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

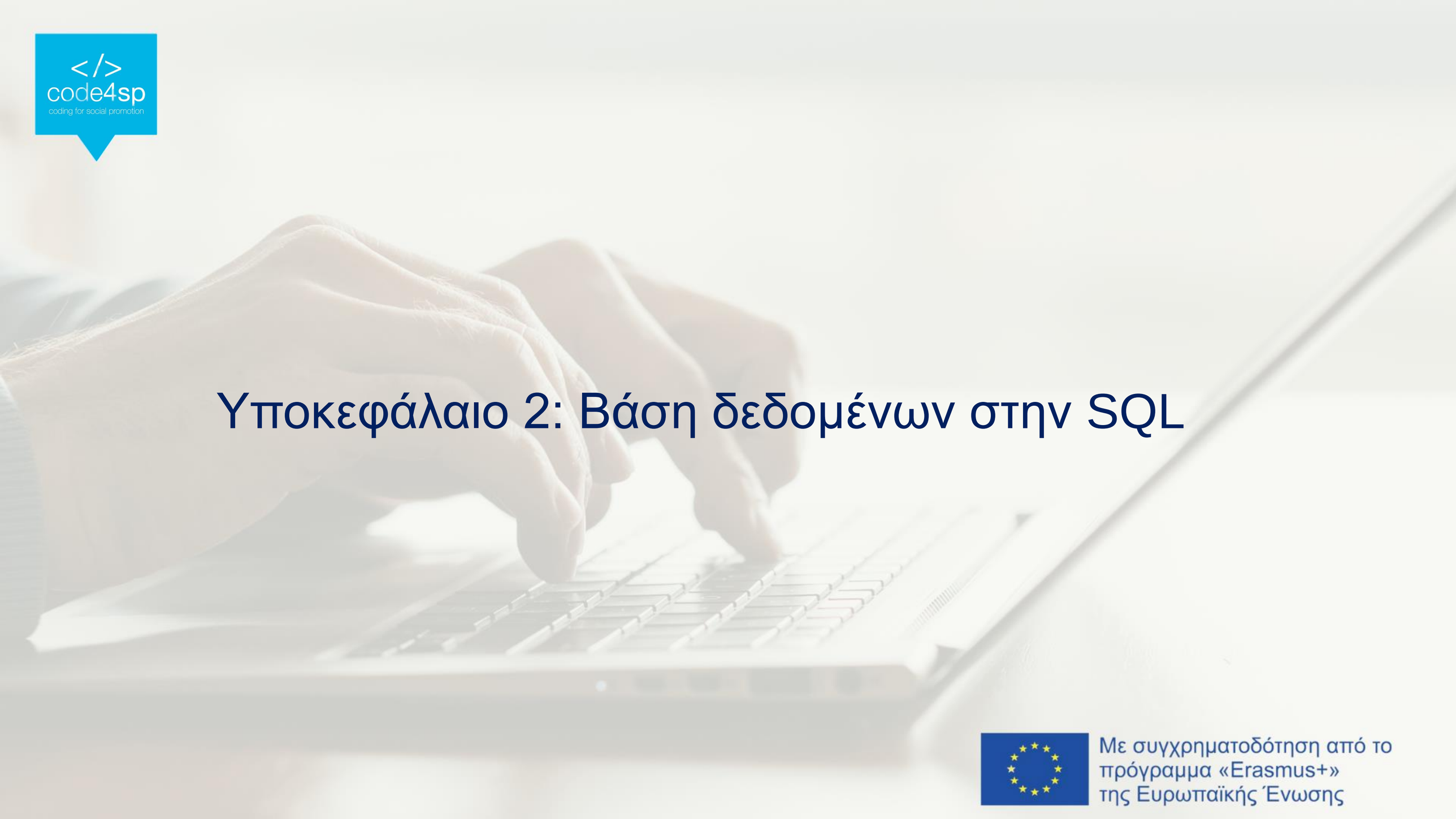
code4sp
coding for social promotion

Εκπαιδευτικό Υλικό SQL
Υποκεφάλαιο 2 – Βάσεις δεδομένων SQL

WP3: Εκπαιδευτικό Υλικό του έργου Code4SP

Συντάχθηκε από: CITIZENS IN POWER

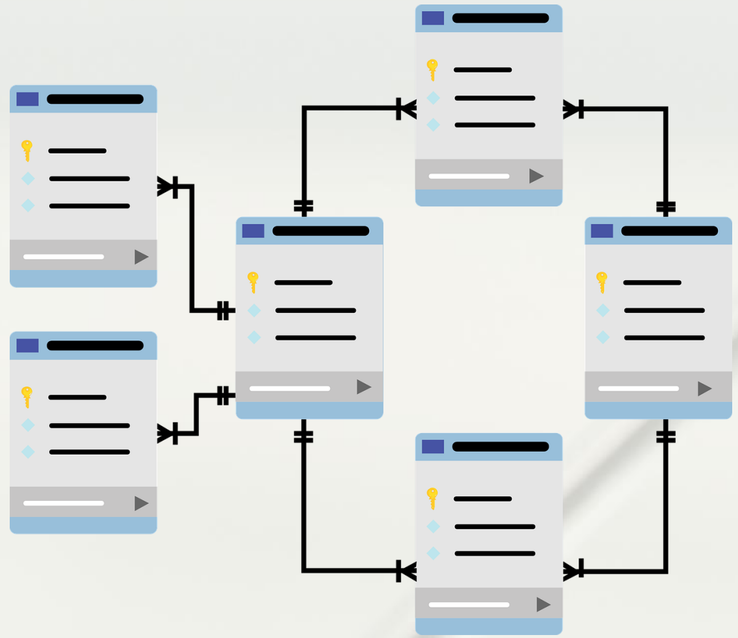
Η υποστήριξη της Ευρωπαϊκής Επιτροπής στην παραγωγή της παρούσας έκδοσης δε συνιστά αποδοχή του περιεχομένου, το οποίο αντικατοπτρίζει αποκλειστικά τις απόψεις των συντακτών της, και η Επιτροπή δεν μπορεί να αναλάβει την ευθύνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν. (Αριθμός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



Υποκεφάλαιο 2: Βάση δεδομένων στην SQL

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

Εισαγωγή στη βάση δεδομένων στην SQL



Όπως έχουμε αναφέρει στην προηγούμενη υποενότητα που ήταν αφιερωμένη στις βασικές λειτουργίες που χρησιμοποιούνται στην SQL, αυτή η γλώσσα προγραμματισμού χρησιμοποιείται κυρίως για σχεσιακές βάσεις δεδομένων.

Ως εκ τούτου, σε αυτή την υποενότητα, θα μάθουμε πώς να δημιουργήσουμε μια βάση δεδομένων, να την τροποποιήσουμε και να την χειριστούμε μέσω της SQL.

Ας ξεκινήσουμε από τις απλές λειτουργίες, προχωρώντας σε ελαφρώς πιο περίπλοκες.

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

Δημιουργία Βάσης Δεδομένων στην SQL

Η εντολή CREATE DATABASE **δημιουργεί μια νέα βάση δεδομένων στην SQL.**

Σύνταξη:

CREATE DATABASE ΌνομαΒάσηςΔεδομένων;

Σημείωση: Να θυμάστε πάντα ότι το όνομα της βάσης δεδομένων πρέπει να είναι μοναδικό εντός του σχεσιακού Συστήματος Διαχείρισης Βάσεων Δεδομένων (ΣΒΔΒ) που χρησιμοποιείτε και βεβαιωθείτε ότι έχετε δικαιώματα διαχειριστή πριν δημιουργήσετε οποιαδήποτε βάση δεδομένων.

Ας πούμε ότι θέλετε να δημιουργήσετε μια δοκιμαστική βάση δεδομένων. Θα χρησιμοποιούσατε την ακόλουθη εντολή:

CREATE DATABASE δοκιμαστικήΒΔ;

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

Διαγραφή Βάσης Δεδομένων στην SQL

Η εντολή DROP DATABASE **διαγράφει μια υπάρχουσα βάση δεδομένων στην SQL.**

Σύνταξη:

DROP DATABASE ΌνομαΒάσηςΔεδομένων;

*Πριν διαγράψετε τη βάση δεδομένων, βεβαιωθείτε ότι δεν χρειάζεστε καμία από τις πληροφορίες που περιέχει, καθώς τη διαγράφει εντελώς.

Θυμάστε τη βάση δεδομένων που μόλις δημιουργήσαμε με την ονομασία «δοκιμαστικήΒΔ»;

Τώρα θα την διαγράψουμε.

Παράδειγμα:

DROP DATABASE δοκιμαστικήΒΔ;

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

Δημιουργία αντιγράφου της Βάσης Δεδομένων στην SQL

Η εντολή BACKUP DATABASE δημιουργεί **ένα πλήρες εφεδρικό αντίγραφο σε μια υπάρχουσα βάση δεδομένων SQL.**

Για να χρησιμοποιήσετε αυτή την εντολή, θα πρέπει να γράψετε δύο πράγματα: το όνομα της βάσης δεδομένων και τη διαδρομή του αρχείου.

Σύνταξη:

BACKUP DATABASE ΌνομαΒάσηςΔεδομένων

TO DISK = ‘διαδρομήαρχείου’

Παράδειγμα:

BACKUP DATABASE δοκιμαστικήΒΔ

TO DISK = 'D:\αντίγραφα\δοκιμαστικήΒΔ.bak';

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

Δημιουργία αντιγράφου της Βάσης Δεδομένων στην SQL

- Για να αποφύγετε τεχνικά προβλήματα, **είναι καλύτερο να δημιουργήσετε εφεδρικά αντίγραφα της βάσης δεδομένων σε μια διαφορετική μονάδα δίσκου από εκείνη στην οποία βρίσκεται η υπάρχουσα βάση δεδομένων.**

- Υπάρχει επίσης μια άλλη επιλογή όπου μπορείτε να εκτελέσετε ένα διαφορικό εφεδρικό αντίγραφο με βάση τις αλλαγές που έχουν γίνει από το τελευταίο πλήρες εφεδρικό αντίγραφο της βάσης δεδομένων. Αυτός ο τύπος εφεδρικών αντιγράφων μειώνει επίσης τον χρόνο δημιουργίας εφεδρικών αντιγράφων.

Σύνταξη:

```
BACKUP DATABASE ΌνομαΒάσηςΔεδομένων
TO DISK = ‘διαδρομήαρχείου'
WITH DIFFERENTIAL;
```

Παράδειγμα:

```
BACKUP DATABASE δοκιμαστικήΔΒ
TO DISK = 'D:\αντίγραφα\δοκιμαστικήΔΒ.bak'
WITH DIFFERENTIAL;
```

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

Δημιουργία πίνακα στην SQL

Η εντολή CREATE TABLE δημιουργεί έναν νέο πίνακα σε μια βάση δεδομένων.

Σύνταξη:

```
CREATE TABLE όνομα_πίνακα(
στήλη1 τύποςδεδομένων,
στήλη2 τύποςδεδομένων,
στήλη3 τύποςδεδομένων,
                .....
);
```

Σε αυτή την εντολή, θα πρέπει να καθορίσετε **τα ονόματα των στηλών** και τον **τύπο των δεδομένων** που θα περιέχει η στήλη.

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

Δημιουργία πίνακα στην SQL

Υπάρχουν πολλοί τύποι δεδομένων όπως integer, date ή varchar. Ανάλογα με τον τύπο των δεδομένων που θέλετε να αποθηκεύσετε, επιλέγετε την πιο κατάλληλη επιλογή. Για παράδειγμα, αν έχετε μια στήλη με το όνομα «Ημερομηνία Γέννησης», τότε πιθανότατα θα επιλέγατε την Ημερομηνία (Date) ως τον τύπο δεδομένων.

Παράδειγμα:

```
CREATE TABLE Άτομα (
ΚωδΑτόμου int,
Επίθετο varchar(255),
Όνομα varchar(255),
Διεύθυνση varchar(255),
Πόλη varchar(255)
);
```

PersonID	LastName	FirstName	Address	City

Κενός Πίνακας - Παράδειγμα της εντολής CREATE TABLE

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

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

Δημιουργία πίνακα στην SQL

Μπορείτε επίσης να δημιουργήσετε έναν πίνακα χρησιμοποιώντας έναν άλλο πίνακα και επιλέγοντας ποιες στήλες θέλετε στον νέο πίνακα. Λάβετε υπόψη ότι τα δεδομένα του υπάρχοντος πίνακα θα συμπληρώσουν τις εγγραφές του νέου πίνακα.

Σύνταξη:

```
CREATE TABLE όνομα_νέου_πίνακα AS
SELECT στήλη1, στήλη2,...
FROM όνομα_υπάρχοντος_πίνακα
WHERE ....;
```

Παράδειγμα:

```
CREATE TABLE ΔοκιμαστικόςΠίνακας AS
SELECT ΌνομαΠελάτη, ΌνομαΕπικοινωνίας
FROM Πελάτες;
```

Όπως μάθαμε στην προηγούμενη ενότητα:

- Η εντολή SELECT επιλέγει τις στήλες από τον υπάρχοντα πίνακα,
- Η εντολή FROM καθορίζει το όνομα του υπάρχοντος πίνακα, και
- Ο όρος WHERE μπορεί να χρησιμοποιηθεί αν θέλετε να επιλέξετε ένα σύνολο εγγραφών που πληρούν μια καθορισμένη συνθήκη.

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

Διαγραφή πίνακα στην SQL

Είναι παρόμοια με την εντολή DROP DATABASE που είδαμε νωρίτερα, όμως η εντολή DROP TABLE διαγράφει έναν υπάρχοντα πίνακα σε μια βάση δεδομένων.

Να θυμάστε ότι θα πρέπει να βεβαιωθείτε ότι δεν χρειάζεστε καμία από τις πληροφορίες που περιέχονται σε έναν πίνακα πριν τον διαγράψετε.

Σύνταξη:

DROP TABLE ΌνομαΠίνακα;

Παράδειγμα:

DROP TABLE Άτομα;

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

Διαγραφή πίνακα στην SQL

Μπορείτε επίσης να επιλέξετε να διαγράψετε τα δεδομένα που περιέχονται σε έναν πίνακα, αλλά όχι και τον ίδιο τον πίνακα.

Μπορεί να δημιουργήσατε έναν νέο πίνακα από έναν υπάρχοντα πίνακα που έχει τη δομή που θέλετε, αλλά να θέλετε να προσθέσετε νέες εγγραφές εξ ολοκλήρου. Σε αυτή την περίπτωση, μπορείτε να χρησιμοποιήσετε την εντολή TRUNCATE TABLE.

Σύνταξη:

TRUNCATE TABLE ΌνομαΠίνακα;

Παράδειγμα:

TRUNCATE TABLE Άτομα;

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

Τροποποίηση πίνακα στην SQL (ALTER TABLE)

Η λειτουργία ALTER TABLE μπορεί να προσθέσει, να διαγράψει ή να τροποποιήσει στήλες σε έναν υπάρχοντα πίνακα. Επίσης, μπορεί να χρησιμοποιηθεί για την προσθήκη και την διαγραφή περιορισμών σε έναν υπάρχοντα πίνακα.

Σύνταξη για προσθήκη στήλης:

```
ALTER TABLE ΌνομαΠίνακα
ADD όνομα_στήλης τύπος δεδομένων;
```

Αυτή η λειτουργία μοιάζει με τη λειτουργία που χρησιμοποιήσαμε για να δημιουργήσουμε έναν πίνακα, καθώς πρέπει να καθορίσουμε το όνομα της στήλης και τον τύπο των δεδομένων που θα περιέχονται σε αυτή την στήλη.

Παράδειγμα:

```
ALTER TABLE Πελάτες
ADD Email varchar(255);
```

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

Τροποποίηση πίνακα στην SQL (ALTER TABLE)

Για να διαγράψουμε μια στήλη σε έναν πίνακα, όπως έχουμε δει στο παρελθόν, χρησιμοποιούμε την εντολή DROP.

Σημειώστε ότι ορισμένα συστήματα βάσεων δεδομένων δεν επιτρέπουν στους χρήστες να διαγράψουν μια στήλη.

Σύνταξη:

```
ALTER TABLE ΌνομαΠίνακα
DROP COLUMN ΌνομαΣτήλης;
```

Για παράδειγμα, ας διαγράψουμε τη στήλη που δημιουργήσαμε:

```
ALTER TABLE Πελάτες
DROP COLUMN Email;
```

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

Τροποποίηση πίνακα στην SQL (ALTER TABLE)

Για να **αλλάξετε τον τύπο δεδομένων μιας στήλης**, μπορείτε να χρησιμοποιήσετε τις ακόλουθες λειτουργίες ανάλογα με το σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων που χρησιμοποιείτε:

- ALTER COLUMN (για SQL Server/MS Access);
- MODIFY COLUMN (για My SQL/ Oracle πριν την έκδοση 10G);
- MODIFY (για την έκδοση 10G Oracle και μεταγενέστερες εκδόσεις).

Σύνταξη:

ALTER TABLE ΌνομαΠίνακα

ALTER COLUMN ΌνομαΣτήλης τύποςδεδομένων;

*Σημειώστε ότι η δεύτερη εντολή είναι αυτή που αλλάζει σύμφωνα με το ΣΣΔΒΔ που χρησιμοποιείτε, μεταξύ ALTER COLUMN, MODIFY COLUMN και MODIFY. Τα υπόλοιπα παραμένουν τα ίδια.

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

Τροποποίηση πίνακα στην SQL (ALTER TABLE)

Παράδειγμα: Προσθέστε μια στήλη με το όνομα «Ημερομηνία γέννησης» στον πίνακα Άτομα

```
ALTER TABLE Άτομα
ADD ΗμερομηνίαΓέννησης ημερομηνία;
```

Η νέα στήλη που προσθέσαμε στον πίνακα έχει τον τύπο δεδομένων της ημερομηνίας, που σημαίνει ότι αποθηκεύει δεδομένα σε μορφή ημερομηνίας. Παρακάτω, μπορείτε να δείτε τον πίνακα με την προσθήκη της νέας στήλης.

ID	LastName	FirstName	Address	City	DateOfBirth
1	Hansen	Ola	Timoteivn 10	Sandnes	
2	Svendson	Tove	Borgvn 23	Sandnes	
3	Pettersen	Kari	Storgt 20	Stavanger	

Παράδειγμα τροποποίησης πίνακα (ALTER TABLE)

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

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

Τροποποίηση πίνακα στην SQL (ALTER TABLE)

Ωστόσο, αν αλλάξατε γνώμη και θέλετε να αλλάξετε τον τύπο δεδομένων της νέας στήλης, τότε μπορείτε να χρησιμοποιήσετε τη λειτουργία ALTER COLUMN.

Για παράδειγμα, μπορούμε να αλλάξουμε τον τύπο της στήλης που προστέθηκε πρόσφατα από ημερομηνία σε έτος:

ALTER TABLE Άτομα

ALTER COLUMN ΗμερομηνίαΓέννησης Έτος;

Ο τύπος δεδομένων έτους περιέχει ένα έτος σε διψήφιο ή τετραψήφιο μορφότυπο.

Για να διαγράψουμε τη στήλη που μόλις τροποποιήσαμε, χρησιμοποιούμε την εντολή DROP COLUMN.

ALTER TABLE Άτομα

DROP COLUMN ΗμερομηνίαΓέννησης;

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

Περιορισμοί στην SQL (Constraints)

Οι περιορισμοί στην SQL χρησιμοποιούνται όταν ο πίνακας δημιουργείται με την εντολή CREATE TABLE ή την εντολή ALTER TABLE κατά την τροποποίησή του.

Οι περιορισμοί χρησιμοποιούνται για **τον καθορισμό ενός συνόλου κανόνων και περιορισμών που ισχύουν για μια στήλη ή έναν πίνακα.** Χρησιμοποιούνται για τη διασφάλιση της ακεραιότητας, της ακρίβειας και της αξιοπιστίας των δεδομένων. Εάν εφαρμόζονται περιορισμοί εσε έναν πίνακα, τότε όλες οι στήλες πρέπει να διαμορφώνονται με βάση αυτούς τους περιορισμούς.

Σύνταξη:

```
CREATE TABLE όνομα_πίνακα(
        στήλη1 τύποςδεδομένων περιορισμός,
        στήλη2 τύποςδεδομένων περιορισμός,
        στήλη3 τύποςδεδομένων περιορισμός,
        ....
);
```

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

Περιορισμοί στην SQL (Constraints)

Οι ακόλουθοι περιορισμοί είναι αυτοί που χρησιμοποιούνται τις περισσότερες φορές:

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT
- CREATE INDEX

Θα εξετάσουμε καθέναν από αυτούς τους περιορισμούς για να εξηγήσουμε τη χρήση και τη σύνταξή τους με παραδείγματα.

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

Περιορισμοί ύπαρξης τιμής στην SQL (Not Null)

Στην SQL, οι στήλες μπορούν να έχουν κενές τιμές από προεπιλογή. Ο περιορισμός NOT NULL χρησιμοποιείται για την αποφυγή των κενών τιμών σε στήλες. Αυτό είναι ιδιαίτερα σημαντικό για να εξασφαλιστεί ότι, όταν προστίθεται μια νέα εγγραφή σε έναν πίνακα, συμπληρώνονται όλα τα απαραίτητα πεδία.

Για παράδειγμα, ας πούμε ότι θέλουμε να δημιουργήσουμε έναν πίνακα με το όνομα «Άτομα» και θέλουμε να διασφαλίσουμε ότι οι στήλες «ΚωδΑτόμου», «Επίθετο» και «Όνομα» δεν θα περιέχουν κενές τιμές:

```
CREATE TABLE Άτομα (
        ID int NOT NULL,
Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255) NOT NULL,
        Ηλικία int
);
```

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

Περιορισμοί ύπαρξης τιμής στην SQL (Not Null)

Εάν για κάποιο λόγο, θέλετε να τροποποιήσετε έναν ήδη υπάρχοντα πίνακα για να προσθέσετε περιορισμούς, μπορείτε να χρησιμοποιήσετε την ακόλουθη εντολή:

```
ALTER TABLE Άτομα
MODIFY Ηλικία int NOT NULL;
```

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

Μοναδικές τιμές στην SQL (Unique)

Ο περιορισμός UNIQUE χρησιμοποιείται για να διασφαλιστεί ότι όλες οι τιμές που αποθηκεύονται σε μια στήλη είναι μοναδικές μεταξύ των σειρών ενός πίνακα. Για να γίνει αυτό σαφέστερο, σκεφτείτε τη μεταβλητή «ΚωδΑτόμου». Για να μην έχουν δύο άτομα την ίδια ΚωδΑτόμου, χρησιμοποιούμε τον περιορισμό UNIQUE.

SQL Server / Oracle / MS Access:

```
CREATE TABLE Άτομα (
        ΚωδΑτόμου int NOT NULL UNIQUE,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int
);
```

My SQL:

```
CREATE TABLE Άτομα (
        ΚωδΑτόμου int NOT NULL,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int,
UNQUE (ΚωδΑτόμοι
);
```

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

Μοναδικές τιμές στην SQL (Unique)

Εάν θέλετε να **ονομάσετε ή να θέσετε έναν περιορισμό UNIQUE σε πολλαπλές στήλες**, χρησιμοποιήστε τα ακόλουθα:

```
CREATE TABLE Άτομα (
        ΚωδΑτόμου int NOT NULL,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int,
CONSTRAINT ΜΠ_Άτομο UNIQUE (ΚωδΑτόμου.Επίθετο)
);
```

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

Μοναδικές τιμές στην SQL (Unique)

Μπορείτε επίσης να προσθέσετε έναν περιορισμό UNIQUE μετά τη δημιουργία του πίνακα χρησιμοποιώντας την εντολή ALTER TABLE που μάθαμε νωρίτερα.

MySQL / SQL Server / Oracle / MS Access:

```
ALTER TABLE Άτομα
ADD UNIQUE (ΚωδΑτόμου);
```

Εάν θέλετε επίσης να **ονομάσετε και να θέσετε έναν περιορισμό UNIQUE σε πολλές ήδη υπάρχουσες στήλες**, χρησιμοποιείτε την ακόλουθη πρόταση:

```
ALTER TABLE Άτομα
ADD CONSTRAINT ΜΠ_Άτομα UNIQUE (ΚωδΑτόμου.Επίθετο);
```

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

Πρωτεύον Κλειδί στην SQL (Primary Key)

Ο περιορισμός PRIMARY KEY χρησιμοποιείται για τη μοναδική αναγνώριση κάθε σειράς ή εγγραφής σε έναν πίνακα. Σημειώστε ότι οι περιορισμοί πρέπει να περιέχουν μοναδικές τιμές, ωστόσο δεν μπορούν να περιέχουν κενές τιμές.

Ένας πίνακας μπορεί να έχει **μόνο** ΕΝΑ πρωτεύον κλειδί το οποίο αυτό μπορεί να αποτελείται από μία ή περισσότερες στήλες.

SQL Server / Oracle / MS Access:

```
CREATE TABLE Άτομα (
        ΚωδΑτόμου int NOT NULL PRIMARY KEY,
        Επίθετο varchar(255) NOT NULL,
        Όνομα archar(255),
        Ηλικία int
);
```

MySQL:

```
CREATE TABLE Άτομα (
        ΚωδΑτόμου int NOT NULL,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int,
    PRIMARY KEY (ΚωδΑτόμου)
);
```

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

Πρωτεύον Κλειδί στην SQL (Primary Key)

Το ακόλουθο παράδειγμα σας επιτρέπει να **ονομάσετε και να θέσετε έναν περιορισμό PRIMARY KEY σε πολλαπλές στήλες:**

```
CREATE TABLE Άτομα (
        ΚωδΑτόμου int NOT NULL,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int,
    CONSTRAINT ΠΚ_Άτομα PRIMARY KEY (Ταυότητα,Επίθετο)
);
```

*** Σημειώστε ότι υπάρχει ένα μόνο PRIMARY KEY, ωστόσο η τιμή του περιλαμβάνει δύο στήλες.**

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

Πρωτεύον Κλειδί στην SQL (Primary Key)

Μπορείτε επίσης να δημιουργήσετε έναν περιορισμό PRIMARY KEY σε έναν υπάρχοντα πίνακα χρησιμοποιώντας την ακόλουθη εντολή:

```
ALTER TABLE Άτομα
ADD PRIMARY KEY (ΚωδΑτόμου);
```

Για να προσθέσετε και να θέσετε έναν περιορισμό PRIMARY KEY σε έναν υπάρχοντα πίνακα, χρησιμοποιήστε την ακόλουθη εντολή:

```
ALTER TABLE Άτομα
ADD CONSTRAINT ΠΚ_Άτομα PRIMARY KEY (ΚωδΑτόμου, Επίθετο);
```

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

Πρωτεύον Κλειδί στην SQL (Primary Key)

Για να διαγράψετε έναν περιορισμό PRIMARY KEY, χρησιμοποιήστε τις ακόλουθες εντολές σύμφωνα με το ΣΣΔΒΔ που χρησιμοποιείτε.

MySQL:

```
ALTER TABLE Άτομα
DROP PRIMARY KEY;
```

SQL Server / Oracle / MS Access:

```
ALTER TABLE Άτομα
DROP PRIMARY KEY;
```

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

Ξένο Κλειδί στην SQL (Foreign Key)

Το FOREING KEY **αντιπροσωπεύει τις στήλες ενός πίνακα που συνδέονται με έναν περιορισμό PRIMARY KEY σε έναν άλλο πίνακα.** Ο πίνακας που έχει τον περιορισμό FOREIGN KEY ονομάζεται child table, ενώ ο πίνακας που έχει το primary key ονομάζεται πίνακας αναφοράς ή parent table.

Αυτός ο τύπος περιορισμού χρησιμοποιείται για να αποτρέψει οποιεσδήποτε ενέργειες που θα κατέστρεφαν τους δεσμούς μεταξύ του parent table και του child table. **Εξετάστε τους ακόλουθους δύο πίνακες: Τι κοινό έχουν;**

PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

Πίνακας ατόμων - Παράδειγμα FOREING KEY
(Πηγή: https://www.w3schools.com/sql/sql_foreignkey.asp)

OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

Πίνακας παραγγελιών - Παράδειγμα FOREING KEY
(Πηγή: https://www.w3schools.com/sql/sql_foreignkey.asp)

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

Ξένο Κλειδί στην SQL (Foreign Key)

Αυτοί οι δύο πίνακες συνδέονται με τη στήλη «ΑναγνωριστικόΑτόμου» (PersonID) που βρίσκεται και στους δύο πίνακες. Τώρα, το primary key βρίσκεται στον πίνακα ΄Ατομα και το foreign key στη στήλη «ΑναγνωριστικόΑτόμου» στον πίνακα Παραγγελίες.

Ο περιορισμός FOREIGN KEY λειτουργεί εμποδίζοντας την εισαγωγή μη έγκυρων δεδομένων στη στήλη του FOREIGN KEY επειδή συνδέεται με τον parent table και οι τιμές του πρέπει να είναι πανομοιότυπες.

SQL Server / Oracle / MS Access:

```
CREATE TABLE Παραγγελίες (
    ΚωδικόςΠαραγγελίας int NOT NULL PRIMARY KEY,
    ΑριθμόςΠαραγγελίας int NOT NULL,
    ΑναγνωριστικόΑτόμου int FOREIGN KEY REFERENCES Άτομα(ΑναγνωριστικόΑτόμου)
);
```

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

Ξένο Κλειδί στην SQL (Foreign Key)

My SQL:

```
CREATE TABLE Παραγγελίες (
        ΚωδικόςΠαραγγελίας int NOT NULL,
        ΑριθμόςΠαραγγελίας int NOT NULL,
        ΑναγνωριστικόΑτόμου int,
         PRIMARY KEY (ΚωδικόςΠαραγγελίας),
         FOREIGN KEY (ΑναγνωριστικόΑτόμου) REFERENCES Άτομα(ΑναγνωριστικόΑτόμου)
);
```

Αυτή η εντολή συνέδεσε τον πίνακα Παραγγελιών με τον πίνακα Ατόμων χρησιμοποιώντας τον περιορισμό FOREIGN KEY με βάση τη στήλη ΑναγνωριστικόΑτόμου.

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

Περιορισμός ελέγχου τιμής στην SQL (CHECK)

Ο περιορισμός CHECK χρησιμοποιείται για να καθορίσει τις τιμές που επιτρέπονται σε μια στήλη ή σε ορισμένες στήλες ενός πίνακα με βάση τις τιμές που βρίσκονται σε άλλες στήλες της ίδιας σειράς.

Παράδειγμα περιορισμού CHECK με την εντολή CREATE TABLE: Βεβαιωθείτε ότι ένα άτομο δεν είναι κάτω από την ηλικία των 18 ετών, προσθέτοντας τον περιορισμό CHECK στη στήλη «Ηλικία».

MySQL:

```
CREATE TABLE Άτομα (
        ΚωδΑτόμου int NOT NULL,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int,
    CHECK (Age>=18)
);
```

SQL Server / Oracle / MS Access:

```
CREATE TABLE  Άτομα (
        ΚωδΑτόμου int NOT NULL,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int CHECK (Age>=18)
);
```

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

Περιορισμός ελέγχου τιμής στην SQL (CHECK)

Εάν θέλετε να καθορίσετε έναν περιορισμό CHECK και να χρησιμοποιήσετε τον περιορισμό σε πολλαπλές στήλες, μπορείτε να χρησιμοποιήσετε την ακόλουθη εντολή:

MySQL / SQL Server / Oracle / MS Access:

```
CREATE TABLE Άτομα (
        ΚωδΑτόμου int NOT NULL,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int,
         Πόλη varchar(255),
         CONSTRAINT ΕΛΓΧ_Άτομο CHECK (Ηλικία>=18 AND Πόλη= ‘Σάντνες’)
);
```

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

Περιορισμός ελέγχου τιμής στην SQL (CHECK)

Παράδειγμα περιορισμού CHECK στην εντολή ALTER TABLE

Για να καθορίσετε έναν περιορισμό σε έναν ήδη υπάρχοντα πίνακα, χρησιμοποιήστε την ακόλουθη εντολή.

MySQL / SQL Server / Oracle / MS Access:

ALTER TABLE Άτομα

ADD CHECK (Ηλικία>=18);

Για να καθορίσετε έναν περιορισμό και να τον προσθέσετε σε πολλαπλές στήλες, μπορείτε να χρησιμοποιήσετε:

ALTER TABLE Άτομα

ADD CONSTRAINT ΕΛΓΧ_Άτομα CHECK (Ηλικία>=18 AND Πόλη= ‘Σάντνες’);

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

Περιορισμός ελέγχου τιμής στην SQL (CHECK)

Παράδειγμα εντολής DROP ενός περιορισμού CHECK

Για να διαγράψετε έναν περιορισμό CHECK, μπορείτε να χρησιμοποιήσετε τα ακόλουθα σύμφωνα με το ΣΣΔΒΔ που χρησιμοποιείτε.

SQL Server / Oracle / MS Access:

```
ALTER TABLE Άτομα
DROP CONSTRAINT ΕΛΓΧ_ΗλικίαΑτόμου;
```

MySQL:

```
ALTER TABLE Άτομα
DROP CHECK ΕΛΓΧ_ΗλικίαΑτόμου;
```

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

Περιορισμός προεπιλεγμένων τιμών στην SQL (DEFAULT)

Ο περιορισμός DEFAULT χρησιμοποιείται για να **καθορίσει μια προεπιλεγμένη τιμή για μια στήλη.** Εάν δεν έχουν καθοριστεί άλλες τιμές, η προεπιλεγμένη τιμή θα προστεθεί σε όλες τις νέες εγγραφές.

Παράδειγμα περιορισμού DEFAULT στη λειτουργία CREATE TABLE: Πρόσθεση μιας προεπιλεγμένης τιμή στη στήλη Πόλη όταν δημιουργείται ο πίνακας Άτομα:

```
CREATE TABLE Άτομα (
        ΚωδΑτόμου int NOT NULL,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int,
        Πόλη varchar(255) DEFAULT ‘Σάντνες'
);
```

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

Περιορισμός προεπιλεγμένων τιμών στην SQL (DEFAULT)

Αυτός ο περιορισμός μπορεί επίσης να χρησιμοποιηθεί για την **εισαγωγή τιμών με τελεστές όπως το GETDATE():**

```
CREATE TABLE Παραγγελίες (
        Κωδικός int NOT NULL,
        ΑρΠαραγγελίας int NOT NULL,
        ΗμερομηνίαΠαραγγελίας ημερομηνία DEFAULT GETDATE()
);
```

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

Περιορισμός προεπιλεγμένων τιμών στην SQL (DEFAULT)

Παράδειγμα περιορισμού DEFAULT με την εντολή ALTER TABLE

Σε αυτό το παράδειγμα, η στήλη «Πόλη» χρησιμοποιείται για τον καθορισμό ενός περιορισμού DEFAULT όταν τροποποιούμε έναν ήδη υπάρχοντα πίνακα.

MySQL:

```
ALTER TABLE Άτομα
ALTER Πόλη SET DEFAULT ‘Σάντνες';
```

SQL Server:

```
ALTER TABLE Άτομα
ADD CONSTRAINT προεπ_Πόλη
DEFAULT ‘Σάντνες' FOR Πόλη;
```

MS Access:

```
ALTER TABLE Άτομα
ALTER COLUMN ΄Πόλη SET DEFAULT ‘Σάντνες’;
```

Oracle:

```
ALTER TABLE Άτομα
MODIFY Πόλη DEFAULT ‘Σάντνες';
```

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

Περιορισμός προεπιλεγμένων τιμών στην SQL (DEFAULT)

Παράδειγμα κατάργησης ενός περιορισμού DEFAULT με την εντολή DROP

Χρησιμοποιείται για τη διαγραφή του περιορισμού DEFAULT στον ήδη υπάρχοντα πίνακα.

MySQL:

```
ALTER TABLE Άτομα
ALTER City DROP DEFAULT;
```

SQL Server / Oracle / MS Access:

```
ALTER TABLE Άτομα
ALTER COLUMN Πόλη DROP DEFAULT;
```

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

Δείκτες στην SQL (Index)

Η εντολή CREATE INDEX **δημιουργεί έναν δείκτη σε έναν πίνακα**. Οι δείκτες είναι χρήσιμοι όταν θέλετε να ανακτήσετε δεδομένα πιο γρήγορα.

Για την εντολή CREATE INDEX σε έναν πίνακα όπου επιτρέπονται διπλές τιμές, χρησιμοποιήστε την ακόλουθη σύνταξη:

```
CREATE INDEX index_name
ON table_name (column1, column2, ...);
```

Για την εντολή CREATE UNIQUE INDEX σε έναν πίνακα όπου επιτρέπονται διπλές τιμές, χρησιμοποιήστε την ακόλουθη σύνταξη:

```
CREATE UNIQUE INDEX όνομα_δείκτη
ON όνομα_πίνακα (στήλη1, στήλη2, ...);
```

- ✓ Σημειώστε ότι οι πίνακες με δείκτες χρειάζονται περισσότερο χρόνο για να ενημερωθούν σε σύγκριση με τους πίνακες χωρίς δείκτες. Ως εκ τούτου, **προτείνεται να δημιουργηθούν μόνο δείκτες σε στήλες όπου αναζητούνται δεδομένα συχνά.**
- ✓ Λάβετε υπόψη **ότι η δημιουργία δεικτών ποικίλλει από βάση δεδομένων σε βάση δεδομένων,** οπότε πάντα να ελέγχετε τη σύνταξη όταν δημιουργείτε έναν δείκτη στη βάση δεδομένων σας.

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

Δείκτες στην SQL (Index)

Παραδείγματα της εντολής CREATE INDEX

Σε αυτό το παράδειγμα, **δημιουργούμε έναν δείκτη στη στήλη Επίθετο** καθορίζοντας το όνομα δκτς_επίθετο:

```
CREATE INDEX δκτς_επίθετο
ON Άτομα (Επίθετο);
```

Για να **δημιουργήσετε έναν δείκτη σε ένα συνδυασμό στηλών,** χρησιμοποιήστε την ακόλουθη εντολή:

```
CREATE INDEX δκτς_όνομα
ON Άτομα (Επίθετο, Όνομα);
```

Αν θέλετε, μπορείτε να προσθέσετε περισσότερες στήλες στην παρένθεση.

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

Δείκτες στην SQL (Index)

Παραδείγματα της εντολής DROP INDEX για τη διαγραφή ενός δείκτη

Εάν θέλετε να διαγράψετε έναν δείκτη, χρησιμοποιήστε την ακόλουθη εντολή σύμφωνα με το ΣΣΔΒΔ που χρησιμοποιείτε.

MS Access:

DROP INDEX όνομα_δείκτη ON όνομα_πίνακα;

SQL Server:

DROP INDEX όνομα_πίνακα.όνομα_δείκτη;

DB2/Oracle:

DROP INDEX όνομα_δείκτη;

MySQL:

ALTER TABLE όνομα_πίνακα

DROP INDEX όνομα_δείκτη;

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

Αυτόματη Αύξηση Τιμής στην SQL (Auto-Increment)

Η αυτόματη αύξηση χρησιμοποιείται για την αυτόματη δημιουργία μοναδικών αριθμών όταν εισάγεται μια νέα εγγραφή σε έναν πίνακα. Αυτό χρησιμοποιείται συνήθως σαν Primary key προκειμένου να διασφαλιστεί ότι κανένα άτομο δεν έχει την ίδια ΚωδΑτόμου.

Αυτή η λειτουργία χρησιμοποιεί διαφορετική σύνταξη στα συστήματα MySQL, SQL Server, Access και Oracle. Ως εκ τούτου, θα εξετάσουμε καθένα από αυτά για να εξηγήσουμε πώς να χρησιμοποιήσετε την αυτόματη αύξηση τιμών.

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

Αυτόματη Αύξηση Τιμής στην SQL (Auto-Increment)

MySQL:

```
CREATE TABLE Άτομα (
        ΤαυτΑτόμου int NOT NULL AUTO_INCREMENT,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int,
         PRIMARY KEY (ΤαυτΑτόμου)
);
```

Στο σύστημα MySQL, **η εντολή AUTO_INCREMENT** προσθέτει τη λειτουργία αυτόματης αύξησης και **η τιμή ορίζεται 1 από προεπιλογή και αυξάνεται κατά 1 κάθε φορά.**

Εάν θέλετε η **ακολουθία να ξεκινήσει από διαφορετική τιμή**, χρησιμοποιήστε την ακόλουθη εντολή:

```
ALTER TABLE Άτομα AUTO_INCREMENT=100;
```

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

Αυτόματη Αύξηση Τιμής στην SQL (Auto-Increment)

Εάν εισαγάγετε μια νέα εγγραφή στον πίνακα «Άτομα», δεν θα χρειαστεί να καθορίσετε μια τιμή για τη στήλη «ΤαυτΑτόμου», καθώς θα δημιουργηθεί αυτόματα:

```
INSERT INTO Άτομα (Όνομα, Επίθετο)
VALUES (‘Λαρς','Μόνσεν');
```

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

Αυτόματη Αύξηση Τιμής στην SQL (Auto-Increment)

SQL Server

Ακολουθούμε το ίδιο παράδειγμα όπως είδαμε στις προηγούμενες διαφάνειες για το MySQL, όπου χρησιμοποιούμε τη στήλη «ΤαυτΑτόμου» ως το primary key στον πίνακα «Άτομα»:

```
CREATE TABLE Άτομα (
        ΤαυτΑτόμου int IDENTITY(1,1)
PRIMARY KEY,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int
);
```

Στο σύστημα SQL Server, η εντολή της αυτόματης αύξησης τιμών χρησιμοποιεί τη λέξη-κλειδί IDENTIFY για να ενεργοποιηθεί. Οι δύο τιμές στην παρένθεση υποδεικνύουν (την αρχική τιμή, προσθέτοντας μια νέα τιμή για κάθε νέα εγγραφή). Ξεκινά από το 1 και αυξάνεται κατά 1 κάθε φορά που εισάγεται μια νέα εγγραφή.

Σε περίπτωση που θέλετε να αλλάξετε την αρχική τιμή σε 10 και να προσθέτετε 5 κάθε φορά που προστίθεται μια νέα εγγραφή, θα πρέπει να την γράψετε ως εξής: IDENTIFY (10,5).

Κατά την εισαγωγή νέων εγγραφών, δεν χρειάζεται να καθορίσετε την τιμή στη στήλη ΤαυτΑτόμου. Θα εμφανιστεί αυτόματα.

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

Αυτόματη Αύξηση Τιμής στην SQL (Auto-Increment)

MS Access

```
CREATE TABLE Άτομα (
ΚωδΑτόμου AUTOINCREMENT PRIMARY KEY,
        Επίθετο varchar(255) NOT NULL,
        Όνομα varchar(255),
        Ηλικία int
);
```

Το MS Access χρησιμοποιεί τη λέξη-κλειδί AUTOINCREMENT για να ενεργοποιήσει τη λειτουργία αυτόματης αύξησης τιμών. Η αρχική τιμή είναι το 1 και αυξάνεται κατά 1 κάθε φορά που προστίθεται μια νέα εγγραφή, όπως και στις άλλες δύο περιπτώσεις.

Μπορείτε να καθορίσετε διαφορετικές τιμές όπως 10 για την αρχική τιμή και να αυξάνεται κατά 5 με κάθε προσθήκη με την εντολή AUTOINCREMENT(10,5).

Και πάλι, σημειώστε ότι κάθε φορά που προσθέτουμε μια νέα εγγραφή, δεν χρειάζεται να καθορίσουμε τον Κωδικό Ατόμου. Εμφανίζεται αυτόματα.

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

Αυτόματη Αύξηση Τιμής στην SQL (Auto-Increment)

Oracle

Στο σύστημα Oracle, ο κώδικας για την αυτόματη αύξηση τιμών είναι λίγο πιο περίπλοκος. Για να δημιουργήσετε ένα πεδίο αυτόματης αύξησης, πρέπει να δημιουργήσετε μια ακολουθία αριθμών:

```
CREATE SEQUENCE ακ_ατόμου
MINVALUE 1
START WITH 1
INCREMENT BY 1
CACHE 10;
```

Αυτή η ακολουθία δημιουργεί ένα αντικείμενο ακολουθίας που ονομάζεται «ακ_ατόμου», ορίζει την ελάχιστη τιμή από την οποία ξεκινά (η οποία είναι 1 σε αυτήν την περίπτωση), έπειτα καθορίζει την αύξηση κατά 1. Η προσωρινή μνήμη καθορίζει πόσες τιμές ακολουθίας θα πρέπει να αποθηκεύονται στη μνήμη για γρηγορότερη πρόσβαση.

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

Αυτόματη Αύξηση Τιμής στην SQL (Auto-Increment)

Σε αντίθεση με τα προηγούμενα παραδείγματα, για να εισαγάγετε μια νέα εγγραφή στον πίνακα Άτομα, θα πρέπει να χρησιμοποιήσετε τη λειτουργία nextval. Αυτή η λειτουργία χρησιμοποιείται για να ανακτήσει την επόμενη τιμή από το αντικείμενο ακολουθίας που δημιουργήσαμε.

```
INSERT INTO Άτομα (ΚωδΑτόμου, Όνομα, Επίθετο)
VALUES (ακ_ατόμου.nextval,’Λαρς',’Μόνσεν');
```

Εδώ, μπορούμε να δούμε ότι η στήλη ΚωδΑτόμου επιλέγεται για να εκχωρηθεί ο επόμενος αριθμός από το αντικείμενο ακολουθίας που δημιουργήσαμε που ονομάζεται «ακ_ατόμου».

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

Ημερομηνίες στην SQL

Μια από τις μεγαλύτερες προκλήσεις όταν εργάζεστε με ημερομηνίες (date) είναι να διασφαλίσετε ότι η μορφή της ημερομηνίας που προσπαθείτε να εισάγετε είναι η ίδια με τη μορφή της στήλης ημερομηνίας στη βάση δεδομένων.

Είναι σημαντικό να σημειωθεί ότι τα δεδομένα που περιέχουν μόνο ημερομηνίες θα λειτουργήσουν όπως αναμένεται στα ερωτήματα. Ωστόσο, αν συμπεριλαμβάνεται και η ώρα (time), τα πράγματα γίνονται λίγο πιο περίπλοκα.

Τύποι χρονολογικών δεδομένων στο MySQL:

- DATE (Ημερομηνία) – μορφή: EEEE-MM-HH
- DATETIME (Ημερομηνία και Ώρα) - μορφή: EEEE-MM-HH ΩΩ: ΛΛ:ΔΔ
- TIMESTAMP (Χρονοσήμανση) - μορφή: EEEE-MM-HH ΩΩ: ΛΛ:ΔΔ
- YEAR (Έτος) – μορφή: EEEE ή EE

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

Ημερομηνίες στην SQL

Τύποι χρονολογικών δεδομένων στο SQL Server:

- DATE – μορφή: EEEE-MM-HH
- DATETIME- μορφή: EEEE-MM-HH ΩΩ: ΛΛ:ΔΔ
- SMALLDATETIME - μορφή: EEEE-MM-HH ΩΩ: ΛΛ:ΔΔ
- TIMESTAMP - μορφή: ένας μοναδικός αριθμός

Λάβετε υπόψη ότι οι τύποι δεδομένων επιλέγονται όταν δημιουργείτε έναν νέο πίνακα στη βάση δεδομένων σας.

Θα χρησιμοποιήσουμε τον πίνακα Παραγγελιών στο παράδειγμά μας για να επιλέξουμε τις εγγραφές με ημερομηνία παραγγελίας “2008-11-11”.

Παράδειγμα:

```
SELECT *
FROM Παραγγελίες
WHERE ΗμερομηνίαΠαραγγελίας='2008-11-11’;
```

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

Ημερομηνίες στην SQL

Σημειώστε ότι δύο ημερομηνίες μπορούν να συγκριθούν εύκολα όταν δεν υπάρχει Χρονοσήμανση (TIMESTAMP).

Ας υποθέσουμε ότι έχετε τον πίνακα Παραγγελιών, αλλά με μια χρονική σήμανση στη στήλη Ημερομηνία Παραγγελίας (OrderDate).

OrderId	ProductName	OrderDate
1	Geitost	2008-11-11 13:23:44
2	Camembert Pierrot	2008-11-09 15:45:21
3	Mozzarella di Giovanni	2008-11-11 11:12:01
4	Mascarpone Fabioli	2008-10-29 14:56:59

Πίνακας παραγγελιών με χρονική σήμανση - Παράδειγμα ημερομηνιών (Πηγή: https://www.w3schools.com/sql/sql_dates.asp)

- Εάν προσπαθήσατε να χρησιμοποιήσετε το ίδιο ερώτημα με αυτό που χρησιμοποιήσαμε στην προηγούμενη διαφάνεια, δεν θα είχατε κανένα αποτέλεσμα. Γιατί; Επειδή το ερώτημα δεν λαμβάνει υπόψη τη χρονοσήμανση.

Συνιστάται να μην χρησιμοποιείτε χρονοσημάνσεις, εκτός αν είναι απολύτως απαραίτητο.

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

Προβολές στην SQL (VIEWS)

Στην SQL, μια προβολή (view) είναι ένας **εικονικός πίνακας του συνόλου αποτελεσμάτων που δημιουργείται από ένα συγκεκριμένο ερώτημα.** Μια προβολή είναι χρήσιμη όταν θέλετε να προβάλετε και να παρουσιάσετε δεδομένα μέσω ενός συνδυασμού πινάκων.

Σύνταξη:

```
CREATE VIEW όνομα_προβολής AS
SELECT στήλη1, στήλη2, ...
FROM όνομα_πίνακα
WHERE συνθήκη;
```

*** Σημειώστε ότι μια προβολή εμφανίζει πάντα ενημερωμένα δεδομένα, καθώς η βάση δεδομένων αναδημιουργεί τον εικονικό πίνακα, κάθε φορά που δημιουργούν το αντίστοιχο ερώτημα οι χρήστες.**

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

Προβολές στην SQL (VIEWS)

Παράδειγμα δημιουργίας προβολής που επιλέγει κάθε προϊόν στον πίνακα Προϊόντων με τιμή υψηλότερη από τη μέση τιμή:

```
CREATE VIEW [Προϊόντα με υψηλότερη τιμή από τη μέση τιμή] AS
SELECT ΌνομαΠροϊόντος, Τιμή
FROM Προϊόντα
WHERE Τιμή > (SELECT AVG(Τιμή) FROM Προϊόντα);
```

Για να υποβάλετε ένα ερώτημα σχετικά με την παραπάνω προβολή:

```
SELECT * FROM [Προϊόντα με υψηλότερη τιμή από τη μέση τιμή];
```

Παράδειγμα για να δημιουργήσετε ερώτημα με όλους τους πελάτες από τη Βραζιλία:

```
CREATE VIEW [Πελάτες από Βαζιλία] AS
SELECT ΌνομαΠελάτη, ΌνομαΕπικοινωνίας
FROM Πελάτες
WHERE Country = ‘Βραζιλία';
```

Για να προβάλετε το ερώτημα:

```
SELECT * FROM [Πελάτες από Βραζιλία];
```

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

Προβολές στην SQL (VIEWS)

Για να ενημερώσετε μια προβολή, χρησιμοποιήστε την εντολή CREATE OR REPLACE VIEW:

```
CREATE OR REPLACE VIEW όνομα_προβολής AS
SELECT στήλη1, στήλη2, ...
FROM όνομα_πίνακα
WHERE συνθήκη;
```

Για να προσθέσετε τη στήλη «Πόλη» στην προβολή με όνομα Πελάτες από Βραζιλία που δημιουργήσαμε νωρίτερα:

```
CREATE OR REPLACE VIEW [Πελάτες από Βραζιλία] AS
SELECT ΌνομαΠελάτη, ΌνομαΕπικοινωνίας, Πόλη
FROM Πελάτες
WHERE Χώρα = ‘Βραζιλία';
```

Για να διαγράψετε μια προβολή, χρησιμοποιήστε την εντολή DROP VIEW:

```
DROP VIEW όνομα_προβολής;
```

Για να διαγράψετε την προβολή «Πελάτες από Βραζιλία»:

```
DROP VIEW [Πελάτες από Βραζιλία];
```

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

Τύποι δεδομένων στην SQL

Γενικά, κάθε στήλη σε έναν πίνακα απαιτεί ένα όνομα και έναν τύπο δεδομένων.

Ένας προγραμματιστής της SQL θα πρέπει να αποφασίσει τον τύπο των δεδομένων που θα αποθηκευτούν μέσα σε κάθε στήλη κατά τη δημιουργία ενός πίνακα. Ο τύπος δεδομένων χρησιμοποιείται στην SQL για την κατανόηση των δεδομένων που θα περιέχονται σε κάθε στήλη και επίσης πώς θα αλληλεπιδρά με αυτά τα δεδομένα.

*** Λάβετε υπόψη ότι οι τύποι δεδομένων μπορεί να έχουν διαφορετικά ονόματα σε κάθε βάση δεδομένων. Πρέπει πάντοτε να ελέγχετε την τεκμηρίωση, ακόμη και αν το όνομα είναι το ίδιο, καθώς άλλες λεπτομέρειες μπορεί να είναι διαφορετικές, όπως το μέγεθος.**

- Για περισσότερες πληροφορίες σχετικά με τους διαφορετικούς τύπους δεδομένων σε διαφορετικά ΣΣΔΒΔ, επισκεφθείτε τον ακόλουθο ιστότοπο: https://www.w3schools.com/sql/sql_datatypes.asp

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

ΕΥΧΑΡΙΣΤΟΥΜΕ!

ΕΠΟΜΕΝΟ ΚΕΦΑΛΑΙΟ: Αναφορές στην SQL Για περισσότερες πληροφορίες, πατήστε εδώ

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