



# Βάσεις Δεδομένων

## Ενότητα 8: Η Γλώσσα SQL

Ιωάννης Μανωλόπουλος, Καθηγητής  
Τμήμα Πληροφορικής ΑΠΘ



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*  
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» 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.





# Η Γλώσσα SQL

Τύποι δεδομένων, ορισμός και χειρισμός  
δεδομένων, όψεις

# Περιεχόμενα ενότητας

---

1. Εισαγωγή – Βασικές Έννοιες
2. Τύποι Δεδομένων
3. Ορισμός Δεδομένων
4. Χειρισμός Δεδομένων
5. Διαχείριση Όψεων



# Σκοποί ενότητας

- Η εισαγωγή της γλώσσας SQL.
- Η περιγραφή των τύπων δεδομένων της SQL.
- Ο ορισμός του σχήματος βάσης, πίνακα και πεδίου ορισμού.
- Ο ορισμός της διατύπωσης ερωτημάτων.
- Ο ορισμός της εισαγωγής, ενημέρωσης και διαγραφής δεδομένων.
- Ο ορισμός και διαχείριση όψεων.



# Εισαγωγή (1/6)

- Η δομημένη γλώσσα ερωτημάτων **SQL (Structured Query Language)** χρησιμοποιείται από τους χρήστες του ΣΔΒΔ με σκοπό τη δημιουργία, ενημέρωση ή διαγραφή δεδομένων και τη διατύπωση ερωτημάτων.
- Η αρχική έκδοση της γλώσσας ονομάστηκε **SEQUEL** (Structured English Query Language) και σχεδιάστηκε από ερευνητική ομάδα της IBM στα πλαίσια του System R (1974).
- Η εταιρεία Oracle χρησιμοποίησε για πρώτη φορά τη γλώσσα σε εμπορικό ΣΔΒΔ το 1979.



# Εισαγωγή (2/6)

- Πρότυπα:
  - SQL-86: ANSI standard
  - SQL-89: προσθήκη περιορισμών ακεραιότητας
  - SQL-92 ή SQL2: ISO standard
  - SQL-1999 ή SQL3: προσθήκη triggers, transitive closure, embedded SQL in Java
  - SQL-2003: προσθήκη XML
  - SQL-2006: προσθήκη XQuery
  - SQL-2008:
  - SQL-2011:





# Εισαγωγή (3/6)

- Ο αναγνώστης παροτρύνεται να εξασκηθεί στον επόμενο ιστοχώρο

<http://w3schools.com/sql/>

The screenshot shows the w3schools.com website. The browser address bar contains the URL [www.w3schools.com/sql/default.asp](http://www.w3schools.com/sql/default.asp). The page title is "SQL Tutorial". The left sidebar lists various SQL topics, with "SQL Tutorial" highlighted. The main content area includes a "SQL Tutorial" section with a "Next Chapter" link, an "Examples in Each Chapter" section with a "Try it yourself" button, and a "SQL Quiz Test" section. The page also features a search bar, a Google logo, and various navigation links.



# Εισαγωγή (4/6)

- Η γλώσσα SQL έχει σημαντικά **πλεονεκτήματα** με βασικότερα τα εξής:
  - υποστηρίζεται από κάθε σχεσιακό σύστημα, οπότε η διατύπωση των ερωτημάτων είναι ανεξάρτητη του ΣΔΒΔ
  - χρησιμοποιείται για όλες τις λειτουργίες του ΣΔΒΔ
  - πολλές γλώσσες προγραμματισμού έχουν επεκταθεί ώστε να υποστηρίζουν διατύπωση ερωτημάτων σε SQL
  - έχει απλή σύνταξη και αποδεσμεύει το χρήστη από λεπτομέρειες υλοποίησης



# Εισαγωγή (5/6)

- Η SQL στηρίζεται στο σχεσιακό λογισμό πλειάδων και τη σχεσιακή άλγεβρα.
- Είναι δηλωτική γλώσσα υψηλού επιπέδου, δηλαδή ορίζουμε τί θέλουμε να γίνει όχι το πώς.
- Η επεξεργασία γίνεται σε επίπεδο συνόλου. Υπάρχουν μηχανισμοί για επεξεργασία σε επίπεδο μεμονωμένων εγγραφών για υποστήριξη διαδικαστικών γλωσσών.
- Η SQL χρησιμοποιεί τους όρους: **πίνακας, γραμμή, στήλη**, που αντιστοιχούν στις έννοιες: **σχέση, πλειάδα, χαρακτηριστικό**.



# Εισαγωγή (6/6)

- Η SQL έχει διάφορα τμήματα:
  - Γλώσσα Ορισμού Δεδομένων (data definition language)
  - Γλώσσα Χειρισμού Δεδομένων (data manipulation language)
  - Ενσωματωμένη Γλώσσα Χειρισμού Δεδομένων (π.χ. C/C++, Java, Cobol, Pascal)
  - Ορισμό Όψεων (view definition)
  - Εξουσιοδότηση (authentication)
  - Ακεραιότητα (integrity)
  - Έλεγχο Συναλλαγών (concurrency control)



# Τύποι Δεδομένων (1/5)

Τύποι Δεδομένων	Δηλώσεις
λογικός	BOOLEAN
χαρακτήρας	CHAR, VARCHAR
δυναμικό ψηφίο	BIT, BIT VARYING
ακριβής αριθμητικός	NUMERIC, DECIMAL, INTEGER, SMALLINT
προσεγγιστικός αριθμητικός	FLOAT, REAL, DOUBLE, PRECISION
ημερομηνία και ώρα	DATE, TIME, TIMESTAMP
διάστημα	INTERVAL

- Είναι ο τύπος δεδομένων κατάλληλος για το είδος δεδομένων που πρέπει να αποθηκεύσουμε;
- Πρέπει να χρησιμοποιήσουμε έναν τύπο δεδομένων καθορισμένου ή μεταβλητού μήκους;



# Τύποι Δεδομένων (2/5)

- **CHARACTER(N)** ή **CHAR(N)** – συμβολοσειρά σταθερού μήκους N χαρακτήρων
- **CHARACTER VARYING (N)** ή **VARCHAR(N)** – συμβολοσειρά μεταβλητού μήκους από 0 έως N χαρακτήρες
- **BIT(N)**, **BIT VARYING(N)** – ίδια λειτουργία με CHAR(N) και VARCHAR(N)
- **INTEGER**, **SMALLINT**, **BIGINT** – για μεγάλους και μικρούς ακέραιους (εξαρτώνται από το υλικό)
- **NUMERIC(N,M)** ή **DECIMAL(N,M)** – αριθμός με N συνολικά ψηφία από τα οποία τα M (ίσως και 0) είναι δεκαδικά (**ακρίβεια και κλίμακα**)



# Τύποι Δεδομένων (3/5)

- **FLOAT(N), REAL, DOUBLE** – για δεκαδικούς κινητής υποδιαστολής. N είναι η ακρίβεια του δεκαδικού μέρους (mantissa). Εξαρτώνται από το υλικό.
- **DATE, TIME, TIMESTAMP, INTERVAL** – χρησιμοποιούνται για τον ορισμό σημείων στο χρόνο με συγκεκριμένη ακρίβεια. Υποδιαιρούνται σε απλούστερα πεδία που είναι: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE\_HOUR, TIMEZONE\_MINUTE



# Τύποι Δεδομένων (4/5)

- **DATE** έχει τη μορφή: YYYY-MM-DD.
- **TIME** έχει τη μορφή: HH-MM-SS.
- **TIME(I)**: με I δεκαδικά για κλάσματα δευτερολέπτου  
HH-MM-SS-F....F.
- **TIME WITH TIME ZONE**: με χρονική ζώνη  
HH-MM-SS-HH-MM.
- **TIME-STAMP** έχει τη μορφή:  
YYYY-MM-DD-HH-MM-SS-F...F{-HH-MM}.
- **INTERVAL**: χρησιμοποιείται για την αναπαράσταση  
χρονικών διαστημάτων με ανάλογη μορφή και ακρίβεια.





# Τύποι Δεδομένων (5/5)

- Υπάρχουν και οι **user-defined** τύποι δεδομένων.

```
CREATE DOMAIN COLOR AS CHAR(5)
```

```
CONSTRAINT VALID COLORS
```

```
DEFAULT 'GREEN'
```

```
CHECK (VALUE IN 'Red', 'Green', 'Blue', '???' )
```

```
DROP DOMAIN COLOR RESTRICT;
```

```
DROP DOMAIN COLOR CASCADE;
```

- Με το `restrict` αποτυγχάνει το `drop` αν το πεδίο χρησιμοποιείται σε ορισμό στηλών.
- Με το `cascade` το `drop` προκαλεί οι στήλες να ορισθούν απευθείας στο αντίστοιχο πεδίο ορισμού.



# Γλώσσα Ορισμού Δεδομένων (1/11)

- Δημιουργία νέου σχήματος:

**CREATE SCHEMA** νέο-σχήμα  
**AUTHORIZATION** όνομα-χρήστη



# Γλώσσα Ορισμού Δεδομένων (2/11)

- Δημιουργία νέου πίνακα μέσω της εντολής `create table` ορίζουμε:
  - το όνομα του πίνακα,
  - τα ονόματα και τους τύπους δεδομένων των χαρακτηριστικών,
  - αν τα χαρακτηριστικά μπορούν να δέχονται ή όχι τιμές NULL,
  - εξ ορισμού τιμές για τα χαρακτηριστικά,
  - πρωτεύοντα κλειδιά και περιορισμούς μοναδικότητας,
  - περιορισμούς τιμών επάνω σε πεδία, και
  - περιορισμούς αναφορικής ακεραιότητας.



# Γλώσσα Ορισμού Δεδομένων (3/11)

- Δημιουργία νέου πίνακα. Σύνταξη SQL:

**CREATE TABLE** νέος-πίνακας  
(στήλη1 τύπος-δεδομένων,  
στήλη2 τύπος-δεδομένων,  
...  
στήληN τύπος-δεδομένων,  
<περιορισμός-ακεραιότητας-1>,  
<περιορισμός-ακεραιότητας-2>,  
...  
<περιορισμός-ακεραιότητας-M>);



# Γλώσσα Ορισμού Δεδομένων (4/11)

- Δημιουργία νέου πίνακα:

```
CREATE TABLE Γνωστική_Περιοχή  
(κωδικός INTEGER NOT NULL,  
τίτλος CHAR(50) NOT NULL,  
αριθμός_συνδρομητών INTEGER DEFAULT 0,  
PRIMARY KEY (κωδικός),  
UNIQUE (τίτλος),  
CHECK (αριθμός_συνδρομητών  $\geq 0$ ));
```



# Γλώσσα Ορισμού Δεδομένων (5/11)

- Δημιουργία νέου πίνακα:

```
CREATE TABLE Πρακτικά_Συνεδρίου  
(κωδικός INTEGER NOT NULL,  
συνέδριο VARCHAR(100) NOT NULL,  
ημερομηνία DATE NOT NULL,  
χώρα CHAR(20),  
κωδικός_εκδοτικού_οίκου INTEGER NOT NULL,  
PRIMARY KEY (κωδικός),  
FOREIGN KEY (κωδικός_εκδοτικού_οίκου)  
REFERENCES Εκδοτικός_Οίκος (κωδικός),  
ON DELETE RESTRICTED,  
ON UPDATE CASCADE);
```



# Γλώσσα Ορισμού Δεδομένων (6/11)

- Δημιουργία νέου πίνακα:
  - αν παραλειφθεί ο προσδιορισμός **NOT NULL**, τότε το χαρακτηριστικό μπορεί να δεχθεί τιμές **NULL**,
  - η πρόταση **DEFAULT** παρατίθεται για να δίνονται εξ ορισμού τιμές κατά την εισαγωγή στοιχείων, όταν οι τιμές των χαρακτηριστικών δεν ορίζονται ρητώς,
  - αν δεν είναι παρούσα πρόταση **DEFAULT** σε ένα χαρακτηριστικό και δεν δοθεί τιμή στην εισαγωγή γι' αυτό, τότε χρησιμοποιείται η τιμή **NULL**,
  - όταν το πρωτεύον κλειδί είναι ένα χαρακτηριστικό μόνον, μπορεί να ορισθεί είτε γράφοντας **PRIMARY KEY** δίπλα από το χαρακτηριστικό είτε παραθέτοντας την πρόταση **PRIMARY KEY** μετά το τέλος των πεδίων,



# Γλώσσα Ορισμού Δεδομένων (7/11)

- Δημιουργία νέου πίνακα:
  - ένας πίνακας μπορεί να έχει μόνον ένα πρωτεύον κλειδί, αλλά επίσης να έχει περιορισμούς μοναδικότητας πάνω σε άλλα πεδία. Αν το πεδίο είναι ένα, δηλώνεται με τη λέξη **UNIQUE** δίπλα από το όνομα του,
  - αν τα μοναδικά χαρακτηριστικά είναι δύο ή περισσότερα, δηλώνονται με ξεχωριστή πρόταση **UNIQUE** μετά τους ορισμούς των πεδίων,
  - αν τα ονόματα των στηλών στους δύο πίνακες ταυτίζονται, μπορεί στον πίνακα που ορίζεται το ξένο κλειδί να παραληφθεί το όνομα της στήλης:

```
department_id CHAR(5) NOT NULL  
REFERENCES department
```





# Γλώσσα Ορισμού Δεδομένων (8/11)

- Δημιουργία νέου πίνακα:
  - η στήλη προς την οποία αναφερόμαστε πρέπει να είναι μοναδική (**UNIQUE**) ή πρωτεύον κλειδί στον πίνακα όπου βρίσκεται,
  - μπορούμε να ορίσουμε ότι η διαγραφή στον αναφερόμενο πίνακα, προκαλεί διαγραφή στον πίνακα όπου ορίζεται ο περιορισμός:

```
department_id CHAR(5) NOT NULL  
REFERENCES department  
ON DELETE CASCADE
```
  - αν έχουμε δύο ή περισσότερες στήλες στο ξένο κλειδί πρέπει να χρησιμοποιήσουμε την πρόταση **FOREIGN KEY**.



# Γλώσσα Ορισμού Δεδομένων (9/11)

- Η αλλαγή δομής πίνακα (alter) χρησιμοποιείται για:
  - προσθήκη νέας στήλης στον πίνακα,
  - διαγραφή υπάρχουσας στήλης,
  - εισαγωγή νέου περιορισμού (constraint),
  - κατάργηση περιορισμού,
  - αλλαγή της εξ ορισμού (default) τιμής στήλης, και
  - κατάργηση εξ ορισμού τιμής στήλης.



# Γλώσσα Ορισμού Δεδομένων (10/11)

- Αλλαγή τύπου, δυνατότητας χρήσης τιμών NULL ή εξ ορισμού τιμής στήλης:

```
ALTER TABLE Πρακτικά_Συνεδρίου  
MODIFY χώρα CHAR(15);
```

- Προσθήκη ή κατάργηση περιορισμών ακεραιότητας:

```
ALTER TABLE Πρακτικά_Συνεδρίου  
ADD CONSTRAINT ημερομηνία  
CHECK (ημερομηνία > 2000-1-1);
```

```
ALTER TABLE Πρακτικά_Συνεδρίου  
DROP CONSTRAINT ημερομηνία  
CHECK (ημερομηνία >2000-1-1);
```



# Γλώσσα Ορισμού Δεδομένων (11/11)

- Καταστροφή πινάκων (DROP TABLE):

**DROP TABLE** Γνωστική\_Περιοχή;

- Ο πίνακας δεν μπορεί να καταστραφεί αν υπάρχουν όψεις ή περιορισμοί ακεραιότητας που τον αναφέρουν:

**DROP TABLE** Πρακτικά\_Συνεδρίου **RESTRICT**;

**DROP TABLE** Πρακτικά\_Συνεδρίου **CASCADE**;



# Περιορισμοί Ακεραιότητας

- Για την επιβολή ακεραιότητας σε επίπεδο στηλών χρησιμοποιούνται οι εξής τεχνικές:
  - Κενά (nulls)
  - Μοναδικές τιμές (unique column values)
  - Τιμές κύριου κλειδιού (primary key values)
  - Αναφορική ακεραιότητα (referential integrity)
  - Σύνθετοι έλεγχοι ακεραιότητας (complex integrity checking)
  - Τιμές εξ ορισμού (default values)



# Πλεονεκτήματα από τη χρήση περιορισμών ακεραιότητας (1/2)

- Εύκολος ορισμός τους χωρίς επιπλέον δηλώσεις.
- Κεντρική διαχείριση καθώς ορίζονται σε πίνακες και αποθηκεύονται στο λεξικό χωρίς να επαναλαμβάνονται στις εφαρμογές.
- Παραγωγικότητα ανάπτυξης εφαρμογών καθώς εύκολα ενημερώνονται οι περιορισμοί ακεραιότητας σε περίπτωση αλλαγής των επιχειρησιακών κανόνων. Αν αλλάξει η φιλοσοφία κάποιου περιορισμού ακεραιότητας, δεν αλλάζουν οι εφαρμογές που χρησιμοποιούν τα συγκεκριμένα δεδομένα.



# Πλεονεκτήματα από τη χρήση περιορισμών ακεραιότητας (2/2)

- Άμεση ανάδραση στο χρήστη με βάση τα μηνύματα του ΣΔΒΔ.
- Δυνατότητα εύκολης ακύρωσης των περιορισμών ακεραιότητας ώστε να διευκολυνθεί η φόρτωση μεγάλων ποσοτήτων δεδομένων. Μπορούν να εφαρμοσθούν οι περιορισμοί και να παραχθούν αυτόματα αναφορές λαθών για τις γραμμές με πρόβλημα.
- Μειονέκτημα: επιβάρυνση της απόδοσης του συστήματος.



# Περιορισμοί Ακεραιότητας – Κενά

- Εφαρμόζεται σε μία συγκεκριμένη στήλη και επιτρέπει ή αποτρέπει εισαγωγές ή ενημερώσεις γραμμών που περιέχουν null σε αυτή τη στήλη.

**NOT NULL**





# Περιορισμοί Ακεραιότητας – Μοναδικές τιμές

- Ο κανόνας της μοναδικής τιμής ορίζεται σε μια στήλη ή σύνολο στηλών και επιτρέπει την εισαγωγή ή ενημέρωση μίας γραμμής μόνον αν περιέχει μία μοναδική τιμή στη συγκεκριμένη ή συγκεκριμένες στήλες.
- Μπορεί να χρησιμοποιηθεί πολλές φορές σε μία εντολή `create table`.

**UNIQUE** <attribute(s)>



# Περιορισμοί Ακεραιότητας – Τιμές πρωτεύοντος κλειδιού

- Η τιμή που ορίζεται σε ένα πρωτεύον κλειδί (απλό ή σύνθετο) πρέπει να είναι μοναδική ώστε να προσδιορίζει μοναδικά κάθε γραμμή του πίνακα.
- Χρησιμοποιείται μία μόνο φορά σε μία εντολή `create table`.

**PRIMARY KEY** (<attribute(s)>)



# Περιορισμοί Ακεραιότητας – Αναφορική ακεραιότητα

- Ορίζεται σε ένα κλειδί (απλό ή σύνθετο) και εξασφαλίζει ότι οι τιμές του ταιριάζουν με τις τιμές σε έναν συσχετιζόμενο πίνακα (referenced value).

**FOREIGN KEY** <attribute> **REFERENCES**  
<table (attribute)>



# Περιορισμοί Ακεραιότητας – Σύνθετος έλεγχος ακεραιότητας

- Ορίζεται για μία ή περισσότερες στήλες και επιτρέπει ή αποτρέπει την εισαγωγή, ενημέρωση ή διαγραφή μίας γραμμής με βάση την τιμή (ή τιμές) της στήλης (ή των στηλών).

**CHECK** <conditional expression>

- Π.χ. YEAR INT (4) **NOT NULL, DEFAULT 2003, CHECK (YEAR BETWEEN 1960 AND 2003)**



# Περιορισμοί Ακεραιότητας – Απλοί κανόνες επιλογής κύριου κλειδιού

- Συνήθως από τη φύση της εφαρμογής το κλειδί είναι γνωστό (ταυτότητα, διαβατήριο, φορολογικό μητρώο, ΑΜΚΑ, ΑΕΜ κλπ).
- Αν δεν συμβαίνει κάτι τέτοιο τότε:
  - Χρησιμοποιούμε μία στήλη με σειριακούς αριθμούς. Ο κανόνας αυτός ικανοποιεί όλους τους επόμενους.
  - Χρησιμοποιούμε ένα πεδίο με τιμές μοναδικές.
  - Χρησιμοποιούμε ένα πεδίο με μικρές αριθμητικές τιμές.
  - Χρησιμοποιούμε ένα πεδίο με τιμές χωρίς κενά.
  - Χρησιμοποιούμε ένα πεδίο με τιμές που δεν αλλάζουν.
  - Αποφεύγουμε τη χρήση σύνθετων κλειδιών γιατί μπορεί να παραβιασθεί κάποιος κανόνας.



# Περιορισμοί Ακεραιότητας – Περιορισμοί ξένου κλειδιού (1/4)

- Ένα κλειδί αναφοράς μπορεί να είναι σύνθετο. Στην περίπτωση αυτή πρέπει να αναφέρεται σε ένα κλειδί που επίσης είναι σύνθετο με αντίστοιχες στήλες ως προς τον αριθμό και τον τύπο των δεδομένων.
- Επιτρέπεται στο κλειδί αναφοράς να υπάρχουν τιμές null, ακόμη και αν δεν υπάρχουν στην αναφορά του primary ή unique key.



# Περιορισμοί Ακεραιότητας – Περιορισμοί ξένου κλειδιού (2/4)

- Μπορούμε να εισάγουμε τρόπους αντίδρασης σε περίπτωση αλλαγής/διαγραφής του κλειδιού αναφοράς στον parent table. Π.χ. να συνδυάσουμε τις εντολές ON DELETE ή ON UPDATE για τον κύριο πίνακα με τις εντολές CASCADE, SET NULL or SET DEFAULT:
  - **ON DELETE CASCADE:** Σε περίπτωση διαγραφής του κλειδιού στον parent table διαγράφονται επίσης οι εγγραφές στον child table.
  - **ON DELETE RESTRICT:** Επιτρέπεται η διαγραφή με βάση τιμές της στήλης του parent table μόνο αν δεν υπάρχουν αντίστοιχες τιμές στον child table.



# Περιορισμοί Ακεραιότητας – Περιορισμοί ξένου κλειδιού (3/4)

- **ON DELETE SET NULL:** Σε περίπτωση διαγραφής του κλειδιού στον parent table, οι τιμές στον child table που αναφέρονται σ' αυτόν παίρνουν την τιμή null.
- **ON DELETE SET DEFAULT:** Σε περίπτωση διαγραφής του κλειδιού στον parent table, οι αντίστοιχες τιμές στον child παίρνουν την default τιμή του. Πρέπει να έχει ορισθεί όμως μία default τιμή, αλλιώς γίνεται null.
- **ON UPDATE CASCADE:** Σε περίπτωση αλλαγής του κλειδιού στον parent table, αλλάζει επίσης και η αντίστοιχη αναφορά του στον child table.





# Περιορισμοί Ακεραιότητας – Περιορισμοί ξένου κλειδιού (4/4)

- **ON UPDATE RESTRICT:** Σε περίπτωση αλλαγής του κλειδιού στον parent table, η αλλαγή επιτρέπεται μόνο αν δεν υπάρχουν αναφορές του στον child table.
- **ON UPDATE SET NULL:** Όταν αλλάξει η τιμή στον parent table, η αντίστοιχη αναφορά στον child table γίνεται null.
- **ON UPDATE SET DEFAULT:** Όταν αλλάξει η τιμή στον parent table, η αντίστοιχη αναφορά του στον child παίρνει την default τιμή. Πρέπει να έχει ορισθεί όμως μια default τιμή, διαφορετικά γίνεται null.



# Γλώσσα Χειρισμού Δεδομένων (1/5)

- Εντολές:
  - **SELECT** για διατύπωση ερωτημάτων,
  - **INSERT** για εισαγωγή νέων δεδομένων,
  - **UPDATE** για ενημέρωση δεδομένων, και
  - **DELETE** για διαγραφή δεδομένων από τη ΒΔ.



# Γλώσσα Χειρισμού Δεδομένων (2/5)

- Σχήμα Βάσης με συνδρομητικά δεδομένα:

*Συγγραφέας(κωδικός, όνομα, οδός, αριθμός, ΤΚ, πόλη, χώρα, σύνολο-άρθρων)*

*Άρθρο(κωδικός, τίτλος, PDF, κωδικός-γνωστικής-περιοχής, κωδικός-συνεδρίου, κωδικός-περιοδικού, αρχική-σελίδα-πρακτικών, τελική-σελίδα-πρακτικών, τεύχος, τόμος, αρχική-σελίδα-περιοδικού, τελική-σελίδα-περιοδικού)*

*Συγγραφή-Άρθρου(κωδικός-συγγραφέα, κωδικός-άρθρου)*

*Συνδρομητής(κωδικός, όνομα, οδός, αριθμός, ΤΚ, πόλη, χώρα, τηλέφωνο)*

*Εκδοτικός-Οίκος(κωδικός, όνομα, οδός, αριθμός, ΤΚ, πόλη, χώρα, τηλέφωνο)*

*Περιοδικό(κωδικός, τίτλος, κωδικός-εκδοτικού-οίκου)*

*Πρακτικά-Συνεδρίου(κωδικός, συνέδριο, πόλη, χώρα, ημερομηνία, κωδικός-εκδοτικού-οίκου)*



# Γλώσσα Χειρισμού Δεδομένων (3/5)

```
SELECT A1, A2, ..., An  
FROM R1, R2, ..., Rm  
WHERE P;
```

- Η πρόταση **SELECT** προσδιορίζει τις στήλες του αποτελέσματος.
- Η πρόταση **FROM** προσδιορίζει τους πίνακες από όπου θα απαντηθεί το ερώτημα.
- Η πρόταση **P** του **WHERE** προσδιορίζει τη συνθήκη που πρέπει να ισχύει ως προς τα περιεχόμενα των στηλών  $A_i$  των πινάκων  $R_j$  της πρότασης **FROM**.



# Γλώσσα Χειρισμού Δεδομένων (4/5)

```
SELECT A1, A2, ..., An  
FROM R1, R2, ... Rm  
WHERE P;
```

- Η πρόταση P μπορεί να είναι μία σύνθετη συνθήκη με:
  - λογικούς τελεστές: and, or, not,
  - τελεστές σύγκρισης: <, <=, >, >=, =, <>, between, not between ανάμεσα σε αριθμητικές εκφράσεις, συμβολοσειρές και ειδικούς τύπους.
  - Προσοχή: X is NULL (σωστό), X=NULL (λάθος).
- Η ισοδύναμη έκφραση σε σχεσιακή άλγεβρα είναι:
  - $\Pi_{A_1, A_2, \dots, A_n}(\sigma_P(R_1 \times R_2 \times \dots \times R_n))$ .



# Γλώσσα Χειρισμού Δεδομένων (5/5)

- Πλήρης σύνταξη SELECT:

**SELECT** [**DISTINCT**] <λίστα στηλών>

**FROM** <λίστα πινάκων>

**WHERE** συνθήκη]

**GROUP BY** <λίστα στηλών>]

**HAVING** συνθήκη]

**ORDER BY** <λίστα στηλών> [**ASCENDING** | **DESCENDING**] ] ;



# Γλώσσα Χειρισμού Δεδομένων – Απλά ερωτήματα (1/4)

- «Να δοθούν όλες οι γραμμές του πίνακα Γνωστική\_Περιοχή».

```
SELECT *  
FROM Γνωστική_Περιοχή;  
ή  
SELECT κωδικός, τίτλος,  
αριθμός_συνδρομητών  
FROM Γνωστική_Περιοχή;
```



# Γλώσσα Χειρισμού Δεδομένων – Απλά ερωτήματα (2/4)

- «Να βρεθούν όλοι οι τίτλοι των γνωστικών περιοχών που είναι αποθηκευμένοι στη ΒΔ».

**SELECT** τίτλος

**FROM** Γνωστική Περιοχή;

- «Να βρεθούν τα διαφορετικά ονόματα των συνδρομητών».

**SELECT DISTINCT** όνομα

**FROM** Συνδρομητής;





# Γλώσσα Χειρισμού Δεδομένων – Απλά ερωτήματα (3/4)

- «Να βρεθούν οι κωδικοί, τα ονόματα και οι αριθμοί πιστωτικών καρτών των συνδρομητών που βρίσκονται στην Ελλάδα, και τα αποτελέσματα να ταξινομηθούν αλφαβητικά ως προς το όνομα, και στη συνέχεια ως προς τον αριθμό πιστωτικής κάρτας».

```
SELECT κωδικός, όνομα, ΑΠΚ  
FROM Συνδρομητής  
WHERE χώρα = 'Ελλάδα'  
ORDER BY όνομα, ΑΠΚ;
```



# Γλώσσα Χειρισμού Δεδομένων – Απλά ερωτήματα (4/4)

- «Να βρεθούν οι κωδικοί και οι τίτλοι των άρθρων που έχουν δημοσιευθεί και σε περιοδικό και σε πρακτικά συνεδρίου».

**SELECT** κωδικός, τίτλος

**FROM** Άρθρο

**WHERE** κωδικός\_περιοδικού **IS NOT NULL**  
**AND** κωδικός\_πρακτικών\_συνεδρίου **IS**  
**NOT NULL;**



# Γλώσσα Χειρισμού Δεδομένων – Ερωτήματα Σύνδεσης (1/4)

- «Να βρεθούν οι κωδικοί και τα ονόματα των συνδρομητών και οι τίτλοι των γνωστικών περιοχών για τις οποίες έχουν πληρώσει συνδρομή».

```
SELECT Συνδρομητής.κωδικός, Συνδρομητής.όνομα,  
Γνωστική_Περιοχή.τίτλος  
FROM Συνδρομητής, Γνωστική_Περιοχή, Συνδρομή  
WHERE Συνδρομή.κωδικός_συνδρομητή =  
Συνδρομητής.κωδικός AND  
Γνωστική_Περιοχή.κωδικός=  
Συνδρομή.κωδικός_γνωστικής_περιοχής;
```



# Γλώσσα Χειρισμού Δεδομένων – Ερωτήματα Σύνδεσης (2/4)

- «Να βρεθούν οι τίτλοι των άρθρων και τα ονόματα των συγγραφέων που τα έχουν συγγράψει. Το αποτέλεσμα να ταξινομηθεί αλφαβητικά ως προς τους τίτλους των άρθρων».

```
SELECT Άρθρο.τίτλος, Συγγραφέας.όνομα  
FROM Άρθρο, Συγγραφέας, Συγγραφή_Άρθρου  
WHERE Συγγραφή_Άρθρου.κωδικός_άρθρου =  
Άρθρο.κωδικός AND Συγγραφέας.κωδικός =  
Συγγραφή_Άρθρου.κωδικός_συγγραφέα  
ORDER BY Άρθρο.τίτλος;
```



# Γλώσσα Χειρισμού Δεδομένων – Ερωτήματα Σύνδεσης (3/4)

- «Να προσδιορισθεί ο τίτλος κάθε άρθρου και ο τίτλος της γνωστικής περιοχής όπου ανήκει».

```
SELECT Άρθρο.τίτλος, Γνωστική Περιοχή.τίτλος  
FROM Άρθρο, Γνωστική Περιοχή  
WHERE Άρθρο.κωδικός_γνωστικής_περιοχής=  
Γνωστική_Περιοχή.κωδικός;
```

ή

```
SELECT Άρθρο.τίτλος, Γνωστική Περιοχή.τίτλος  
FROM Άρθρο JOIN Γνωστική Περιοχή ON  
Γνωστική_Περιοχή.κωδικός=  
Άρθρο.κωδικός_γνωστικής_περιοχής;
```



# Γλώσσα Χειρισμού Δεδομένων – Ερωτήματα Σύνδεσης (4/4)

- Η εντολή JOIN ON επιστρέφει τη φυσική σύνδεση δύο πινάκων. Ακόμη, η SQL προσφέρει τις εξής τρεις εντολές για την εξωτερική σύνδεση δύο πινάκων:
  - **R1 LEFT JOIN R2 ON R1.a = R2.b**  
(αριστερή εξωτερική σύνδεση),
  - **R1 RIGHT JOIN R2 ON R1.a = R2.b**  
(δεξιά εξωτερική σύνδεση),
  - **R1 FULL JOIN R2 ON R1.a = R2.b**  
(πλήρης εξωτερική σύνδεση).



# Γλώσσα Χειρισμού Δεδομένων – Ένωση

- «Να προσδιορισθούν οι τίτλοι των συνεδρίων και οι τίτλοι των περιοδικών».

```
SELECT συνέδριο  
FROM Πρακτικά_Συνεδρίου  
UNION  
SELECT τίτλος  
FROM Περιοδικό;
```

- Με **UNION ALL** διατηρούνται οι διπλοεγγραφές.



# Γλώσσα Χειρισμού Δεδομένων – Τομή

- «Να βρεθούν (αν υπάρχουν) συνδρομητές που έχουν το ίδιο όνομα με κάποιο συγγραφέα».

```
SELECT όνομα  
FROM Συνδρομητής  
INTERSECT  
SELECT όνομα  
FROM Συγγραφέας;
```

- Με **INTERSECT ALL** διατηρούνται οι διπλοεγγραφές.





# Γλώσσα Χειρισμού Δεδομένων – Διαφορά (1/2)

- «Να προσδιορισθούν οι κωδικοί και οι τίτλοι των άρθρων που δεν έχουν δημοσιευθεί σε κάποιο περιοδικό».

**SELECT** κωδικός, τίτλος

**FROM** Άρθρο

**EXCEPT**

**SELECT** κωδικός, τίτλος

**FROM** Άρθρο

**WHERE** κωδικός περιοδικού **IS NOT NULL;**

- Με **EXCEPT ALL** διατηρούνται οι διπλοεγγραφές.



# Γλώσσα Χειρισμού Δεδομένων – Διαφορά (2/2)

- «Να προσδιορισθούν οι κωδικοί και οι τίτλοι των άρθρων που δεν έχουν δημοσιευθεί σε κάποιο περιοδικό».
- Ισοδύναμα:

```
SELECT κωδικός, τίτλος
```

```
FROM Άρθρο
```

```
WHERE κωδικός_περιοδικού IS NULL;
```



# Γλώσσα Χειρισμού Δεδομένων – Συναθροιστικές Συναρτήσεις (1/3)

- Η SQL υποστηρίζει τις ακόλουθες συναρτήσεις:
  - **MIN** για την εύρεση της ελάχιστης τιμής μίας στήλης,
  - **MAX** για την εύρεση της μέγιστης τιμής μίας στήλης,
  - **AVG** για τον υπολογισμό της μέσης τιμής μίας στήλης,
  - **SUM** για τον υπολογισμό του αθροίσματος των τιμών μίας στήλης,
  - **COUNT** για τη μέτρηση του αριθμού των γραμμών.



# Γλώσσα Χειρισμού Δεδομένων – Συναθροιστικές Συναρτήσεις (2/3)

- «Να βρεθεί ο αριθμός των άρθρων που έχει γράψει κάθε συγγραφέας».

```
SELECT Συγγραφέας.όνομα,  
COUNT (Συγγραφή_Άρθρου.κωδικός_άρθρου)  
FROM Συγγραφέας, Συγγραφή_Άρθρου  
WHERE Συγγραφή_Άρθρου.κωδικός_συγγραφέα  
= Συγγραφέας.κωδικός  
GROUP BY Συγγραφέας.όνομα;
```

- Το πεδίο ως προς το οποίο εκτελείται η ομαδοποίηση στο Group by πρέπει να βρίσκεται στο τμήμα Select.



# Γλώσσα Χειρισμού Δεδομένων – Συναθροιστικές Συναρτήσεις (3/3)

- «Να βρεθεί ο μέσος όρος των συνδρομητών για όλες τις γνωστικές περιοχές».

```
SELECT τίτλος, AVG(αριθμός_συνδρομητών)  
FROM Γνωστική Περιοχή  
GROUP BY τίτλος ;
```



# Γλώσσα Χειρισμού Δεδομένων – Σειρές Χαρακτήρων (1/2)

- Για την αναζήτηση σειρών χαρακτήρων χρησιμοποιούνται οι τελεστές LIKE και NOT LIKE.

**SELECT** \*

**FROM** Συνδρομητής

**WHERE** όνομα **LIKE** '%όπουλος' ;



# Γλώσσα Χειρισμού Δεδομένων – Σειρές Χαρακτήρων (2/2)

Πρότυπο	Επεξήγηση	Παραδείγματα
'Βασ%'	Να εντοπιστούν οι συμβολοσειρές που αρχίζουν με 'Βασ'	'Βασιλάκης', 'Βασιλείου'
'%όπουλος'	Να εντοπιστούν οι συμβολοσειρές που τελειώνουν σε 'όπουλος'	'Βασιλακόπουλος', 'Νανόπουλος'
'%που%'	Να εντοπιστούν οι συμβολοσειρές που περιέχουν το 'που'	'Μανωλόπουλος', 'Παπαδόπουλος'
'Τιάκ_ς'	Να εντοπιστούν οι συμβολοσειρές με ακριβώς ένα χαρακτήρα μεταξύ του 'Τιακ' και του 'ς'	'Τιάκας', 'Τιάκος'
'%γγέλο_'	Να εντοπιστούν οι συμβολοσειρές που περιέχουν το 'γγέλο' και ακολουθεί ένας χαρακτήρας	'Ευαγγέλου', 'Αγγέλου'



# Γλώσσα Χειρισμού Δεδομένων – Υποερωτήματα (1/6)

- «Να βρεθούν οι τίτλοι των γνωστικών περιοχών με αριθμός συνδρομητών μεγαλύτερο από το μέσο όρο».

```
SELECT τίτλος  
FROM Γνωστική_Περιοχή  
WHERE αριθμός_συνδρομητών >  
  ( SELECT AVG (αριθμός_συνδρομητών)  
FROM Γνωστική_Περιοχή ) ;
```





# Γλώσσα Χειρισμού Δεδομένων – Υποερωτήματα (2/6)

- «Να προσδιορισθούν οι τίτλοι των γνωστικών περιοχών όπου δεν αντιστοιχεί κάποιο άρθρο».

```
SELECT τίτλος  
FROM Γνωστική_Περιοχή  
WHERE κωδικός NOT IN  
  SELECT κωδικός_γνωστικής_περιοχής  
  FROM Άρθρο);
```



# Γλώσσα Χειρισμού Δεδομένων – Υποερωτήματα (3/6)

- «Να βρεθούν οι κωδικοί των συγγραφέων που έχουν έστω και ένα άρθρο δημοσιευμένο και σε περιοδικό και σε πρακτικά συνεδρίου».

```
SELECT κωδικός_συγγραφέα  
FROM Συγγραφή_Άρθρου AS ΣΑ  
WHERE EXISTS  
  (SELECT *  
   FROM Άρθρο  
   WHERE κωδικός=ΣΑ.κωδικός_άρθρου AND  
   κωδικός_συνεδρίου IS NOT NULL AND  
   κωδικός_περιοδικού IS NOT NULL);
```



# Γλώσσα Χειρισμού Δεδομένων – Υποερωτήματα (4/6)

- Τα κατηγορήματα **SOME** ή **ANY** και **ALL** αντιστοιχούν στον καθολικό και υπαρξιακό ποσοδείκτη ( $\exists$  και  $\forall$ ).
- Πριν από τα κατηγορήματα **SOME** και **ALL** μπορεί να προηγείται τελεστής σύγκρισης ( $=, >, <, >=, <=, <>$ ).
- «Να βρεθούν οι γνωστικές περιοχές με μεγαλύτερο αριθμό συνδρομητών από τουλάχιστον μία άλλη γνωστική περιοχή».

**SELECT** τίτλος

**FROM** Γνωστική\_Περιοχή

**WHERE** αριθμός\_συνδρομητών > **SOME**

(**SELECT** αριθμός\_συνδρομητών

**FROM** Γνωστική\_Περιοχή) ;



# Γλώσσα Χειρισμού Δεδομένων – Υποερωτήματα (5/6)

- «Να βρεθούν οι γνωστικές περιοχές με μεγαλύτερο αριθμό συνδρομητών από όλες τις άλλες γνωστικές περιοχές».

```
SELECT τίτλος  
FROM Γνωστική_Περιοχή  
WHERE αριθμός_συνδρομητών > ALL  
  (SELECT αριθμός_συνδρομητών  
FROM Γνωστική_Περιοχή) ;
```



# Γλώσσα Χειρισμού Δεδομένων – Υποερωτήματα (6/6)

- «Να βρεθούν οι κωδικοί των συγγραφέων και το σύνολο των άρθρων που έχουν γράψει, έτσι ώστε ο αριθμός των άρθρων να είναι μεγαλύτερος από το ένα δέκατο των συνολικών δημοσιεύσεων».

```
SELECT κωδικός_συγγραφέα,  
COUNT (κωδικός_άρθρου) AS αριθμός_άρθρων  
FROM Συγγραφή_Άρθρου  
GROUP BY κωδικός_συγγραφέα  
HAVING αριθμός_άρθρων >  
    (SELECT COUNT (*) / 10  
FROM Συγγραφή_Άρθρου) ;
```



# Γλώσσα Χειρισμού Δεδομένων – Μετονομασία (1/2)

- «Να υπολογισθεί ο μέσος αριθμός συνδρομητών για κάθε γνωστική περιοχή».

```
SELECT τίτλος, AVG (αριθμός_συνδρομητών)  
FROM Γνωστική Περιοχή  
GROUP BY τίτλος;
```

```
SELECT τίτλος, AVG (αριθμός_συνδρομητών)  
AS μέσος_αριθμός_συνδρομητών  
FROM Γνωστική Περιοχή  
GROUP BY τίτλος;
```



# Γλώσσα Χειρισμού Δεδομένων – Μετονομασία (2/2)

- «Να βρεθούν τα ζεύγη κωδικών γνωστικών περιοχών έτσι ώστε ο αριθμός συνδρομητών της πρώτης να είναι μικρότερος του αριθμού συνδρομητών της δεύτερης».

```
SELECT A.κωδικός AS κωδικόςA, B.κωδικός  
AS κωδικόςB
```

```
FROM Γνωστική Περιοχή AS A, Γνωστική  
Περιοχή AS B
```

```
WHERE A.αριθμός_συνδρομητών <  
B.αριθμός_συνδρομητών;
```

- Τα A και B είναι μεταβλητές πλειάδων (tuple variables).



# Γλώσσα Χειρισμού Δεδομένων – Εισαγωγή (1/2)

```
INSERT INTO Όνομα_Πίνακα [(στήλ1, ..., στήλN) ]  
VALUES (τιμή1, ..., τιμήN);
```

Π.χ.

```
INSERT INTO Γνωστική_Περιοχή  
VALUES (10, 'Διαστημική Τεχνολογία', 0);
```

```
INSERT INTO Γνωστική_Περιοχή(τίτλος, κωδικός)  
VALUES ('Διαστημική Τεχνολογία', 10);
```





# Γλώσσα Χειρισμού Δεδομένων – Εισαγωγή (2/2)

- Χρήση υποερωτήματος σε εισαγωγή.

```
INSERT INTO Άρθρο_Περιοδικού  
AS  
SELECT κωδικός, τίτλος  
FROM Άρθρο  
WHERE κωδικός_περιοδικού IS NOT NULL  
      AND κωδικός_συνεδρίου IS NULL;
```



# Γλώσσα Χειρισμού Δεδομένων – Ενημέρωση

```
UPDATE Όνομα_Πίνακα  
SET (στήλη1=τιμή1, ..., στήληN=τιμήN)  
WHERE συνθήκη;
```

Π.χ.

```
UPDATE Γνωστική_Περιοχή  
SET (τίτλος='Διάστημα')  
WHERE κωδικός=10;
```

```
UPDATE Γνωστική_Περιοχή  
SET (αριθμός_συνδρομητών=0);
```



# Γλώσσα Χειρισμού Δεδομένων – Διαγραφή

```
DELETE FROM Όνομα_Πίνακα  
WHERE συνθήκη;
```

Π.χ.

```
DELETE FROM Άρθρο;
```

```
DELETE FROM Γνωστική_Περιοχή  
WHERE αριθμός_συνδρομητών = 0 AND  
Γνωστική_Περιοχή.κωδικός NOT IN  
(SELECT DISTINCT  
κωδικός_γνωστικής_περιοχής FROM Άρθρο);
```



# Γλώσσα Χειρισμού Δεδομένων – Διαχείριση Όψεων

```
CREATE VIEW Όνομα_Όψης  
AS (υποερώτημα SQL);
```

Π.χ.

```
CREATE VIEW Ονόματα_Περιοδικών_Συνεδρίων  
AS  
(SELECT τίτλος FROM Περιοδικό  
UNION  
SELECT συνέδριο FROM Πρακτικά_Συνεδρίου);
```

```
DROP VIEW Ονόματα_Περιοδικών_Συνεδρίων;
```



# Σύνοψη

- Η SQL προτυποποιήθηκε για πρώτη φορά 1986 και αποτελεί τη γλώσσα χειρισμού ΒΔ. Βασίζεται στη σχεσιακή άλγεβρα και διακρίνεται σε γλώσσα ορισμού δεδομένων και γλώσσα χειρισμού δεδομένων.
- Κάθε σχεσιακό σύστημα υποστηρίζει SQL, οπότε η διατύπωση των ερωτημάτων είναι ανεξάρτητη από το χρησιμοποιούμενο ΣΔΒΔ.
- Η γλώσσα χρησιμοποιείται για όλες τις λειτουργίες του ΣΔΒΔ (δημιουργία πινάκων, εισαγωγή και αναζήτηση δεδομένων, δημιουργία νέων χρηστών, κλπ.)
- Πολλές γλώσσες προγραμματισμού έχουν τροποποιηθεί ώστε να υποστηρίζουν διατύπωση ερωτημάτων σε SQL.
- Η SQL είναι έχει απλή σύνταξη, και αποδεσμεύει το χρήστη από λεπτομέρειες υλοποίησης.



# Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Ιωάννης  
Μανωλόπουλος. «Βάσεις Δεδομένων. Η γλώσσα SQL». Έκδοση: 1.0.  
Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:  
<http://eclass.auth.gr/courses/OCRS263/>.



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

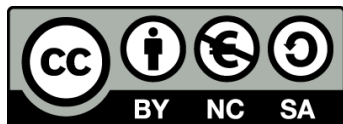
[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>





# Τέλος Ενότητας

Επεξεργασία: Ανδρέας Κοσματόπουλος  
Θεσσαλονίκη, Νοέμβριος 2014



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ





ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

---

# Σημειώματα

# Σημείωμα Ιστορικού Εκδόσεων Έργου

---

Το παρόν έργο αποτελεί την έκδοση **1.0**.



# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

