Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

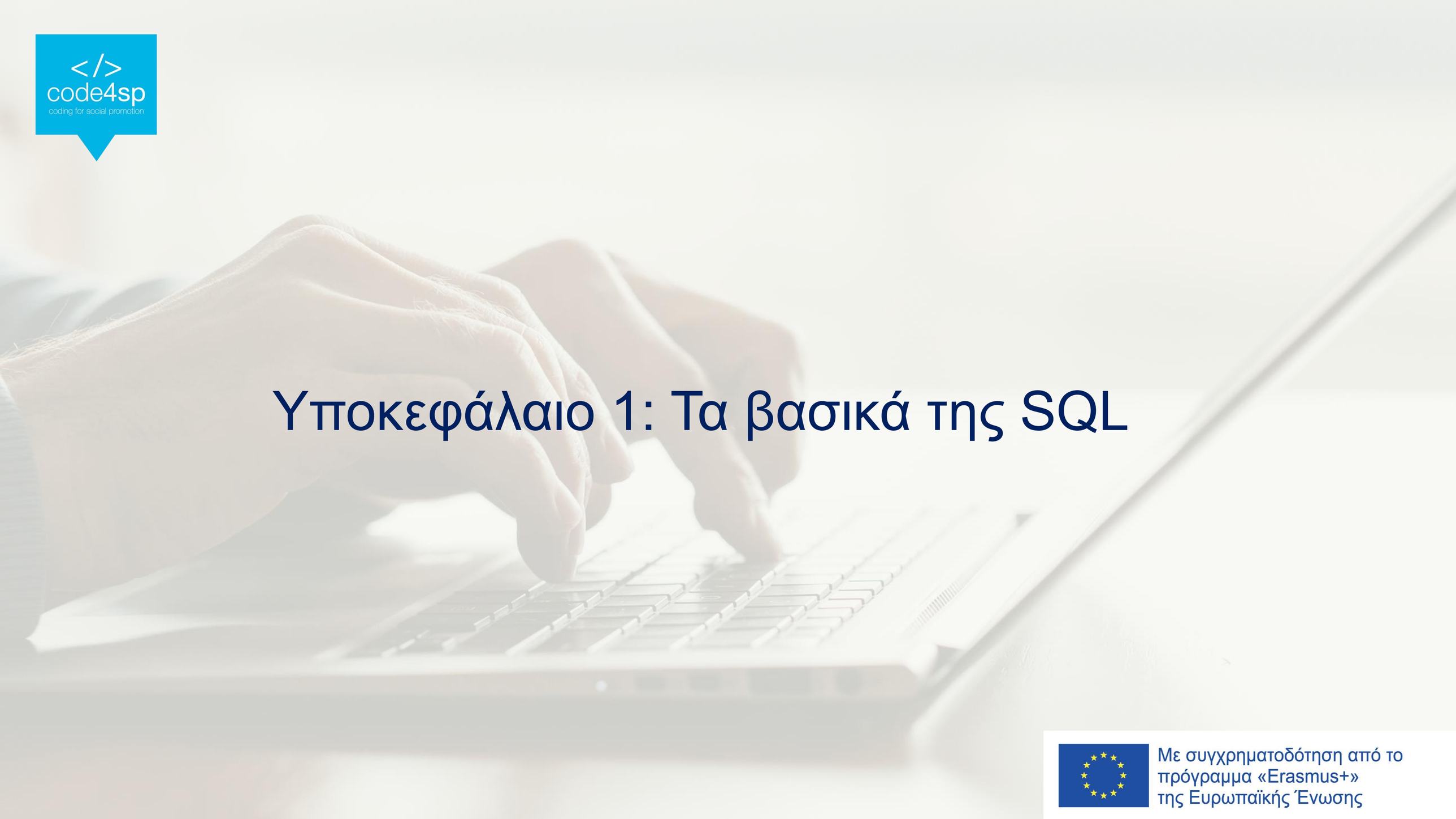
code4sp
coding for social promotion

Εκπαιδευτικό Υλικό SQL
Υποκεφάλαιο 1 – Τα βασικά της SQL

WP3: Εκπαιδευτικό Υλικό του έργου Code4SP

Συντάχθηκε από: CITIZENS IN POWER

Η υποστήριξη της Ευρωπαϊκής Επιτροπής στην παραγωγή της παρούσας έκδοσης δε συνιστά αποδοχή του περιεχομένου, το οποίο αντικατοπτρίζει αποκλειστικά τις απόψεις των συντακτών της, και η Επιτροπή δεν μπορεί να αναλάβει την ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός Έργου: 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)



Υποκεφάλαιο 1: Τα βασικά της SQL

Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης

Εισαγωγή στη γλώσσα SQL

- Η SQL (Structured Query Language) είναι μια γλώσσα υπολογιστή για την αποθήκευση, το **χειρισμό και την ανάκτηση δεδομένων που είναι αποθηκευμένα σε μια σχεσιακή βάση δεδομένων.**

Μια σχεσιακή βάση δεδομένων είναι μια συλλογή στοιχείων δεδομένων με προκαθορισμένες σχέσεις μεταξύ τους.

Μερικές από τις πολλές εφαρμογές της είναι:

- Να επιτρέπει στους χρήστες να **έχουν πρόσβαση σε δεδομένα** στα σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων.
- Να επιτρέπει στους χρήστες να **περιγράψουν τα δεδομένα**.
- Να επιτρέπει στους χρήστες να **ορίσουν τα δεδομένα** σε μια βάση δεδομένων και να χειρίζονται τα δεδομένα αυτά.

Να την δοκιμάσουμε;
https://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Σύνταξη στην SQL

Οι περισσότερες από τις ενέργειες που πρέπει να εκτελέσετε σε μια βάση δεδομένων γίνονται με εντολές SQL

Μια εντολή στην SQL αποτελείται από μια ακολουθία λέξεων-κλειδιών, αναγνωριστικών κ.λπ., που τερματίζονται από ένα ελληνικό ερωτηματικό (;)

Παράδειγμα:

SELECT όνομα_εργαζομένου, ημερομηνία_πρόσληψης, μισθός FROM Εργαζόμενοι WHERE μισθός > 5000;

Για πιο εύκολη αναγνωσιμότητα, μπορείτε να γράψετε την ίδια εντολή ως εξής:

```
SELECT όνομα_εργαζομένου, ημερομηνία_πρόσληψης, μισθός
FROM Εργαζόμενοι
WHERE μισθός > 5000;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Διάκριση πεζών-κεφαλαίων στην SQL

Οι λέξεις-κλειδιά SQL **δεν κάνει διάκριση** πεζών-κεφαλαίων, που σημαίνει ότι η εντολή SELECT είναι ίδια με την select.

- Εξετάστε μια άλλη εντολή στην SQL που ανακτά εγγραφές από τον πίνακα Εργαζομένων:

SELECT όνομα_εργαζομένου, ημερομηνία_πρόσληψης, μισθός FROM Εργαζόμενοι;

- Η ίδια εντολή μπορεί επίσης να γραφτεί ως εξής:

select όνομα_εργαζομένου, ημερομηνία_πρόσληψης, μισθός from εργαζόμενοι;

*** Σημειώστε ότι αυτό εξαρτάται από το λειτουργικό σύστημα, δηλαδή οι πλατφόρμες Unix ή Linux είναι ευαίσθητες στη διάκριση πεζών-κεφαλαίων, ενώ οι πλατφόρμες Windows δεν είναι.**

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Επιλογή δεδομένων στην SQL (SELECT)

Η εντολή SELECT **επιλέγει ή ανακτά δεδομένα από έναν ή περισσότερους πίνακες.**

Μπορείτε να χρησιμοποιήσετε αυτή την εντολή για να ανακτήσετε όλες τις σειρές από έναν πίνακα με μία κίνηση ή να ανακτήσετε μόνο εκείνες τις σειρές που ικανοποιούν μια συγκεκριμένη συνθήκη ή έναν συνδυασμό συνθηκών.

- *Επιλογή όλων από τον πίνακα:* επιλέγει όλες τις σειρές από τον πίνακα των εργαζομένων

SELECT * FROM Εργαζόμενοι;

- *Επιλογή συγκεκριμένων στηλών από τον πίνακα: επιλέγει* όλες τις σειρές από τις καθορισμένες στήλες

SELECT ταυτότητα_εργαζομένου, όνομα_εργαζομένου, ημερομηνία_πρόσληψης, μισθός
FROM εργαζόμενοι;

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Επιλογή ορισμένων δεδομένων στην SQL (SELECT DISTINCT)

Στην προηγούμενη διαφάνεια, είδαμε πώς να επιλέξουμε όλες τις τιμές από έναν πίνακα ή από συγκεκριμένες στήλες.

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Πίνακας πελατών - ΠΑΡΑΔΕΙΓΜΑ ΤΗΣ ΕΝΤΟΛΗΣ SELECT DISTINCT

(Πηγή: https://www.w3schools.com/sql/sql_distinct.asp)

- *Γράψτε μια εντολή για να επιλέξετε όλες τις τιμές από τη στήλη Χώρα (Country) στον πίνακα Πελάτες*

  Τι παρατηρείτε; Είναι οι τιμές διαφορετικές ή οι ίδιες;

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Επιλογή ορισμένων δεδομένων στην SQL (SELECT DISTINCT)

Η εντολή SELECT DISTINCT **παραλείπει τις διπλές τιμές** όταν χρησιμοποιείται σε ένα ερώτημα.

Σύνταξη:

SELECT DISTINCT στήλη1, στήλη2, ...

FROM όνομα_πίνακα;

***Σημειώστε ότι αυτό το παράδειγμα δεν θα λειτουργήσει στο Firefox, καθώς το COUNT(DISTINCT όνομα_στήλης) δεν υποστηρίζεται στο σύστημα MS Access.**

Παράδειγμα 1: Επιλέξτε τις διαφορετικές χώρες από τον πίνακα Πελατών

SELECT DISTINCT Χώρα FROM Πελάτες;

Παράδειγμα 2: Καταγράψτε τον αριθμό των διαφορετικών χωρών των πελατών

SELECT COUNT(DISTINCT Χώρα) FROM Πελάτες;

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Όρος WHERE στην SQL

Σε πραγματικές περιπτώσεις, πρέπει γενικά να επιλέξουμε, να ενημερώσουμε ή να διαγράψουμε μόνο εκείνα τα αρχεία που ικανοποιούν καθορισμένες συνθήκες, όπως οι χρήστες που ανήκουν σε μια συγκεκριμένη ηλικιακή ομάδα, χώρα κ.λπ.

Ο όρος WHERE χρησιμοποιείται με τις εντολές SELECT, UPDATE, και DELETE.

Ο όρος WHERE χρησιμοποιείται με την εντολή SELECT για την εξαγωγή μόνο εκείνων των εγγραφών που ικανοποιούν καθορισμένες συνθήκες.

Σύνταξη:

```
SELECT στήλη_λίστα
FROM όνομα_πίνακα
WHERE συνθήκη;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Όρος WHERE στην SQL

Παράδειγμα 1: Επιλέξτε όλους τους υπαλλήλους από τον πίνακα Εργαζομένων των οποίων ο μισθός είναι μεγαλύτερος από 7000

```
SELECT * FROM εργαζόμενοι WHERE μισθός > 7000;
```

Παράδειγμα 2: Επιλέξτε όλους τους υπαλλήλους με κωδικό τμήματος =1:

```
SELECT * FROM εργαζόμενοι WHERE κωδικό_τμήματος=1;
```

Ο όρος WHERE απλώς φιλτράρει τα ανεπιθύμητα δεδομένα.

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Όρος WHERE στην SQL

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
BETWEEN	Between a certain range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

Πίνακας των λειτουργιών που χρησιμοποιούνται στο σημείο του όρου WHERE

(Πηγή: https://www.w3schools.com/sql/sql_where.asp)

Εξάσκηση:

- Επιλέξτε όλους τους εργαζομένους από τον πίνακα εργαζομένων των οποίων ο μισθός είναι μικρότερος από 7000
- Επιλέξτε όλους τους εργαζομένους από τον πίνακα εργαζομένωνν των οποίων το αναγνωριστικό_τμήματος είναι το 5

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

AND, OR και NOT στην SQL

Ο όρος WHERE μπορεί να συνδυαστεί με τους τελεστές AND, OR, και NOT.

Ο τελεστής **AND** εμφανίζει μια εγγραφή εάν όλες οι συνθήκες που χρησιμοποιούν AND είναι TRUE.

Σύνταξη:

```
SELECT στήλη1, στήλη2, ...
FROM όνομα_πίνακα
WHERE συνθήκη1 AND συνθήκη2 AND συνθήκη3 ...;
```

Παράδειγμα: Επιλέξτε όλα τα πεδία από τον πίνακα πελατών όπου η χώρα είναι η «Γερμανία» ΚΑΙ η πόλη είναι το «Βερολίνο».

```
SELECT * FROM Πελάτες
WHERE Χώρα='Γερμανία' AND Πόλη='Βερολίνο';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

AND, OR και NOT στην SQL

Ο τελεστής **OR** εμφανίζει μια εγγραφή εάν οποιαδήποτε από τις συνθήκες που χρησιμοποιούν το OR είναι TRUE.

Σύνταξη:

```
SELECT στήλη1, στήλη2, ...
FROM όνομα_πίνακα
WHERE συνθήκη1 OR συνθήκη2 OR συνθήκη3 ...;
```

Παράδειγμα 1: Επιλέξτε όλα τα πεδία από τον πίνακα πελατών όπου η πόλη είναι το «Βερολίνο» ή το «Μόναχο»

```
SELECT * FROM Πελάτες
WHERE Πόλη='Βερολίνο' OR Πόλη='Μόναχο';
```

Παράδειγμα 2: Επιλέξτε όλα τα πεδία από τον πίνακα πελατών όπου η χώρα είναι «Γερμανία» ή «Ισπανία».

```
SELECT * FROM Πελάτες
WHERE Χώρα='Γερμανία' OR Χώρα='Ισπανία';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

AND, OR και NOT στην SQL

Ο **τελεστής NOT** εμφανίζει μια εγγραφή εάν οι συνθήκες είναι NOT TRUE.

Σύνταξη:

```
SELECT στήλη1, στήλη2, ...
FROM όνομα_πίνακα
WHERE NOT συνθήκη;
```

Παράδειγμα: Επιλέξτε όλα τα πεδία από τον πίνακα πελατών όπου η χώρα ΔΕΝ είναι η «Γερμανία».

```
SELECT * FROM Πελάτες
WHERE NOT Χώρα='Γερμανία';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

AND, OR και NOT στην SQL

Συνδυασμός των AND, OR και NOT

Παράδειγμα 1: Επιλέξτε όλες τις σειρές από τον πίνακα Πελάτες όπου η χώρα είναι η Γερμανία και η πόλη πρέπει να είναι είτε το Βερολίνο είτε το Μόναχο.

```
SELECT * FROM Πελάτες
WHERE Χώρα='Γερμανία' AND (Πόλη='Βερολίνο' OR Πόλη='Μόναχο');
```

Παράδειγμα 2: Επιλέξτε όλες τις σειρές από τον πίνακα Πελάτες όπου η χώρα ΔΕΝ είναι η Γερμανία και ούτε οι ΗΠΑ.

```
SELECT * FROM Πελάτες
WHERE NOT Χώρα='Γερμανία' AND NOT Χώρα='ΗΠΑ';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ταξινόμηση στην SQL (ORDER BY)

Η λέξη-κλειδί ORDER BY ταξινομεί το σύνολο των αποτελεσμάτων κατά αύξουσα ή φθίνουσα σειρά. Η λέξη-κλειδί ORDER BY ταξινομεί τις εγγραφές με αύξουσα σειρά από προεπιλογή.

Σύνταξη:

```
SELECT στήλη1, στήλη, ...
FROM όνομα_πίνακα
ORDER BY στήλη1, στήλη2, ... ASC|DESC;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ταξινόμηση στην SQL (ORDER BY)

Για τα παραδείγματα μας, θα χρησιμοποιήσουμε τον πίνακα Πελατών που μπορείτε να βρείτε εδώ στα Αγγλικά.

Παράδειγμα 1: Επιλέγει όλους τους πελάτες από τον πίνακα Πελατών και τους ταξινομεί κατά τη στήλη Χώρα

```
SELECT * FROM Πελάτες
ORDER BY Χώρα:
```

Παράδειγμα 2: Επιλέγει όλους τους πελάτες από τον ίδιο πίνακα και τους ταξινομεί με φθίνουσα σειρά κατά τη στήλη Χώρα

```
SELECT * FROM Πελάτες
ORDER BY Χώρα DESC;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ταξινόμηση στην SQL (ORDER BY)

Παράδειγμα 3: Επιλέγει όλους τους πελάτες από τον ίδιο πίνακα και τους ταξινομεί ανά χώρα και όνομα πελάτη.

```
SELECT * FROM Πελάτες
ORDER BY Χώρα, ΌνομαΠελάτη;
```

Εδώ, η ταξινόμηση γίνεται αρχικά ανά χώρα. Ωστόσο, εάν υπάρχουν ορισμένες σειρές που έχουν την ίδια χώρα, τότε ταξινομούνται ανά Όνομα Πελάτη.

Παράδειγμα 4: Επιλέγει όλους τους πελάτες από τον ίδιο πίνακα και τους ταξινομεί κατά αύξουσα σειρά ανά χώρα και κατά φθίνουσα σειρά ανά όνομα πελάτη

```
SELECT * FROM Πελάτες
ORDER BY Χώρα ASC, ΌνομαΠελάτη DESC;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ταξινόμηση στην SQL (ORDER BY)

Παράδειγμα 3: Επιλέγει όλους τους πελάτες από τον ίδιο πίνακα και τους ταξινομεί ανά χώρα και όνομα πελάτη.

```
SELECT * FROM Πελάτες
ORDER BY Χώρα, ΌνομαΠελάτη;
```

Εδώ, η ταξινόμηση γίνεται αρχικά ανά χώρα. Ωστόσο, εάν υπάρχουν ορισμένες σειρές που έχουν την ίδια χώρα, τότε ταξινομούνται ανά Όνομα Πελάτη.

Παράδειγμα 4: Επιλέγει όλους τους πελάτες από τον ίδιο πίνακα και τους ταξινομεί κατά αύξουσα σειρά ανά χώρα και κατά φθίνουσα σειρά ανά όνομα πελάτη

```
SELECT * FROM Πελάτες
ORDER BY Χώρα ASC, ΌνομαΠελάτη DESC;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ταξινόμηση στην SQL (ORDER BY)

Εξάσκηση:

- Επιλέξτε όλους τους πελάτες από τον πίνακα Πελατών και ταξινομήστε τους με αύξουσα σειρά της στήλης Χώρα
- Επιλέξτε όλους τους πελάτες από τον πίνακα Πελατών και ταξινομήστε τους με φθίνουσα σειρά ανά Όνομα Πελάτη
- Επιλέξτε όλους τους πελάτες από τον πίνακα Πελατών και ταξινομήστε τους κατά φθίνουσα σειρά ανά Πόλη και κατά αύξουσα σειρά ανά Όνομα Πελάτη

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Εισαγωγή Δεδομένων στην SQL (INSERT INTO)

Η εντολή INSERT INTO εισάγει νέες εγγραφές σε έναν πίνακα. Για να τρέξει σωστά ο κώδικάς σας, καθορίστε τα ονόματα των στηλών και τις τιμές που θα εισαχθούν.

Σύνταξη:

INSERT INTO όνομα_πίνακα (στήλη1, στήλη2, στήλη3, ...)

VALUES (τιμή1, τιμή2, τιμή3, ...);

Παράδειγμα: Προσθέστε μια νέα εγγραφή στον πίνακα «Πελάτες»

INSERT INTO Πελάτες (ΌνομαΠελάτη, ΌνομαΕπικοινωνίας, Διεύθυνση, Πόλη, ΤαχυδρομικόςΚώδικας, Χώρα)

VALUES (‘Κάρντιναλ', ‘Τομ Μπ. Έριχσεν', ‘Σκάγκεν 21', ‘Στάβανγκερ', '4006', ‘Νορβηγία');

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Κενές τιμές στην SQL (NULL)

Ένα πεδίο με τιμή NULL είναι **ένα πεδίο χωρίς τιμή**. Εάν ένα πεδίο σε έναν πίνακα είναι προαιρετικό, είναι δυνατό να εισαχθεί μια νέα εγγραφή ή να ενημερωθεί μια εγγραφή χωρίς να προστεθεί μια τιμή σε αυτό το πεδίο.

*** Μια τιμή NULL είναι διαφορετική από μια μηδενική τιμή ή ένα πεδίο με κενά.**

Ένα πεδίο με τιμή NULL έχει μείνει κενό κατά την καταχώρηση εγγραφών!

Για να ελέγξετε για τιμές NULL, δεν μπορείτε να χρησιμοποιήσετε τελεστές σύγκρισης όπως =, <>.

Μπορείτε να χρησιμοποιήσετε είτε τους τελεστές **IS NULL** είτε NOT NULL.

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Κενές τιμές στην SQL (NULL)

Σύνταξη IS NULL:

```
SELECT όνομα_στήλης
FROM όνομα_πίνακα
WHERE όνομα_στήλης IS NULL;
```

Σύνταξη IS NOT NULL:

```
SELECT όνομα_στήλης
FROM όνομα_πίνακα
WHERE όνομα_στήλης IS NOT NULL;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Κενές τιμές στην SQL (NULL)

Παράδειγμα της τιμής IS NULL: Επιλέγει όλους τους πελάτες με τιμές NULL στη στήλη Διεύθυνση

```
SELECT ΌνομαΠελάτη, ΌνομαΕπικοινωνίας, Διεύθυνση
FROM Πελάτες
WHERE Διεύθυνση IS NULL;
```

Παράδειγμα της τιμής IS NOT NULL: Επιλέγει όλους τους πελάτες με τιμές NOT NULL στη στήλη Διεύθυνση

```
SELECT ΌνομαΠελάτη, ΌνομαΕπικοινωνίας, Διεύθυνση
FROM Πελάτες
WHERE Διεύθυνση IS NOT NULL;
```

Εξάσκηση:

- Αναζητήστε κενές τιμές (NULL) στις στήλες Ταχυδρομικός Κώδικας και Πόλη
- Αναζητήστε μη-κενές τιμές (NOT NULL) στη στήλη ΌνομαΕπαφής

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ενημέρωση δεδομένων στην SQL (UPDATE)

Η εντολή UPDATE **τροποποιεί τις υπάρχουσες εγγραφές** ενός πίνακα.

***Να είστε προσεκτικοί όταν ενημερώνετε εγγραφές σε έναν πίνακα! Παρατηρήστε τον όρο WHERE στην εντολή UPDATE. Ο όρος WHERE προσδιορίζει ποια εγγραφή(-ές) πρέπει να ενημερωθεί (-ούν).**
***Εάν παραλείψετε τον όρο «WHERE», θα ενημερωθούν όλες οι εγγραφές στον πίνακα!**

Σύνταξη:

```
UPDATE όνομα_πίνακα
SET στήλη1 = τίμη1, στήλη2 = τιμή2, ...
WHERE συνθήκη;
```

Παράδειγμα: Ενημέρωση της στήλης ΚωδικόςΠελάτη=1 από τον πίνακα «Πελάτες» στη δειγματική βάση δεδομένων

```
UPDATE Πελάτες
SET ΌνομαΕπικοινωνίας = ‘Άλφρεντ Σμιντ’, Πόλη= ‘Φρανκφούρτη'
WHERE ΚωδικόςΠελάτη = 1;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ενημέρωση δεδομένων στην SQL (UPDATE)

Ο όρος **WHERE καθορίζει πόσες εγγραφές** θα ενημερωθούν.

Παράδειγμα: Ενημέρωση της στήλης ΌνομαΕπικοινωνίας σε «Χουάν» για όλες τις εγγραφές όπου η χώρα είναι «Μεξικό» στον πίνακα Πελατών

```
UPDATE Πελάτες
SET ΌνομαΕπικοινωνίας='Χουάν'
WHERE Χώρα='Μεξικό';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Διαγραφή δεδομένων στην SQL (DELETE)

Η εντολή DELETE **διαγράφει υπάρχουσες εγγραφές** σε έναν πίνακα.

Σύνταξη:

DELETE FROM όνομα_πίνακα WHERE συνθήκη;

Παράδειγμα:

DELETE FROM Πελάτες WHERE ΌνομαΠελάτη='Αλφρεντς Φούτερκιστε';

Λάβετε υπόψη ότι πρέπει να είστε προσεκτικοί όταν διαγράφετε καταχωρήσεις σε έναν πίνακα! Παρατηρήστε τον όρο WHERE στην εντολή DELETE. Ο όρος WHERE προσδιορίζει ποια εγγραφή(-ές) πρέπει να διαγραφεί (-ούν).
***Εάν παραλείψετε τον όρο "WHERE", θα διαγραφούν όλες οι καταχωρήσεις του πίνακα!**

Είναι δυνατή η **διαγραφή όλων των σειρών σε έναν πίνακα χωρίς να διαγραφεί ο ίδιος ο πίνακας**. Αυτό σημαίνει ότι η δομή, τα χαρακτηριστικά και οι δείκτες του πίνακα θα παραμείνουν άθικτα:

DELETE FROM όνομα_πίνακα;

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Διαγραφή δεδομένων στην SQL (DELETE)

Η εντολή DELETE **διαγράφει υπάρχουσες εγγραφές** σε έναν πίνακα.

Σύνταξη:

DELETE FROM όνομα_πίνακα WHERE συνθήκη;

Παράδειγμα:

DELETE FROM Πελάτες WHERE ΌνομαΠελάτη='Άλφρεντς Φούτερκιστε';

***Να θυμάστε ότι πρέπει να είστε προσεκτικοί όταν διαγράφετε εγγραφές σε έναν πίνακα! Παρατηρήστε τον όρο WHERE στην εντολή DELETE. Ο όρος WHERE προσδιορίζει ποια εγγραφή(-ές) πρέπει να διαγραφεί (-ούν).**
***Εάν παραλείψετε τον όρο «WHERE», θα διαγραφούν όλες οι εγγραφές του πίνακα!**

Είναι δυνατή η **διαγραφή όλων των σειρών σε έναν πίνακα χωρίς να διαγραφεί ο ίδιος ο πίνακας**. Αυτό σημαίνει ότι η δομή, τα χαρακτηριστικά και οι δείκτες του πίνακα θα παραμείνουν άθικτα:

DELETE FROM όνομα_πίνακα;

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Επιλογή συγκεκριμένου αριθμού δεδομένων στην SQL (SELECT TOP)

Ο όρος SELECT TOP **καθορίζει τον αριθμό των εγγραφών που θα επιλεχθούν.**

Σύνταξη SQL Server/MS Access:

```
SELECT TOP αριθμός|ποσοστό όνομα(τα)_στήλης(ων)
FROM όνομα_πίνακα
WHERE συνθήκη;
```

Σύνταξη MySQL:

```
SELECT όνομα(τα)_στήλης(ων)
FROM όνομα_πίνακα
WHERE συνθήκη
LIMIT αριθμός;
```

***Σημειώστε ότι δεν υποστηρίζουν όλα τα συστήματα βάσης δεδομένω τον όρο SELECT TOP.**
***Το MySQL υποστηρίζει τον όρο LIMIT για την επιλογή ενός περιορισμένου αριθμού εγγραφών, ενώ το Oracle χρησιμοποιεί τους όρους FETCH FIRST αριθμός ROWS ONLY και ROWNUM.**

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Επιλογή συγκεκριμένου αριθμού δεδομένων στην SQL (SELECT TOP)

Σύνταξη Oracle 12:

SELECT όνομα(τα)_στήλης(ων)

FROM όνομα_πίνακα

ORDER BY όνομα(τα)_στήλης(ων)

FETCH FIRST αριθμός ROWS ONLY;

Για τη σύνταξη σε παλαιότερη έκδοση του Oracle και τη σύνταξη με ORDER BY σε παλαιότερη έκδοση Oracle, κάντε κλικ εδώ

Παράδειγμα: Επιλέξτε τις τρεις πρώτες εγγραφές από τον πίνακα Πελατών

SQL Server/MS Access: SELECT TOP 3 * FROM Πελάτες;

MySQL: SELECT * FROM Πελάτες LIMIT 3;

Oracle: SELECT * FROM Πελάτες FETCH FIRST 3 ROWS ONLY;

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Επιλογή συγκεκριμένου αριθμού δεδομένων στην SQL (SELECT TOP)

ΑΝΩΤΑΤΟ ΠΟΣΟΣΤΟ

Για να επιλέξετε το πρώτο 50% των εγγραφών που βρίσκονται στον πίνακα Πελατών:

SQL Server/MS Access: SELECT TOP 50 PERCENT * FROM Πελάτες;

Oracle: SELECT * FROM Πελάτες FETCH FIRST 50 PERECNT ROWS ONLY;

Προσθήκη του όρου WHERE

Επιλέξτε τις τρεις πρώτες εγγραφές από τον Πίνακα Πελατών, όπου η Χώρα είναι η «Γερμανία»:

- SQL Server/MS Access: SELECT TOP 3 * FROM Πελάτες WHERE Χώρα='Γερμανία';
- MySQL: SELECT * FROM Πελάτες WHERE Χώρα='Γερμανία' LIMIT 3;
- Oracle: SELECT * FROM Πελάτες WHERE Χώρα='Γερμανία' FETCH FIRST 3 ROWS ONLY;

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Min και Max στην SQL

Η συνάρτηση MIN() επιλέγει **τις μικρότερες τιμές από τις** επιλεγμένες στήλες.

Σύνταξη της συνάρτησης MIN()

```
SELECT MIN(όνομα_στήλης)
FROM όνομα_πίνακα
WHERE συνθήκη;
```

Η συνάρτηση MAX() επιλέγει **τη μεγαλύτερη τιμή** από τις επιλεγμένες στήλες.

Σύνταξη της συνάρτησης MAX()

```
SELECT MAX(όνομα_στήλης)
FROM όνομα_πίνακα
WHERE συνθήκη;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Min και Max στην SQL

Στα παρακάτω παραδείγματα, χρησιμοποιούμε τον πίνακα Προϊόντα που μπορείτε να βρείτε εδώ.

Παράδειγμα 1: Βρίσκει την τιμή του φθηνότερου προϊόντος

```
SELECT MIN(Τιμή) AS ΦθηνότερηΤιμή
FROM Προϊόντα;
```

Παράδειγμα 2: Βρίσκει την τιμή του ακριβότερου προϊόντος

```
SELECT MAX(Τίμη) AS ΑκριβότερηΤιμή
FROM Προϊόντα;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Count, Avg, Sum στην SQL

Η λειτουργία COUNT() επιλέγει **τον αριθμό των σειρών που ταιριάζουν σε ένα καθορισμένο κριτήριο.**

Σύνταξη COUNT()

```
SELECT COUNT(όνομα_στήλης)
FROM όνομα_πίνακα
WHERE συνθήκη;
```

Η λειτουργία AVG() επιλέγει **τη μέση τιμή μιας αριθμητικής στήλης.**

Σύνταξη AVG()

```
SELECT AVG(όνομα_στήλης)
FROM όνομα_πίνακα
WHERE συνθήκη;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Count, Avg, Sum στην SQL

Η συνάρτηση SUM() επιστρέφει **το συνολικό άθροισμα μιας αριθμητικής στήλης.**

Σύνταξη SUM()

```
SELECT SUM(όνομα_στήλης)
FROM όνομα_πίνακα
WHERE συνθήκη;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Count, Avg, Sum στην SQL

Για τα παραδείγματα, θα χρησιμοποιήσουμε τον ίδιο πίνακα στο που χρησιμοποιήσαμε με τα Min και Max, τον πίνακα Προϊόντα:

Παράδειγμα του COUNT(): Εκτελέστε ένα ερώτημα για να βρείτε τον αριθμό των προϊόντων

```
SELECT COUNT(ΚωδικόςΠροϊόντων)
FROM Προϊόντα;
```

Παράδειγμα της AVG(): Εκτελέστε ένα ερώτημα για να βρείτε τη μέση τιμή όλων των προϊόντων

```
SELECT AVG(Τιμή)
FROM Προϊόντα;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Count, Avg, Sum στην SQL

Στο παρακάτω παράδειγμα, θα χρησιμοποιήσουμε τον πίνακα ΛεπτομέρειεςΠαραγγελιών, που μπορείτε να βρείτε εδώ, για να βρούμε το άθροισμα της στήλης «Ποσότητα».

Παράδειγμα του SUM():

```
SELECT SUM(Ποσότητα)
FROM ΛεπτομέρειεςΠαραγγελιών;
```

Εξάσκηση:

- Βρείτε τη μέση ποσότητα από τον πίνακα ΛεπτομέρειεςΠαραγγελιών
- Βρείτε τον αριθμό των παραγγελιών από τον πίνακα ΛεπτομέρειεςΠαραγγελιών

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής LIKE στην SQL

Ο τελεστής LIKE χρησιμοποιείται σε ένα όρο WHERE **για να αναζητήσει ένα καθορισμένο μοτίβο σε μια στήλη.**

Σύνταξη του τελεστή LIKE

SELECT στήλη1, στήλη2, ...
FROM όνομα_πίνακα
WHERE στήλη LIKE μοτίβο;

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

Τελεστές LIKE με τους χαρακτήρες μπαλαντέρ ‘%’ και ‘_’

(Πηγή: https://www.w3schools.com/sql/sql_like.asp)

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής Like στην SQL

Παράδειγμα 1: Επιλογή όλωω των πελατών με Όνομα Πελάτη που αρχίζει με ‘α’:

```
SELECT * FROM Πελάτες
WHERE ΌνομαΠελάτη LIKE 'α%';
```

Παράδειγμα 2: Επιλογή όλων των ονομάτων πελατών που τελειώνουν με «α»:

```
SELECT * FROM Πελάτες
WHERE ΌνομαΠελάτη LIKE '%α';
```

Παράδειγμα 3: Επιλογή όλων των ονομάτων πελατών που περιέχουν ‘ή’ στο όνομά τους σε οποιαδήποτε θέση.

```
SELECT * FROM Πελάτες
WHERE Όνομα Πελάτη LIKE '%ή%';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής Like στην SQL

Παράδειγμα 4: Επιλογή όλων των ονομάτων πελατών που περιέχουν ‘ρ’ στη δεύτερη θέση:

```
SELECT * FROM Πελάτες
WHERE ΌνομαΠελάτη LIKE '_ρ%';
```

Παράδειγμα 5: Επιλογή όλων των ονομάτων πελατών που αρχίζουν με ‘α’ και έχουν τουλάχιστον τρεις χαρακτήρες στο σύνολο

```
SELECT * FROM Πελάτες
WHERE ΌνομαΠελάτη LIKE 'α__%';
```

Παράδειγμα 6: Επιλογή όλων των ονομάτων πελατών που αρχίζουν με ‘α’ και τελειώνουν με ‘ο’ :

```
SELECT * FROM Πελάτες
WHERE ΌνομαΠελάτη LIKE 'α%ο';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Χαρακτήρες Μπαλαντέρ στην SQL

Ένας χαρακτήρας μπαλαντέρ **αντικαθιστά έναν ή περισσότερους χαρακτήρες σε μια** συμβολοσειρά. Χρησιμοποιείται με τον τελεστή LIKE.

Για περισσότερες πληροφορίες σχετικά με τους χαρακτήρες μπαλαντέρ που χρησιμοποιούνται στα σχεσιακά ΣΔΒΔ, κάντε κλικ εδώ.

Παραδείγματα με τον χαρακτήρα μπαλαντέρ %

Σε αυτό το παράδειγμα, επιλέγουμε όλους τους πελάτες με Πόλη που ξεκινά με «βερ»:

```
SELECT * FROM Πελάτες
WHERE Πόλη LIKE ‘βερ%';
```

Στο παρακάτω παράδειγμα, επιλέγουμε όλους τους πελάτες με Πόλη που περιέχει το «εσ»:

```
SELECT * FROM Πελάτες
WHERE Πόλη LIKE ‘%εσ%';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Χαρακτήρες Μπαλαντέρ στην SQL

Παραδείγματα με τον χαρακτήρα μπαλαντέρ «_»

Εδώ, επιλέγουμε όλους τους πελάτες που μένουν σε Πόλη που ξεκινά με οποιονδήποτε χαρακτήρα ακολουθούμενο από “ονδίνο”:

```
SELECT * FROM Πελάτες
WHERE Πόλη LIKE ‘_ονδίνο';
```

Σε αυτό το παράδειγμα, επιλέγουμε ξανά όλους τους πελάτες από Πόλη που ξεκινά με ‘Λ’, ακολουθούμενο από οποιονδήποτε χαρακτήρα, ακολουθούμενο από "νδ", ακολουθούμενο από οποιονδήποτε χαρακτήρα, ακολουθούμενο από "νο":

```
SELECT * FROM Πελάτες
WHERE Πόλη LIKE ‘Λ_νδ_νο';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Χαρακτήρες Μπαλαντέρ στην SQL

Παραδείγματα με τον χαρακτήρα μπαλαντέρ [charlist]

Εδώ, επιλέγουμε όλους τους πελάτες από Πόλη που ξεκινά από «β», «σ» ή «π»:

```
SELECT * FROM Πελάτες
WHERE Πόλη LIKE ‘[βσπ]%';
```

Στο παρακάτω παράδειγμα, επιλέγουμε όλους τους πελάτες από Πόλη που να ξεκινά από «α», «β» ή «γ»:

```
SELECT * FROM Πελάτες
WHERE Πόλη LIKE '[α-γ]%';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Χαρακτήρες Μπαλαντέρ στην SQL

Παραδείγματα με τον χαρακτήρα μπαλαντέρ [!charlist]

Το θαυμαστικό εμφανίζει χαρακτήρες που δεν περιέχουν μια καθορισμένη συμβολοσειρά. Για παράδειγμα, θέλουμε να επιλέξουμε όλους τους πελάτες από Πόλη που δεν ξεκινά από «β», «σ» ή «π»:

```
SELECT * FROM Πελάτες
WHERE Πόλη LIKE ‘[!βσπ]%';
```

Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε τα ακόλουθα:

```
SELECT * FROM Πελάτες
WHERE Πόλη NOT LIKE ‘[βσπ]%';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής In στην SQL

Ο τελεστής IN **καθορίζει πολλαπλές τιμές σε έναν όρο WHERE**. Μπορεί να θεωρηθεί ότι πληροί διάφορες προϋποθέσεις.

Σύνταξη 1:

```
SELECT όνομα(τα)_στήλης(ών)
FROM όνομα_πίνακα
WHERE όνομα_στήλης IN (τιμή1, τιμή2, ...);
```

Σύνταξη 2:

```
SELECT όνομα(τα)_στήλης(ών)
FROM όνομα_πίνακα
WHERE όνομα_στήλης IN (εντολή SELECT);
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής In στην SQL

Ας υποθέσουμε ότι έχουμε έναν πίνακα που ονομάζεται «Πελάτες» που περιέχει τις ακόλουθες στήλες: ΚωδικόςΠελάτη, ΌνομαΠελάτη, ΌνομαΕπικοινωνίας, Διεύθυνση, Πόλη, Ταχυδρομικός Κώδικας και Χώρα (δείτε τον πίνακα εδώ).

Παράδειγμα 1: Επιλογή όλων των πελατών που βρίσκονται στη Γερμανία, τη Γαλλία ή το Ηνωμένο Βασίλειο

```
SELECT * FROM Πελάτες
WHERE Χώρα IN (‘Γερμανία’, ‘Γαλλία’, ‘Ηνωμένο Βασίλειο’)
```

Παράδειγμα 2: Επιλογή όλων των πελατών που δεν βρίσκονται στη Γερμανία, τη Γαλλία ή το Ηνωμένο Βασίλειο

```
SELECT * FROM Πελάτες
WHERE Χώρα NOT IN ( ‘Γερμανία’,  ‘Γαλλία’, ‘Ηνωμένο Βασίλειο’)
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής In στην SQL

Παράδειγμα 3: Επιλέξτε πελάτες που προέρχονται από τις ίδιες χώρες με τους προμηθευτές

```
SELECT * FROM Πελάτες
WHERE Χώρα IN (SELECT Χώρα FROM Προμηθευτές)
```

Εξάσκηση:

- Επιλέξτε πελάτες που βρίσκονται στην πόλη του Βερολίνου
- Επιλέξτε πελάτες που βρίσκονται στην πόλη του Λονδίνου και της Μαδρίτης
- Επιλέξτε πελάτες που δεν είναι από την ίδια χώρα με τους προμηθευτές

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής Between στην SQL

Ο τελεστής BETWEEN παρέχει ένα εύρος τιμών από τις οποίες μπορείτε να επιλέξετε. Οι τιμές μπορεί να είναι κείμενο, αριθμοί ή ημερομηνίες. Ο τελεστής BETWEEN **περιλαμβάνει τις τιμές έναρξης και λήξης.**

Σύνταξη:

```
SELECT όνομα(τα)_στήλης(ών)
FROM όνομα_πίνακα
WHERE όνομα_στήλης BETWEEN τιμή1 AND τιμή2;
```

Θα υπάρχει μια σειρά παραδειγμάτων με τους ακόλουθους τελεστές: BETWEEN, NOT BETWEEN, BETWEEN με IN, BETWEEN και NOT BETWEEN με τιμές κειμένου και ημερομηνίες.

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής Between στην SQL

Παράδειγμα BETWEEN: Εμφάνιση όλων των προϊόντων με εύρος τιμών 10 έως 20.

```
SELECT * FROM Πρόϊόντα
WHERE Τιμή BETWEEN 10 AND 20;
```

Παράδειγμα NOT BETWEEN: Εμφάνιση όλων των προϊόντων εκτός της σειράς που ορίσαμε στο προηγούμενο παράδειγμα.

```
SELECT * FROM Προϊόντα
WHERE Τιμή NOT BETWEEN 10 AND 20;
```

Παράδειγμα BETWEEN με IN: Εμφάνιση όλων των προϊόντων με εύρος τιμών 10 έως 20 και δεν εμφανίζει προϊόντα με αναγνωριστικό κατηγορίας 1, 2 ή 3.

```
SELECT * FROM Προϊόντα
WHERE Τιμή BETWEEN 10 AND 20
AND ΑριθμόςΚατηγορίας NOT IN (1,2,3);
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής Between στην SQL

Παράδειγμα BETWEEN με τιμές κειμένου: Εμφάνιση όλων των πρόϊόντων με Όνομα προϊόντος μεταξύ Carnarvon Tigers και Mozzarella di Giovanni.

```
SELECT * FROM Προϊόντα
WHERE ΌνομαΠροϊόντος BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni'
ORDER BY ΌνομαΠροϊόντος;
```

Παράδειγμα NOT BETWEEN με τιμές κειμένου: Εμφάνιση όλων των προϊόντων με Όνομα προϊόντος που δεν είναι μεταξύ Carnarvon Tigers και Mozzarella di Giovanni.

```
SELECT * FROM Προϊόντα
WHERE ΌνομαΠροϊόντος NOT BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni'
ORDER BY ΌνομαΠροϊόντος;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Τελεστής Between στην SQL

Παράδειγμα Between σε ημερομηνίες: Εμφάνιση όλων των παραγγελιών με ημερομηνία παραγγελίας μεταξύ '01-Ιουλίου-1996' και '31-Ιουλίου-1996' (για να βρείτε τον πίνακα που χρησιμοποιείται σε αυτό το παράδειγμα, μεταβείτε εδώ)

Υπάρχουν δύο τρόποι για να γίνει αυτό, χρησιμοποιώντας είτε ένα hashtag (#) ή εισαγωγικά (‘’):

```
SELECT * FROM Παραγγελίες
WHERE ΗμερομηνίαΠαραγγελίας BETWEEN #07/01/1996# AND #07/31/1996#;
```

Ή

```
SELECT * FROM Παραγγελίες
WHERE ΗμερομηνίαΠαραγγελίας BETWEEN '1996-07-01' AND '1996-07-31';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ψευδώνυμα στην SQL (Aliases)

Τα ψευδώνυμα δίνουν ένα **προσωρινό όνομα σε έναν πίνακα ή μια στήλη μέσα σε έναν πίνακα.** Ένα ψευδώνυμο υπάρχει μόνο για τη διάρκεια ενός ερωτήματος και χρησιμοποιείται συνήθως για να κάνει τα ονόματα των στηλών πιο ευανάγνωστα. Ένα ψευδώνυμο δημιουργείται χρησιμοποιώντας τη λέξη-κλειδί AS.

Σύνταξη για ψευδώνυμο στήλης:

```
SELECT όνομα_στήλης AS όνομα_ψευδώνυμου
FROM όνομα_πίνακα;
```

Σύνταξη για ψευδώνυμο πίνακα:

```
SELECT όνομα(τα)_στήλης(ών)
FROM όνομα_πίνακα AS όνομα_ψευδώνυμου;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ψευδώνυμα στην SQL (Aliases)

Ψευδώνυμα στηλών

Ας δούμε ένα παράδειγμα που δημιουργεί δύο ψευδώνυμα, ένα για κάθε στήλη:

```
SELECT ΚωδικόςΠελάτη AS ID, ΌνομαΠελάτη AS Πελάτης
FROM Πελάτες;
```

Ένα άλλο παράδειγμα δημιουργεί και πάλι δύο ψευδώνυμα:

```
SELECT ΌνομαΠελάτη AS Πελάτης, ΌνομαΕπικοινωνίας AS [Άτομο Επικοινωνίας]
FROM Πελάτες;
```

***Σημειώστε ότι τοποθετείται σε αγκύλες ([ ]) επειδή το ψευδώνυμο περιέχει κενά. Τα εισαγωγικά μπορούν επίσης να χρησιμοποιηθούν αντί για τις αγκύλες.**

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ψευδώνυμα στην SQL (Aliases)

Έχετε επίσης την επιλογή να δημιουργήσετε ένα ψευδώνυμο που περιέχει μία ή περισσότερες στήλες. Ας δούμε το παρακάτω παράδειγμα για να δούμε πώς λειτουργεί:

```
SELECT ΌνομαΠελάτη, Διεύθυνση + ', ' + ΤαχυδρομικόςΚώδικας + ' ' + Πόλη + ', ' + Χώρα AS Διεύθυνση
FROM Πελάτες;
```

Η παραπάνω εντολή αλλάζει λίγο στο MySQL:

```
SELECT ΌνομαΠελάτη, CONCAT(Διεύθυνση,', ',ΤαχυδρομικόςΚώδικας,', ',Πόλη,', ',Χώρα) AS Διεύθυνση
FROM Πελάτες;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ψευδώνυμα στην SQL (Aliases)

Ψευδώνυμα Πίνακα: Το παρακάτω παράδειγμα επιλέγει όλες τις παραγγελίες από τον πίνακα πελατών με ΚωδικόΠελάτη =4 δίνοντας το ψευδώνυμο “Around the Horn”.

Εδώ χρησιμοποιούνται ψευδώνυμα για τη συντόμευση του ερωτήματος:

```
SELECT o.ΚωδικόςΠαραγγελίας, o.ΗμερομηνίαΠαραγγελίας, c.ΌνομαΠελάτη
FROM Πελάτες AS c, Παραγγελίες AS o
WHERE c.Όνομα Πελάτη='Around the Horn' AND c.ΚωδικόςΠελάτη=o.ΚωδικόςΠελάτη;
```

Ένα ερώτημα χωρίς ψευδώνυμα θα έμοιαζε κάπως έτσι:

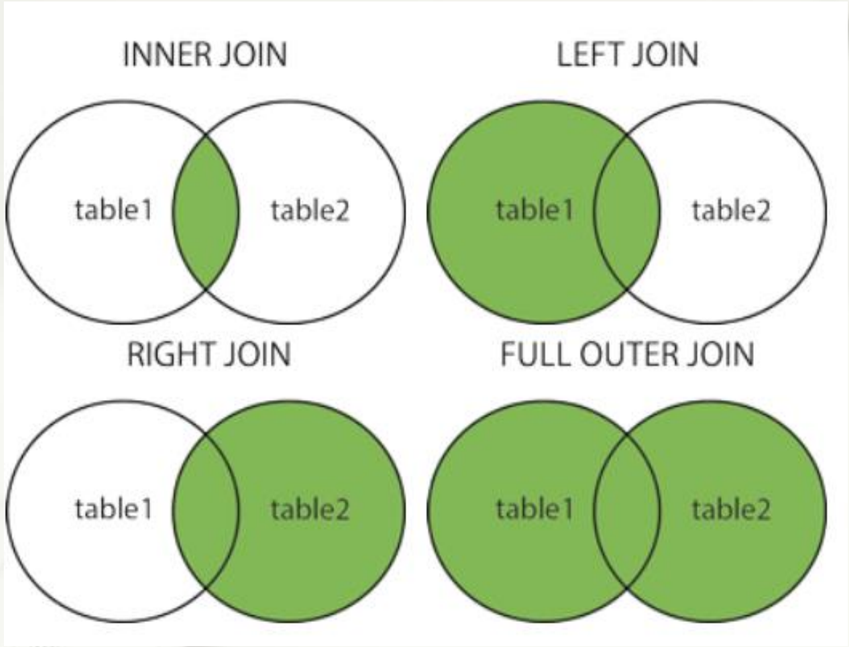
```
SELECT Παραγγελίες.ΚωδικόςΠαραγγελίας, Παραγγελίες.ΗμερομηνίαΠαραγγελίας, Πελάτες.ΌνομαΠελάτη
FROM Πελάτες, Παραγγελίες
WHERE Πελάτες.ΌνομαΠελάτη='Around the Horn' AND
Πελάτες.ΚωδικόςΠελάτη=Παραγγελίες.ΚωδικόςΠελάτη;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Συνενώσεις στην SQL (JOIN)

Μια εντολή JOIN **συνδυάζει σειρές από δύο ή περισσότερους πίνακες με βάση μια σχετική στήλη που βρίσκεται και στους δύο πίνακες.** Υπάρχουν τέσσερις διαφορετικές εντολές JOIN στην SQL:

1. *(INNER) JOIN:* Επιλέγει εγγραφές που έχουν αντίστοιχες τιμές και στους δύο πίνακες.
2. *LEFT (OUTER) JOIN:* Επιλέγει όλες τις εγγραφές από τον πίνακα στα αριστερά και τις αντίστοιχες αντιστοιχισμένες εγγραφές από τον πίνακα στα δεξιά.
3. *RIGHT (OUTER) JOIN:* Επιλέγει όλες τις εγγραφές από τον πίνακα στα δεξιά και τις αντίστοιχες αντιστοιχισμένες εγγραφές από τον πίνακα στα αριστερά.
4. *FULL (OUTER) JOIN:* Επιλέγει όλες τις εγγραφές όταν υπάρχει αντιστοιχία είτε στον πίνακα στα αριστερά ή στα δεξιά.



Εικόνα - Διαφορετικοί τύποι εντολών JOIN
(Πηγή: https://www.w3schools.com/sql/sql_join.asp)

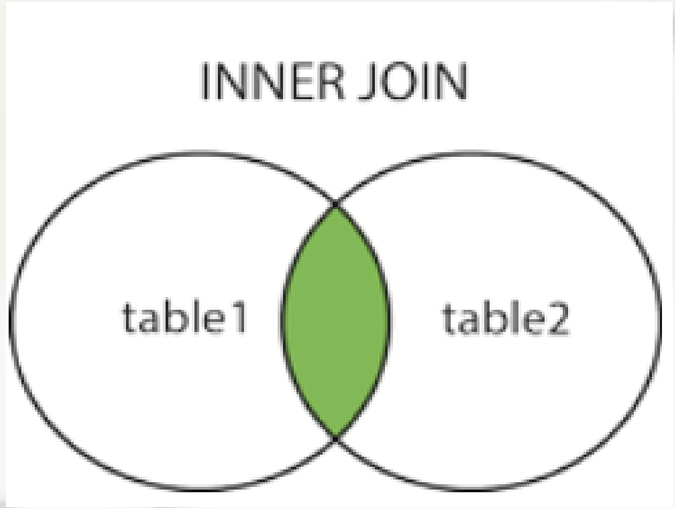
Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Συνένωση Inner join στην SQL

Η λέξη-κλειδί INNER JOIN επιλέγει εγγραφές που έχουν ίδιες τιμές και από τους δύο πίνακες.

Σύνταξη:

```
SELECT όνομα(τα)_στήλης(ών)
FROM πίνακας1
INNER JOIN πίνακας2
ON πίνακας1.όνομα_στήλης = πίνακας2.όνομα_στήλης;
```



Εντολή Inner join

(Πηγή: https://www.w3schools.com/sql/sql_join.asp)

Για να ανακτήσετε τα ονόματα των πελατών και τους αντίστοιχους κωδικούς παραγγελίας τους:

```
SELECT Παραγγελίες.ΚωδικόςΠαραγγελίας, Πελάτες.ΌνομαΠελάτη
FROM Παραγγελίες
INNER JOIN Πελάτες ON Παραγγελίες.ΚωδικόςΠελάτη = Πελάτες.ΚωδικόςΠελάτη;
```

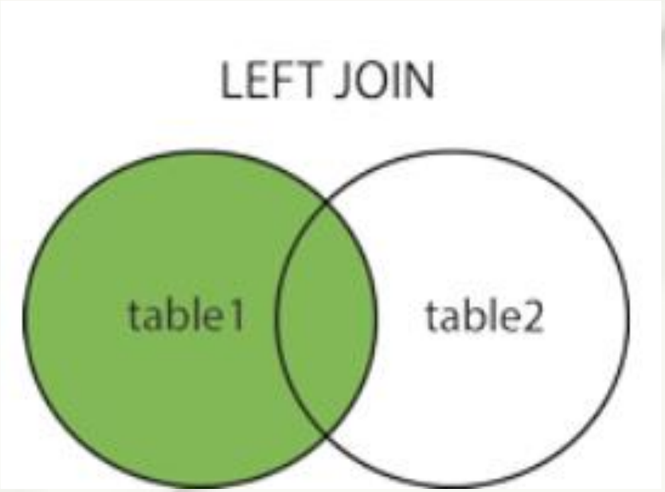
Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Συνένωση Left Join στην SQL

Η λέξη-κλειδί LEFT JOIN επιλέγει όλες τις εγγραφές από τον πίνακα στα αριστερά και τις αντίστοιχες εγγραφές από τον πίνακα στα δεξιά.

Σύνταξη:

```
SELECT όνομα(τα)_στήλης(ών)
FROM πίνακας1
LEFT JOIN πίνακας2
ON πίνακας1.όνομα_στήλης = πίνακας2.όνομα_στήλης;
```



Εντολή Left join
(Πηγή: https://www.w3schools.com/sql/sql_join.asp)

Επιλέξτε όλους τους πελάτες και τυχόν παραγγελίες που μπορεί να έχουν:

```
SELECT Πελάτες.ΌνομαΠελάτη, Παραγγελίες.ΚωδικόςΠαραγγελίας
FROM Πελάτες
LEFT JOIN Παραγγελίες ON Πελάτες.ΚωδικόςΠελάτη = Παραγγελίες.ΚωδικόςΠελάτη
ORDER BY Πελάτες.ΌνομαΠελάτη;
```

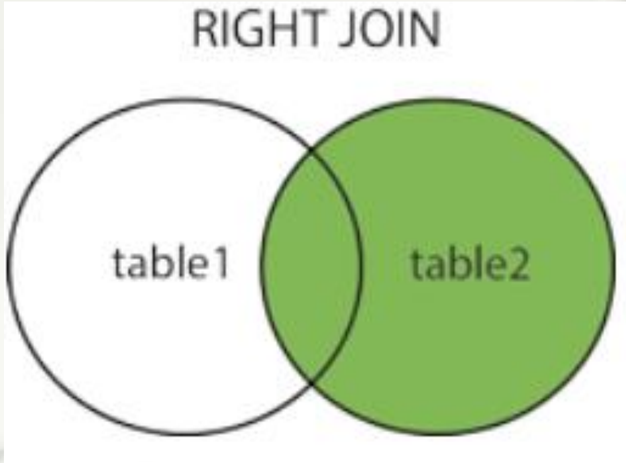
Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Συνένωση Right join στην SQL

Η λέξη-κλειδί RIGHT JOIN ακολουθεί ουσιαστικά την ίδια λογική ξεκινώντας από τη δεξιά πλευρά αντί για την αριστερή όπως περιγράφεται στην προηγούμενη υποενότητα.

Σύνταξη:

```
SELECT όνομα(τα)_στήλης(ών)
FROM πίνακας1
RIGHT JOIN πίνακας2
ON πίνακας1.όνομα_στήλης = πίνακας2.όνομα_στήλης;
```



Εντολή Right join

(Πηγή: https://www.w3schools.com/sql/sql_join.asp)

Επιλέγει όλους τους εργαζομένους και τυχόν παραγγελίες που μπορεί να έχουν κάνει:

```
SELECT Παραγγελίες.ΚωδικόςΠαραγγελίας, Εργαζόμενοι.Επίθετο, Εργαζόμενοι.Όνομα
FROM Παραγγελίες
RIGHT JOIN Εργαζόμενοι ON Παραγγελίες.ΚωδικόςΕργαζομένου =
Εργαζόμενοι.ΚωδικόςΕργαζομένου
ORDER BY Παραγγελίες.ΚωδικόςΠαραγγελίας;
```

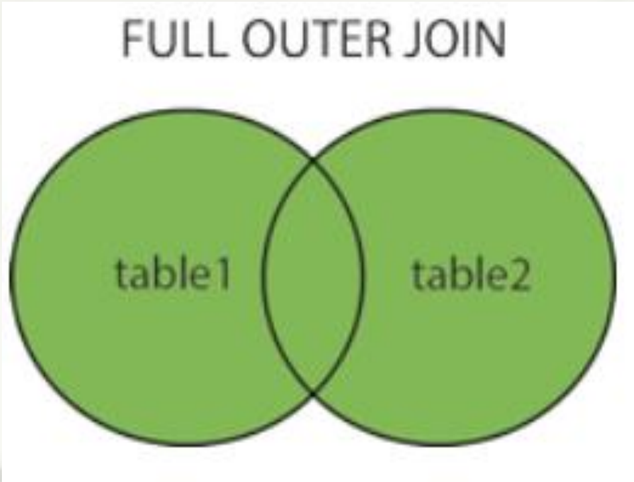
Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Συνένωση Full Join στην SQL

Η λέξη-κλειδί FULL JOIN επιλέγει όλες τις εγγραφές όταν οι εγγραφές που είναι ίδιες βρίσκονται είτε στον πίνακα στα δεξιά είτε στα αριστερά.

Σύνταξη:

```
SELECT όνομα(τα)_στήλης(ών)
FROM πίνακας1
FULL OUTER JOIN πίνακας2
ON πίνακας1.όνομα_στήλης = πίνακας2.όνομα_στήλης
WHERE συνθήκη;
```



Εντολή Full (outer) join
(Πηγή: https://www.w3schools.com/sql/sql_join.asp)

* Σημειώστε ότι επιλέγει όλες τις εγγραφές που ταιριάζουν και από τους δύο πίνακες, ακόμη και αν δεν υπάρχουν κοινές αντιστοιχίες μεταξύ των δύο πινάκων. Σε περίπτωση που δεν υπάρχουν κοινές αντιστοιχίσεις, ορίζεται μηδενική τιμή.

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Συνένωση Self Join στην SQL

Η συνένωση SELF-JOIN θεωρείται ως μια κανονική συνένωση JOIN, ωστόσο οι εγγραφές στον πίνακα συσχετίζονται αυτόματα.

Σύνταξη:

SELECT όνομα(τα)_στήλης(ών)

FROM πίνακας1 T1, πίνακας1 T2 *(Τα T1 καιT2 είναι ψευδώνυμα που χρησιμοποιούνται για τον ίδιο πίνακα)*

WHERE συνθήκη;

Παράδειγμα: Επιλέξτε πελάτες που προέρχονται από την ίδια πόλη

```
SELECT A.ΌνομαΠελάτη AS ΌνομαΠελάτη1, B.ΌνομαΠελάτη AS ΌνομαΠελάτη2, A.Πόλη
FROM Πελάτες A, Πελάτες B
WHERE A.ΚωδικόςΠελάτη <> B.ΚωδικόςΠελάτη
AND A.Πόλη = B.Πόλη
ORDER BY A.Πόλη;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ένωση στην SQL (UNION)

Ο τελεστής UNION χρησιμοποιείται για να συνδυάσει το σύνολο δύο ή περισσότερων εντολών SELECT.

Υπάρχουν ορισμένες **απαιτήσεις** για να καταστεί δυνατός ένας τελεστής UNION:

1. Κάθε εντολή SELECT εντός του τελεστή UNION πρέπει να έχει τον ίδιο αριθμό στηλών

2. Οι στήλες πρέπει να έχουν παρόμοιους τύπους δεδομένων.

3. Οι στήλες σε κάθε εντολή SELECT πρέπει να είναι στην ίδια σειρά.

Σύνταξη:

```
SELECT όνομα_στήλης(ών) FROM πίνακας1
UNION
SELECT όνομα_στήλης(ών) FROM πίνακας2;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ένωση στην SQL (UNION)

Σημειώστε ότι η λειτουργία **UNION επιλέγει μόνο διακριτές τιμές από προεπιλογή.**

Για να επιτρέψετε τις διπλές τιμές, χρησιμοποιήστε την λειτουργία UNION ALL:

```
SELECT όνομα(τα)_στήλης(ών) FROM πίνακας1
UNION ALL
SELECT όνομα(τα)_στήλης(ών) FROM πίνακας2;
```

Σημειώστε ότι τα ονόματα των στηλών στις δύο εντολές SELECT είναι συνήθως ίσα.

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ένωση στην SQL (UNION)

Παράδειγμα 1: Επιλογή ξεχωριστών πόλεων

```
SELECT Πόλη FROM Πελάτες
UNION
SELECT Πόλη FROM Προμηθευτές
ORDER BY Πόλη;
```

Παράδειγμα 2: Επιλογή διπλών τιμών

```
SELECT Πόλη FROM Πελάτες
UNION ALL
SELECT Πόλη FROM Προμηθευτές
ORDER BY Πόλη;
```

Παράδειγμα 3: Επιλογή των ξεχωριστών γερμανικών πόλεων από τους πίνακες «Πελάτες» και «Προμηθευτές» με τη χρήση του όρου WHERE:

```
SELECT Πόλη, Χώρα FROM Πελάτες
WHERE Χώρα='Γερμανία'
UNION
SELECT Πόλη, Χώρα FROM Πελάτες
WHERE Χώρα = 'Γερμανία'
ORDER BY Πόλη;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ομαδοποίηση στην SQL (Group By)

Η εντολή GROUP BY ομαδοποιεί τις σειρές με τις ίδιες τιμές σε συνοπτικές γραμμές. Για παράδειγμα, σκεφτείτε ότι θέλετε να βρείτε τον αριθμό των πελατών σε κάθε χώρα.

Επίσης, η εντολή GROUP BY χρησιμοποιείται συχνά με αθροιστικές συναρτήσεις όπως COUNT(), MAX(), MIN(), SUM(), AVG() για την ομαδοποίηση του αποτελέσματος κατά μία ή περισσότερες στήλες.

Σύνταξη:

```
SELECT όνομα_στήλης(ών)
FROM όνομα_πίνακα
WHERE συνθήκη
GROUP BY όνομα_στήλης(ών)
ORDER BY όνομα_στήλης(ών);
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ομαδοποίηση στην SQL (Group By)

Παράδειγμα 1: Λίστες με τον αριθμό των πελατών που βρέθηκαν σε κάθε χώρα

```
SELECT COUNT(ΚωδικόςΠελάτη), Χώρα
FROM Πελάτες
GROUP BY Χώρα;
```

Παράδειγμα 2: Απαριθμεί τον αριθμό των πελατών σε κάθε χώρα, αλλά με φθίνουσα σειρά.

```
SELECT COUNT(ΚωδικόςΠελάτη), Χώρα
FROM Πελάτες
GROUP BY Χώρα;
ORDER BY COUNT(ΚωδικόςΠελάτη) DESC;
```

Παράδειγμα 3: Απαριθμεί τον αριθμό των παραγγελιών που αποστέλλονται από κάθε αποστολέα, συγχωνεύοντας δύο πίνακες

```
SELECT Αποστολέας.ΌνομαΑποστολέα,
COUNT(Παραγγελίες.ΚωδικόςΠαραγγελίας) AS
ΑριθμόςΠαραγγελιών
FROM Παραγγελίες
LEFT JOIN Αποστολείς ON
Παραγγελίες.ΚωδικόςΑποστολέα=
Αποστολείς.ΚωδικόςΑποστολέα
GROUP BY ΌνομαΑποστολέα;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ο όρος HAVING στην SQL

Ο όρος HAVING προστέθηκε στην SQL επειδή ο όρος WHERE δεν μπορεί να χρησιμοποιηθεί με αθροιστικές συναρτήσεις.

Σύνταξη:

```
SELECT όνομα_στήλης(ών)
FROM όνομα_πίνακα
WHERE συνθήκη
GROUP BY όνομα_στήλης(ών)
HAVING συνθήκη
ORDER BY όνομα_στήλης(ών);
```

Παράδειγμα: Απαριθμεί τον αριθμό των πελατών που βρίσκονται σε κάθε χώρα και χώρες στις οποίες έχουν περισσότερους από πέντε πελάτες

```
SELECT COUNT(ΚωδικόςΠελάτη), Χώρα
FROM Πελάτες
GROUP BY Χώρα
HAVING COUNT(ΚωδικόςΠελάτη) > 5;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ο όρος HAVING στην SQL

Παράδειγμα 1: Απαριθμεί τους εργαζόμενους με επίθετο «Νταβόλιο» ή «Φούλερ» εάν έχουν καταχωρίσει παραγγελίες περισσότερες από 25 φορές:

```
SELECT Εργαζόμενοι.Επίθετο,
COUNT(Παραγγελίες.ΚωδικόςΠαραγγελίας) AS
ΑριθμόςΠαραγγελιών
FROM (Παραγγελίες
INNER JOIN Εργαζόμενοι ON
Παραγγελίες.ΚωδικόςΕργαζομένου =
Εργαζόμενοι.ΚωδικόςΕργαζομένων)
WHERE Επίθετο = ‘Νταβόλιο’ OR Επίθετο = ‘Φούλερ’
GROUP BY Επίθετο
HAVING COUNT(Παραγγελίες.ΚωδικόςΠαραγγελιών) > 25;
```

Παράδειγμα 2: Απαριθμεί τους εργαζόμενους που έχουν καταχωρίσει περισσότερες από δέκα παραγγελίες, συγχωνεύοντας πληροφορίες από δύο πίνακες

```
SELECT Εργαζόμενοι.Επίθετο,
COUNT(Παραγγελίες.ΚωδικόςΠαραγγελιών) AS
ΑριθμόςΠαραγγελιών
FROM (Παραγγελίες
INNER JOIN Εργαζόμενοι ON
Παραγγελίες.ΚωδικόςΕργαζομένου =
Εργαζόμενοι.ΚωδικόςΕργαζομένου)
GROUP BY Επίθετο
HAVING COUNT(Παραγγελίες.ΚωδικόςΠαραγγελιών) > 10;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Εντολή Select Into στην SQL

Η εντολή SELECT INTO αντιγράφει τα δεδομένα από έναν πίνακα σε έναν νέο πίνακα.

Σύνταξη για αντιγραφή όλων των στηλών σε νέο πίνακα:

```
SELECT *
INTO νέοςπίνακας [IN externaldb]
FROM παλιόςπίνακας
WHERE συνθήκη;
```

Σύνταξη για αντιγραφή μόνο μερικών στηλών σε νέο πίνακα:

```
SELECT στήλη1, στήλη2, ...
INTO νέοςπίνακας [IN externaldb]
FROM παλιόςπίνακας
WHERE συνθήκη;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Εντολή Select Into στην SQL

Παράδειγμα δημιουργίας εφεδρικού αντιγράφου της στήλης Πελατών:

```
SELECT * INTO ΑντίγραφοΠελατών2017
FROM Πελάτες;
```

Παράδειγμα χρήσης του όρου IN για την αντιγραφή του πίνακα σε νέο πίνακα σε άλλη βάση δεδομένων:

```
SELECT * INTO ΑντίγραφοΠελατών2017 IN 'Backup.mdb'
FROM Πελάτες;
```

Παράδειγμα αντιγραφής μόνο μερικών στηλών σε νέο πίνακα:

```
SELECT ΌνομαΠελάτη, ΌνομαΕπικοινωνίας INTO ΑντίγραφοΠελατών2017
FROM Πελάτες;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Εντολή Select Into στην SQL

Παράδειγμα αντιγραφής μόνο των Γερμανών πελατών σε νέο πίνακα:

```
SELECT * INTO ΠελάτεςΓερμανία
FROM Πελάτες
WHERE Χώρα = ‘Γερμανία';
```

Παράδειγμα αντιγραφής δεδομένων από πολλαπλούς πίνακες σε νέο πίνακα:

```
SELECT Πελάτες.ΌνομαΠελάτη, Παραγγελία.ΚωδικόςΠαραγγελίας
INTO ΑντίγραφοΠαραγγελιώνΠελατών2017
FROM Πελάτες
LEFT JOIN Παραγγελίες ON Πελάτες.ΚωδικόςΠελάτη=
Παραγγελίες.ΚωδικόςΠελάτη;
```

Η εντολή SELECT INTO μπορεί επίσης να χρησιμοποιηθεί για τη δημιουργία ενός νέου, κενού πίνακα χρησιμοποιώντας το σχήμα ενός άλλου.

Για να το κάνετε αυτό, προσθέστε έναν όρο WHERE που δεν εμφανίζει δεδομένα:

```
SELECT * INTO νέοςπίνακας
FROM παλιόςπίνακας
WHERE 1 = 0;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Η εντολή Insert Into Select στην SQL

Η εντολή INSERT INTO SELECT αντιγράφει δεδομένα από έναν πίνακα και τα εισάγει σε έναν άλλον. Απαιτεί την αντιστοίχιση των τύπων δεδομένων στον πίνακα πηγής και στόχου.

Σύνταξη για αντιγραφή όλων των στηλών από έναν πίνακα σε έναν άλλο:

```
INSERT INTO πίνακας2
SELECT * FROM πίνακας1
WHERE συνθήκη;
```

Σύνταξη για αντιγραφή μόνο μερικών στηλών από έναν πίνακα σε έναν άλλο:

```
INSERT INTO πίνακας2 (στήλη1, στήλη2, στήλη3, ...)
SELECT στήλη1, στήλη2, στήλη3, ...
FROM πίνακας1
WHERE συνθήκη;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Η εντολή Insert Into Select στην SQL

Παράδειγμα 1: Αντιγράφει Προμηθευτές σε Πελάτες (σημειώστε ότι οι στήλες που δεν έχουν δεδομένα θα περιέχουν μηδενικές τιμές)

```
INSERT INTO Πελάτες (ΌνομαΠελάτη, Πόλη, Χώρα)
SELECT ΌνομαΠρομηθευτή, Πόλη, Χώρα FROM Προμηθευτές;
```

Παράδειγμα 2: Αντιγράφει Προμηθευτές σε Πελάτες για να συμπληρώσει όλες τις στήλες:

```
INSERT INTO Πελάτες (ΌνομαΠελάτη, ΌνομαΕπικοινωνίας, Διεύθυνση, Πόλη, ΤαχυδρομικόςΚώδικας, Χώρα)
SELECT ΌνομαΠρομηθευτή, ΌνομαΕπικοινωνίας, Διεύθυνση, Πόλη, ΤαχυδρομικόςΚώδικας, Χώρα FROM Προμηθευτές;
```

Παράδειγμα 3: Αντιγράφει μόνο τους Γερμανούς προμηθευτές σε Πελάτες:

```
INSERT INTO Πελάτες (ΌνομαΠελάτη, Πόλη, Χώρα)
SELECT ΌνομαΠρομηθευτή, Πόλη, Χώρα FROM Προμηθευτές
WHERE NOT Χώρα='Γερμανία';
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Εντολή Case στην SQL

Η εντολή CASE περνά από μια σειρά προϋποθέσεων και επιστρέφει μια τιμή όταν πληρούται η πρώτη προϋπόθεση. Σκεφτείτε το σαν μια εντολή με τα if, then, else.

Όταν ένας όρος ισχύει, θα σταματήσει να περνάει μέσα από τον βρόχο. Εάν δεν ισχύει, θα εμφανίσει την τιμή στον όρο ELSE.

***Σημειώστε ότι αν δεν υπάρχει ο όρος ELSE και δεν υπάρχουν όροι που να ισχύουν, θα εμφανίσει το NULL.**

Σύνταξη:

```
CASE
WHEN συνθήκη1 THEN αποτέλεσμα1
WHEN συνθήκη2 THEN αποτέλεσμα2
WHEN συθήκηN THEN αποτέλεσμαN
ELSE αποτέλεσμα
END;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Εντολή Case στην SQL

Παράδειγμα 1: Περνάει από μια σειρά συνθηκών και εμφανίζει μια τιμή όταν πληρούται η πρώτη συνθήκη

```
SELECT ΚωδικόςΠαραγγελίας, Ποσότητα,
CASE
WHEN Ποσότητα > 30 THEN ‘Ο ποσότητα είναι μεγαλύτερη από 30'
WHEN Ποσότητα = 30 THEN ‘Η ποσότητα είναι ίση με 30’
ELSE ‘Η ποσότητα είναι μικρότερη από 30'
END AS ΑριθμόςΠοσότητας
FROM ΛεπτομέριεςΠαραγγελίας;
```

Παράδειγμα 2: Παραγγελίες πελατών ανά πόλη. Αν η Πόλη έχει την τιμή NULL, θα ταξινομηθούν ανά Χώρα.

```
SELECT ΌνομαΠελάτη, Πόλη, Χώρα
FROM Πελάτες
ORDER BY
(CASE
WHEN Πόλη IS NULL THEN Χώρα
ELSE Πόλη
END);
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Λειτουργίες κενών τιμών στην SQL (Null Functions)

Οι λειτουργίες NULL περιλαμβάνουν τα ακόλουθα: IFNULL(), ISNULL(), COALESCE(), and NVL().

Θεωρήστε ότι η στήλη «ΤεμάχιαΠαραγγελίας» είναι προαιρετική και μπορεί να περιέχει κενές τιμές (NULL).

Παράδειγμα:

```
SELECT ΌνομαΠροϊόντος, ΤιμήΤεμαχίου * (ΤεμάχιαΣεΑπόθεμα + ΤεμάχιαΠαραγγελίας)
FROM Προϊόντα;
```

Εδώ μπορούμε να δούμε ότι εάν οποιαδήποτε από τις τιμές ΤεμάχιαΠαραγγελίας είναι κενή, το αποτέλεσμα θα είναι επίσης κενό (NULL).

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Λειτουργίες κενών τιμών στην SQL (Null Functions)

Στο σύστημα MySQL, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση ISNULL() που μας επιτρέπει να επιλέξουμε μια εναλλακτική τιμή αν μια συνάρτηση είναι μηδενική:

```
SELECT ΌνομαΠροϊόντος, ΤιμήΤεμαχίου * (ΤεμάχιαΣεΑπόθεμα + IFNULL(ΤεμάχιαΠαραγγελίας, 0))
FROM Προϊόντα;
```

Ή μπορούμε να χρησιμοποιήσουμε τη λειτουργία COALESCE() :

```
SELECT ΌνομαΠροϊόντος, ΤιμήΤεμαχίου * (ΤεμάχιαΣεΑπόθεμα + COALESCE(ΤεμάχιαΠαραγγελίας, 0))
FROM Προϊόντα;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Λειτουργίες κενών τιμών στην SQL (Null Functions)

Στον διακομιστή SQL, η λειτουργία ISNULL() κάνει το ίδιο πράγμα όπως στο MySQL:

```
SELECT ΌνομαΠροϊόντος, ΤιμήΤεμαχίου * (ΤεμάχιαΣεΑπόθεμα + IFNULL(ΤεμάχιαΠαραγγελίας, 0))
FROM Προϊόντα;
```

Στη λειτουργία IsNull() στο MS Access η συνάρτηση TRUE(-1) προκύπτει αν η έκφραση είναι μια κενή τιμή, αλλιώς προκύπτει η συνάρτηση FALSE (0):

```
SELECT ΌνομαΠροϊόντος, ΤιμήΤεμαχίου * (ΤεμάχιαΣεΑπόθεμα + IIF(IsNull(ΤεμάχιαΠαραγγελίας), 0, ΤεμάχιαΠαραγγελίας))
FROM Προϊόντα;
```

Στο σύστημα Oracle, η λειτουργία NVL() κάνει το ίδιο πράγμα:

```
SELECT ΌνομαΠροϊόντος, ΤιμήΤεμαχίου * (ΤεμάχιαΣεΑπόθεμα + NVL(ΤεμάχιαΠαραγγελίας, 0))
FROM Προϊόντα;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Σχόλια στην SQL (Comments)

Τα σχόλια στην **SQL εξηγούν τις ενότητες των εντολών στην SQL ή εμποδίζουν την εκτέλεσή τους.**

Τα σχόλια μιας γραμμής στην SQL ξεκινούν με - - (δύο παύλες):

```
--Select all:

SELECT * FROM Πελάτες;
```

***Σημειώστε ότι τα παραδείγματα σε αυτήν την ενότητα δεν υποστηρίζονται στο Firefox και το Microsoft Edge, τα οποία είναι βάσεις δεδομένων της Microsoft Access. Τα σχόλια (comments) γενικά δεν υποστηρίζονται στις βάσεις δεδομένων της Microsoft Access.**

Ή μπορεί να χρησιμοποιηθεί με αυτόν τον τρόπο για να αγνοήσει το τέλος της γραμμής:

```
SELECT * FROM Πελάτες -- WHERE Πόλη=‘Βερολίνο';
```

Ή για να αγνοήσετε μια πρόταση:

```
--SELECT * FROM Πελάτες;

SELECT * FROM Προϊόντα;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Σχόλια στην SQL (Comments)

Τα σχόλια πολλαπλών γραμμών ξεκινούν με /* και τελειώνουν με */. Οποιοδήποτε κείμενο γραφτεί μεταξύ αυτών των δύο θα αγνοηθεί.

Παράδειγμα 1:

```
/*Επιλογή όλων των στηλών
όλων των εγγραφών
στον Πίνακα Πελατών:*/
SELECT * FROM Πελάτες;
```

Για να αγνοήσετε μέρος μιας πρότασης, μπορείτε επίσης να χρησιμοποιήσετε το/**/.

Παράδειγμα 2:

```
SELECT ΌνομαΠελάτη, /*Πόλη,*/ Χώρα FROM Πελάτες;
```

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

Ας εξασκηθούμε

Έχετε μάθει πολλά νέα πράγματα μέχρι τώρα, οπότε ήρθε η ώρα να εφαρμόσουμε αυτά που μάθαμε στην πράξη!

Για να το κάνετε αυτό, κάντε κλικ εδώ.

Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

code4sp
coding for social promotion

ΕΥΧΑΡΙΣΤΟΥΜΕ!

ΕΠΟΜΕΝΟ ΚΕΦΑΛΑΙΟ: Βάση δεδομένων SQL

Με συγχρηματοδότηση από το
πρόγραμμα «Erasmus+»
της Ευρωπαϊκής Ένωσης