Κωδικός Κατηγορίας</u>	Ονομασία
κ1	1	Είδη γραφείου
κ2	2	Είδη υπολογιστή
κ3	3	Υπολογιστής



SELECT *
FROM ΠΡΟΪΟΝΤΑ LEFT JOIN
ΚΑΤΗΓΟΡΙΕΣ
ON Κατηγορία = ΚωδικόςΚατηγορίας

	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία	Κωδικός Κατηγορίας	Ονομασία
πκ1: π1 με κ1	1001	Στυλό	0.75	Π200	1	1	Είδη γραφείου
πκ4: π2 με κ1	1002	Μολύβι	1	Π250	1	1	Είδη γραφείου
πκ11: π4 με κ2	1004	Δισκέτα 3,5"	0.8	Π120	2	2	Είδη υπολογιστή
	1003	Μαρκαδόρος	1.3	Π130	5	NULL	NULL

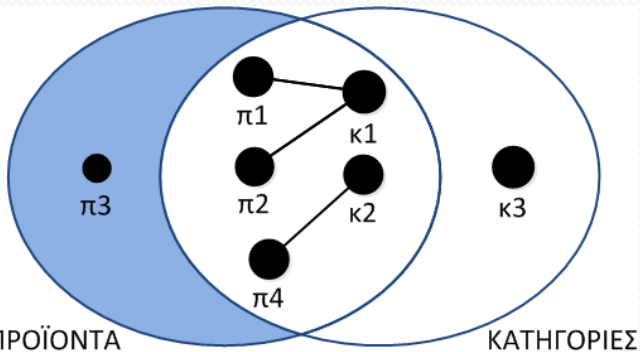
Παράδειγμα αριστερής σύζευξης πινάκων για μη σχετιζόμενες εγγραφές

ΠΡΟΪΟΝΤΑ

Εγγραφή	<u>Κωδικός Προϊόντος</u>	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία
π1	1001	Στυλό	0.75	Π200	1
π2	1002	Μολύβι	1	Π250	1
π3	1003	Μαρκαδόρος	1.3	Π130	5
π4	1004	Δισκέτα 3,5"	0.8	Π120	2

ΚΑΤΗΓΟΡΙΕΣ

Εγγραφή	<u>Κωδικός Κατηγορίας</u>	Όνομασία
κ1	1	Είδη γραφείου
κ2	2	Είδη υπολογιστή
κ3	3	Υπολογιστής



```
SELECT *
FROM ΠΡΟΪΟΝΤΑ LEFT JOIN
ΚΑΤΗΓΟΡΙΕΣ
ON Κατηγορία = ΚωδικόςΚατηγορίας
WHERE ΚωδικόςΚατηγορίας IS NULL
```

Συνδυασμός
εγγραφών
Κ3 με καμία

Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία	Κωδικός Κατηγορίας	Όνομασία
1003	Μαρκαδόρος	1.3	Π130	5	NULL	NULL

Δεξιά σύζευξη

- Η **δεξιά σύζευξη** (right join) ανάμεσα σε δύο πίνακες που έχουν μία κοινή στήλη είναι η πράξη με την οποία εμφανίζονται **όλες** τις εγγραφές του δεξιού (δηλαδή του δεύτερου) πίνακα.
- Η δεξιά σύζευξη περιλαμβάνει:
 - τις εγγραφές των δύο πινάκων που στη κοινή στήλη τους έχουν την ίδια τιμή καθώς και
 - τις εγγραφές του δεξιού (δηλαδή του δεύτερου στη σειρά) πίνακα οι οποίες στην κοινή στήλη έχουν τιμή που δεν υπάρχει στον αριστερό (πρώτο) πίνακα.

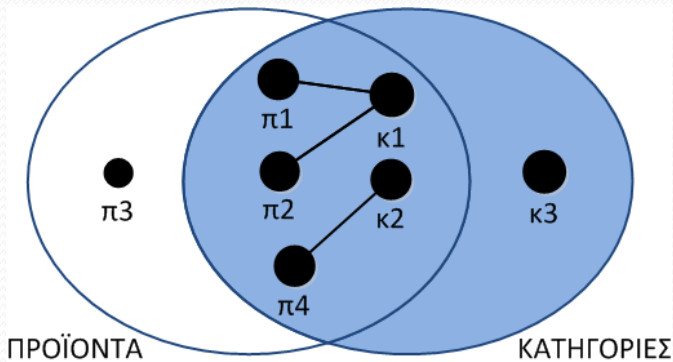
Παράδειγμα δεξιάς σύζευξης πινάκων με τον όρο RIGHT JOIN

ΠΡΟΪΟΝΤΑ

Εγγραφή	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία
π1	1001	Στυλό	0.75	Π200	1
π2	1002	Μολύβι	1	Π250	1
π3	1003	Μαρκαδόρος	1.3	Π130	5
π4	1004	Δισκέτα 3,5"	0.8	Π120	2

ΚΑΤΗΓΟΡΙΕΣ

Εγγραφή	Κωδικός Κατηγορίας	Ονομασία
κ1	1	Είδη γραφείου
κ2	2	Είδη υπολογιστή
κ3	3	Υπολογιστής



SELECT *
FROM ΠΡΟΪΟΝΤΑ RIGHT JOIN
ΚΑΤΗΓΟΡΙΕΣ
ON Κατηγορία =
ΚωδικόςΚατηγορίας

	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία	Κωδικός Κατηγορίας	Ονομασία
πκ1: π1 με κ1	1001	Στυλό	0.75	Π200	1	1	Είδη γραφείου
πκ4: π2 με κ1	1002	Μολύβι	1	Π250	1	1	Είδη γραφείου
πκ11: π4 με κ2	1004	Δισκέτα 3,5"	0.8	Π120	2	2	Είδη υπολογιστή
	NULL	NULL	NULL	NULL	NULL	3	Υπολογιστή

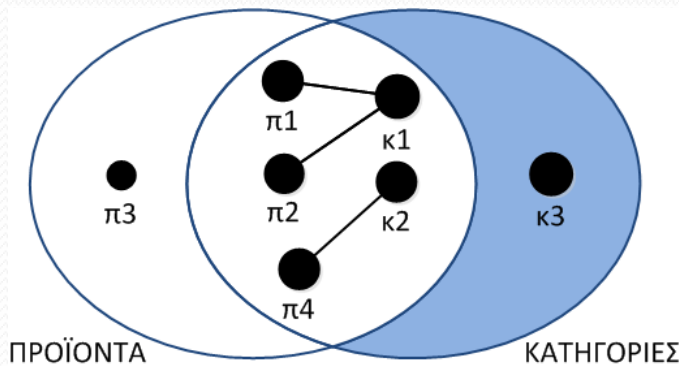
Παράδειγμα δεξιάς σύζευξης πινάκων για τις μη σχετιζόμενες εγγραφές

ΠΡΟΪΟΝΤΑ

Εγγραφή	<u>Κωδικός Προϊόντος</u>	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία
π1	1001	Στυλό	0.75	Π200	1
π2	1002	Μολύβι	1	Π250	1
π3	1003	Μαρκαδόρος	1.3	Π130	5
π4	1004	Δισκέτα 3,5"	0.8	Π120	2

ΚΑΤΗΓΟΡΙΕΣ

Εγγραφή	<u>Κωδικός Κατηγορίας</u>	Ονομασία
κ1	1	Είδη γραφείου
κ2	2	Είδη υπολογιστή
κ3	3	Υπολογιστής



```
SELECT *
FROM ΠΡΟΪΟΝΤΑ RIGHT JOIN
ΚΑΤΗΓΟΡΙΕΣ
ON Κατηγορία = ΚωδικόςΚατηγορίας
WHERE Κατηγορία IS NULL
```

Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία	Κωδικός Κατηγορίας	Ονομασία
NULL	NULL	NULL	NULL	NULL	3	Υπολογιστή

Πλήρης σύζευξη

Η πλήρης σύζευξη ανάμεσα σε δύο πίνακες που έχουν μία κοινή στήλη είναι η πράξη με την οποία εμφανίζονται όλα τα στοιχεία όλων των πινάκων:

- Εγγραφές που συνδυάζονται
- Εγγραφές του ενός πίνακα που δεν συνδέονται με καμία εγγραφή του άλλου. Τότε οι εγγραφές του άλλου πίνακα εμφανίζουν τιμές **NULL** στα πεδία που δεν υπάρχει αντιστοίχιση.

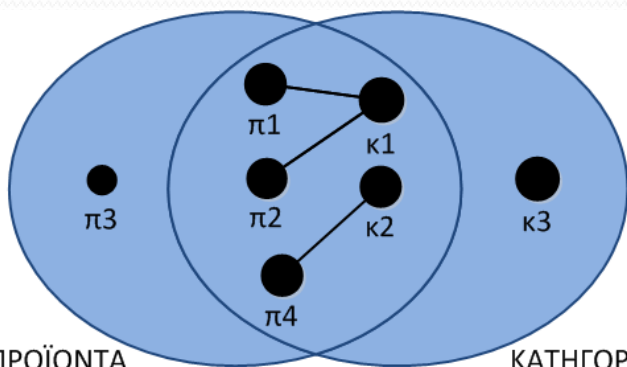
Παράδειγμα πλήρους σύζευξης πινάκων όταν υποστηρίζεται ο όρος FULL JOIN

ΠΡΟΪΟΝΤΑ

Εγγραφή	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία
π1	1001	Στυλό	0.75	Π200	1
π2	1002	Μολύβι	1	Π250	1
π3	1003	Μαρκαδόρος	1.3	Π130	5
π4	1004	Δισκέτα 3,5"	0.8	Π120	2

ΚΑΤΗΓΟΡΙΕΣ

Εγγραφή	Κωδικός Κατηγορίας	Όνομασία
κ1	1	Είδη γραφείου
κ2	2	Είδη υπολογιστή
κ3	3	Υπολογιστής



```
SELECT *
FROM ΠΡΟΪΟΝΤΑ FULL OUTER JOIN
ΚΑΤΗΓΟΡΙΕΣ
ON Κατηγορία = ΚωδικόςΚατηγορίας
```

πκ1: π1 με κ1
 πκ4: π2 με κ1
 πκ11: π4 με κ2

	Κωδικός Προϊόντος	Περιγραφή	Τιμή Αγοράς	Κωδικός Προμηθευτή	Κατηγορία	Κωδικός Κατηγορίας	Όνομασία
πκ1: π1 με κ1	1001	Στυλό	0.75	Π200	1	1	Είδη γραφείου
πκ4: π2 με κ1	1002	Μολύβι	1	Π250	1	1	Είδη γραφείου
πκ11: π4 με κ2	1004	Δισκέτα 3,5"	0.8	Π120	2	2	Είδη υπολογιστή
	1003	Μαρκαδόρος	1.3	Π130	5	NULL	NULL
	NULL	NULL	NULL	NULL	NULL	3	Υπολογιστή

Πλήρης σύζευξη στην SQL όταν δεν υποστηρίζεται ο όρος FULL JOIN

- Σε συστήματα που δεν υποστηρίζεται η δεσμευμένη λέξη **FULL** χρησιμοποιείται η εντολή **UNION** :

```
SELECT * FROM ΠΡΟΪΟΝΤΑ LEFT JOIN ΚΑΤΗΓΟΡΙΕΣ  
ON Κατηγορία = ΚωδικόςΚατηγορίας  
UNION
```

```
SELECT * FROM ΠΡΟΪΟΝΤΑ RIGHT JOIN ΚΑΤΗΓΟΡΙΕΣ  
ON Κατηγορία = ΚωδικόςΚατηγορίας
```

Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

