



# Εισαγωγή στις Βάσεις Δεδομένων

**Αθ. Ανδρούτσος**



# Βάση Δεδομένων

## Βάσεις Δεδομένων

- Βάση Δεδομένων (Database) είναι μία οργανωμένη συλλογή από δεδομένα
- Πιο συγκεκριμένα, Βάση Δεδομένων (ΒΔ) είναι μία συλλογή από **αρχεία** (π.χ. Καθηγητές, Μαθητές, Προϊόντα, Παραγγελίες, κλπ.) τα οποία περιέχουν **εγγραφές**
- Τα αρχεία σε μία ΒΔ που αποθηκεύονται με συστηματικό και οργανωμένο τρόπο και επιτρέπουν τη διενέργεια πράξεων όπως **εισαγωγή, διαγραφή, ενημέρωση** και κυρίως **αναζήτηση** των εγγραφών



# DB Management Systems

## Βάσεις Δεδομένων

- Δύο θέματα θα πρέπει να εξετάσουμε στο χώρο των Βάσεων Δεδομένων:
  1. Σχεδιασμός Βάσεων Δεδομένων
  2. Υλοποίηση ΒΔ σε ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων
- **Σύστημα Διαχείρισης Βάσεων Δεδομένων (Database Management System – DBMS)** είναι ένα πρόγραμμα που μας δίνει τη δυνατότητα να ορίζουμε Βάσεις Δεδομένων, να ορίζουμε χρήστες με δικαιώματα πρόσβασης στις ΒΔ, να μπορούμε να εισάγουμε / ενημερώνουμε / διαγράφουμε εγγραφές καθώς και να αναζητούμε εγγραφές σε μία Βάση Δεδομένων



# Σχεδιασμός Βάσεων Δεδομένων (1)

Βάσεις Δεδομένων

- Η πιο σημαντική πράξη που μας δίνεται από τα συστήματα Βάσεων Δεδομένων είναι η αναζήτηση
- Τα δεδομένα όμως στον πραγματικό κόσμο συνήθως δεν είναι οργανωμένα με τρόπο ώστε να μπορούν να είναι εύκολα αναζητήσιμα
- Ο σχεδιασμός Βάσεων Δεδομένων αφορά την οργάνωση και αποθήκευση των δεδομένων με τέτοιο τρόπο ώστε να μειώνεται ο πλεονασμός (redundancy - να μην αποθηκεύονται δηλαδή τα ίδια δεδομένα δύο και τρεις φορές) αλλά και να εξασφαλίσει τη συνέπεια και ακεραιότητα της ΒΔ



# Σχεδιασμός Βάσεων Δεδομένων (2)

Βάσεις Δεδομένων

- Σε σωστά σχεδιασμένες και οργανωμένες Βάσεις Δεδομένων, τα δεδομένα θα μπορούν να εισάγονται, ενημερώνονται και διαγράφονται χωρίς ανωμαλίες που θα επηρέαζαν την ακεραιότητα και συνέπεια της Βάσης Δεδομένων
- Ενώ οι αναζητήσεις (Queries) θα εκτελούνται γρήγορα και αποτελεσματικά για μεγάλους όγκους δεδομένων



# Σχεδιασμός Βάσεων Δεδομένων (3)

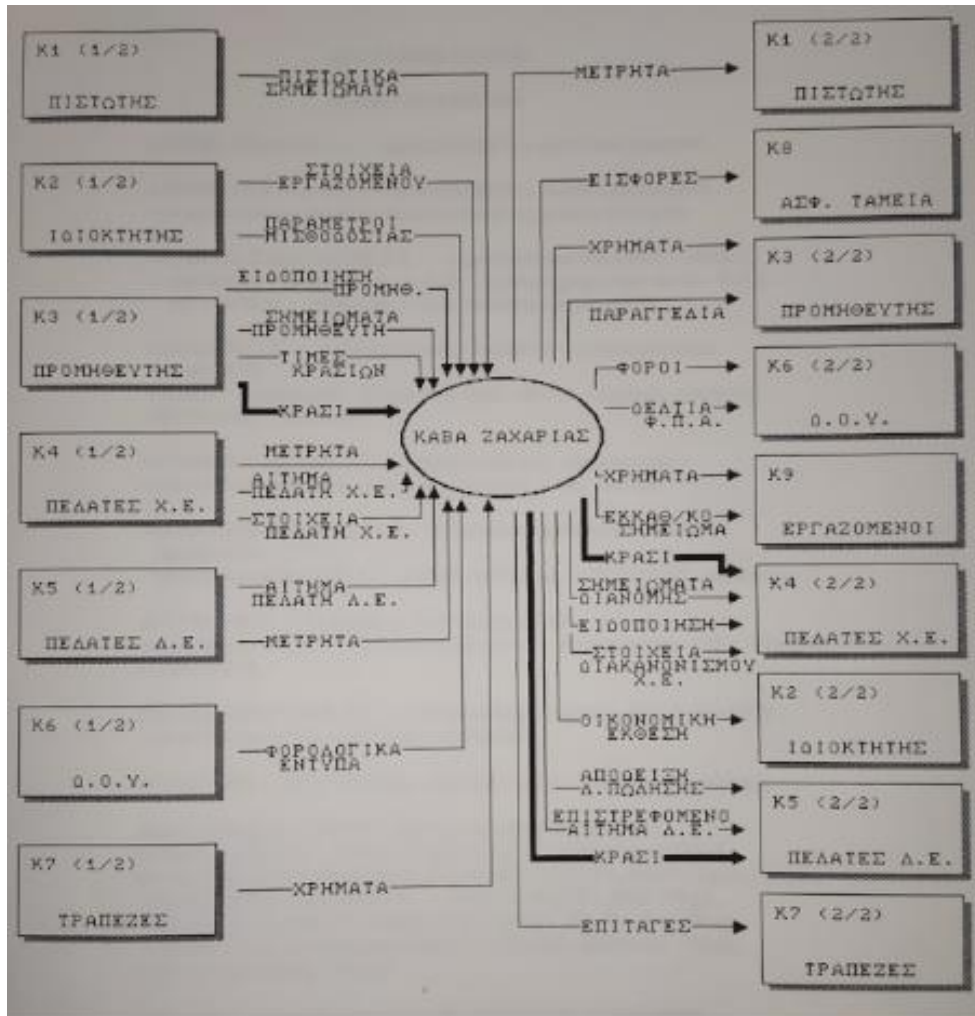
Βάσεις Δεδομένων

- Τυπικά ο σχεδιασμός μίας ΒΔ ξεκινά από την ανάλυση ενός συστήματος που αποτυπώνεται σε διαγράμματα ροής δεδομένων και conceptual models
- Στη συνέχεια μετατρέπουμε το conceptual model μέσα από μία συστηματική διαδικασία σε ERD (Entity-Relationship Diagram)



# Διάγραμμα ροής δεδομένων

## Βάσεις Δεδομένων



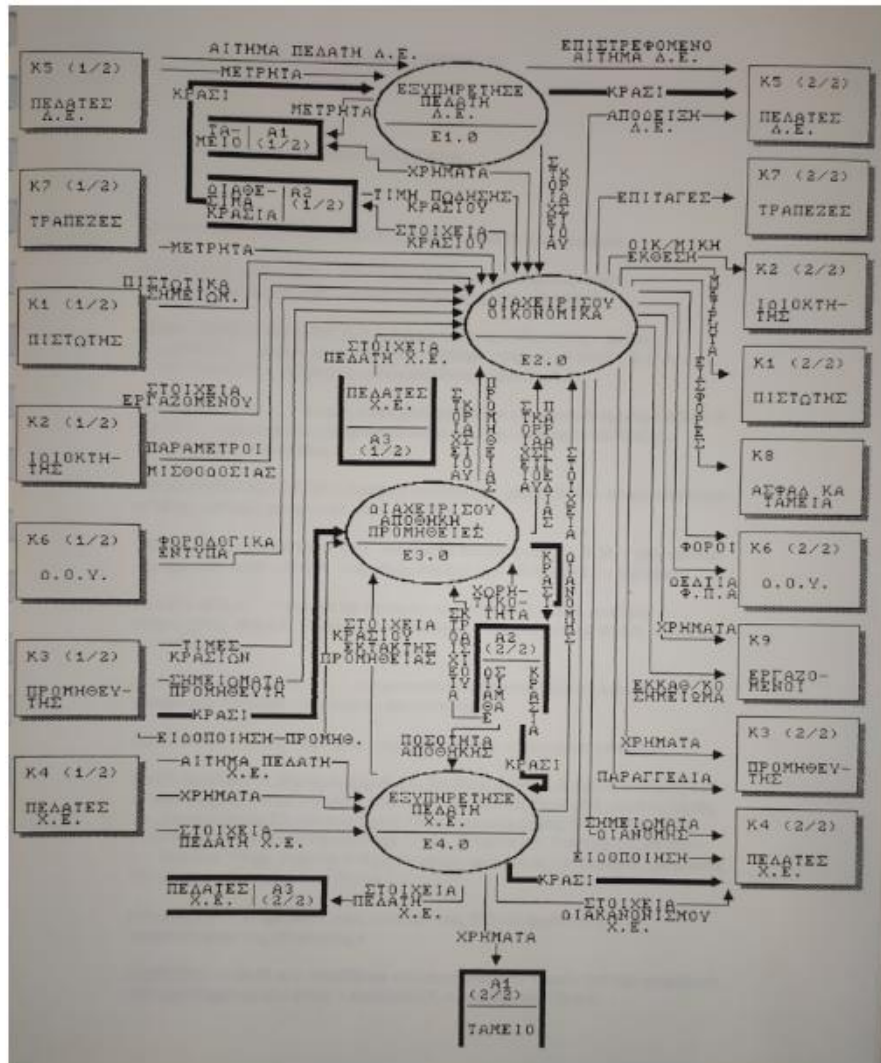
- Ένα διάγραμμα ροής δεδομένων είναι μέρος της ανάλυσης ενός συστήματος και προκύπτει με συνεντεύξεις στα μέλη τους συστήματος (εργαζόμενους)
- Καταγράφει Οντότητες που δίνουν input στο σύστημα και Οντότητες που λαμβάνουν output από το σύστημα ενώ καταγράφει και τις ροές δεδομένων
- Οι ροές δεδομένων συνοδεύονται από ένα λεξικό όρων (βλ. μεθεπόμενη διαφάνεια)
- Το παρόν είναι ένα Διάγραμμα Πλαίσιο (Μηδενικού Επιπέδου)





# Διάγραμμα Επιπέδου 1

## Βάσεις Δεδομένων



- Πρόκειται για πιο αναλυτικό διάγραμμα ροής δεδομένων που καταγράφει και τα υποσυστήματα / διαδικασίες που επεξεργάζονται τα δεδομένα
- Καταγράφει τις ροές ανά υποσύστημα, ενώ μπορεί να υπάρχει και σε μεγαλύτερο βάθος εκλέπτυνση μιας και τα υποσυστήματα αποτελούνται από μικρότερα υποσυστήματα
- Οι ροές δεδομένων συνοδεύονται από ένα λεξικό όρων (βλ. επόμενη διαφάνεια)





# Λεξικό Δεδομένων

## Βάσεις Δεδομένων

- Το Λεξικό Δεδομένων αναλύει τις ροές δεδομένων. Τελικά τα στοιχεία των **Οντοτήτων** και των **Ροών Δεδομένων** θα αποτελέσουν τα **αρχεία του συστήματος**
- Τα αρχεία και οι σχέσεις μεταξύ τους καταγράφονται και δημιουργείται ένα πρώτο conceptual model (κατανοούμε το σύστημα, τις διαδικασίες, τις απαιτήσεις, τα αρχεία και τις σχέσεις μεταξύ τους) και στη συνέχεια δημιουργούμε ένα σχεσιακό μοντέλο / διάγραμμα των δεδομένων που αποτελεί και το σχεδιασμό της ΒΔ.
- Στη συνέχεια θα δούμε το σχεσιακό μοντέλο δεδομένων και πως δημιουργείται, ώστε να σχεδιάζουμε ΒΔ, που είναι το 1<sup>ο</sup> βήμα στη διαδικασία σχεδιασμού και υλοποίησης μιας ΒΔ

### ΛΕΞΙΚΟ ΔΕΔΟΜΕΝΩΝ

ΑΠΗΜΑ ΠΕΛΑΤΗ Α.Ε. = ΕΙΔΟΣ ΚΡΑΣΙΟΥ + ποσότητα κρασιού

ΑΠΗΜΑ ΠΕΛΑΤΗ Χ.Ε. = ΕΙΔΟΣ ΚΡΑΣΙΟΥ + ποσότητα κρασιού + ονοματεπώνυμο πελάτη + διακανονισμένος χρόνος διανομής

ΑΠΟΔΕΙΞΗ ΕΙΣΠΡΑΞΗΣ Χ.Ε. = ονοματεπώνυμο ιδιοκτήτη + αριθμός + απόδειξης + ημερομηνία + Α.Φ.Μ. + καθαρό χρηματικό ποσό + Φ.Π.Α. + ΔΙΕΥΘΥΝΣΗ + τηλέφωνο + ονοματεπώνυμο πελάτη Χ.Ε.

ΑΠΟΔΕΙΞΗ Α.Ε. = \* έντυπο που περιλαμβάνει \* ονοματεπώνυμο ΙΔΙΟΚΤΗΤΗ + Α.Φ.Μ. + ημερομηνία + χρηματικό ποσό + Φ.Π.Α. + αριθμός απόδειξης + ΔΙΕΥΘΥΝΣΗ + τηλέφωνο

ΑΠΟΔΕΙΞΗ ΠΙΣΤΩΤΩΝ = Αριθμός απόδειξης + ημερομηνία + ονοματεπώνυμο πελάτη + καθαρό χρηματικό ποσό + Φ.Π.Α.

ΑΠΟΔΕΙΞΗ ΠΡΟΜΗΘΕΥΤΗ = ονομασία προμηθευτή + Α.Φ.Μ. + ημερομηνία + χρηματικό ποσό + Φ.Π.Α. + αριθμός απόδειξης + ΔΙΕΥΘΥΝΣΗ + τηλέφωνο

ΔΕΛΤΙΑ Φ.Π.Α. = \* έντυπα με τα οποία αποδίδεται το Φ.Π.Α. από τις επιχειρήσεις στη Δ.Ο.Υ. και περιλαμβάνουν \* Δίμηνο έντυπο Φ.Π.Α. \* Ετήσιο έντυπο Φ.Π.Α.

ΔΕΛΤΙΟ ΑΠΟΣΤΟΛΗΣ Χ.Ε. = ονοματεπώνυμο πελάτη + ΕΙΔΟΣ ΚΡΑΣΙΟΥ + ποσότητα κρασιού + αφετηρία + προορισμός

ΔΙΕΥΘΥΝΣΗ = οδός + αριθμός + περιοχή + πόλη + ταχυδρομικός κώδικας

ΕΙΔΟΣ ΚΡΑΣΙΟΥ = [Ερυθρός ξηρός 5 λ., Ερυθρός ξηρός 3 λ., Ερυθρός ξηρός 2 λ., Ερυθρός ξηρός 750 μλ., Ερυθρός ξηρός 375 μλ., Λευκός ξηρός 5 λ., Λευκός ξηρός 3 λ., Λευκός ξηρός 2 λ., Λευκός ξηρός 750 μλ., Λευκός ξηρός 375 μλ., Ροζέ 5 λ., Ροζέ 3 λ., Ροζέ 2 λ., Ροζέ 750 μλ., Ροζέ 375 μλ., Ρετσίνα 5 λ., Ρετσίνα 3 λ., Ρετσίνα 2 λ., Ρετσίνα 750 μλ., Ρετσίνα 375 μλ., Ερυθρός ημίγλυκος 2 λ., Ερυθρός ημίγλυκος 750 μλ., Ερυθρός ημίγλυκος 375 μλ.]

Ειδοποίηση = \* ενημέρωση προς τους πελάτες Χ.Ε. σε περίπτωση καθυστέρησης της εξόφλησης \*

ΕΙΣΦΟΡΕΣ = \* ποσό που αποδίδεται στα ασφαλιστικά ταμεία για την ασφάλιση των εργαζομένων και είναι \* εισφορά Ι.Κ.Α. + εισφορά Τ.Ε.Β.Ε.

ΕΚΚΑΘΑΡΙΣΤΙΚΟ Δ.Ο.Υ. = \* Αναλυτική απόδειξη πληρωμής φόρων από τη



# Σχεσιακό Μοντέλο

Βάσεις Δεδομένων

- Υπάρχουν αρκετοί διαφορετικοί τρόποι περιγραφής του σχήματος μιας ΒΔ, που είναι γνωστοί ως Μοντέλα ΒΔ ή Μοντέλα Δεδομένων
- Το πιο κοινό σε χρήση μοντέλο δεδομένων σήμερα είναι το **Σχεσιακό Μοντέλο** (Relational Model) που προτάθηκε από τον E.F Codd το 1970)
- Το σχεσιακό μοντέλο στοχεύει στην μείωση του πλεονασμού σε μία ΒΔ, στην συνέπεια και ορθότητα των δεδομένων μιας ΒΔ



# Σχεσιακό Μοντέλο

## Βάσεις Δεδομένων

### Teachers and Students File

TEACHER ID	TEACHER FIRSTNAME	TEACHER LASTNAME	STUDENT ID	STUDENT FIRSTNAME	STUDENT LASTNAME
1	Athana	Androutsos	1	Anna	Giannou
1	Athana	Androutsos	2	John	Papa

### Teachers

### Students

TEACHER ID	TEACHER FIRSTNAME	TEACHER LASTNAME
1	Athana	Androutsos

TEACHER ID	STUDENT ID	STUDENT FIRSTNAME	STUDENT LASTNAME
1	1	Anna	Giannou
1	2	John	Papa

Diagram showing a relationship between the Teachers table and the Students table. An arrow points from the Teachers table to the Students table, indicating a foreign key relationship.

- Το Teachers and Students File θα μπορούσε να είναι ένα αρχείο κατά τη φάση της ανάλυσης, που όμως έχει πλεονασμό γιατί ο ίδιος Teacher καταγράφεται δύο φορές
- Μπορούμε να μειώσουμε τον πλεονασμό στο Σχεσιακό Μοντέλο Δεδομένων ορίζοντας δύο οντότητες **Teachers** και **Students** και μία **σχέση** μεταξύ τους ώστε να μπορεί να συνδέεται ένας Καθηγητής με πολλούς μαθητές
- Η σύνδεση γίνεται μέσω του **TEACHER\_ID**



# Πίνακες / Αρχεία

## Βάσεις Δεδομένων

### STUDENTS

ID	FIRSTNAME	LASTNAME
1	Ανδρέας	Ανδρούτσος
2	Άννα	Ανδρούτσου
3	Ελένη	Γιαννούτσου

- Στο σχεσιακό μοντέλο κατά τη φάση του σχεδιασμού μίας ΒΔ (Λογικό Σχήμα της ΒΔ) αναφερόμαστε σε **πίνακες (tables)**
- Κατά τη φάση της υλοποίησης (Φυσικό Σχήμα) αντίστοιχα αναφερόμαστε σε **αρχεία (Files)**
- Για παράδειγμα το αντικείμενο αριστερά στο διάγραμμα είναι ένας **πίνακας STUDENTS**, ο οποίος στη φάση της υλοποίησης θα είναι το **πραγματικό αρχείο STUDENST**



# Πεδία / Πλειάδες

Βάσεις Δεδομένων

STUDENTS

ID	FIRSTNAME	LASTNAME
1	Ανδρέας	Ανδρούτσος
2	Άννα	Ανδρούτσου
3	Ελένη	Γιαννούτσου

- Οι πίνακες αποτελούνται από γραμμές και στήλες
- Οι στήλες ενός Πίνακα ονομάζονται ιδιότητες (**properties**) ενώ οι γραμμές ονομάζονται πλειάδες (**tuples**)
- Κατά τη φάση της υλοποίησης, στα φυσικά αρχεία, αντίστοιχα έχουμε πεδία (**fields**) και εγγραφές (**records**)



# Σχεσιακές Βάσεις Δεδομένων

## Βάσεις Δεδομένων

- Μία Σχεσιακή ΒΔ μπορεί να περιέχει πολλούς πίνακες, που κάθε ένας αποτελεί ένα δυσδιάστατο πίνακα στοιχείων, όπου όλα τα μέλη μιας **στήλης** έχουν τιμές που ανήκουν στον ίδιο τύπο δεδομένων και όλα τα μέλη μιας **γραμμής** σχετίζονται το ένα με το άλλο σχηματίζοντας μία εγγραφή
- Οι στήλες ενός πίνακα μπορεί να έχουν διάφορους τύπους δεδομένων, όπως χαρακτήρες, ακέραιοι, ημερομηνία, κλπ.



# Σχεδιασμός ΒΔ

- Σχεδιασμός μιας Βάσης Δεδομένων (Database Design) είναι μια συστηματική διαδικασία καταγραφής και παρουσίασης με τυπικές μεθόδους (διαγράμματα) των δεδομένων μιας εφαρμογής
- Ο σχεδιασμός μιας ΒΔ είναι μια σημαντική διαδικασία (ίσως η σημαντικότερη διαδικασία κατά την ανάπτυξη μιας εφαρμογής) και έχει ως **βασικό στόχο να απεικονίσει τυπικά με διαγράμματα οντοτήτων-σχέσεων ( Entity-Relationship models – ER Models) την οργάνωση και δομή της Βάσης Δεδομένων**





# ER-Model (1)

- Το βασικό διαγραμματικό εργαλείο απεικόνισης του σχήματος (DB Schema) μίας Βάσης Δεδομένων είναι το ER-Diagram, το διάγραμμα οντοτήτων-συσχετίσεων
- Οι οντότητες (entities) είναι οι **Πίνακες (Tables)** της Βάσης μας και οι συσχετίσεις είναι οι **σχέσεις (Relationships)** μεταξύ των πινάκων



# ER-Model (2)

- Για να μετασχηματιστούν τα δεδομένα του συστήματος από ένα αρχικό **εννοιολογικό μοντέλο δεδομένων** ενός οργανισμού - περιγραφή του συστήματος και των αναγκών των χρηστών, των βασικών πινάκων και των σχέσεων- σε **ER Model** -που τυπικά απεικονίζεται σε ένα ER Diagram- ακολουθούμε μία σειρά βημάτων και μερικές απλές βασικές αρχές



# Οντότητες - Σχέσεις

Βάσεις Δεδομένων

- Έστω ένας εκπαιδευτικός οργανισμός που διαχειρίζεται στοιχεία Καθηγητών (TEACHERS) και Μαθητών (Students)
- Είναι άμεσα προφανές ότι το σύστημα αποτελείται από δύο Οντότητες TEACHERS και STUDENTS
- Έστω ότι κάθε Καθηγητής διδάσκει σε πολλούς μαθητές, αλλά κάθε μαθητής διδάσκεται από ένα μόνο καθηγητή οπότε η σχέση είναι Ένα-Προς-Πολλά, από τον TEACHERS προς τους STUDENTS
- Αυτή είναι μια απλή περιγραφή του συστήματος



# Στόχος ER Model

- Σκοπός της μετάβασης από την περιγραφή του συστήματος και το εννοιολογικό μοντέλο στο ER Model είναι να:
  - Μειώσει την πολυπλοκότητα και τον πλεονασμό (redundancy) των δεδομένων της Βάσης
  - Να εξαλείψει τυχόν ανωμαλίες που θα μπορούσαν να προκύψουν κατά την εισαγωγή, ενημέρωση, διαγραφή, αναζήτηση δεδομένων



# Κανόνες ER Model (1)

- Ένα ER-Model αποτελείται από Πίνακες (Αρχεία κατά τη φάση της υλοποίησης) και Σχέσεις Πινάκων. Μερικοί κανόνες για τη δημιουργία του ER-Model είναι:
  - Κάθε Πίνακας έχει ένα **διαφορετικό όνομα, συνήθως στον πληθυντικό** (π.χ. TEACHERS, STUDENTS, κλπ)
  - Κάθε πίνακας έχει γραμμές -εγγραφές κατά τη φάση της υλοποίησης. Π.χ. το 1 Ανδρέας Ανδρούτσος είναι μία εγγραφή (record)

STUDENTS		
ID	FIRSTNAME	LASTNAME
1	Ανδρέας	Ανδρούτσος
2	Άννα	Ανδρούτσου
3	Ελένη	Γιαννούτσου



# Κανόνες ER Model (2)

- Κάθε Πίνακας έχει στήλες/πεδία (π.χ. ID, FIRSTNAME, LASTNAME), που έχουν μονές τιμές (όχι πολλαπλές, δεν μπορούμε να έχουμε πεδίο ADDRESS γιατί έχει πολλαπλές τιμές, ΟΔΟΣ, ΑΡΙΘΜΟΣ κλπ. Θα πρέπει να έχουμε STREET, NUMBER, που περιέχουν απλές μονές τιμές)
- Για παράδειγμα στον Πίνακα STUDENTS τα πεδία ID, FIRSTNAME, LASTNAME περιέχουν απλές/μονές τιμές, όχι σύνθετες/πολλαπλές

STUDENTS		
ID	FIRSTNAME	LASTNAME
1	Ανδρέας	Ανδρούτσος
2	Άννα	Ανδρούτσου
3	Ελένη	Γιαννούτσου



# Πρωτεύον Κλειδί

- Πρέπει επίσης σε κάθε Πίνακα να υπάρχει ένα **πεδίο/στήλη που να έχει μοναδικές τιμές** ώστε να μπορούμε να αναγνωρίζουμε μοναδικά κάθε εγγραφή του Πίνακα (π.χ. Αριθμός Ταυτότητας, Αριθμός Μητρώου ή ακόμα και ένα ID με ένα απλό ακέραιο αριθμό χωρίς σημασιολογία που απλά να αυξάνει κατά ένα σε κάθε νέα εγγραφή)
- Το πεδίο αυτό ονομάζεται **ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ (Primary Key)**. Π.χ. στον Πίνακα STUDENTS, το Πρωτεύον Κλειδί είναι το ID, που απλά αυξάνει κατά ένα σε κάθε εγγραφή

STUDENTS		
ID	FIRSTNAME	LASTNAME
1	Ανδρέας	Ανδρούτσος
2	Άννα	Ανδρούτσου
3	Ελένη	Γιαννούτσου





# Παράδειγμα Σχέσεων (2)

## Βάσεις Δεδομένων

- Θα πρέπει επίσης να εξετάσουμε αν η σχέση του Πρωτεύοντος Κλειδιού με τα υπόλοιπα πεδία του Πίνακα είναι μοναδική (υπάρχει συναρτησιακή εξάρτηση, όπως λέμε) ή είναι σχέση πολλαπλών τιμών (εξάρτηση πολλαπλών τιμών, όπως λέμε). Δεν επιτρέποντας εξαρτήσεις πολλαπλών τιμών σε ένα Πίνακα
- Για παράδειγμα αν ξέρουμε το ID του Καθηγητή μπορούμε με μοναδικό τρόπο να οδηγηθούμε στα υπόλοιπα στοιχεία του Καθηγητή, αν όμως ξέρουμε το ID του Καθηγητή δεν μπορούμε με μοναδικό τρόπο να οδηγηθούμε στα στοιχεία ενός συγκεκριμένου Μαθητή (μιας και ένας Καθηγητής σχετίζεται με πολλούς μαθητές), οπότε δεν θα μπορούσαν οι μαθητές να είναι μέρος του Πίνακα των Καθηγητών



# Σχέσεις πινάκων

## Βάσεις Δεδομένων

- Υπάρχουν τριών ειδών σχέσεις μεταξύ πινάκων:
  - **1:1** (Ένα-προς-ένα) όπου μία εγγραφή ενός πίνακα σχετίζεται με μία εγγραφή ενός άλλου πίνακα
  - **1:N** (Ένα-προς-πολλά) όπου μία γραμμή του ενός πίνακα σχετίζεται με πολλές γραμμές ενός άλλου πίνακα
  - **N:N** (Πολλά-προς-πολλά) όπου μία εγγραφή του ενός πίνακα σχετίζεται με πολλές εγγραφές ενός άλλου πίνακα, αλλά και μία εγγραφή του άλλου πίνακα σχετίζεται με πολλές εγγραφές του αρχικού



# Σχέσεις ένα προς πολλά

Βάσεις Δεδομένων












- Οι σχέσεις ένα προς πολλά είναι οι πιο σημαντικές σχέσεις σε ένα ER-Model
- Τα ορθά ER-Models έχουν κυρίως **σχέσεις ένα-προς-πολλά**
- Αυτό σημαίνει ότι οι δύο άλλοι τύποι σχέσεων, 1:1 (ένα-προς-ένα) και N:N (πολλά-προς-πολλά) θα πρέπει να αναχθούν τελικά σε σχέση 1:N



# Συμβολισμοί Σχέσεων Πινάκων

Βάσεις Δεδομένων

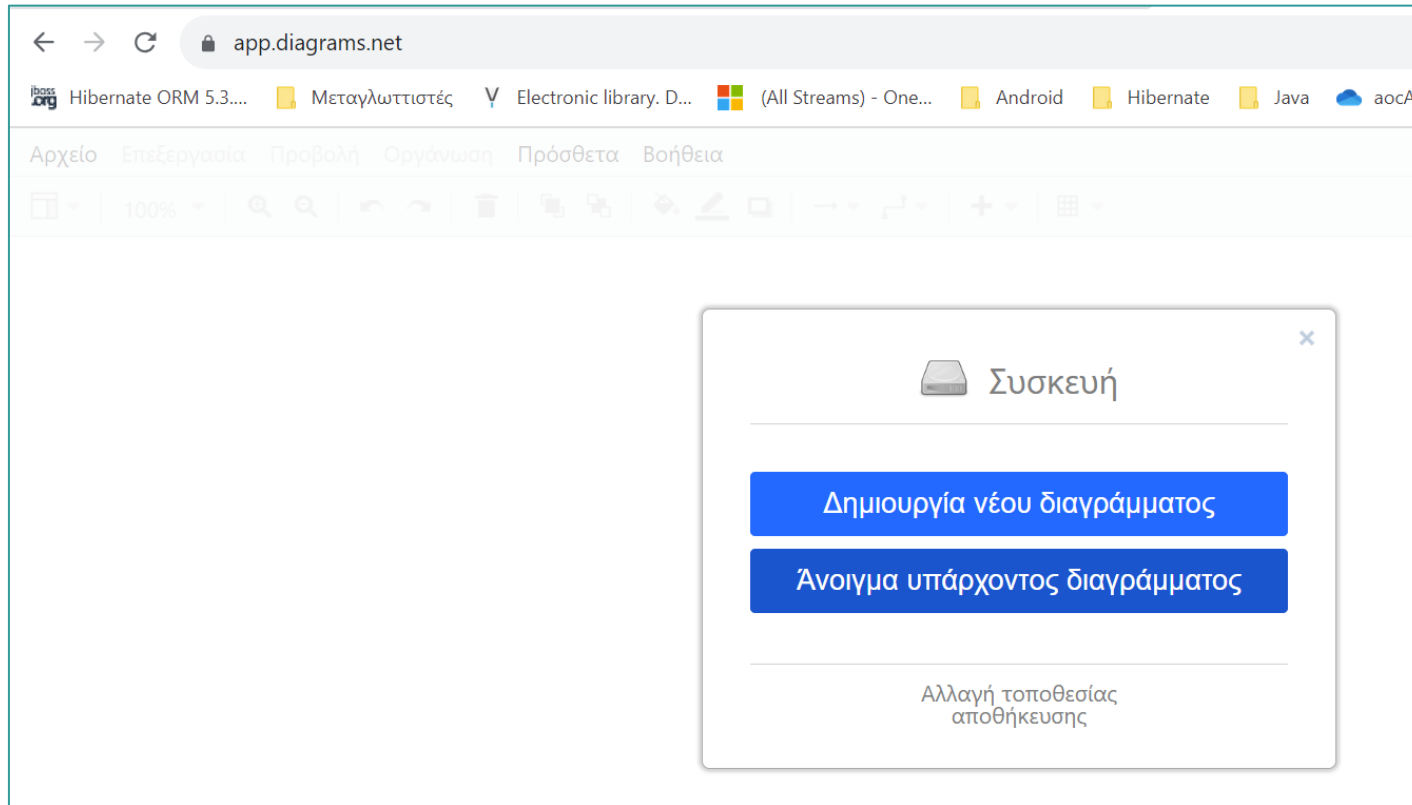
## Data Model Notations

Notation	Barker Notation	Bachman Notation	Information Engineering
Zero or one			
One only			
Zero or more			
One or more			
Primary Key/Unique key	#	P	



# draw.io

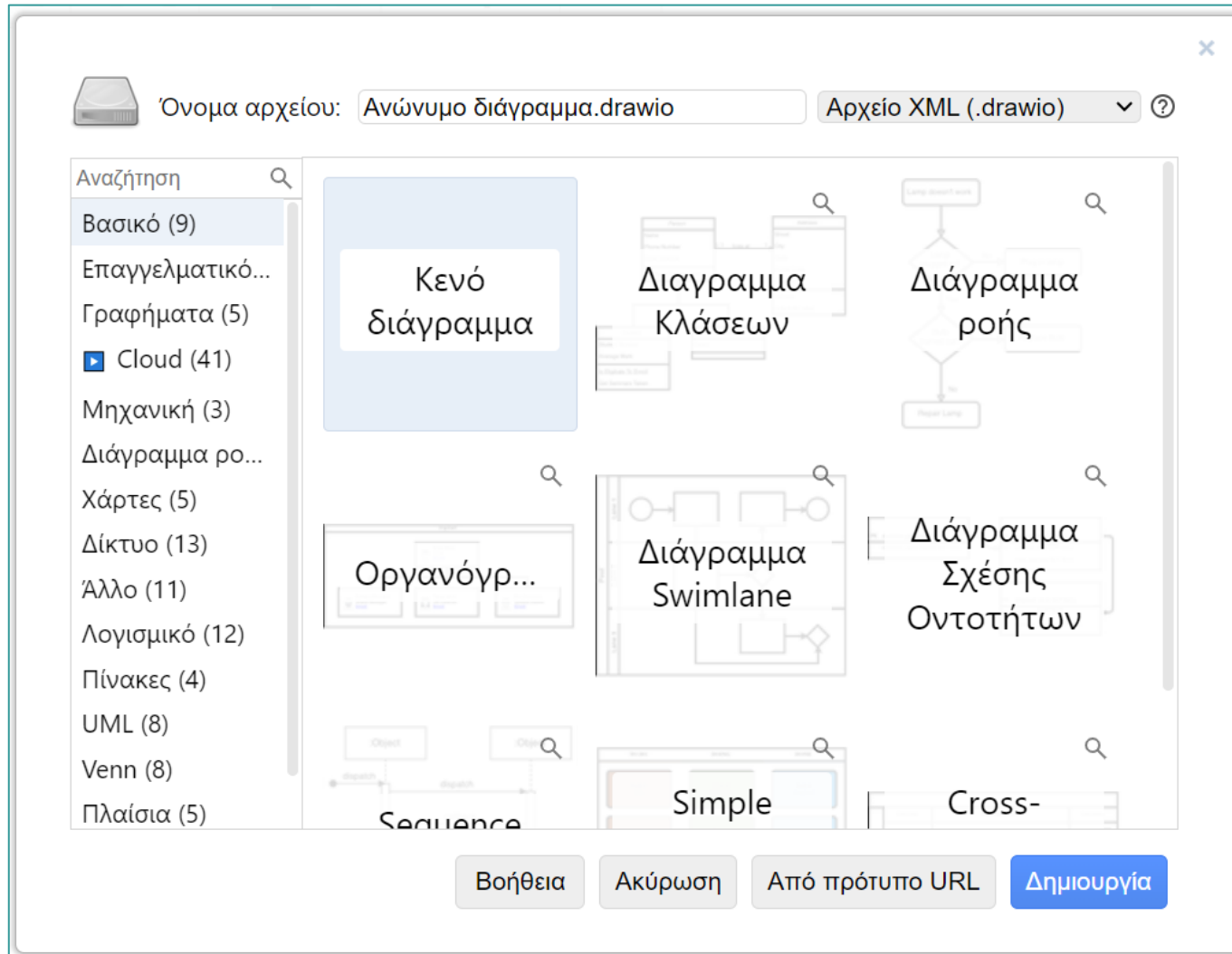
Βάσεις Δεδομένων



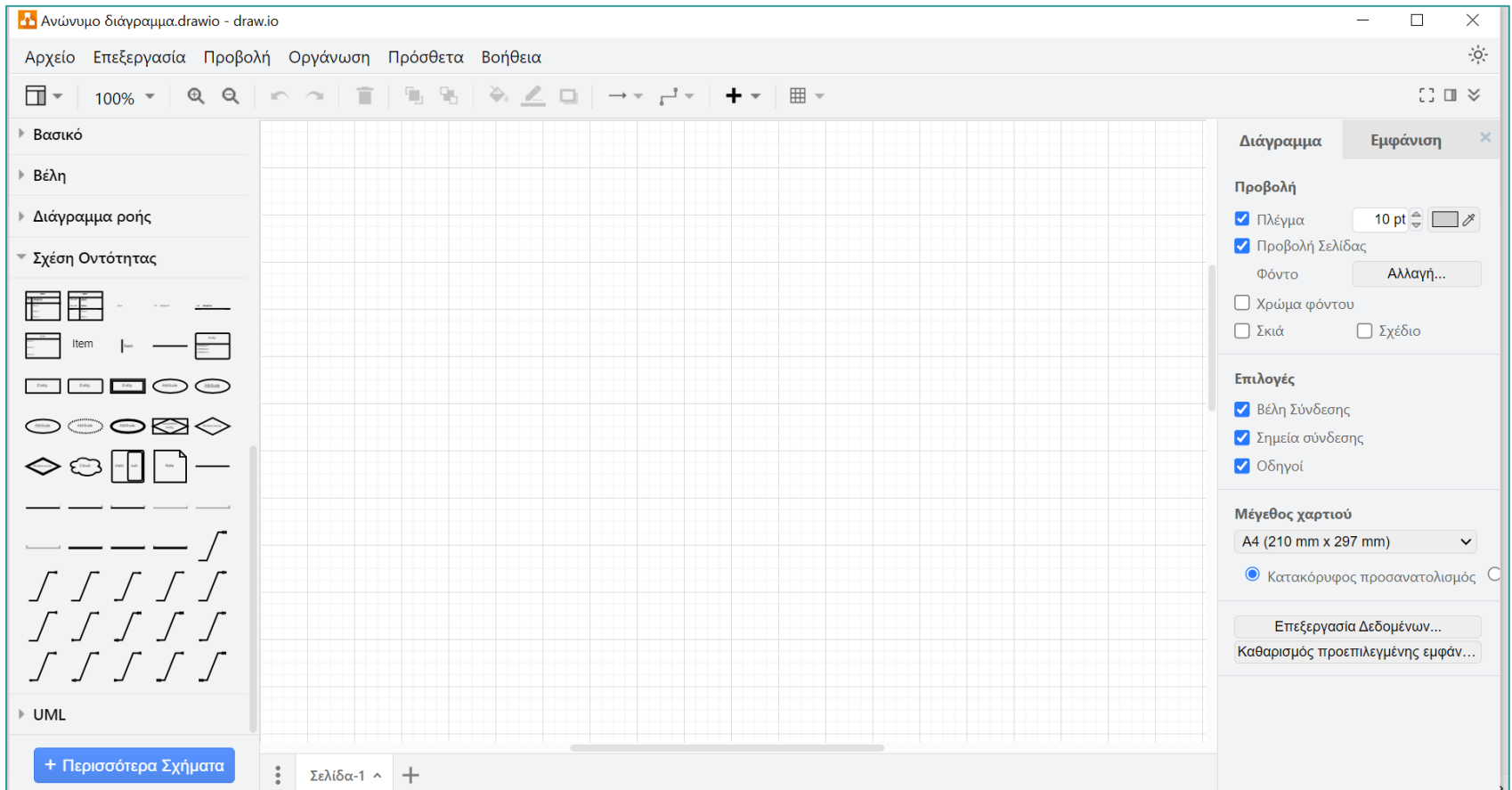
- Το draw.io είναι ένα online διαγραμματικό εργαλείο για τη δημιουργία ER-Models. <https://app.diagrams.net/>



# ER Diagram (ERD)



- Επιλέγουμε Κενό διάγραμμα και δημιουργία



- Επιλέγουμε από το αριστερό μενού, την κατηγορία 'Σχέση Οντότητας'





# Teachers - Students

Βάσεις Δεδομένων

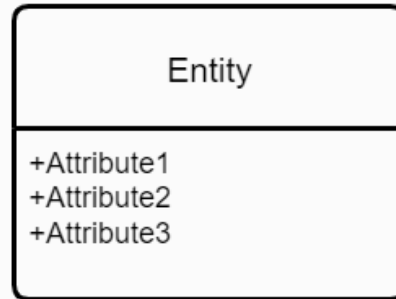
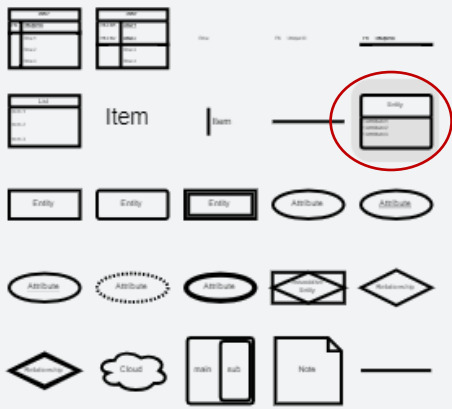
- Θα εισάγουμε δύο πίνακες, τον πίνακα TEACHERS και τον πίνακα STUDENTS
- Θα εισάγουμε σε κάθε πίνακα τα πεδία του πίνακα
- Θα εισάγουμε τη σχέση μεταξύ των δύο πινάκων. Εδώ υποθέτουμε ότι ο κάθε Teacher σχετίζεται με πολλούς Students ενώ ο κάθε Student σχετίζεται με ένα Teacher, οπότε η σχέση είναι 1:N (ένα προς πολλά) από τον Teacher προς τον Student



# Πίνακας TEACHERS

Βάσεις Δεδομένων

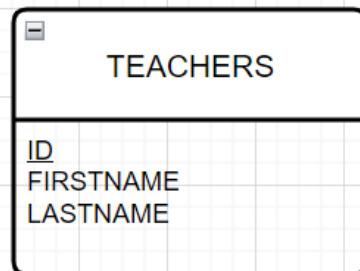
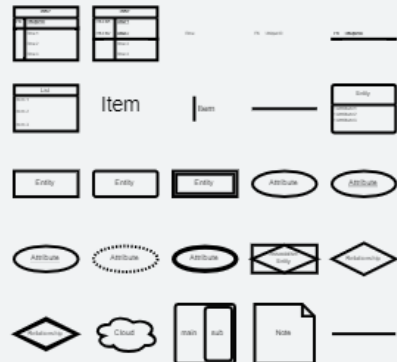
## Σχέση Οντότητας



Entity

- Εισάγουμε με κλικ από το αριστερό μενού ένα απλό πίνακα (έχει επισημανθεί με κόκκινο κύκλο) και στη συνέχεια αλλάζουμε (με διπλό κλικ πάνω στο όνομα) τα υπάρχοντα ονόματα

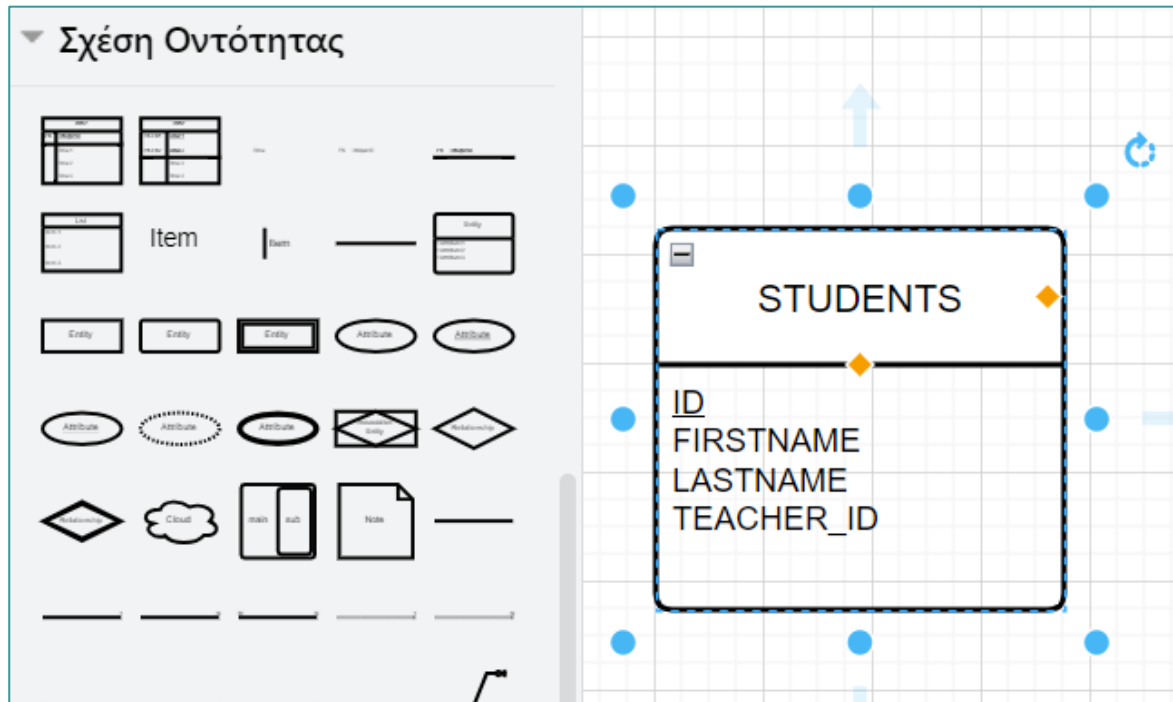
## Σχέση Οντότητας



- Υπογραμμίζουμε το πρωτεύον κλειδί ID (με CTRL + u)



# Ξένο Κλειδί



- Το ίδιο κάνουμε και με τον πίνακα STUDENTS. Εισάγουμε ένα πίνακα από το μενού, μετά εισάγουμε τα πεδία του Student και μετά εισάγουμε ένα ακόμα πεδίο, το TEACHER\_ID. Το TEACHER\_ID υλοποιεί τη σχέση TEACHERS-STUDENTS και ονομάζεται **Ξένο Κλειδί**

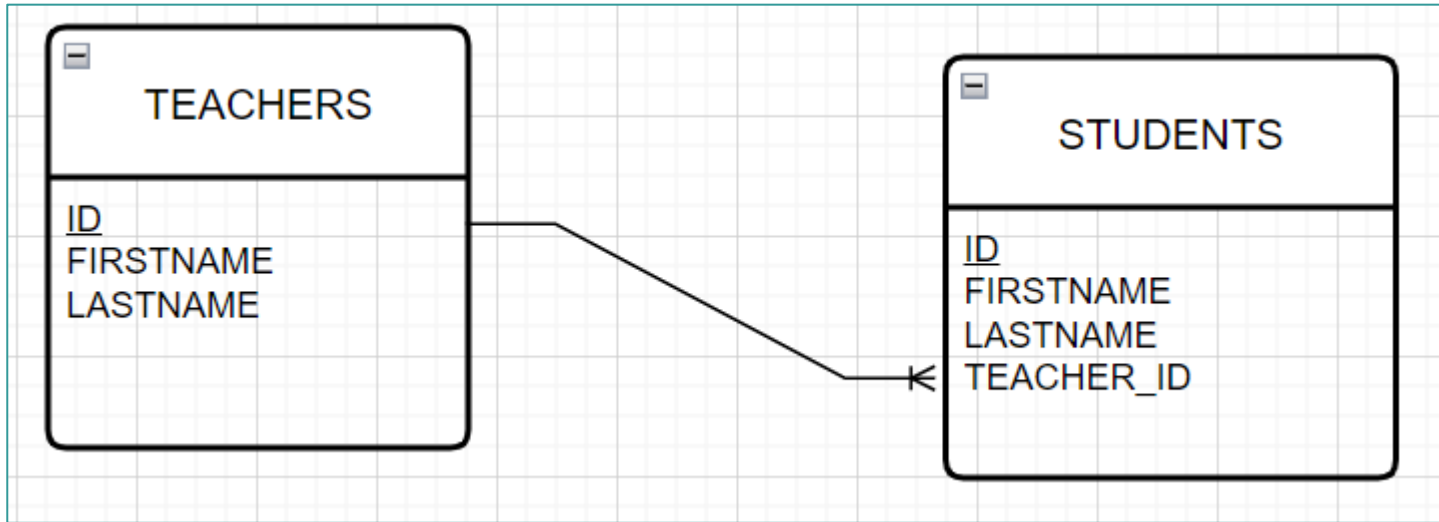


# Ξένα κλειδιά (1)

- Τα ξένα κλειδιά, υλοποιούν τις σχέσεις μεταξύ πινάκων. **Εισάγονται πάντα στο μέρος του προς-πολλά**, όπως εδώ εισάγαμε στο μέρος του Student το TEACHER\_ID που αντιστοιχεί στο ID του πίνακα TEACHERS
- Τα ξένα κλειδιά αντιστοιχούν στο πρωτεύον κλειδί του βασικού πίνακα
- Για εγγραφές που συνδέονται, η τιμή του πρωτεύοντα κλειδιού του βασικού πίνακα **πρέπει να είναι η ίδια με την τιμή του ξένου κλειδιού του προς-πολλά Πίνακα**



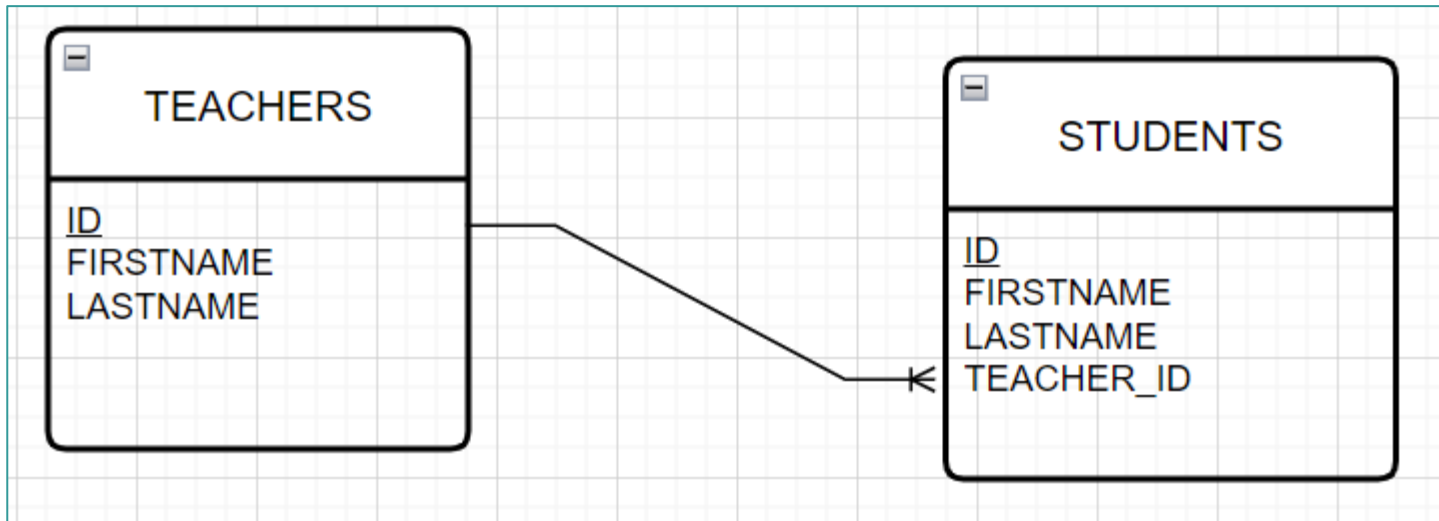
# Ξένα κλειδιά (2)



- Για παράδειγμα στην περίπτωση του παραδείγματος, αν ένας καθηγητής έχει  $ID = 1$  και ένας μαθητής έχει  $TEACHER\_ID = 1$ , τότε σημαίνει πως αυτές οι δύο εγγραφές συσχετίζονται, ότι δηλαδή ο καθηγητής διδάσκει αυτόν τον μαθητή (π.χ. είναι στην τάξη του) και ο μαθητής έχει ως καθηγητή, τον καθηγητή με κωδικό 1



# Σχέση 1:N



- Έτσι υλοποιούνται οι σχέσεις 1:N, εισάγοντας ένα ξένο κλειδί στο μέρος του προς πολλά



# Κανονικοποίηση

## Βάσεις Δεδομένων

- Τεχνικά και σύμφωνα με τη Σχεσιακή Θεωρία για να είναι ένα ER Model ορθό θα πρέπει να περνάει από μία διαδικασία Κανονικοποίησης
- Στόχος της διαδικασίας της κανονικοποίησης είναι η μείωση του όγκου των δεδομένων που αποθηκεύουμε, αποφεύγοντας να έχουμε επαναλαμβανόμενα δεδομένα (redundancy) στον ίδιο ή άλλους πίνακες



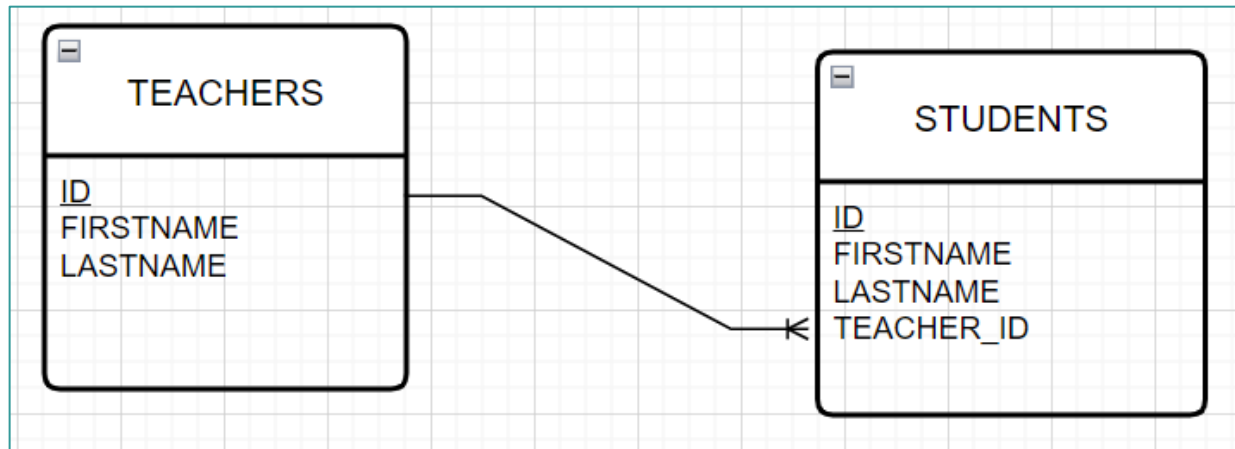


# Κανονικοποίηση

- Η κανονικοποίηση είναι μία διαδικασία που περιλαμβάνει πέντε κανονικές μορφές με βάση τις οποίες εξετάζουμε αν ένα ER Model είναι ορθό
- Συνήθως ένα ER Model αρκεί να είναι στην 3<sup>η</sup> κανονική μορφή



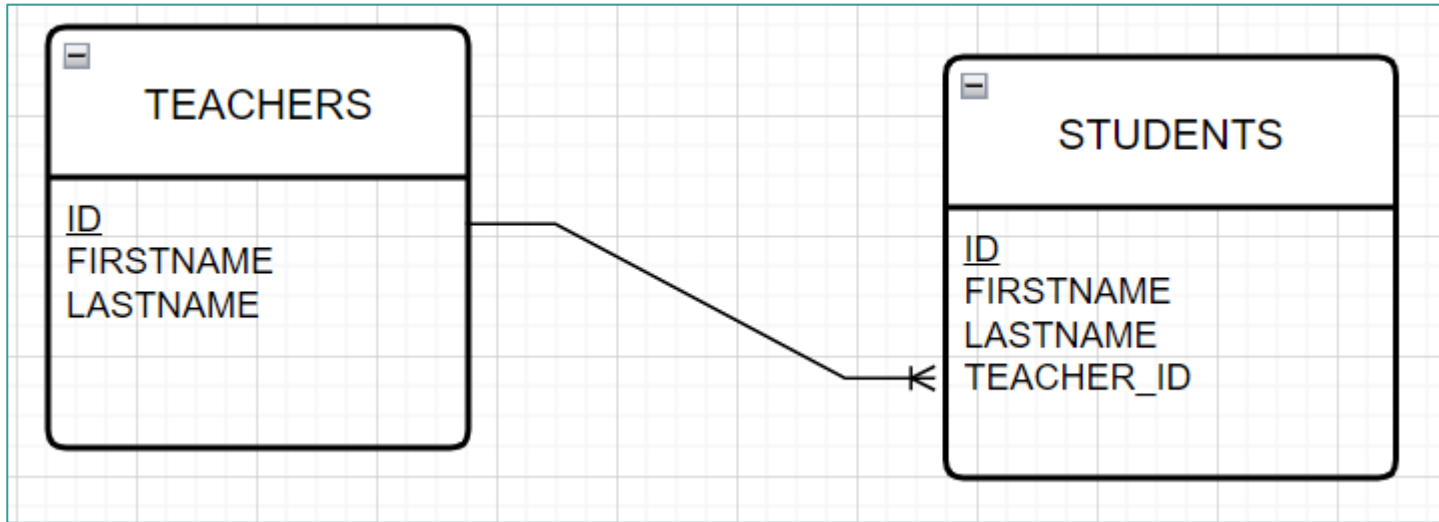
# 1<sup>st</sup> , 2<sup>nd</sup> Normal Form (1NF, 2NF)



- Ο κάθε ένας από τους δύο παραπάνω πίνακες είναι στη 1<sup>η</sup> και 2<sup>η</sup> κανονική μορφή (1NF, 2NF) γιατί:
  - Στην 1NF, υπάρχει ένα υποψήφιο πρωτεύον κλειδί (TEACHER\_ID και STUDENT\_ID) ενώ δεν υπάρχουν σύνθετα πεδία (όπως ονοματεπώνυμο) αλλά απλά πεδία, όπως όνομα και επώνυμο, και
  - Στην 2NF, γιατί δεν υπάρχουν συναρτησιακές εξαρτήσεις πολλαπλών τιμών, σε κάποιο υποσύνολο του υποψήφιου πρωτεύοντος κλειδιού, μιας και το πρωτεύον κλειδί δεν είναι σύνθετο



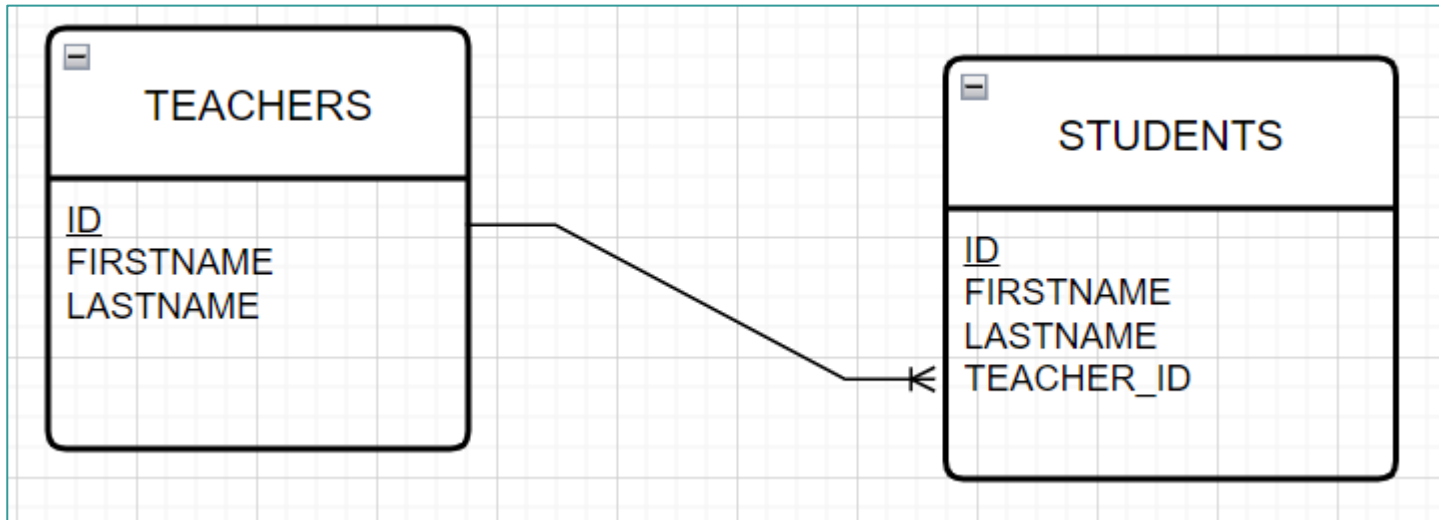
# 3NF - Πρωτεύον κλειδί (1)



- Για να πάμε στην 3<sup>η</sup> Κανονική Μορφή που είναι και ο βασικός στόχος στην διαδικασία μετασχηματισμού του εννοιολογικού μοντέλου σε ER-Model, θα πρέπει να εξασφαλίσουμε ότι ο σε κάθε πίνακα, **το κλειδί και μόνο το κλειδί**, μοναδικοποιεί την κάθε εγγραφή του



# Πρωτεύον κλειδί (2)



- Το πεδίο ID στον πίνακα TEACHERS μοναδικοποιεί την κάθε εγγραφή, επειδή κάθε καθηγητής έχει ένα μοναδικό ID. Το πεδίο αυτό **ονομάζεται Πρωτεύον Κλειδί (Primary Key)**
- Το ίδιο ισχύει και για το πεδίο ID, στον πίνακα STUDENTS
- Τα πρωτεύοντα κλειδιά, στο διάγραμμα, τα υπογραμμίζουμε



# Υποψήφια Πρωτεύοντα

Βάσεις Δεδομένων

- Αν ένας πίνακας έχει περισσότερα από ένα **υποψήφια πρωτεύοντα κλειδιά**, δηλαδή περισσότερα από ένα πεδία που μοναδικοποιούν την κάθε εγγραφή, π.χ. ΑΦΜ, Αριθμός Ταυτότητας, κλπ., μπορούμε να επιλέξουμε ένα από αυτά ως πρωτεύον κλειδί
- Τα παραπάνω υποψήφια πρωτεύοντα κλειδιά ονομάζονται ‘φυσικά’ γιατί αντιστοιχούν σε πραγματικά δεδομένα του πραγματικού κόσμου



# Surrogate key

- Σε άλλες περιπτώσεις μπορεί να μην υπάρχει φυσικό πρωτεύον κλειδί, όπως για παράδειγμα αν θέλαμε να οργανώσουμε κάποια αντικείμενα χωρίς κάποιο ειδικό μοναδικό πεδίο (π.χ. παιχνίδια ή CD), τότε μπορούμε να χρησιμοποιήσουμε το λεγόμενο τεχνητό κλειδί (surrogate key)
- Πρόκειται για ένα πεδίο τύπου *Ακεραίου αριθμού* που η τιμή του αυξάνεται αυτόματα και επομένως κάθε αντικείμενο (εγγραφή) θα μοναδικοποιείται



# Σύνθετα Πρωτεύοντα

## Βάσεις Δεδομένων

- Στις περιπτώσεις που δεν αρκεί ένα πεδίο για να μοναδικοποιήσει κάθε εγγραφή, μπορούμε να χρησιμοποιήσουμε δύο ή περισσότερα πεδία, δηλαδή ένα σύνθετο πρωτεύον κλειδί
- Για παράδειγμα αν σε ένα πίνακα DELIVERIES έχουμε πεδία DELIVERY\_DATE και PRODUCT\_ID, μπορούμε να μοναδικοποιήσουμε την κάθε παραλαβή προϊόντος
- Θα μπορούσαμε ωστόσο να χρησιμοποιήσουμε ένα Surrogate Key και να έχουμε αυτά τα δύο πεδία, ως απλά πεδία του Πίνακα



# Surrogate Key + Unique

- Συνήθως χρησιμοποιούμε Surrogate Keys ως πρωτεύοντα κλειδιά κι όχι φυσικά κλειδιά (όπως ΑΦΜ, ΑΔΤ, κλπ.)
- Ενώ τα φυσικά κλειδιά υπεισέρχονται στον πίνακα ως απλά πεδία, που χαρακτηρίζονται όμως ως *Unique*, δηλαδή μοναδικά. Δεν μπορεί, για παράδειγμα άλλος Teacher να έχει τον ίδιο ΑΔΤ (Αριθμός Δελτίου Ταυτότητας) με άλλο Teacher





# Τεχνητά Κλειδιά

## Βάσεις Δεδομένων

- Υπάρχει ένας απλός κανόνας ώστε το ER-Model μας **να είναι πάντα στην 2<sup>η</sup> και 3<sup>η</sup> κανονική μορφή**
- **Να χρησιμοποιείτε Surrogate Keys ως πρωτεύοντα κλειδιά**, και όχι φυσικά κλειδιά (όπως Αρ. Ταυτότητας, κλπ)
- Δηλαδή απλά τεχνητά κλειδιά όπως το ID που δεν έχει κάποιο φυσικό νόημα, αλλά απλά αυξάνεται κατά ένα και μοναδικοποιεί την κάθε εγγραφή
- Τότε, εφόσον έχουμε εξασφαλίσει και το ότι δεν έχουμε πεδία πολλαπλών τιμών, είμαστε στην 1<sup>η</sup> NF, εφόσον δεν έχουμε σύνθετα κλειδιά είμαστε στην 2<sup>η</sup> NF και εφόσον μόνο το Surrogate Key προσδιορίζει μοναδικά την κάθε εγγραφή είμαστε στην 3<sup>η</sup> NF (Normal Form)



# Surrogate Keys

- Τα τεχνητά κλειδιά (surrogate keys) έχουν πλεονεκτήματα σε σχέση με τα φυσικά κλειδιά
- Θα μπορούσαμε δηλαδή ακόμα και σε πίνακες που έχουν φυσικά κλειδιά, να χρησιμοποιούμε surrogate keys, και τα πεδία των φυσικών κλειδιών να είναι απλά πεδία του πίνακα που απλά να έχουν μοναδικές τιμές (unique)



# Σχέσεις πινάκων

Βάσεις Δεδομένων

- Όπως αναφέραμε εκτός από τις σχέσεις ένα-προς-πολλά, υπάρχουν και σχέσεις ένα-προς-ένα καθώς και πολλά-προς-πολλά
- Σε ένα ορθό ER Model σχέσεις ένα-προς-ένα έχουμε σε λίγες ειδικές περιπτώσεις ενώ πολλά-προς-πολλά δεν πρέπει να έχουμε



# Σχέσεις 1:1 (1)

- Αν υπάρχουν σχέσεις 1:1 μεταξύ δύο πινάκων μπορούμε να ενοποιήσουμε τους πίνακες σε ένα μόνο πίνακα
- Αν είχαμε δηλαδή STUDENTS και ADDRESSES και κάθε STUDENT είχε μία ADDRESS και μία ADDRESS σχετιζόταν με ένα μόνο STUDENT, τότε μπορούμε να συγχωνεύσουμε τους δύο πίνακες



# Σχέσεις 1:1 (2)

- Υπάρχουν όμως και μία εξαίρεση, στις περιπτώσεις που έχουμε σχήματα **γενικού-ειδικότερου**, όπως για παράδειγμα, αν σε ένα σύστημα έχουμε USERS και ADMINISTRATORS και PARTICIPANTS, όπου όλοι είναι USERS, κάποιοι είναι Administrators και κάποιοι είναι PARTICIPANTS. Σε αυτή την περίπτωση το να συγχωνεύαμε όλα σε ένα πίνακα USERS θα δημιουργούσε redundancy (πλεονασμό)
- Αντίθετα αν είχαμε τρεις πίνακες με σχέση ένα-προς-ένα από το USERS στο ADMINISTRATORS και ένα-προς-ένα από το USERS στο PARTICIPANTS, τότε θα μειώναμε τον πλεονασμό

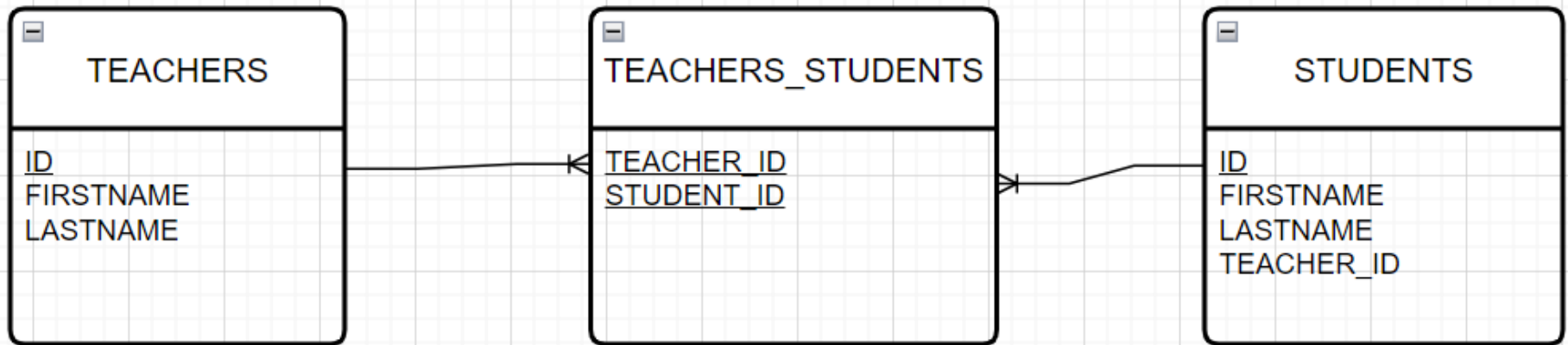


# Σχέσεις N:N (1)

- Οι σχέσεις N:N είναι συνηθισμένες. Για παράδειγμα, αν στο παράδειγμά μας, κάναμε την υπόθεση ότι ένας μαθητής σχετίζεται με πολλούς καθηγητές αλλά και ένας καθηγητής σχετίζεται με πολλούς μαθητές, τότε η σχέση είναι πολλά-προς-πολλά
- Τις σχέσεις πολλά-προς-πολλά θα πρέπει να τις αποσυνθέσουμε εισάγοντας ένα ενδιάμεσο πίνακα



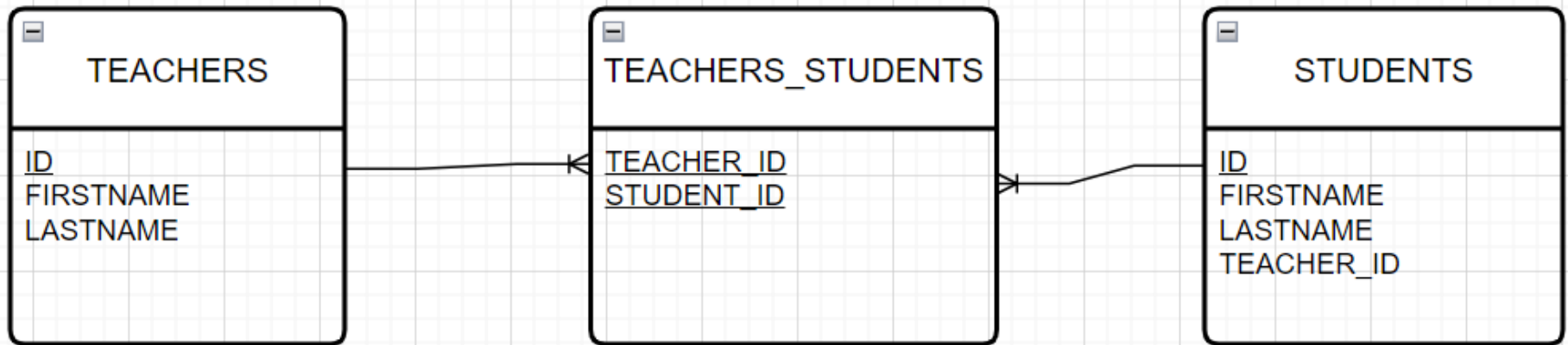
# Σχέσεις N:N (2)



- Επειδή δεν επιτρέπεται να έχουμε σχέσεις πολλά-προς-πολλά, εισάγουμε **ένα ενδιάμεσο πίνακα**, ο οποίος έχει ως **πρωτεύον κλειδί**, **ένα σύνθετο κλειδί** που αποτελείται από τα δύο πρωτεύοντα κλειδιά των βασικών πινάκων
- Το κάθε πεδίο του σύνθετου κλειδιού είναι ξένο κλειδί των βασικών πινάκων και η **σχέση μεταξύ των βασικών πινάκων και του ενδιάμεσου είναι ένα-προς-πολλά**



# Σχέση πολλά-προς-πολλά (1)

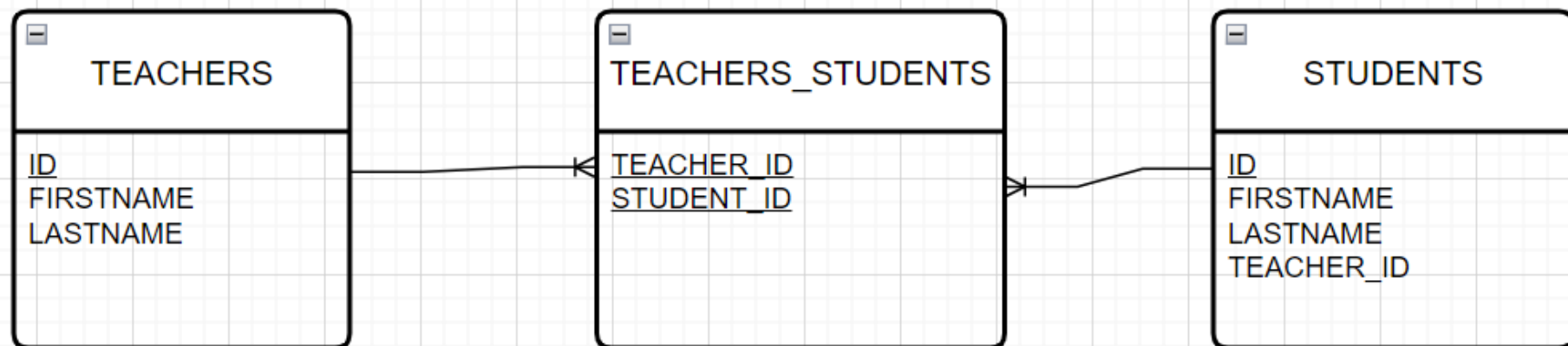


- Ο ενδιάμεσος πίνακας περιέχει ως πρωτεύον κλειδί ένα σύνθετο πρωτεύον (TEACHER\_ID, STUDENT\_ID), που αποτελείται από τα δύο πρωτεύοντα κλειδιά των βασικών πινάκων





# Σχέση πολλά-προς-πολλά (2)



- Επίσης, το κάθε ένα πεδίο του σύνθετου κλειδιού, είναι Ξένο Κλειδί των βασικών πινάκων
- Το STUDENT\_ID είναι ξένο κλειδί του STUDENTS/ID και το COURSE\_ID είναι ξένο κλειδί του COURSES/ID



# MS SQL Server (1)

- Θα δούμε στη συνέχεια πως μπορούμε να υλοποιήσουμε αυτό το σχήμα με τη χρήση ενός εμπορικού RDBMS (Relational Database Management System), του Microsoft SQL Server (MS SQL Server)
- Υπάρχουν και άλλα συστήματα RDBMS, όπως Oracle SQL, MySQL, PostgreSQL, κλπ.



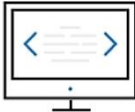
# MS SQL Server (2)

- Υπάρχουν τρεις βασικές εκδόσεις του MS SQL Server:
  - **Enterprise Edition**, που επιτρέπει την ανάπτυξη επαγγελματικών εφαρμογών όλων των τύπων
  - **SQL Server Development**, που περιέχει όλο το functionality του Enterprise Edition αλλά έρχεται με license μόνο για development και test, όχι για production use
  - **SQL Express**, που είναι μία δωρεάν έκδοση για ανάπτυξη εφαρμογών με λιγότερα χαρακτηριστικά από την Enterprise Edition



# SQL Express 2022


Βάσεις Δεδομένων



**Developer**

SQL Server 2022 Developer is a full-featured free edition, licensed for use as a development and test database in a non-production environment.

[Download now](#)



**Express**

SQL Server 2022 Express is a free edition of SQL Server, ideal for development and production for desktop, web, and small server applications.

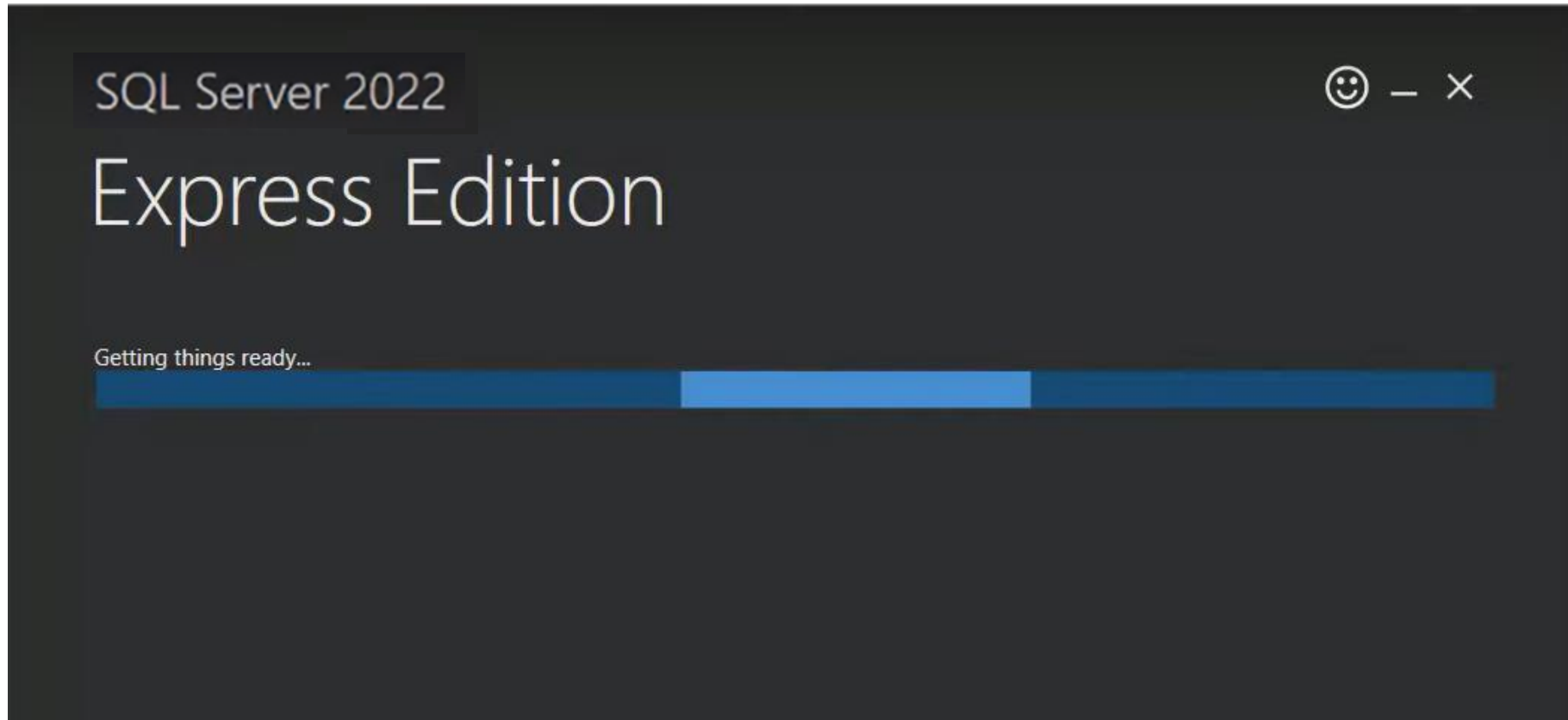
[Download now](#)

- <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>
- Θα εγκαταστήσουμε τον **SQL Express 2022**. Χρειάζεται δικαιώματα **Administrator**. Επομένως, αφού το κατεβάσετε, ξεκινάτε την εγκατάσταση με δεξί κλικ και **Run As Administrator** (σε Windows)



# Εγκατάσταση

Βάσεις Δεδομένων



- Στα Windows, κάντε δεξί κλικ run as administrator. Ακολουθείτε τις αντίστοιχες διαδικασίες εγκατάστασης στα άλλα Λειτουργικά Συστήματα



# Installation Type

Βάσεις Δεδομένων

SQL Server 2022

## Express Edition

Select an installation type:

### Basic

Select Basic installation type to install the SQL Server Database Engine feature with default configuration.

### Custom

Select Custom installation type to step through the SQL Server installation wizard and choose what you want to install. This installation type is detailed and takes longer than running the Basic install.

### Download Media

Download SQL Server setup files now and install them later on a machine of your choice.

SQL Server transmits information about your installation experience, as well as other usage and performance data, to Microsoft to help improve the product. To learn more about data processing and privacy controls, and to turn off the collection of this information after installation, see the [documentation](#)

16.2211.5693.3

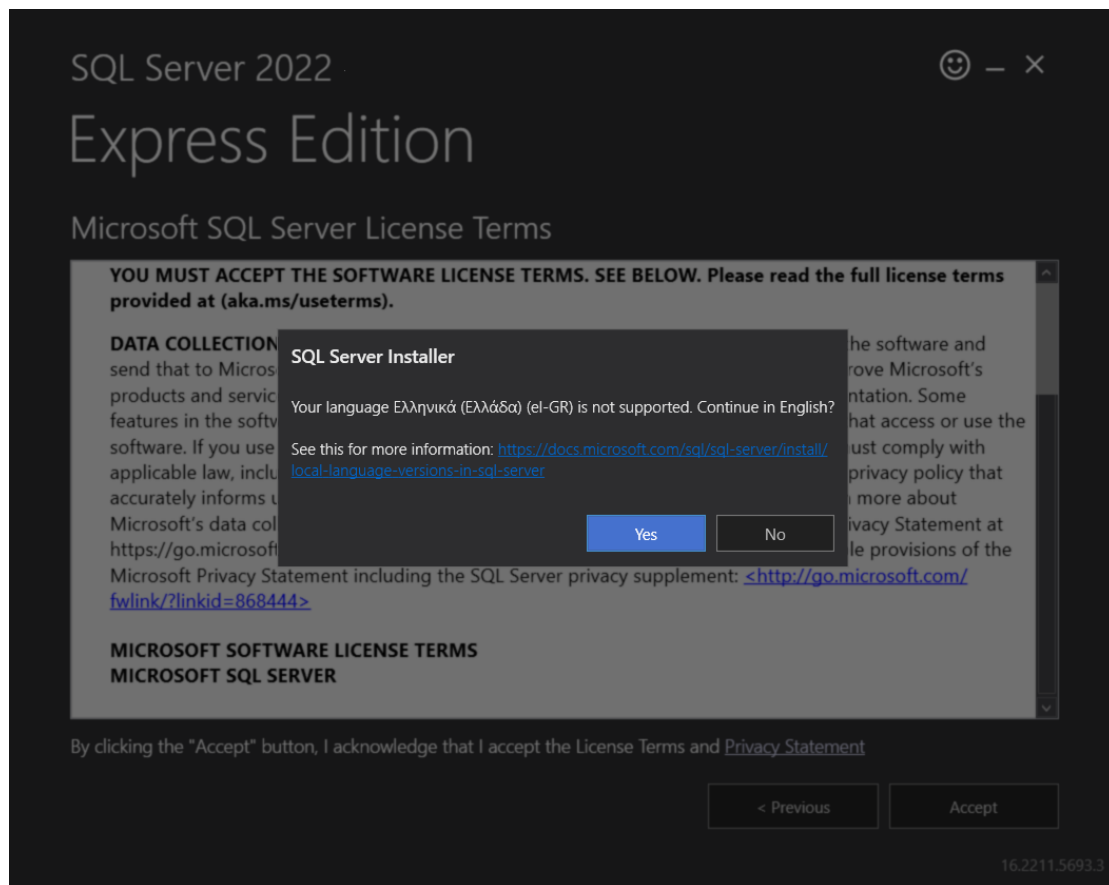
- Θα επιλέξουμε το basic installation



# Install

## Βάσεις Δεδομένων

- Yes και accept το license agreement

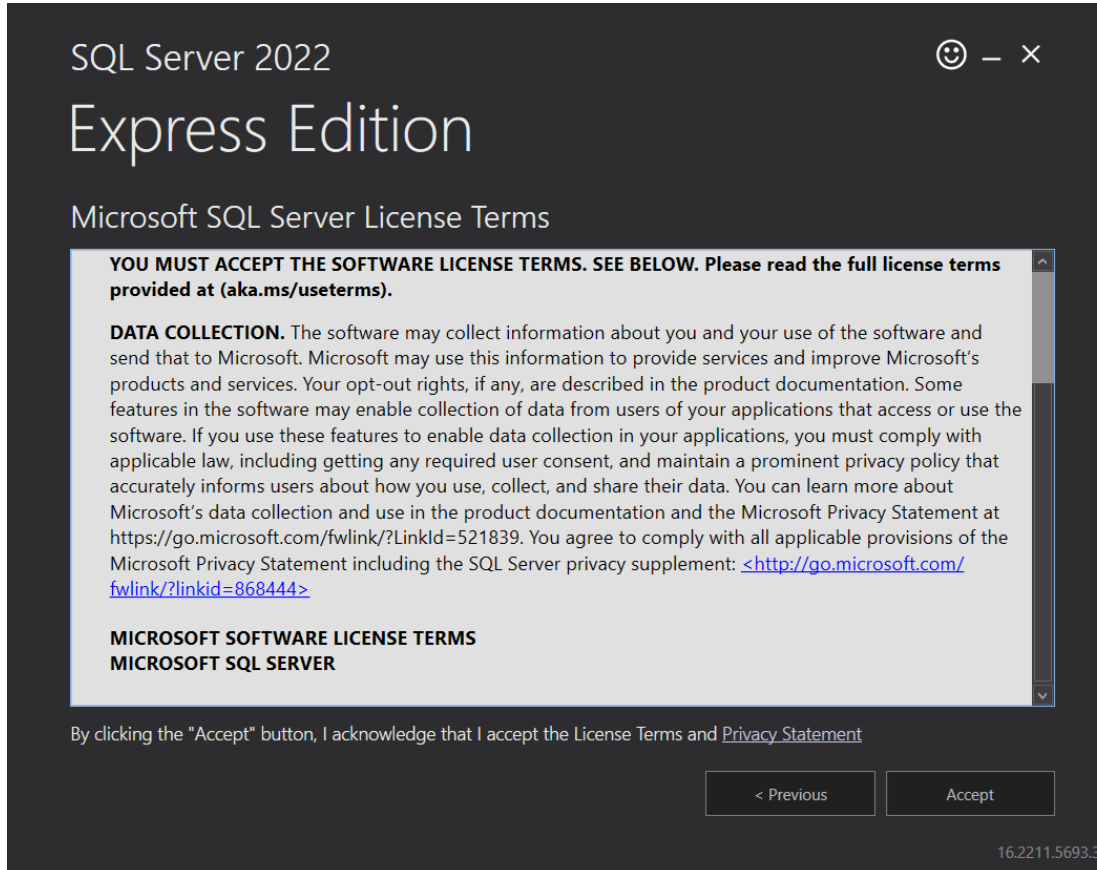




# License Agreement

Βάσεις Δεδομένων

- Accept license agreement to

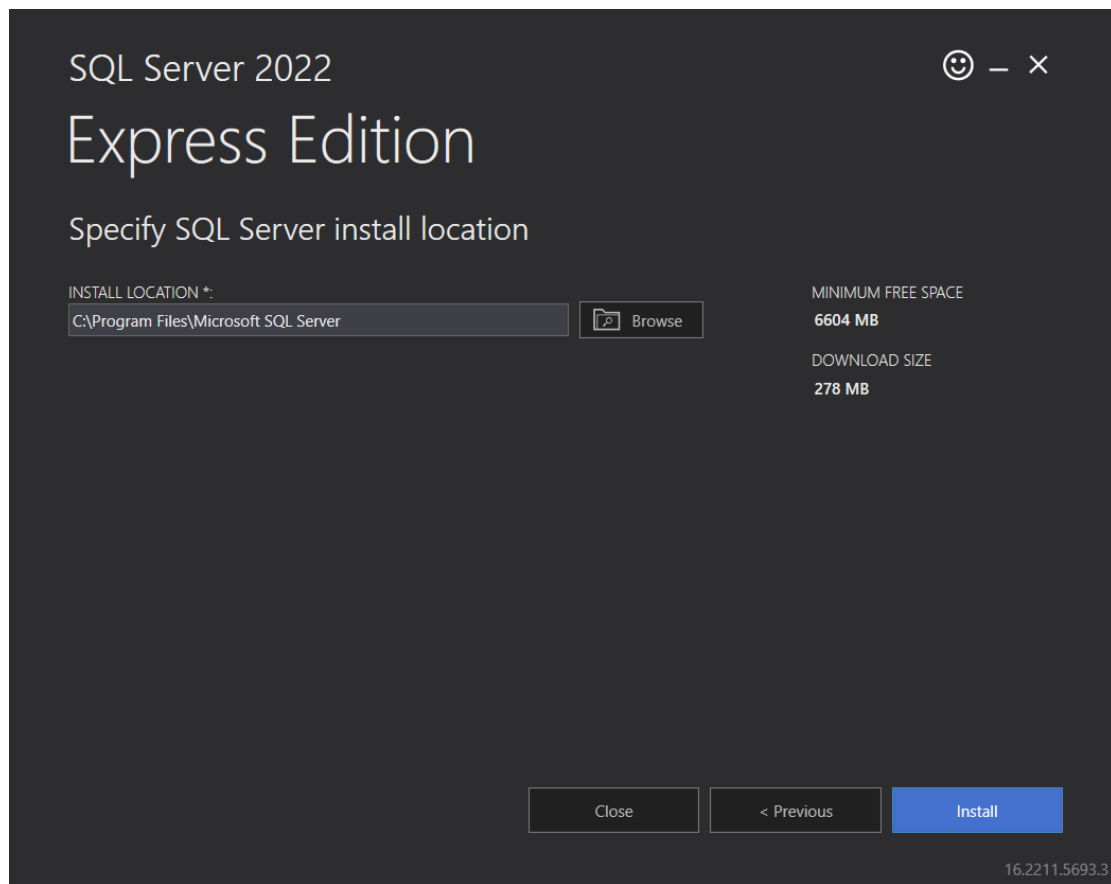






# Install

Βάσεις Δεδομένων

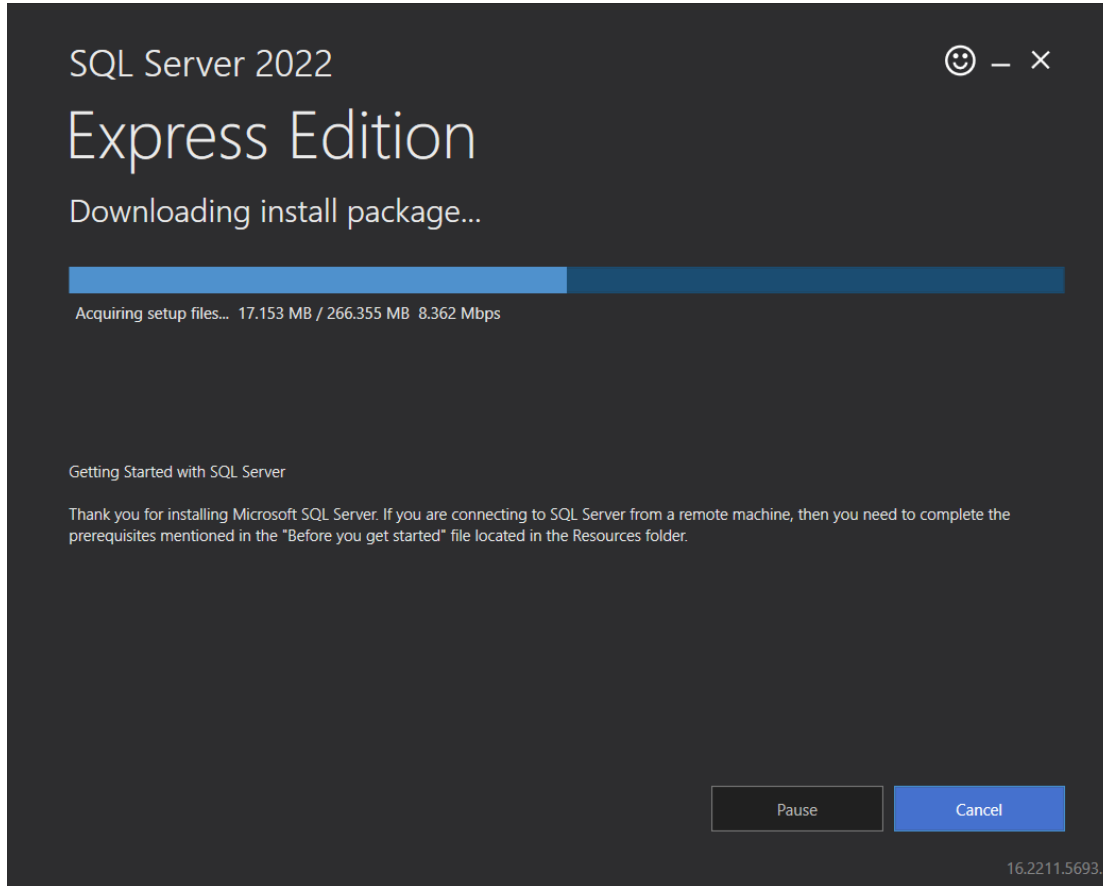


- Install



# Installing

Βάσεις Δεδομένων

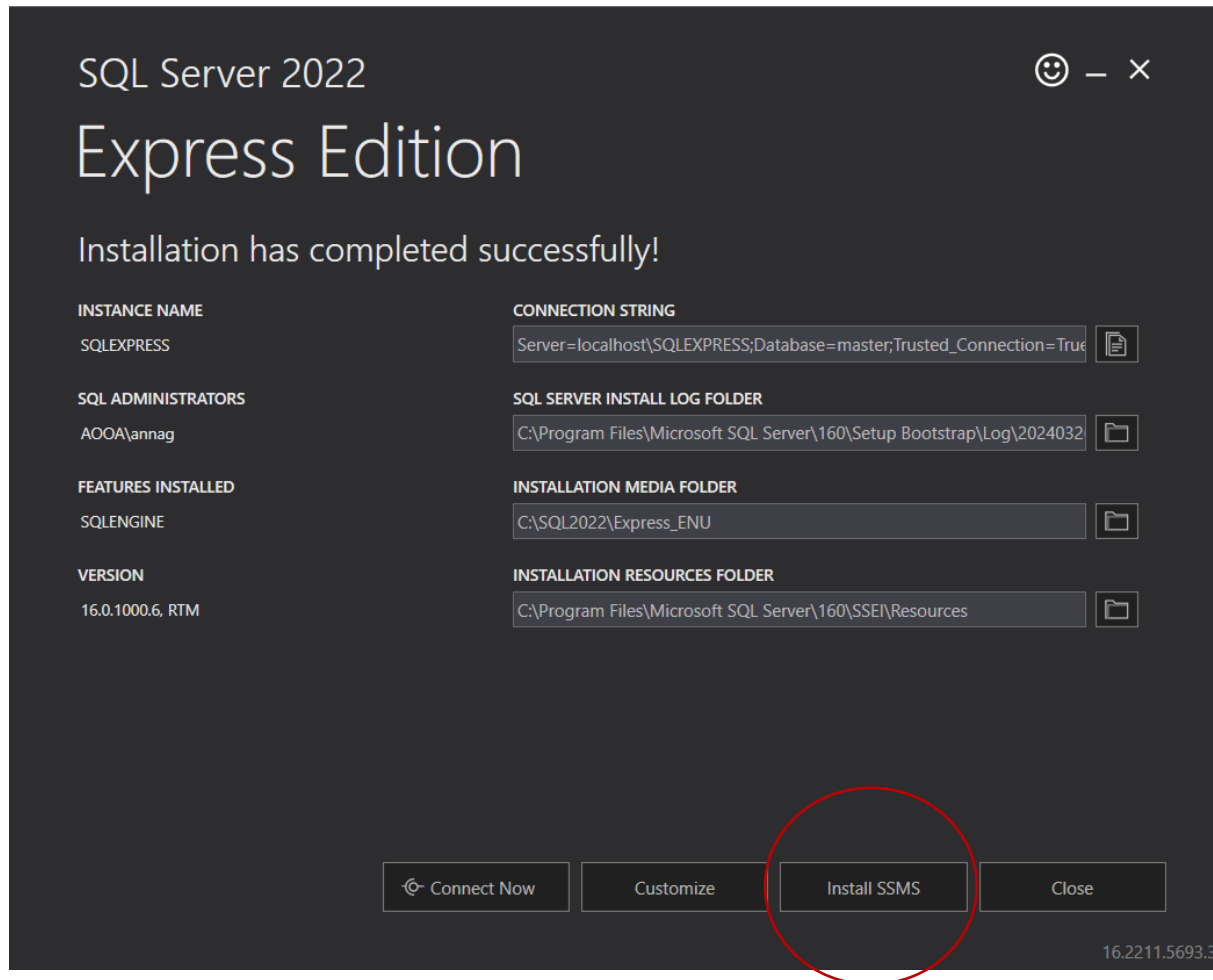


- Ίσως αργήσει λίγο να εγκατασταθεί



# Install SSMS

Βάσεις Δεδομένων



- Επιλέγουμε Install SSMS



# Εγκατάσταση SSMS ver. 20

Βάσεις Δεδομένων

## Download SSMS

↓ [Download SQL Server Management Studio \(SSMS\) 20.0](#) ↗

SSMS 20.0 is the latest generally available (GA) version. If you have a *preview* version of SSMS 20 installed, uninstall it before installing SSMS 20.0. Installing SSMS 20 doesn't upgrade or replace SSMS 19.x and earlier versions.

- Release number: 20.0
- Build number: 20.0.70.0
- Release date: March 19, 2024

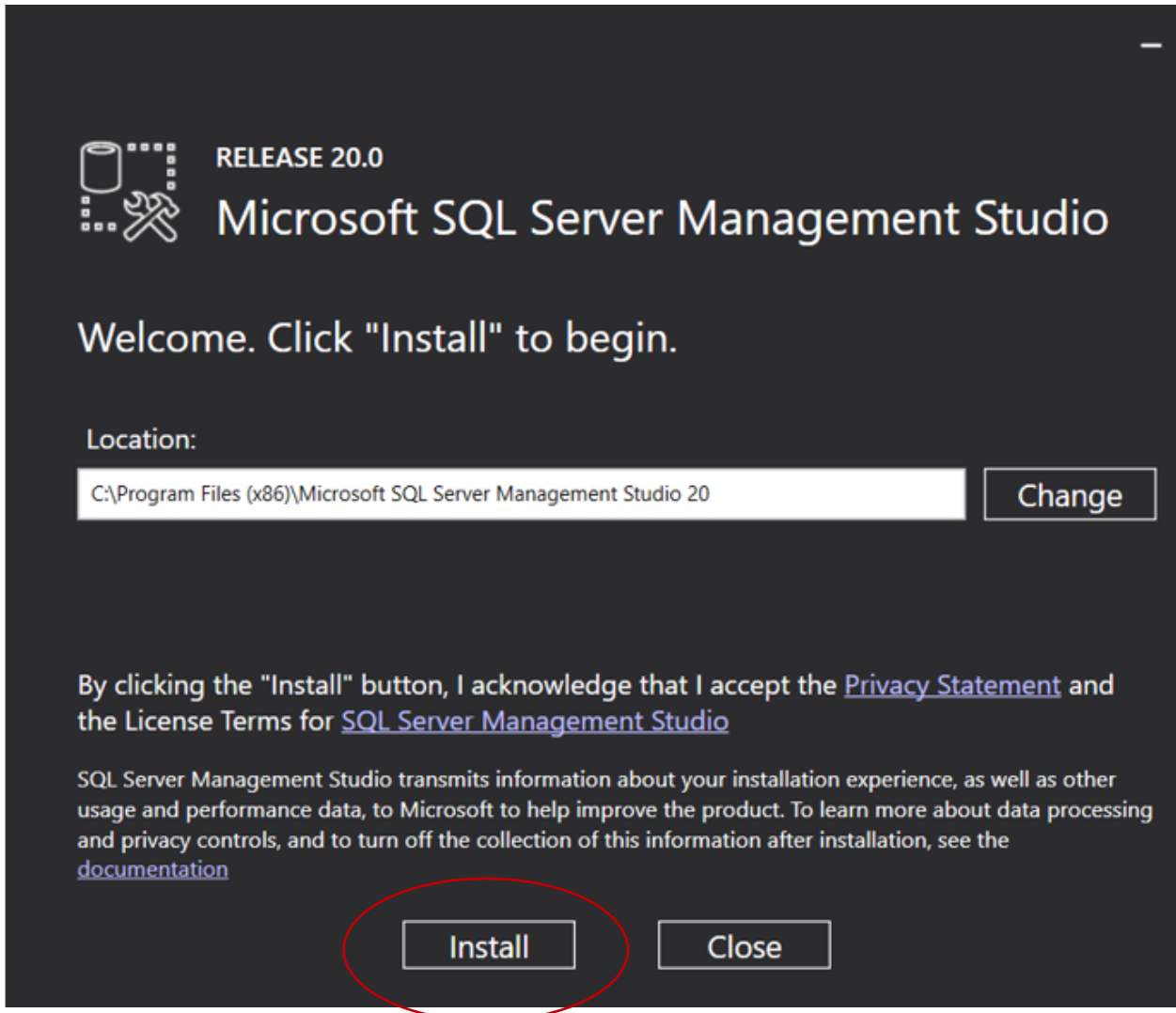
- <https://aka.ms/ssmsfullsetup>
- Το SQL Server Management Studio (SSMS) είναι το Διαχειριστικό πρόγραμμα του SQL Server όπου μπορούμε να δημιουργήσουμε Βάσεις Δεδομένων, Πίνακες, Σχέσεις Πινάκων, κλπ.



# SSMS Install (1)

Βάσεις Δεδομένων

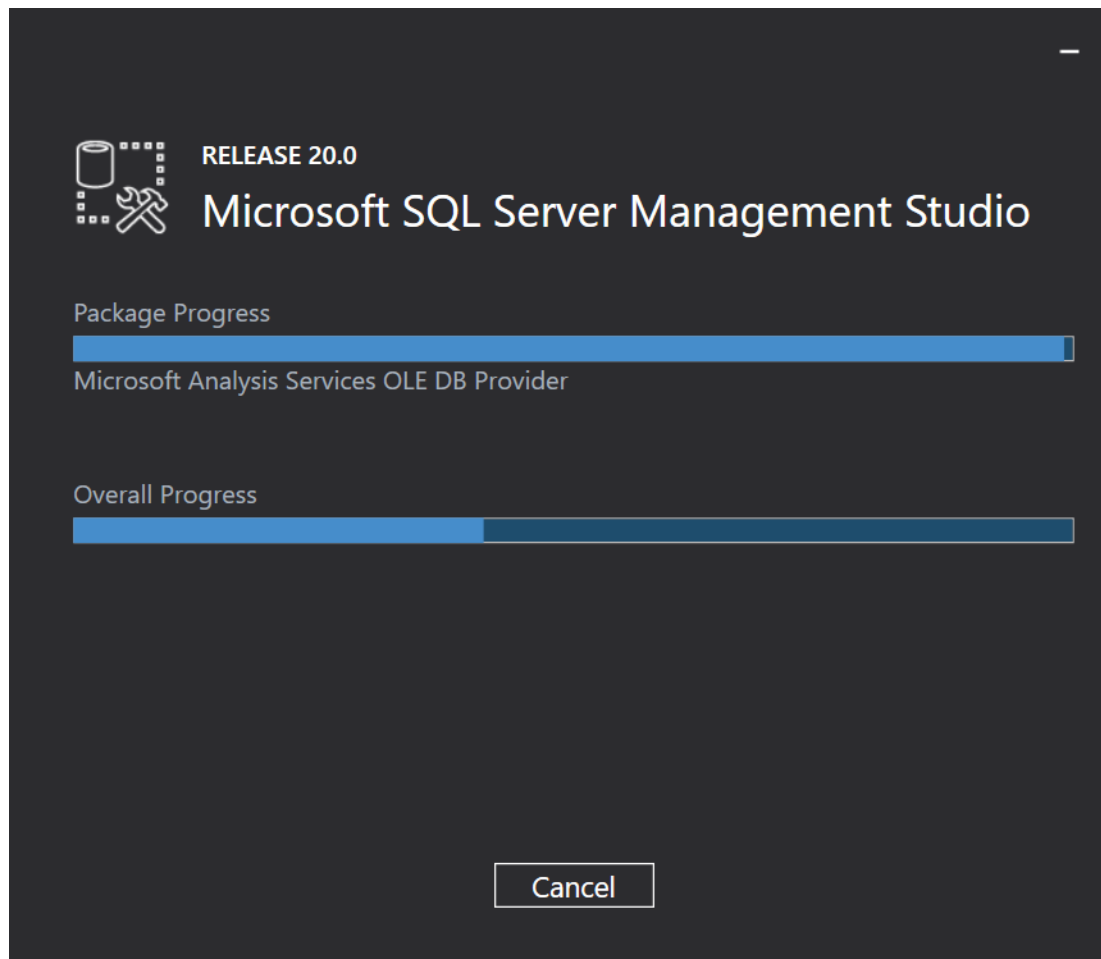
- Εγκαθιστούμε το SQL Server Management Studio





# SSMS Install (2)

Βάσεις Δεδομένων



- Διαδικασία εγκατάστασης



# SSMS Install (3)

Βάσεις Δεδομένων

- Επιτυχής εγκατάσταση



RELEASE 20.0

Microsoft SQL Server Management Studio

Setup Completed

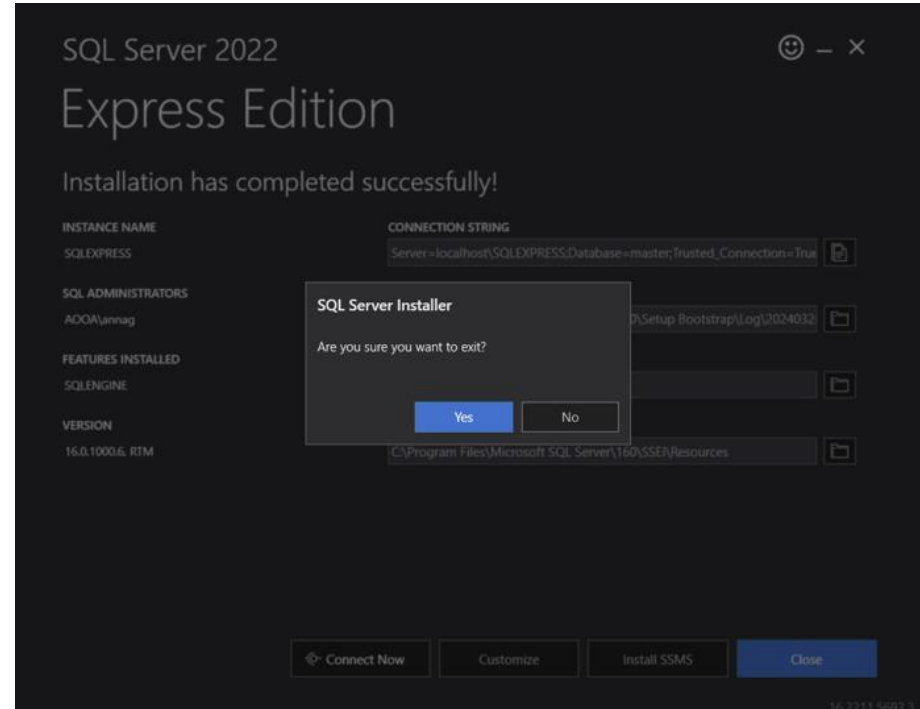
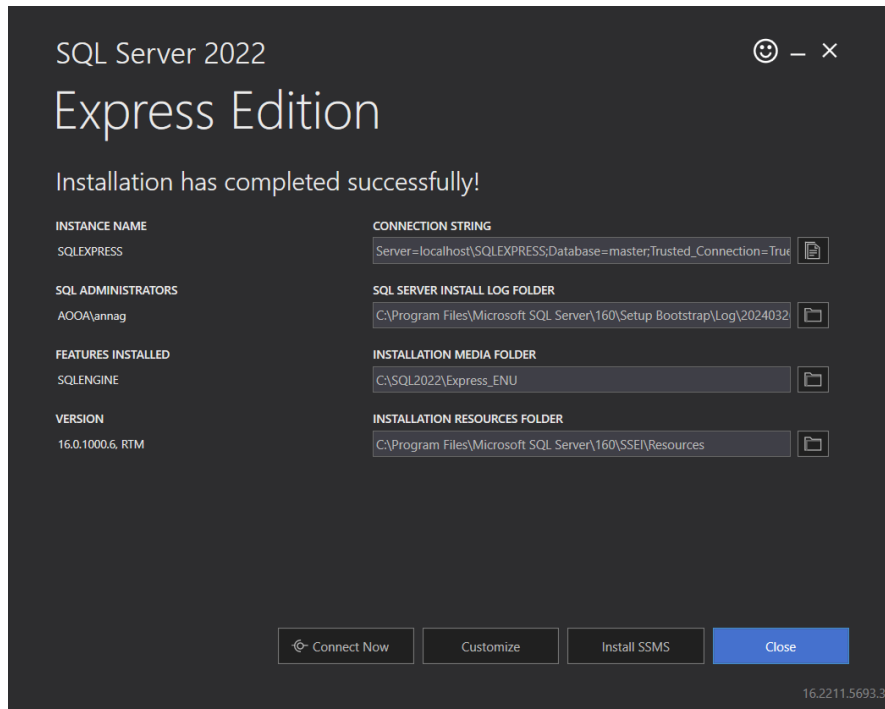
All specified components have been installed successfully.

Close



# SSMS Install (4)

Βάσεις Δεδομένων



- Έξοδος από το setup και επιτυχής εγκατάσταση





# Connect με Windows

Βάσεις Δεδομένων

Connect to Server

SQL Server

Login | Connection Properties | Always Encrypted | Additional Connection Parameters

Server

Server type: Database Engine

Server name: .\sqlexpress

Authentication: Windows Authentication

User name: A00A\annag

Password:

☐ Remember password

Connection Security

Encryption: Mandatory

☒ Trust server certificate

Host name in certificate:

Connect Cancel Help Options <<

- Συνδεόμαστε στο  
.\sqlexpress με  
Windows  
authentication και  
trust server  
certificate αν το  
encryption είναι  
mandatory




# Linux, MacOS

## Βάσεις Δεδομένων

microsoft.com/en-us/sql-server/sql-server-downloads

### Install SQL Server 2019 on Windows, Linux, and Docker containers


Αποθήκευση



Windows

Run SQL Server on Windows or as a Virtual Machine in Azure


[Choose your installation setup >](#)



Linux

Run SQL Server 2019 on Linux

[Choose your installation setup >](#)



Docker

Run SQL Server 2019 container image with Docker

[Choose your installation setup >](#)

- Παρέχονται οδηγίες για εγκατάσταση σε Linux και Mac. Σε Mac θα πρέπει να εγκατασταθεί το **Docker** version του SQL Server και στη συνέχεια το Azure Data Studio αντί για το SSMS



# Docker

## Βάσεις Δεδομένων

← → ↻ docker.com/get-started/

Hibernate ORM 5.3.... Μεταγλωττιστές Electronic library, D... (All Streams) - One... Android Hibernate Java aocAI - OneDrive C#-E-Learning » Όλοι οι σελιδοδείκτες

## Get Started with Docker

Build applications faster and more securely with Docker for developers

[Learn how to install Docker](#)

**Download for Windows**

Download for Mac - Intel Chip

Download for Mac - Apple Chip

Download for Linux

- Το Docker (Docker Engine) παρέχει μία πλατφόρμα εκτέλεσης εφαρμογών. Εγκαθιστούμε το Docker στο Λειτουργικό μας Σύστημα



# Docker SQL Server

## Βάσεις Δεδομένων

- Στο google αναζητούμε και στο docker hub εντοπίζουμε το Microsoft SQL Server – Ubuntu based images



docker sql server



Εικόνες

Βίντεο

Mac

Windows

License

Trusted connection

Ειδήσεις

Χάρτες

Περίπου 65.800.000 αποτελέσματα (0,32 δευτερόλεπτα)



Docker Hub

<https://hub.docker.com> > ...

### Microsoft SQL Server - Ubuntu based images

Official container images for Microsoft **SQL Server** on Linux for **Docker** Engine. How to use this Image. You can now get started with the **SQL Server 2022** on Ubuntu ...



hub.docker.com/search?q=sql+server



Hibernate ORM 5.3....



Μεταγλωττιστές



Electronic library. D...



(All Streams) - One...



Android



Hibernate



Java



aocAI - Or



Microsoft SQL Server - Ubuntu based images • 0 • 0

By Microsoft • Updated an hour ago

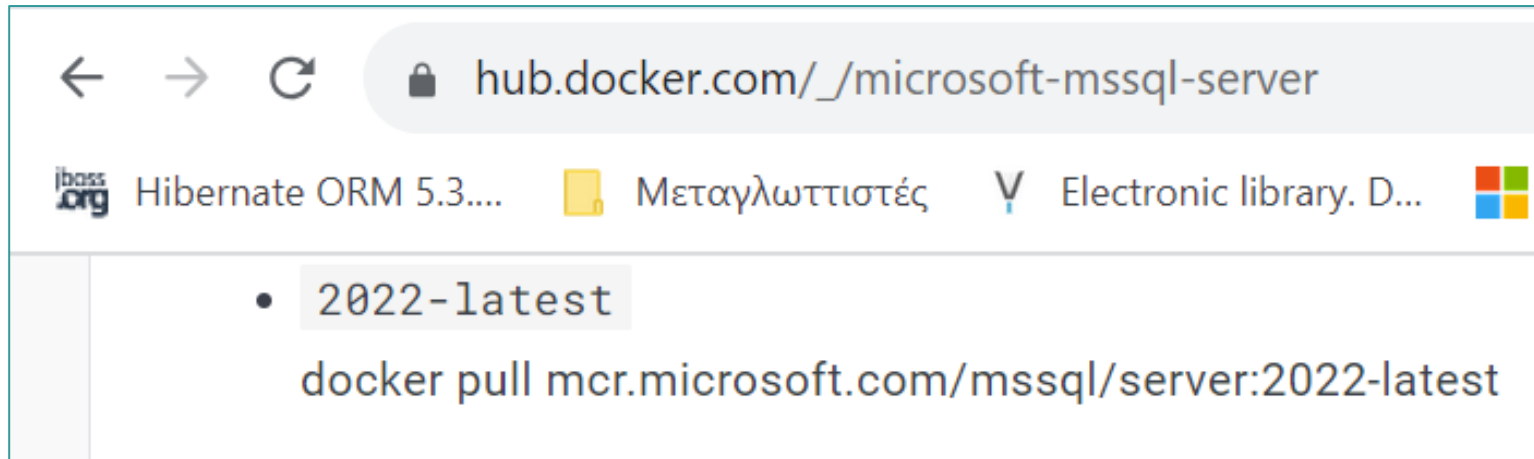
Official images for Microsoft SQL Server based on Ubuntu

x86-64



# docker pull (το image)

Βάσεις Δεδομένων



- [https://hub.docker.com/\\_/microsoft-mssql-server](https://hub.docker.com/_/microsoft-mssql-server)

- Μετά  
σε CLI
- Copy-paste την εντολή



# Run the image

```
docker run -e "ACCEPT_EULA=Y" -e "MSSQL_SA_PASSWORD=yourStrong(!)Password" -p 1433:1433 -d mcr.microsoft.com/mssql/server:2022-latest
```

- Κάνετε copy-paste σε CLI το παραπάνω εισάγοντας ως password στη θέση 'yourStrong(!)Password' **ένα δικό σας password** (με τουλάχιστον 8 χαρακτήρες εκ των οποίων ένα πεζό, ένα κεφαλαίο, ένα αριθμό και ένα ειδικό χαρακτήρα)
- Μπορείτε επίσης να προσθέσετε το `--name sqlserver`, ώστε να δώσετε ένα custom όνομα (sqlserver) στον server σας



# Containers – Running

Βάσεις Δεδομένων

The screenshot shows the Docker Desktop application window. The top bar includes 'Docker Desktop', 'Update to latest', a search bar, and a 'Ctrl+K' button. The left sidebar contains navigation options: 'Containers' (selected), 'Images', 'Volumes', 'Dev Environments' (marked BETA), 'Extensions', and 'Add Extensions'. The main panel is titled 'Containers' and includes a 'Give feedback' link. Below the title, a description states: 'A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)'. A toggle switch for 'Only show running containers' is present. A search bar is also available. The container list table shows one running container:

	Name	Image	Status	Port(s)	Started	Actions
<input type="checkbox"/>	sqlserver fc6d38a39059	<a href="https://mcr.microsoft.com/mssql/server">mcr.microsoft.com/mssql/server</a>	Running	1433:1433	2 seconds ago	


- Sqlserver is running



# Azure Data Studio

Βάσεις Δεδομένων

← → ↻ [azure.microsoft.com/en-us/products/data-studio](https://azure.microsoft.com/en-us/products/data-studio)

 **Azure** Explore ▾ Products ▾ Solutions ▾ Pricing ▾ Partners ▾ Resources ▾

## Azure Data Studio

A modern open-source, cross-platform hybrid data analytics tool designed to simplify the data landscape.

[Try Azure for free](#) [Download now](#)

- Κάνουμε download now





# Install Azure Data Studio

Βάσεις Δεδομένων

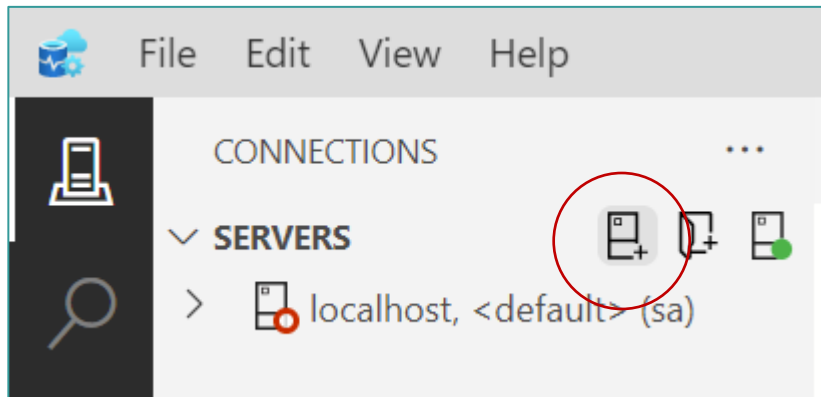
Platform	Type	Download
Windows	User Installer	<a href="#">64-bit ↗</a> <a href="#">ARM64 ↗</a>
	System Installer	<a href="#">64-bit ↗</a> <a href="#">ARM64 ↗</a>
	.zip	<a href="#">64 bit ↗</a> <a href="#">ARM64 ↗</a>
Linux	.tar.gz	<a href="#">64-bit ↗</a>
	.deb	<a href="#">64-bit ↗</a>
	.rpm <sup>1</sup>	<a href="#">64-bit ↗</a>
macOS	.zip	<a href="#">Universal ↗</a> <a href="#">Intel Chip ↗</a> <a href="#">Apple Silicon ↗</a>

- Install ανάλογα με το Λειτουργικό Σύστημα

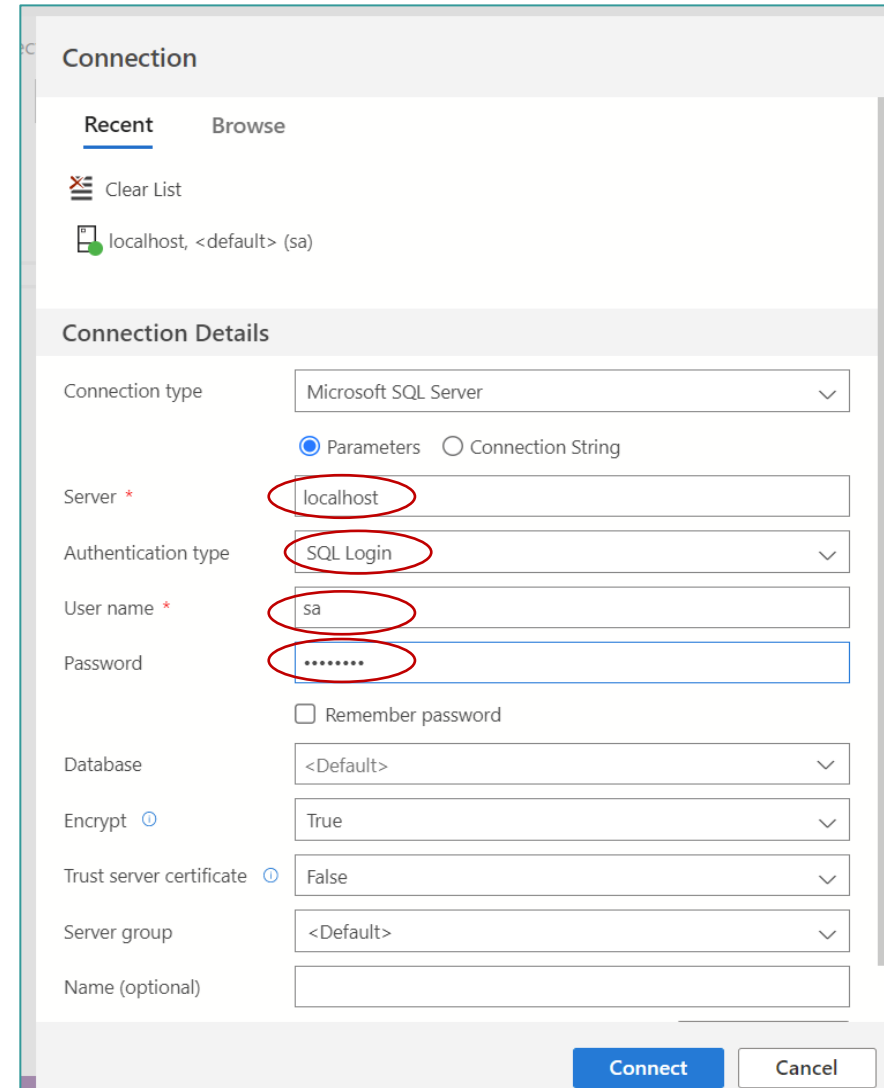


# Connect

## Βάσεις Δεδομένων

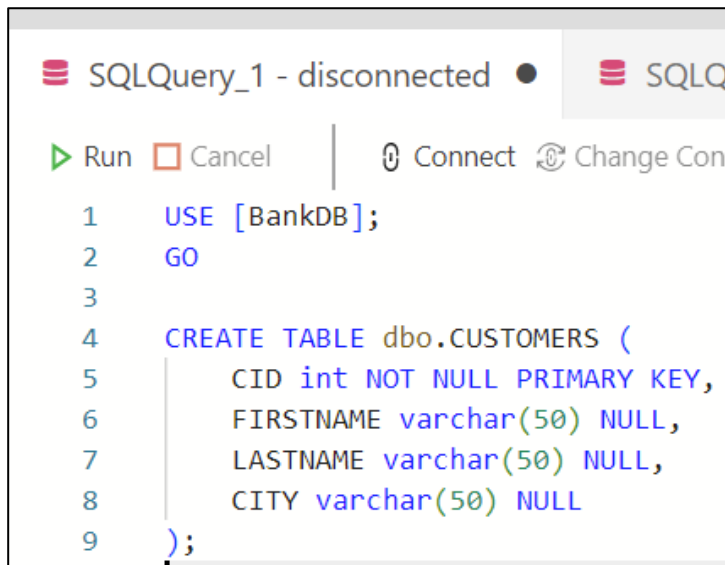
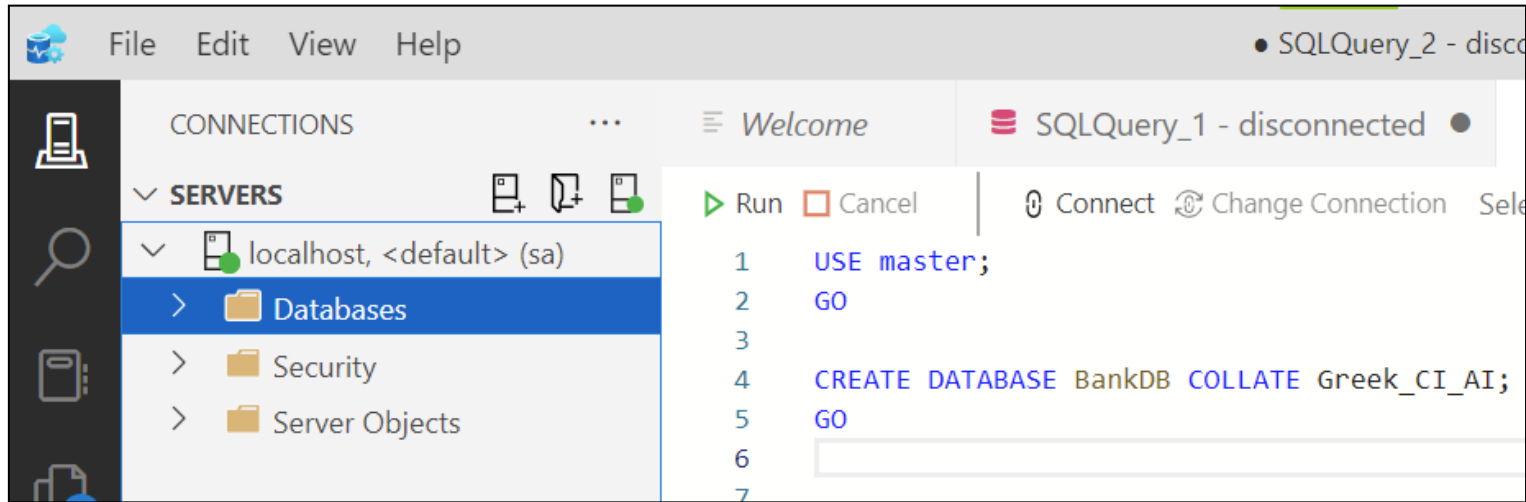


- Κάνουμε connect με Server: localhost, Authentication Type: SQL Login, User name: sa κα password το password που δώσατε στο run του SQL Server στον Docker





# Create Database



- Ανοίγουμε ένα New Query και τρέχουμε καταρχήν το CREATE DATABASE ...
- Μετά τρέχουμε τα CREATE TABLE ...



# Authentication

- Αν ωστόσο έχουμε εγκαταστήσει σε Windows, υπάρχουν δύο τρόποι εισόδου (Authentication) στον SQL Server
  - Ως χρήστης των Windows
  - Ως χρήστης του SQL Server

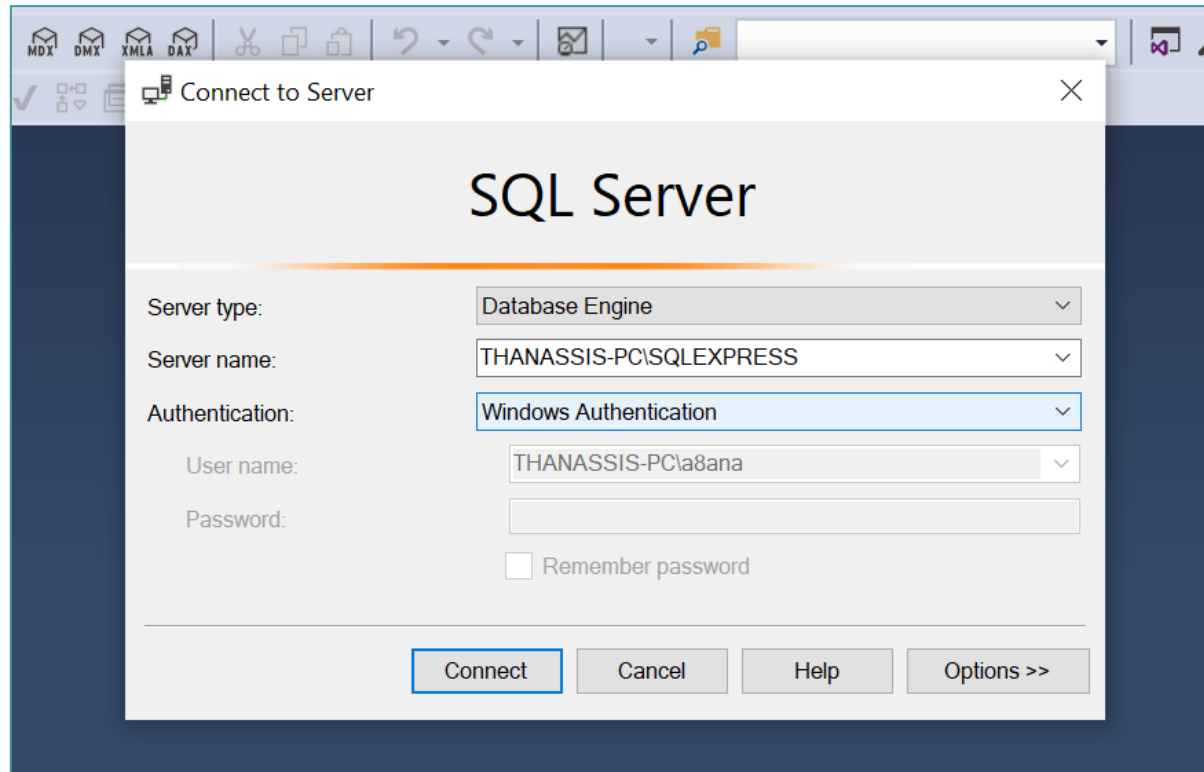


# Server Name

- Το όνομα του SQL Express Server είναι **localhost\sqlexpress** όπου localhost είναι ο τοπικός μας υπολογιστής
- Εναλλακτικά μπορεί να δοθεί το όνομα του υπολογιστή, π.χ. THANASSIS-PC\SQLEXPRESS ή .\sqlexpress



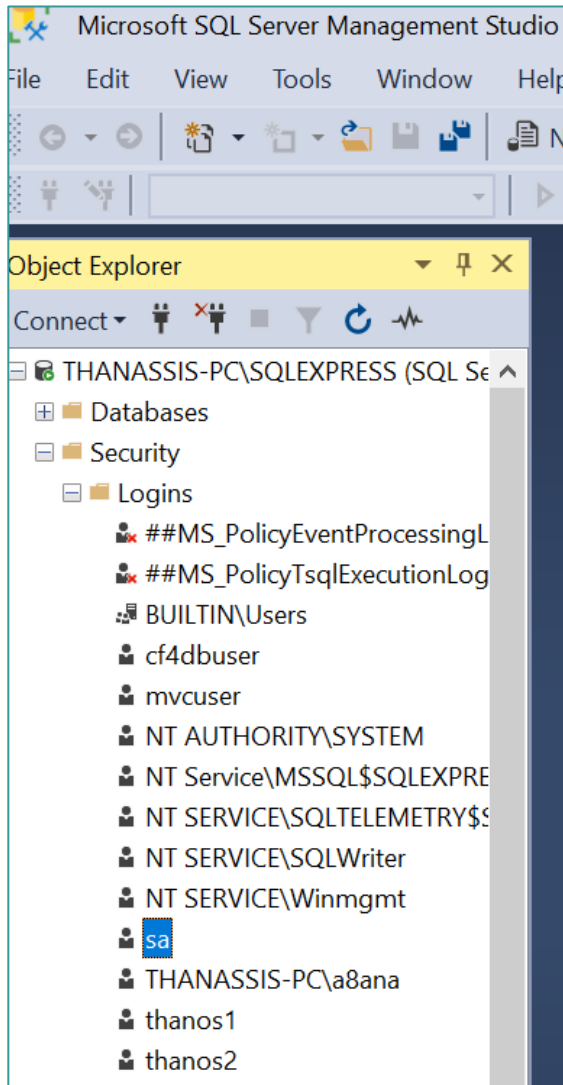
# Σύνδεση με SQL Server (1)



- Ανοίγουμε το SSMS, και κάνουμε σύνδεση με τον SQL Express με **Windows Authentication**



# Sysadmin (1)

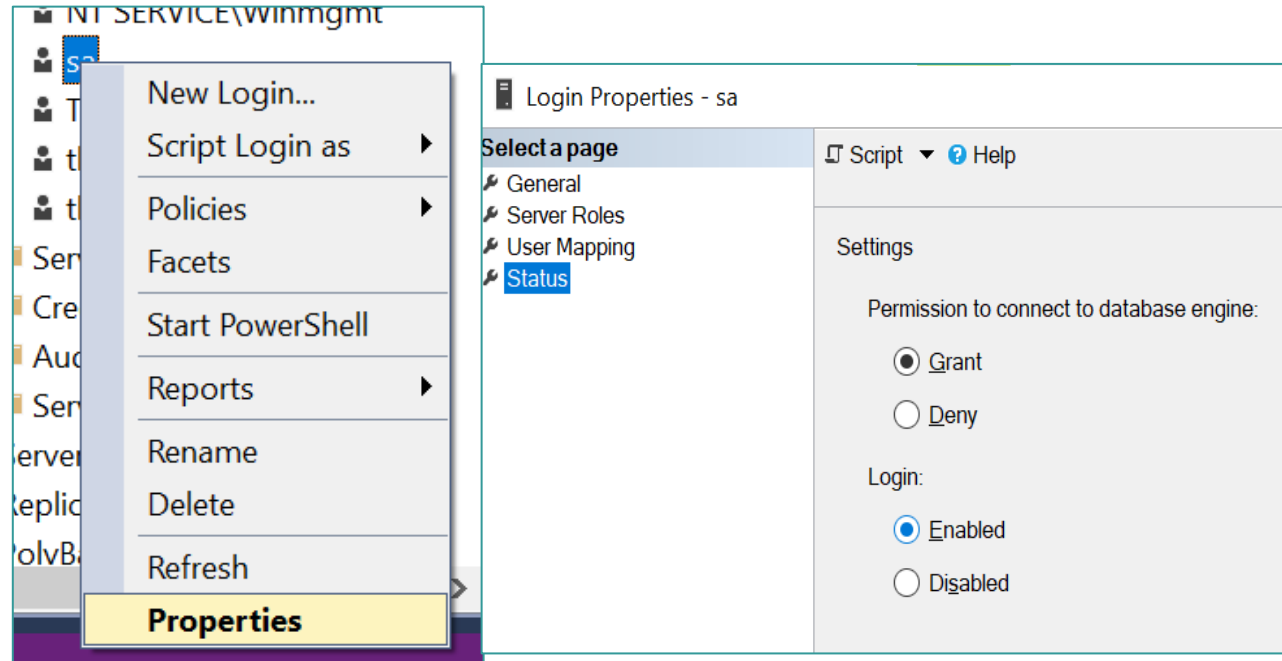


- Επίσης και σε Windows συστήματα αλλά και γενικότερα μπορούμε να συνδεόμαστε ως **sa** που είναι ο system Administrator
- Ο **sa** είναι ο superuser ή αλλιώς sysadmin, και έχει δικαιώματα διαχείρισης όλου του DB Server και όλων των ΒΔ σε αυτόν
- Πηγαίνουμε στο Security Logins για να αλλάξουμε κάποια config settings του **sa**

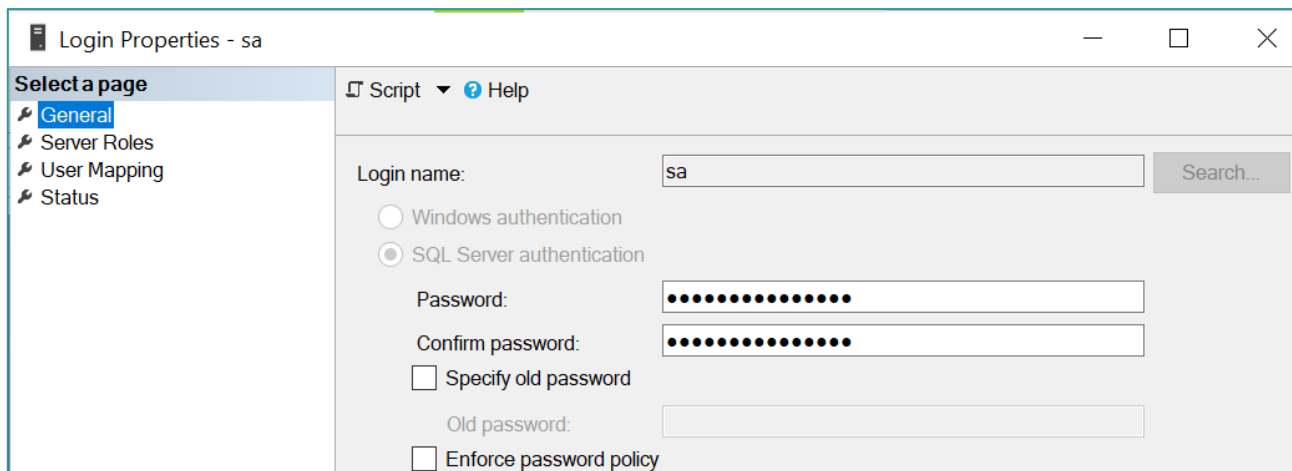


# Sysadmin (2)

## Βάσεις Δεδομένων



- Στο Properties / Status, αν είναι disabled τον κάνουμε enabled και στο General του δίνουμε και ένα password χωρίς *Enforce Password Policy* για να μη χρειάζεται να δίνουμε δύσκολα passwords (όταν αναπτύσσουμε για test purposes, όπως εδώ)

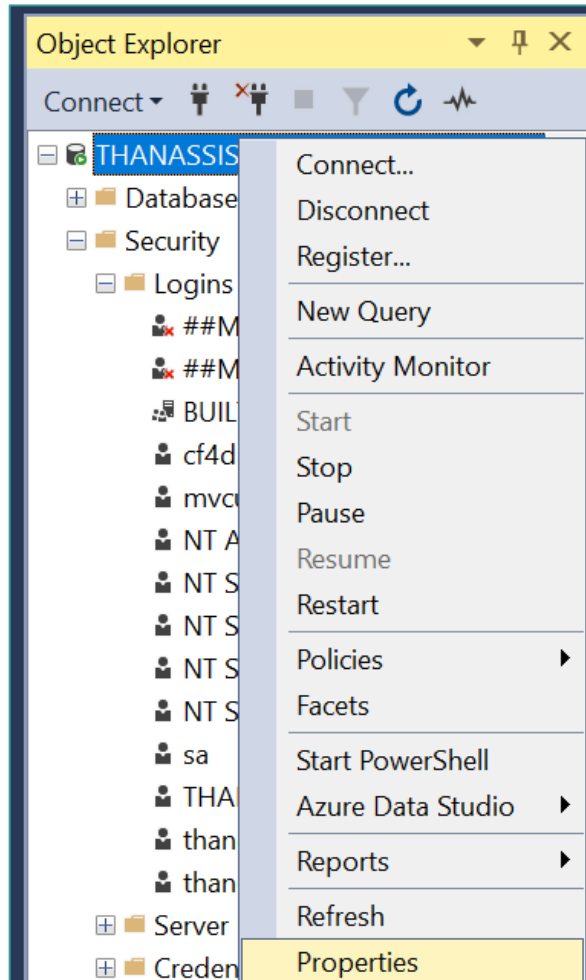




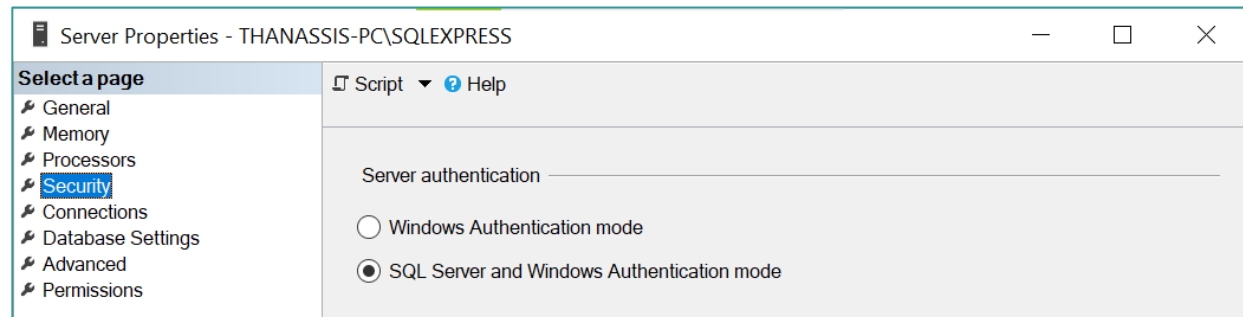


# Security Settings

Βάσεις Δεδομένων



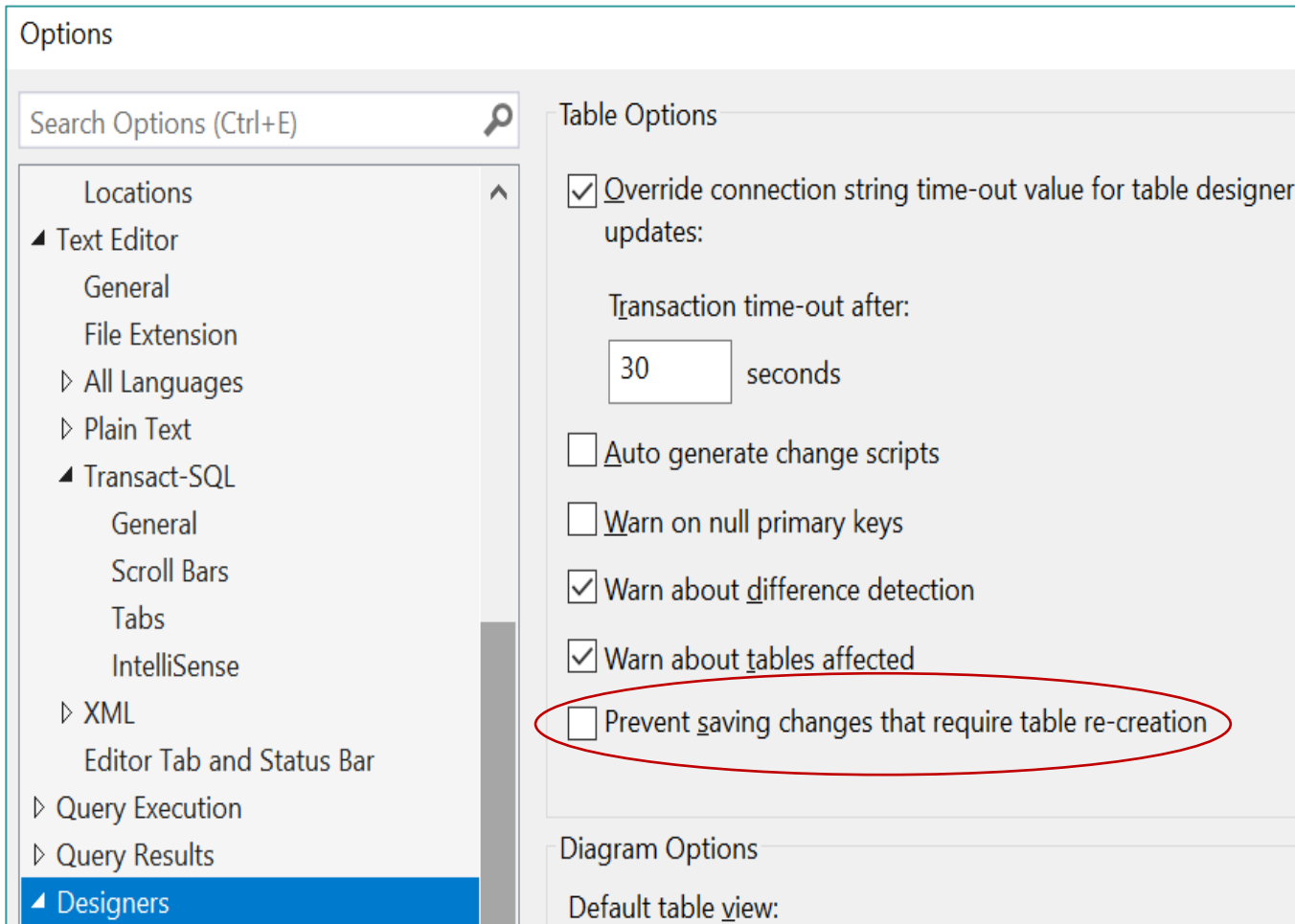
- Θα πρέπει στα **Properties** όλου του Server (δεξί κλικ/Properties), στο **Security** να ορίσουμε 'SQL Server and Windows Authentication Mode'





# Tools / Options

## Βάσεις Δεδομένων



- Επίσης στα Tools/Options/Designers δεν επιλέγουμε **Prevent saving changes that require table re-creation** ώστε να μπορούμε στις επόμενες ενότητες να κάνουμε αλλαγές στους πίνακες που θα δημιουργούμε χωρίς προβλήματα



# Σύνδεση (2)

- Σε non-Windows συνδεόμαστε με SQL Server Authentication και με τον χρήστη sa και το password που έχουμε ορίσει

Connect to Server

SQL Server

Server type: Database Engine

Server name: THANASSIS-PC\SQLEXPRESS

Authentication: SQL Server Authentication

Login:

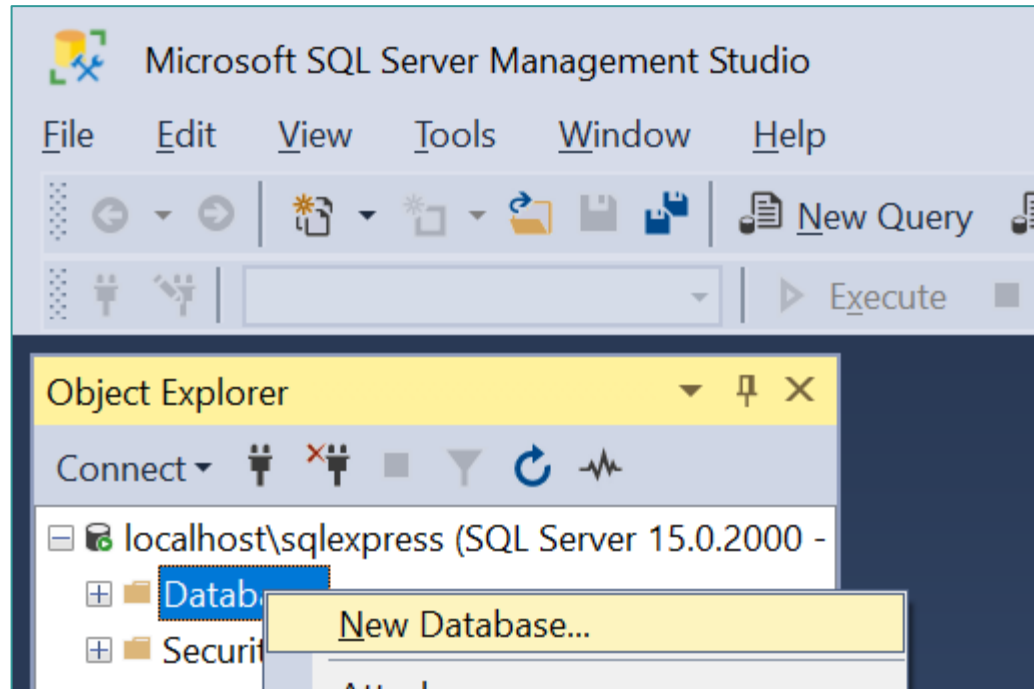
Password:

☐ Remember password

Connect Cancel Help Options >>



# Νέα Βάση Δεδομένων (1)



- Αριστερά στο Μενού, επιλέγουμε το Databases και μετά δεξί κλικ και new Database



# Νέα Βάση Δεδομένων (2)

Βάσεις Δεδομένων

New Database

Select a page

- General
- Options
- Filegroups

Script Help

Database name: CodingFactoryDB

Owner: <default>

☒ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (M...	Autogrowth / Maxsize	Path
CodingFacto...	ROWS ...	PRIMARY	8	By 64 MB, Unlimited	C:\Progra
CodingFacto...	LOG	Not Applicable	8	By 64 MB, Unlimited	C:\Progra

- Δίνουμε όνομα της ΒΔ μας (Database name), έστω CodingFactoryDB



# Νέα Βάση Δεδομένων (3)

New Database

Select a page

- General
- Options
- Filegroups

Script Help

Collation: Greek\_CI\_AS

Recovery model: Simple

Compatibility level: SQL Server 2019 (150)

Containment type: None

Other options:

- Επιλέγουμε Options Αριστερά και μετά Collation **Greek\_CI\_AS** ή **Greek\_100\_CI\_AI**. Το Greek είναι για να υποστηρίζει και Ελληνικούς χαρακτήρες εκτός από Λατινικούς. Το 100 είναι πιο καινούργιος αλγόριθμος ταξινόμησης και σύγκρισης. Το CI σημαίνει Case Insensitive, δηλαδή στις συγκρίσεις τιμών να μην λαμβάνει υπόψη πεζά-κεφαλαία, και το AS σημαίνει Accent Sensitive, δηλαδή στις συγκρίσεις να λαμβάνει υπόψη τους τόνους, αν υπάρχουν σε αλφαριθμητικά και να διαφοροποιούνται από τιμές αλφαριθμητικών χωρίς τόνους. Το AI, δεν λαμβάνει υπόψη τόνους



# Collation (1)

- Τα Collations τα συναντάμε σε όλα τα συστήματα Βάσεων Δεδομένων
- Ένα Collation είναι ένας συγκεκριμένος τρόπος κωδικοποίησης (bit-patterns) των χαρακτήρων μία γλώσσας. Κάθε χαρακτήρας αναπαρίσταται από ένα bit-pattern
- Ένα Collation μπορεί να είναι CS (Case Sensitive) ή CI (Case Insensitive) και να διαχωρίζει πεζά-κεφαλαία ή όχι, AC (Accent Sensitive) ή AI (Accent Insensitive) και να διαχωρίζει λέξεις με τόνους και χωρίς τόνους, ή όχι, BIN, που ταξινομεί τα δεδομένα στους Πίνακες της ΒΔ και by default είναι CS και AS



# Collation (2)

- Το default Collation στον SQL Server εξαρτάται από το Locale (την γλώσσα του Λειτουργικού Συστήματος)
- Αν η γλώσσα είναι Αγγλικά (en-US) τότε το Default Collation είναι SQL\_Latin1\_General\_CP1\_CI\_AS, δηλαδή Λατινικοί χαρακτήρες με κωδικοποίηση CP1, δηλαδή CodePage 1252 που είναι Latin
- Αν σε αυτή την κωδικοποίηση δώσουμε Ελληνικούς χαρακτήρες, δεν αναγνωρίζονται
- Για αυτό χρησιμοποιούμε ρητά Collation Greek





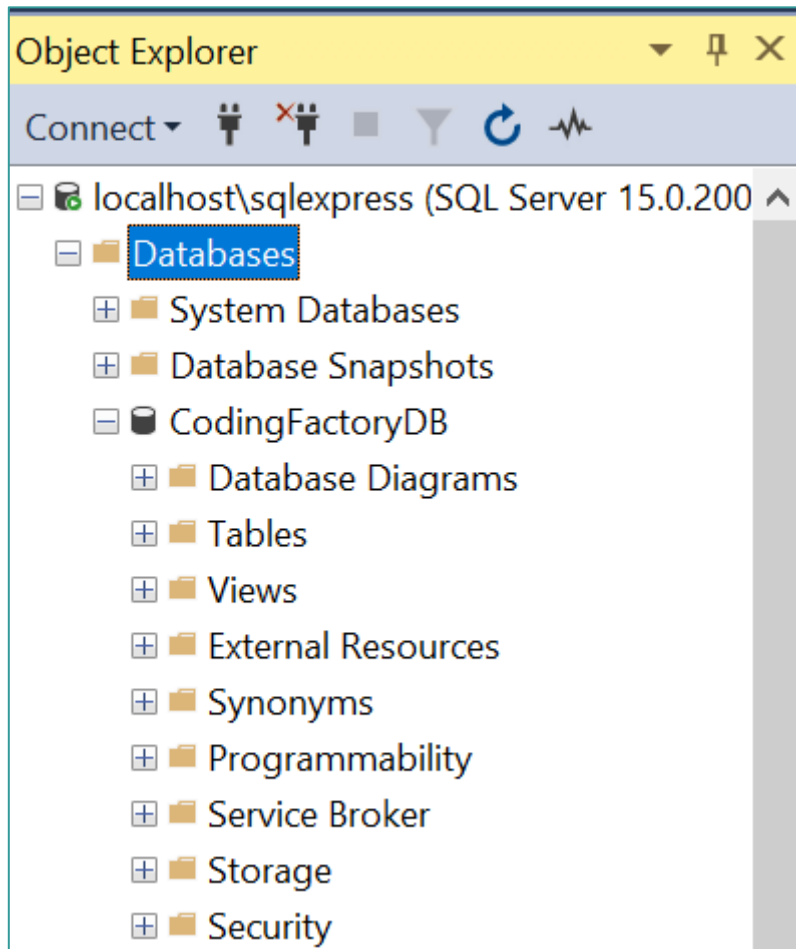
# Collation (3)

- Αν ωστόσο θέλαμε να απευθυνθούμε στην παγκόσμια αγορά και να μπορούμε εκτός από ελληνικούς / λατινικούς χαρακτήρες, να αποθηκεύουμε και international characters (για παράδειγμα Ασιατικούς χαρακτήρες), τότε θα πρέπει να χρησιμοποιούμε UTF-8 (Unicode charsets) Collations
- Τα Unicode Collations χρησιμοποιούνε 2 bytes για την αναπαράσταση χαρακτήρων αντί 1 byte που χρησιμοποιούν τα Non-Unicode Collations



# Νέα Βάση Δεδομένων (4)

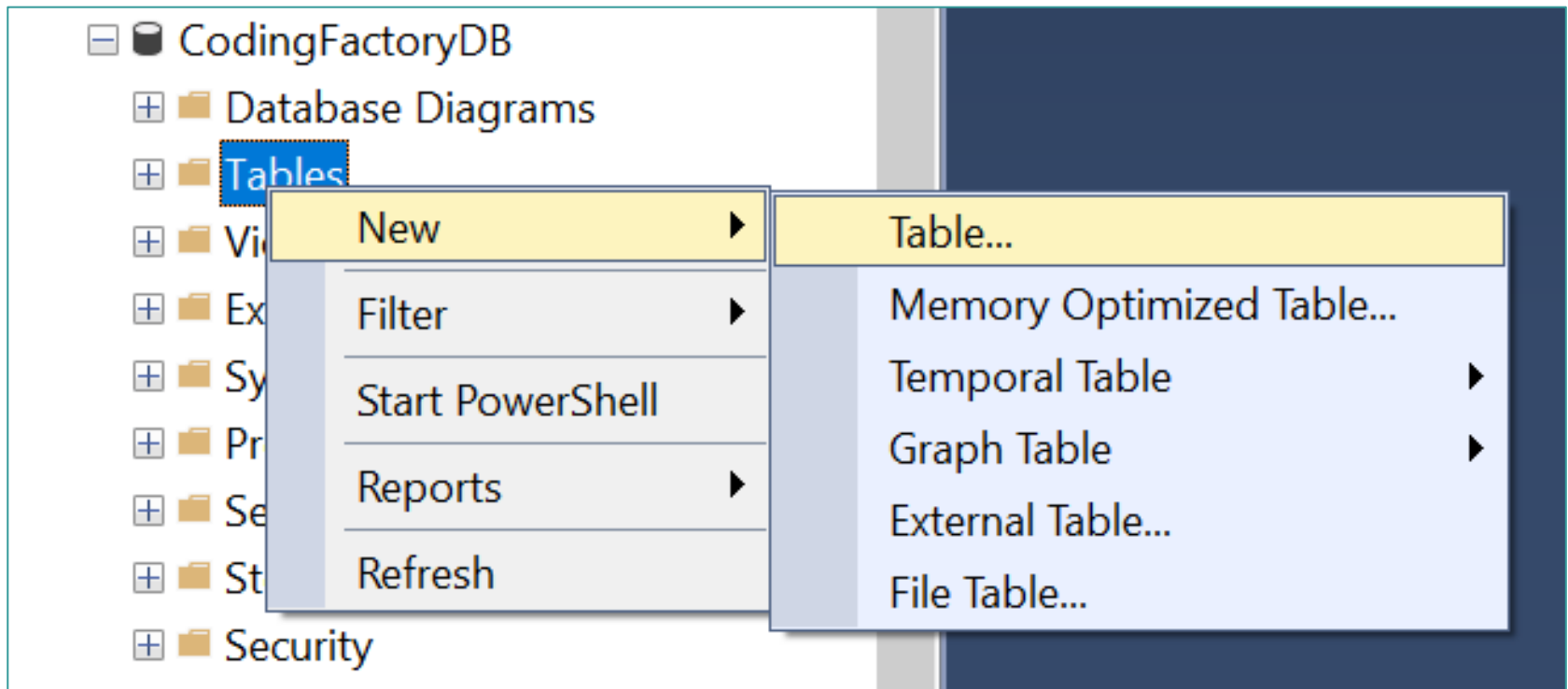
Βάσεις Δεδομένων



- Δημιουργήσαμε τη ΒΔ
- Κάτω από τη ΒΔ υπάρχει μία δομή φακέλων
- Στον φάκελο Tables θα δημιουργήσουμε τους πίνακες



# Πίνακες



- Δημιουργούμε νέο Πίνακα επιλέγοντας Tables / New / Table...



# Πεδία Πίνακα

## Βάσεις Δεδομένων

The screenshot displays the SQL Server Enterprise Manager interface. The main window shows the design of a table named 'Table\_1' in the 'dbo' schema. The table has a single column 'ID' of type 'int' with 'Allow Nulls' set to 'No'. The 'Properties' pane on the right shows the table's settings, including 'Identity' and 'Table Designer' sections.

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>

Properties

[Tbl] dbo.TEACHERS

(Identity)

(Name) TEACHERS

Database Name CodingFactoryDB

Description

Schema dbo

Server Name thanassis-pc\squlexpress

Table Designer

Identity Column ID

Indexable Yes

Lock Escalation Table

Regular Data Space Specifica PRIMARY

Replicated No

Row GUID Column

Text/Image Filegroup PRIMARY

- Εισάγουμε το πεδίο ID, μετά δεξί κλικ πάνω του και Set Primary Key, δίνουμε τύπο int (είναι surrogate key, οπότε είναι ακέραιος), δίνουμε δεξιά στα properties identity Column το ID, ώστε να αυξάνεται αυτόματα το ID (auto-increment). Επίσης, δεν θέλουμε Nulls (το NULL σημαίνει ΤΙΠΟΤΑ, απολύτως τίποτα, όχι απλά κενό αλφαριθμητικό) μιας και το ID είναι Πρωτεύον κλειδί και θέλουμε σε κάθε περίπτωση να έχει τιμή



# Identity Specification

## Βάσεις Δεδομένων

- Όσο αφορά το Identity Column και τη δυνατότητα το ID να λαμβάνει αυτόματα τιμές αυξανόμενες (κατά 1 σε κάθε νέα εγγραφή), ξεκινώντας από το 1, μπορεί να οριστεί και από τα Column Properties του πεδίου στο κάτω μέρος

THANASSIS-PC\SQL...yDB - dbo.Table\_1\* [Icon] [X]

	Column Name	Data Type
[Icon]	ID	int

Column Properties

[Icon] [Icon] [Icon]

Deterministic	Yes
DTS-published	No
> Full-text Specification	No
Has Non-SQL Server Subscriber	No
▼ Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1



# Surrogate Keys

- Κατά αυτό τον τρόπο (με Identity Specification) ορίζουμε Surrogate keys

The screenshot shows the SQL Server Enterprise Manager interface. The main window displays the 'dbo.Table\_1\*' table. The 'Column Properties' window is open for the 'ID' column, showing the 'Identity Specification' tab. The 'Properties' window is open for the table 'dbo.TEACHERS', showing the 'Table Designer' tab.

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>

Property	Value
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1

Property	Value
(Name)	TEACHERS
Database Name	CodingFactoryDB
Description	
Schema	dbo
Server Name	thanassis-pc\sqlexpress
Table Designer	
Identity Column	ID
Indexable	Yes
Lock Escalation	Table
Regular Data Space Specifica	PRIMARY
Replicated	No
Row GUID Column	
Text/Image Filegroup	PRIMARY



# Table TEACHERS

THANASSIS-PC\SQL...yDB - dbo.Table_1* X			
	Column Name	Data Type	Allow Nulls
🔑	ID	int	<input type="checkbox"/>
	FIRSTNAME	nvarchar(50)	<input checked="" type="checkbox"/>
	LASTNAME	nvarchar(50)	<input checked="" type="checkbox"/>
	PHONE_NUMBER	nvarchar(50)	<input checked="" type="checkbox"/>
	ADDRESS	nchar(255)	<input checked="" type="checkbox"/>

- Κατά τον ίδιο τρόπο εισάγουμε και τα υπόλοιπα πεδία. Πρώτα το όνομα και μετά τον τύπο του πεδίου. Ο τύπος των πεδίων κειμένου (αλφαριθμητικών / strings) είναι `nvarchar(n)` όπου `n` το μέγεθος του πεδίου. Ο τύπος δεδομένων **nvarchar** επιτρέπει αλφαριθμητικά Unicode (δηλαδή από όλες τις γλώσσες του κόσμου)
- Δίνουμε Ctrl + S για Save και το όνομα του πίνακα ως TEACHERS



# Αναζήτηση (1)

- Η βασική και θεμελιώδης λειτουργικότητα που παρέχει μία ΒΔ είναι η **αναζήτηση**
- Η αναζήτηση πρέπει να γίνεται γρήγορα, κάτι που είναι δύσκολο ιδιαίτερα αν έχουμε εκατομμύρια ή δισεκατομμύρια εγγραφές





# Αναζήτηση (2)

- Γενικά, αναζήτηση σε συλλογές δεδομένων μπορεί να γίνει τρεις τρόπους
  - **Σειριακά.** Δηλαδή αρχίζουμε από την αρχή και αναζητάμε με βάση ένα πεδίο (το επώνυμο του teacher, για παράδειγμα) αν υπάρχει αυτός ο Καθηγητής και αν υπάρχει να μας προσκομίσει τα στοιχεία του. Το αντίστοιχο της σειριακής αναζήτησης είναι να ξεφυλλίζουμε ένα τηλεφωνικό κατάλογο από την αρχή μέχρι το τέλος για να βρούμε ένα επώνυμο που αναζητούμε



# Αναζήτηση (3)

- Δημιουργώντας ένα **ευρετήριο (Index)**
  - Όπως σε ένα τηλεφωνικό κατάλογο, μπορούμε στην αρχή να έχουμε ένα ευρετήριο, όπου να βρούμε γρήγορα ένα επώνυμο και να μας παραπέμπει σε μία σελίδα που υπάρχουν τα στοιχεία που αναζητούμε, έτσι και στα Συστήματα ΒΔ μπορούμε να δημιουργήσουμε Ευρετήρια (Indexes) ώστε να επιταχύνουμε την διαδικασία αναζήτησης
  - Τα ευρετήρια στις ΒΔ υλοποιούνται με μία δομή δεδομένων που ονομάζεται B-Δένδρα (B-Trees) τα οποία παρέχουν πολύ γρήγορους χρόνους αναζήτησης



# Αναζήτηση (4)

- Συνακόλουθα αν ένα ευρετήριο βρει με βάση το επώνυμο, την σελίδα στον τηλεφωνικό κατάλογο, στην οποία βρίσκεται η εγγραφή που αναζητούμε (αναφερόμαστε με όρους τηλεφωνικού καταλόγου), τότε θα πρέπει να μπορούμε να πάμε στη σελίδα με μία κίνηση, όχι ξεφυλλίζοντας πάλι τον κατάλογο μέχρι να φτάσουμε στη σελίδα






# Αναζήτηση (5)




- Κατ' αναλογία για να μπορέσουμε να πάμε μέσω ενός Index, στη θέση στο σκληρό δίσκο που βρίσκεται η πραγματική εγγραφή, υπάρχει ένα μηχανισμός που ονομάζεται *HashCode*, όπου η κάθε τιμή ενός πεδίου αναζήτησης (για παράδειγμα, Lastname) αντιστοιχίζεται με μοναδικό τρόπο σε ένα αριθμό (hash) και αυτός ο αριθμός μας παραπέμπει αυτόματα στη θέση στον δίσκο όπου υπάρχει η εγγραφή





# Index (Ευρετήριο)

THANASSIS-PC\SQL...B - dbo.TEACHERS\*  

	Column Name	Data Type
	ID	int
	FIRSTNAME	nvarchar(50)
▶	LASTNAME	nvarchar(50)
	PHONE_NUMBER	
	ADDRESS	
	SPECIALITY_ID	

 Set Primary Key  
 Insert Column  
 Delete Column  

---

 Relationships...  
 Indexes/Keys...

- Μπορούμε να ορίσουμε Index (Ευρετήριο) σε πεδία που χρησιμοποιούνται σε αναζητήσεις ώστε να επιταχύνουμε την αναζήτηση
- Επειδή θα θέλαμε ένα Teacher να μπορούμε να τον αναζητούμε (γρήγορα) και με το Επώνυμο, θα ορίσουμε ένα Index (Ευρετήριο) στο LASTNAME



# Index (Ευρετήριο)

## Βάσεις Δεδομένων

- Το PK\_TEACHERS υπάρχει ήδη για γρήγορες αναζητήσεις με βάση το πρωτεύον κλειδί
- Θα κάνουμε Add ένα νέο Index IX\_TEACHERS και μετά θα πατήσουμε τις τελίτσες για να ορίσουμε το Index

Indexes/Keys

Selected Primary/Unique Key or Index:

IX\_TEACHERS\*  
PK\_TEACHERS

Editing properties for new unique key or index.

▼ (General)	
Columns	ID (ASC) <span>...</span>
Is Unique	No
Type	Index
▼ Identity	
(Name)	IX_TEACHERS
Description	
▼ Table Designer	
Create As Clustered	No
▶ Data Space Specification	PRIMARY
▶ Fill Specification	
Ignore Duplicate Keys	No
Included Columns	

Add Delete Close



# Ορισμός Index

Indexes/Keys

Selected Primary/Unique Key or Index:

IX\_TEACHERS\*  
PK\_TEACHERS

Editing properties for new unique key or index.

▼ (General)	
Columns	LASTNAME (ASC)
Is Unique	No
Type	Index
▼ Identity	
(Name)	IX_TEACHERS
Description	
▼ Table Designer	
Create As Clustered	No
➤ Data Space Specification	PRIMARY
➤ Fill Specification	
Ignore Duplicate Keys	No
Included Columns	

Add Delete Close

- Ορίζουμε το Index στο Επώνυμο (LASTNAME) και κάνουμε Close



# Ευρετήρια

- Τα ευρετήρια, όπως αναφέραμε, δημιουργούν μία δομή δεδομένων που ονομάζεται B-Tree, η οποία είναι μία δομή που περιέχει το πεδίο καθώς και τη θέση στο δίσκο (hashCode) της αντίστοιχης εγγραφής, επιταχύνοντας πάρα πολύ τη διαδικασία αναζήτησης με βάση αυτό το πεδίο





# Naming Conventions (1)

- Ως convention στο όνομα του Πίνακα και των πεδίων μπορούμε να δίνουμε είτε όλα κεφαλαία ή όλα πεζά ή το πρώτο κεφαλαίο και τα υπόλοιπα πεζά
- Επειδή τα ονόματα πινάκων αντιστοιχούν πολλές φορές σε ονόματα αρχείων και επειδή διαφορετικά Λειτουργικά Συστήματα χρησιμοποιούν διαφορετικούς τρόπους διάκρισης ή μη μεταξύ πεζών-κεφαλαίων καλό είναι ότι convention χρησιμοποιούμε, να είναι πάντα το ίδιο



# Naming Conventions (2)

- Στο παράδειγμα χρησιμοποιούμε το convention, **όλα κεφαλαία**, όπως προτείνει και το SQL-92 standard, παρότι οι επιμέρους υλοποιήσεις δεν το ακολουθούν πάντα
- Επίσης, όσο αφορά τον ενικό ή πληθυντικό στο όνομα του Πίνακα, πάλι μπορεί να διαφέρει το convention. Το convention που χρησιμοποιούμε στα παραδείγματα είναι **με πληθυντικό (plural)** (π.χ. TEACHERS) μιας και οι πίνακες είναι συλλογές εγγραφών



# Varchar vs nvarchar (1)

- Ως τύπος ενός αλφαριθμητικού πεδίου μπορεί να είναι ο **varchar** που δεν είναι Unicode, αλλά σε συνδυασμό με το Collation που είναι Greek μπορεί να δέχεται Ελληνικούς χαρακτήρες
- Ωστόσο το πλήρες σύνολο χαρακτήρων Unicode (Unicode είναι το σύνολο χαρακτήρων όλων των γλωσσών του κόσμου) το δίνει ο τύπος **nvarchar**



# Varchar vs nvarchar (2)

- Ο τύπος `varchar` δεσμεύει 1 byte ανά χαρακτήρα, ενώ ο `nvarchar` 2 bytes ανά χαρακτήρα
- Και οι δύο τύποι είναι **μεταβλητού μεγέθους**, δηλαδή δεσμεύουν τόσο χώρο όσο είναι το μήκος του αλφαριθμητικού, με `max` το πλήθος των χαρακτήρων που ορίζουμε, π.χ. `nvarchar(50)` δεσμεύει `max` 50 χαρακτήρες, 100 bytes, αλλά αν ένα αλφαριθμητικό είναι 6 χαρακτήρες θα δεσμευτεί χώρος για 6 χαρακτήρες και όχι για 50
- Πλέον στις καινούργιες εφαρμογές χρησιμοποιούμε **`nvarchar`** ώστε να μπορούμε να αποθηκεύουμε οποιονδήποτε χαρακτήρα από το παγκόσμιο σύστημα χαρακτήρων

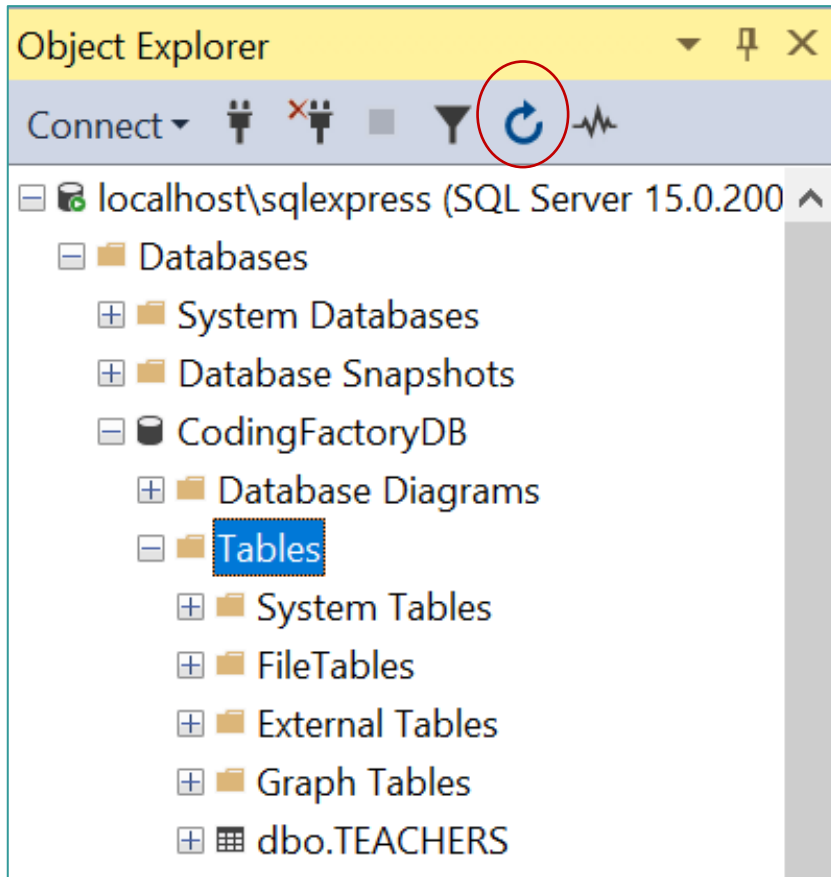


# nchar / nvarchar

- Ο τύπος `nchar(n)` είναι παρόμοιος με τον `nvarchar` αλλά σταθερού μεγέθους και όχι μεταβλητού όπως ο `nvarchar(n)`.
- Ο `nchar(10)` δεσμεύει  $10 * 2 = 20$  θέσεις ακόμα και αν το αλφαριθμητικό εισόδου είναι μικρότερο, ενώ ο τύπος `nvarchar(10)` δεσμεύει  $n * 2$  θέσεις, όπου  $n$  είναι το πραγματικό μήκος του αλφαριθμητικού εισόδου, με  $\max$  τους 10 χαρακτήρες, άρα  $\max$  20 bytes
- Το  $*2$  αφορά το γεγονός ότι στο σύστημα Unicode ο κάθε χαρακτήρας δεσμεύει δύο bytes



# Refresh



- Αν κάνουμε refresh, εμφανίζεται ο πίνακας TEACHERS ως dbo.TEACHERS



# SPECIALITIES

## Βάσεις Δεδομένων

The screenshot displays the SQL Server Enterprise Designer interface. The main window shows the design of a table named 'TEACHERS' in the 'dbo' schema. The table has two columns: 'ID' of type 'int' and 'TITLE' of type 'nvarchar(50)'. The 'ID' column is marked as the primary key and has the 'Allow Nulls' checkbox unchecked. The 'TITLE' column has the 'Allow Nulls' checkbox checked. The 'Properties' window on the right shows the details for the selected table, including its name, database name, schema, and server name. The 'Table Designer' tab is active, showing the 'ID' column as the identity column.

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
TITLE	nvarchar(50)	<input checked="" type="checkbox"/>

**Properties**

**[Tbl] dbo.Table2**

**(Identity)**

(Name)	Table2
Database Name	CodingFactoryDB
Description	
Schema	dbo
Server Name	thanassis-pc\squlexpress

**Table Designer**

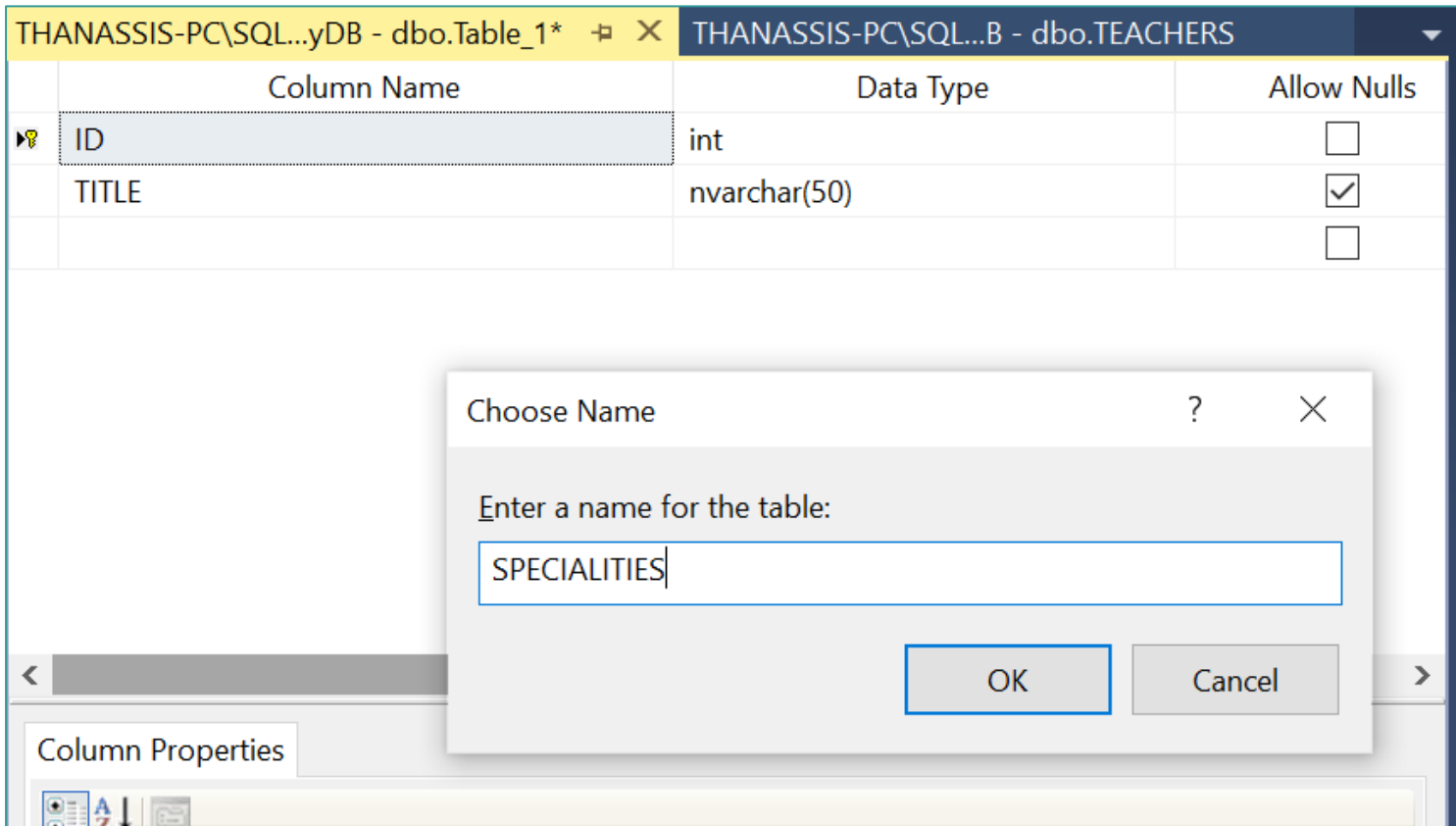
Identity Column	ID
-----------------	----

- Προσθέτουμε άλλο ένα πίνακα με όνομα SPECIALITIES που αποθηκεύει τις Ειδικότητες, δηλαδή την Ειδικότητα που μπορεί να έχει ένας Καθηγητής (Φυσικός, Φιλόλογος, κλπ.)
- Περιλαμβάνει ένα ID που είναι τεχνητό κλειδί, auto-increment, και ένα πεδίο TITLE που είναι ο τίτλος της ειδικότητας



# Ctrl + S - Αποθήκευση

Βάσεις Δεδομένων



- Πατάμε Ctrl + S, δίνουμε το όνομα του πίνακα και αποθηκεύουμε





# Σχέση Πινάκων

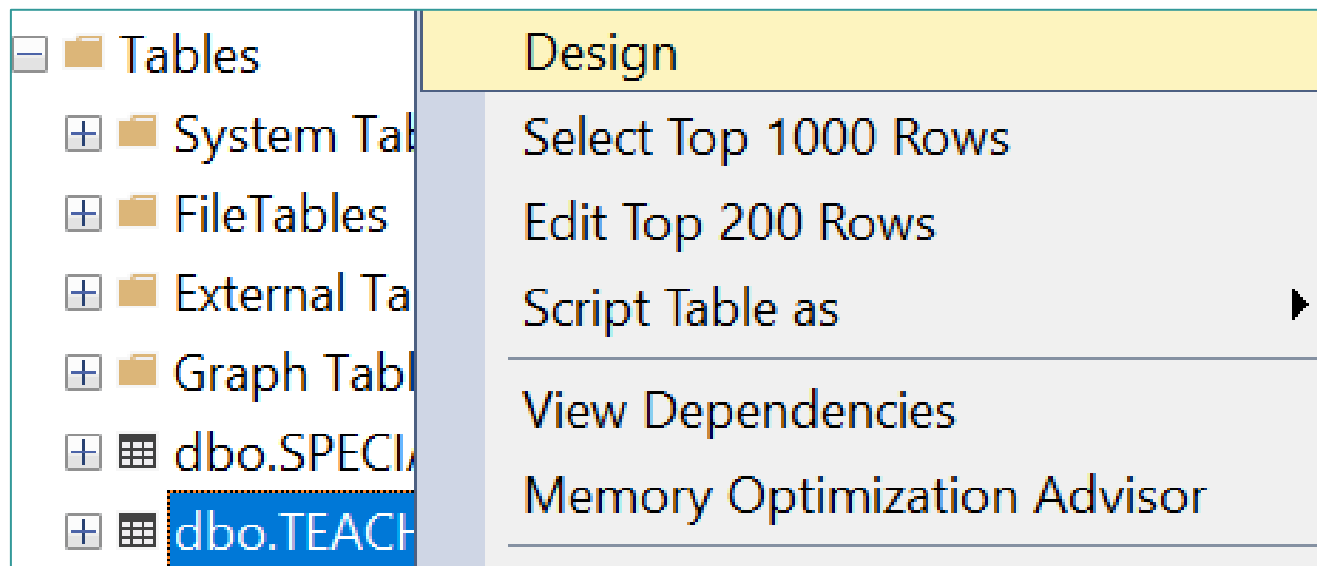
Βάσεις Δεδομένων

- Ένας καθηγητής μπορεί να έχει μία ειδικότητα
- Μία ειδικότητα μπορεί να την έχουν πολλοί Καθηγητές
- Επομένως, η σχέση είναι **ένα-προς-πολλά** από τον SPECIALITIES προς τον TEACHERS
- Θα προσθέσουμε στο μέρος του προς-πολλά δηλαδή στον πίνακα TEACHERS ένα πεδίο SPECIALITY\_ID που θα είναι το ξένο κλειδί και θα κάνει Reference το ID του SPECIALITIES



# Ξένο Κλειδί (1)

- Δεξί κλικ στον TEACHERS και επιλέγουμε Design





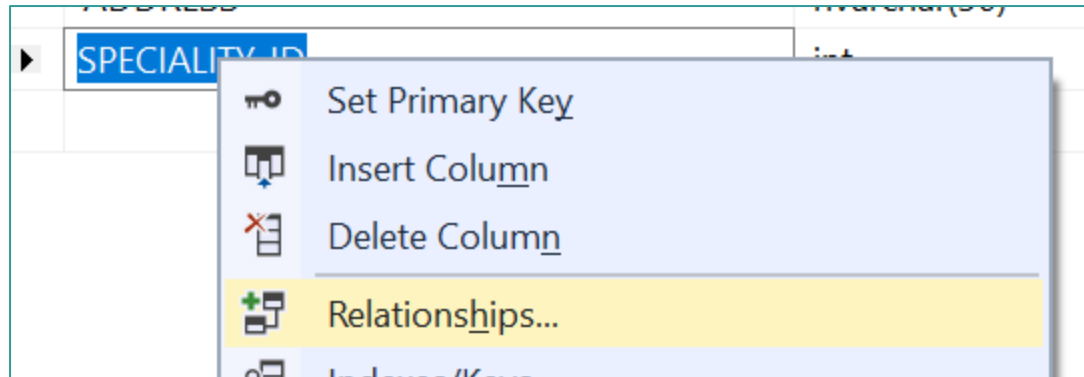
## Ξένο Κλειδί (2)

THANASSIS-PC\SQL...B - dbo.TEACHERS			
	Column Name	Data Type	Allow Nulls
🔑	ID	int	<input type="checkbox"/>
	FIRSTNAME	nvarchar(50)	<input checked="" type="checkbox"/>
	LASTNAME	nvarchar(50)	<input checked="" type="checkbox"/>
	PHONE_NUMBER	nvarchar(50)	<input checked="" type="checkbox"/>
	ADDRESS	nvarchar(50)	<input checked="" type="checkbox"/>
▶	SPECIALITY_ID	int	<input type="checkbox"/>
			<input type="checkbox"/>

- Εισάγουμε το πεδίο SPECIALITY\_ID με τύπο δεδομένων (Data Type) ακριβώς τον ίδιο όπως το ID του TEACHERS, δηλαδή **int** και προαιρετικά χωρίς NULL



# Ξένο Κλειδί (3)



- Δεξί κλικ πάνω στο SPECIALITY\_ID και επιλέγουμε Relationships



# Ξένο Κλειδί (4)

## Βάσεις Δεδομένων

Foreign Key Relationships

Selected Relationship: FK\_TEACHERS\_SPECIALITIES

Editing properties for existing relationship.

▼ (General)  
Check Existing Data On Cr Yes

▼ Tables And Columns Spec  
Foreign Key Base Table TEACHERS  
Foreign Key Columns ID  
Primary/Unique Key B TEACHERS  
Primary/Unique Key C ID

▼ Identity  
(Name) FK\_TEACHERS\_SPECIALITIES  
Description

▼ Table Designer  
Enforce For Replication Yes  
Enforce Foreign Key Const Yes

Add Delete Close

- Αλλάζουμε το όνομα του FK Constraint σε FK\_TEACHERS\_SPECIALITIES για να υπάρχει συνάφεια
- Πατάμε τις τρεις τελίτσες για να ορίσουμε τη σχέση του ξένου κλειδιού



# Ξένο Κλειδί (5)

Tables and Columns

Relationship name:  
FK\_TEACHERS\_SPECIALITIES

Primary key table:  
SPECIALITIES  
ID

Foreign key table:  
TEACHERS  
SPECIALITY\_ID

- Ορίζουμε τη σχέση ώστε το Primary Key Table να είναι το SPECIALITIES, και ως Foreign Key Table, που είναι το μέρος του προς-πολλά, επιλέγουμε TEACHERS και ως πεδίο το SPECIALITY\_ID και δίνουμε OK
- Επομένως, ορίζουμε τη σχέση (για την ακρίβεια το Constraint) SPECIALITIES/ID (πρωτεύον κλειδί) με TEACHERS/SPECIALITY\_ID (ξένο κλειδί)



# Ξένο Κλειδί (6)

- Πατάμε Close

Foreign Key Relationships

Selected Relationship: FK\_TEACHERS\_SPECIALITIES

Editing properties for existing relationship.

▼ **(General)**

Check Existing Data On Cr Yes

▼ **Tables And Columns Spec** ...

Foreign Key Base Table TEACHERS

Foreign Key Columns SPECIALITY\_ID

Primary/Unique Key Base SPECIALITIES

Primary/Unique Key Columns ID

▼ **Identity**

(Name) FK\_TEACHERS\_SPECIALITIES

Description

▼ **Table Designer**

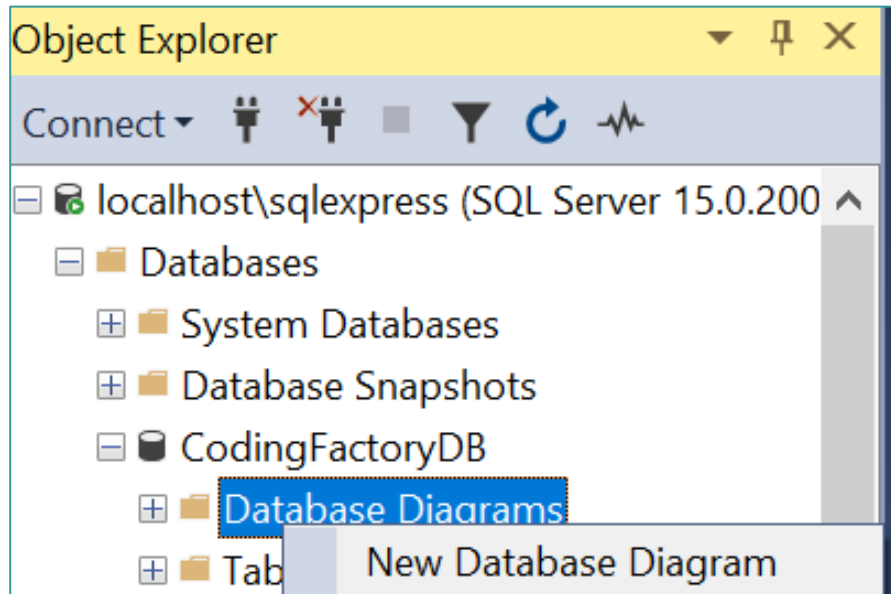
Enforce For Replication Yes

Enforce Foreign Key Constraints Yes

Add Delete Close



# ER-Diagram (1)



- Μπορούμε να φτιάξουμε και το ER-Diagram. Δεξί κλικ στο Database Diagrams και New Database Diagram





# ER-Diagram (2)

Microsoft SQL Server Management Studio



This database does not have one or more of the support objects required to use database diagramming. Do you wish to create them?



Copy message

Yes

No

- Αποδεχόμαστε (Yes) ώστε το SSMS να προσθέσει τα support objects to use database diagramming, όπως μας ενημερώνει



# ER-Diagram (3)

Add Table

Tables

SPECIALITIES

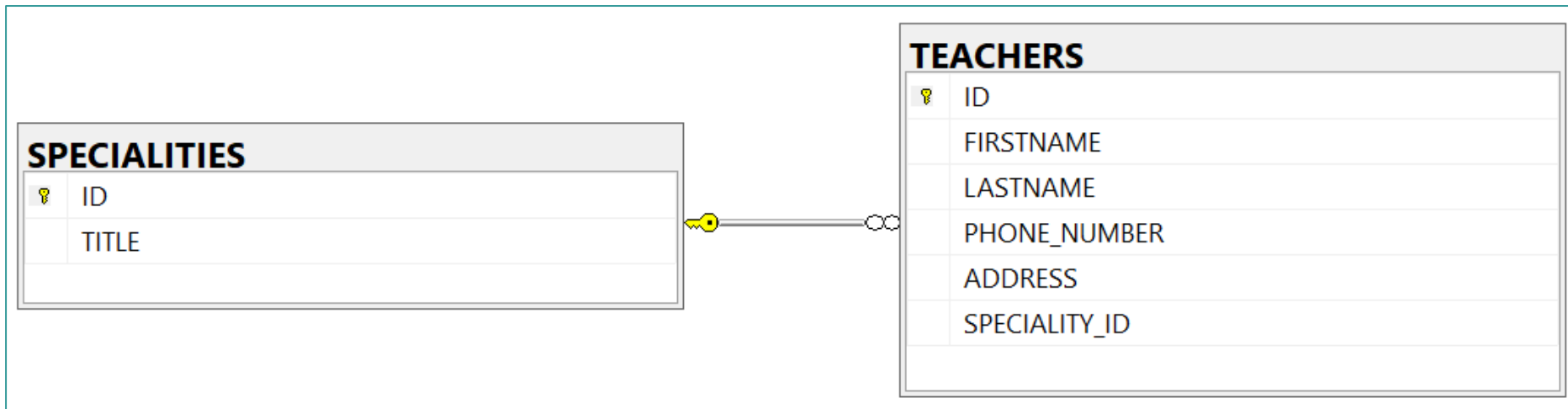
TEACHERS

Refresh Add Close

- Επιλέγουμε και προσθέτουμε (Add) και τους δύο πίνακες



# ER-Diagram (4)



- Εμφανίζεται το ER Diagram



# Αποθήκευση

Choose Name

Enter a name for the diagram:

ERD1

OK Cancel

- Αποθηκεύουμε ως ERD1



# SQL (1)

- Είδαμε πως μπορούμε εύκολα να δημιουργήσουμε μία ΒΔ στον SQL Express χάρη στο γραφικό περιβάλλον δημιουργίας πινάκων και σχέσεων που μας δίνει
- Στην πραγματικότητα, στο παρασκήνιο (under the hood) το γραφικό περιβάλλον δημιουργεί κώδικα σε μία γλώσσα προγραμματισμού ΒΔ, που ονομάζεται SQL (Structured Query Language)

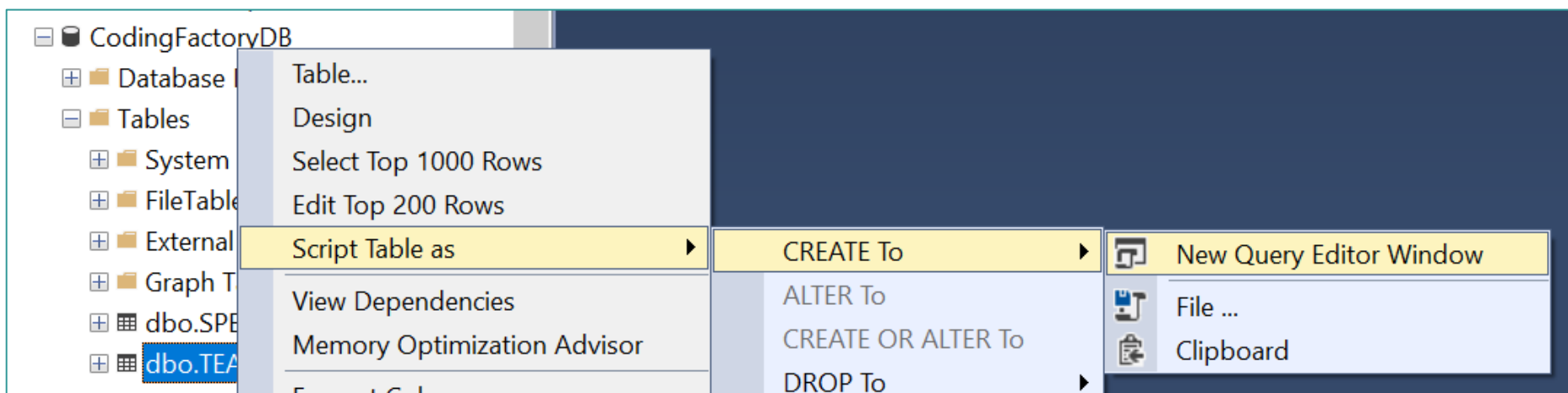


# SQL (2)

- Η SQL όπως θα δούμε είναι μία δηλωτική (declarative) γλώσσα προγραμματισμού
- Δηλωτικές είναι οι γλώσσες προγραμματισμού, που τις 'προστάζουμε' να κάνουν κάτι χωρίς να προγραμματίζουμε εμείς το πως θα το κάνουν



# CREATE TABLE (1)



- Κάνουμε δεξί κλικ πάνω στον Πίνακα TEACHERS και επιλέγουμε **Script Table as / CREATE To / New Query Editor Window**



# CREATE TABLE (2)

```
CREATE TABLE [dbo].[TEACHERS](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [FIRSTNAME] [nvarchar](50) NULL,
    [LASTNAME] [nvarchar](50) NULL,
    [PHONE_NUMBER] [nvarchar](50) NULL,
    [ADDRESS] [nvarchar](50) NULL,
    [SPECIALITY_ID] [int] NOT NULL,
    CONSTRAINT [PK_TEACHERS] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE
) ON [PRIMARY]
GO
```

- Η CREATE TABLE είναι η εντολή της SQL για τη δημιουργία πινάκων. Παρατηρούμε ότι ο έτοιμος κώδικας έχει όλα τα identifiers (ονόματα πινάκων και στηλών) καθώς και τους τύπους δεδομένων μέσα σε brackets []
- Brackets χρειάζονται μόνο αν έχουμε κενά στα identifiers ή είναι keywords. Προσεκτικά επιλεγμένα identifiers δεν χρειάζονται []





# Identifiers (Αναγνωριστικά)

Βάσεις Δεδομένων

- Σε αντιδιαστολή με τις δεσμευμένες λέξεις της γλώσσας (reserved words ή Keywords), όπως CREATE TABLE, CONSTRAINT, PRIMARY KEY, CLUSTERED, κλπ., τα αναγνωριστικά ή identifiers είναι ονόματα που **επιλέγουμε εμείς, όπως για παράδειγμα** το όνομα του πίνακα, των πεδίων των πινάκων και γενικά για τα database objects



# Κανόνες Σύνταξης Identifiers

Βάσεις Δεδομένων

- Τα Αναγνωριστικά έχουν κανόνες σύνταξης:
  - Ξεκινάνε με γράμμα (και όχι αριθμό) ή με το underscore (`_`), at sign (`@`), ή το hash (`#`) και ακολουθούνται από γράμμα ή αριθμό ή `_`, `@`, `#`, `$`
  - Τα identifiers δεν μπορεί να είναι reserved words
  - Τα αλφαριθμητικά που δεν συμμορφώνονται με τους παραπάνω κανόνες πρέπει να εισάγονται μέσα σε brackets `[]` ή σε double quotes `" "`



# CREATE TABLE (2)

```
CREATE TABLE dbo.TEACHERS(  
    ID int IDENTITY(1,1) NOT NULL,  
    FIRSTNAME nvarchar(50) NULL,  
    LASTNAME nvarchar(50) NULL,  
    PHONE_NUMBER nvarchar(50) NULL,  
    ADDRESS nvarchar(50) NULL,  
    SPECIALITY_ID int NOT NULL,  
    CONSTRAINT PK_TEACHER PRIMARY KEY CLUSTERED  
    (  
        ID ASC  
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE  
    ) ON [PRIMARY]  
GO
```

- Χωρίς [] ο κώδικας είναι πιο ευανάγνωστος
- Κάθε γραμμή της CREATE TABLE έχει τον *identifier* (το όνομα του πεδίου της στήλης), τον τύπο δεδομένων του πεδίου και NULL ή NOT NULL αντίστοιχα.



# Primary Key constraint

```
CREATE TABLE [dbo].[TEACHERS](  
    [ID] [int] IDENTITY(1,1) NOT NULL,  
    [FIRSTNAME] [nvarchar](50) NULL,  
    [LASTNAME] [nvarchar](50) NULL,  
    [PHONE_NUMBER] [nvarchar](50) NULL,  
    [ADDRESS] [nvarchar](50) NULL,  
    [SPECIALITY_ID] [int] NOT NULL,  
    CONSTRAINT [PK_TEACHERS] PRIMARY KEY ([ID] ASC)  
)
```

- Τα PRIMARY KEYS ορίζουν περιορισμούς (constraints), ότι δηλαδή οι τιμές του αντίστοιχου πεδίου πρέπει να είναι μοναδικές. Στην απλή τους μορφή ορίζονται όπως παραπάνω
- Αν όμως ορίζουν και ένα ευρετήριο για την ταχύτερη αναζήτηση εγγραφών με βάση το πρωτεύον κλειδί (και πρέπει να ορίζουν), τότε πρέπει να έχουμε και τον ορισμό του Index ταυτόχρονα (βλ. επόμενη διαφάνεια)



# Index (1)

- Ο clustered Index ορίζει την διάταξη (**αύξουσα Ascending/ φθίνουσα Descending**) με την οποία αποθηκεύονται οι γραμμές του πίνακα στον φυσικό δίσκο
- Επειδή οι εγγραφές μπορούν να αποθηκεύονται με μία μόνο διάταξη, επιτρέπεται μόνο ένα clustered index, που ορίζεται (**ON**) συνήθως στο πρωτεύον κλειδί
- Όλα τα άλλα ευρετήρια που δημιουργούμε είναι non-clustered

```
CONSTRAINT [PK_TEACHERS] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE =
) ON [PRIMARY]
GO
```



# Index (2)

- Όλα τα άλλα ευρετήρια είναι non-clustered και δημιουργούν Β-Δένδρα που αναφέρονται στις πραγματικές εγγραφές στο σκληρό δίσκο

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'CodingFactoryDB' database is expanded, showing a tree view of its components: Database Diagrams, Tables, System Tables, FileTables, External Tables, Graph Tables, and several system tables (dbo.SPECIALITIES, dbo.TEACHERS). Under 'dbo.TEACHERS', the 'Indexes' folder is expanded, showing two indexes: 'IX\_TEACHERS (Non-Unique, Non-Cluster)' and 'PK\_TEACHERS (Clustered)'. The 'IX\_TEACHERS' index is circled in red. On the right, a T-SQL query is shown in a text editor, creating a non-clustered index on the 'TEACHERS' table:

```
CREATE NONCLUSTERED INDEX [IX_TEACHERS] ON [dbo].[TEACHERS]
(
    [LASTNAME] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_
GO
```



# SPECIALITIES – CREATE TABLE

Βάσεις Δεδομένων

```
CREATE TABLE [dbo].[SPECIALITIES](  
    [ID] [int] IDENTITY(1,1) NOT NULL,  
    [TITLE] [nvarchar](50) NULL,  
    CONSTRAINT [PK_SPECIALITIES] PRIMARY KEY CLUSTERED  
    (  
        [ID] ASC  
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF  
    ) ON [PRIMARY]  
GO
```

- Η ίδια σύνταξη ακολουθείται και στο CREATE TABLE του πίνακα SPECIALITIES



# WITH (PAD\_INDEX = OFF, ...)

```
CREATE TABLE [dbo].[SPECIALITIES](  
    [ID] [int] IDENTITY(1,1) NOT NULL,  
    [TITLE] [nvarchar](50) NULL,  
    CONSTRAINT [PK_SPECIALITIES] PRIMARY KEY CLUSTERED  
    (  
        [ID] ASC  
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF  
    ) ON [PRIMARY]  
GO
```

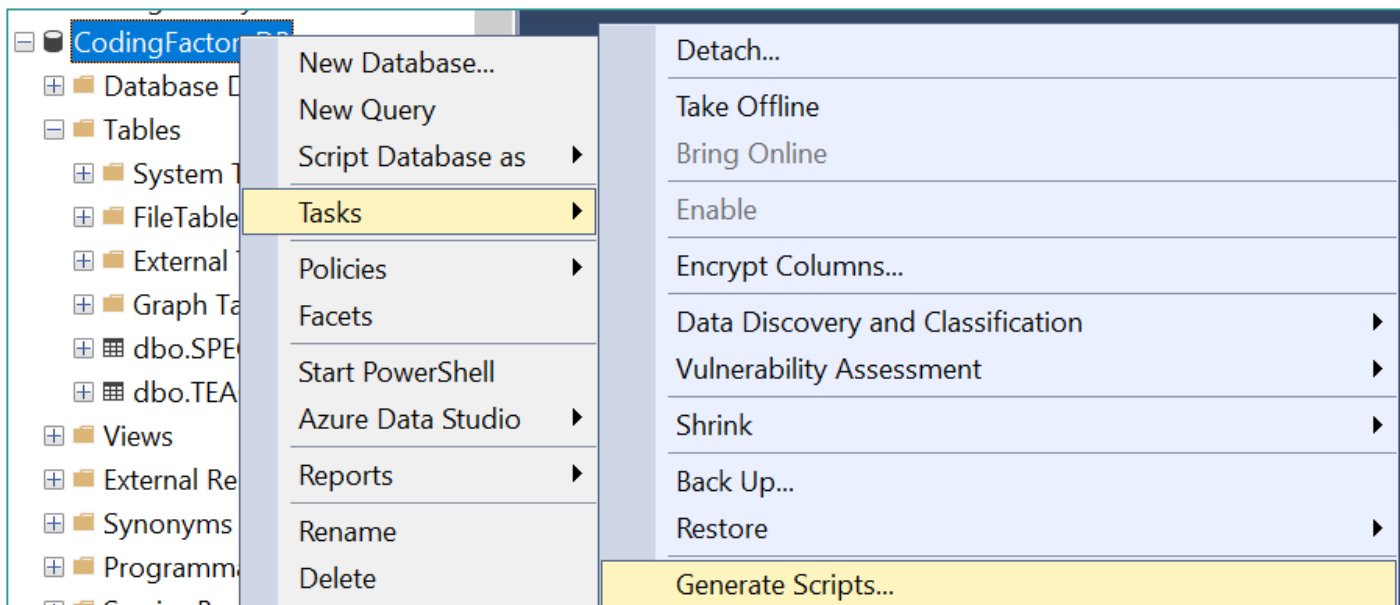
- Περισσότερες πληροφορίες σχετικά με τα flags που εμφανίζονται στο **WITH** μπορείτε να δείτε στον παρακάτω σύνδεσμο
- <https://learn.microsoft.com/en-us/sql/t-sql/statements/alter-table-index-option-transact-sql>





# SQL database dump (1)

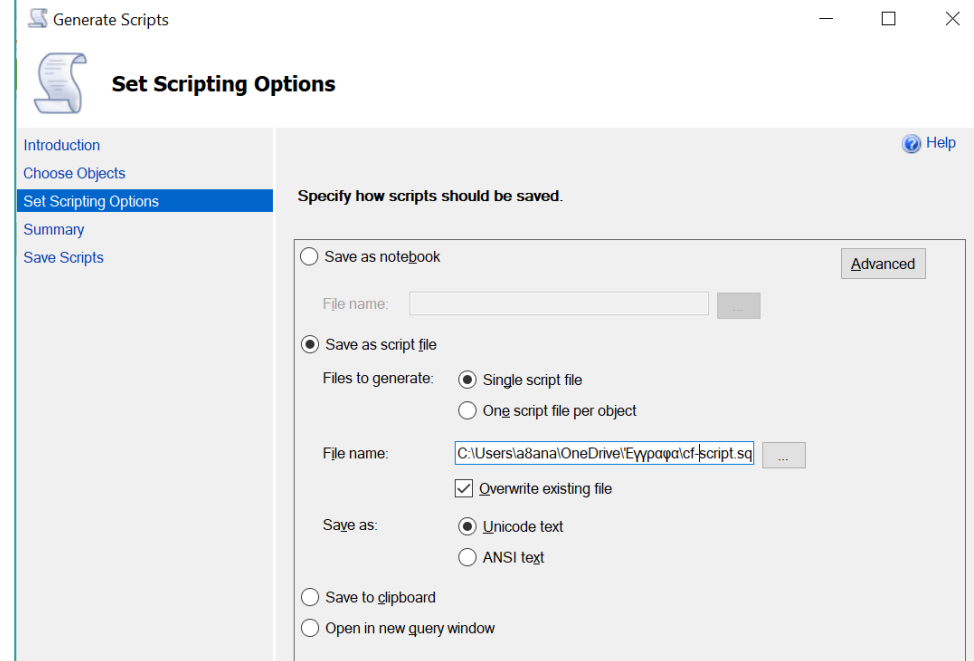
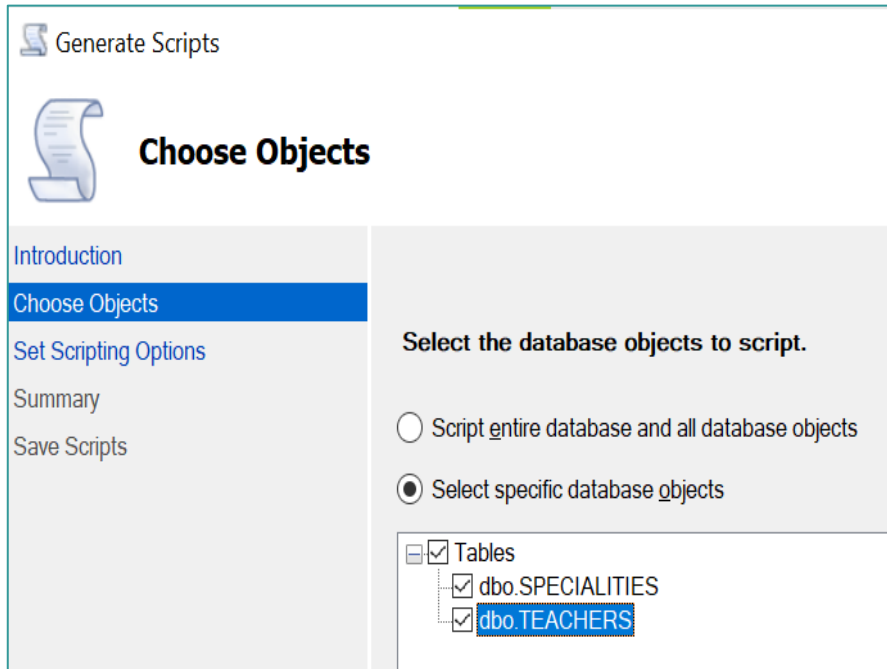
- Το SQL dump είναι ένα αρχείο (.sql) που περιέχει όλες τις SQL εντολές ώστε να μπορούμε να επαναδημιουργήσουμε μία Βάση Δεδομένων
- Δεξί κλικ στην ΒΔ και μετά Tasks / Generate Scripts...





# SQL database dump (2)

Βάσεις Δεδομένων




- Επιλέγουμε τους Πίνακες και στο επόμενο βήμα το όνομα του αρχείου (π.χ. cf-script.sql)



# SQL database dump (3)

Βάσεις Δεδομένων

Generate Scripts

 **Save Scripts**

Introduction  
Choose Objects  
Set Scripting Options  
Summary  
**Save Scripts**

Help

**Save Scripts**

Action	Result
✓ Getting the list of objects from 'CodingFactoryDB'.	Success
✓ Preparing dbo.SPECIALITIES	Success
✓ Preparing dbo.TEACHERS	Success
✓ Save to file	Success

Open Save Report

< Previous Next > **Finish** Cancel

- Επιτυχές SQL Dump



# SQL database dump (4)

Βάσεις Δεδομένων

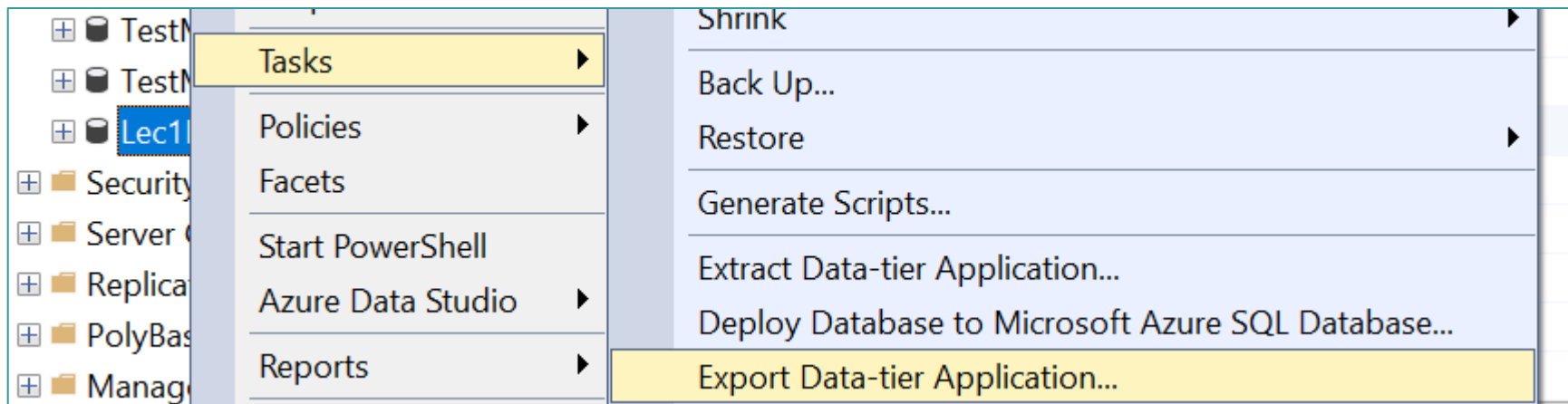
OneDrive > Έγγραφα >			cf
<input type="checkbox"/> Όνομα	Κατάσταση	Ημερομηνία τροποποίησης	Τύπος
<input checked="" type="checkbox"/> cf-script.sql	✓	15/10/2023 11:02 μμ	Microsoft SQL Server...

```
cf-script.sql - THA...SSIS-PC\αθانا (56)
USE [CodingFactoryDB]
GO
/***** Object: Table [dbo].[SPECIALITIES]    Script Date: 15/10/2023 11:02:44 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SPECIALITIES](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [TITLE] [nvarchar](50) NULL,
    CONSTRAINT [PK_SPECIALITIES] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_L
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[TEACHERS]    Script Date: 15/10/2023 11:02:44 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TEACHERS](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [FIRSTNAME] [nvarchar](50) NULL,
    [LASTNAME] [nvarchar](50) NULL,
    [PHONE_NUMBER] [nvarchar](50) NULL,
    [ADDRESS] [nvarchar](50) NULL,
    [SPECIALITY ID] [int] NOT NULL,
```

- Το .sql file created
- Ανοίγουμε το .sql file στο SSMS



# Export Data-tier (1)



- Επίσης, αν θέλουμε, μπορούμε να πάρουμε back-up (export) της Βάσης.
- Δηλαδή όλη τη δομή και τα δεδομένα, ώστε σε περίπτωση που χρειαστεί να μπορούμε να κάνουμε restore



# Export Data-tier (2)

Βάσεις Δεδομένων

Export Data-tier Application 'Lec1DB'

**Export Settings**

Introduction  
Export Settings  
Summary  
Results

Help

**Export Settings**

This operation will create a BACPAC file that contains the logical contents of your database. To continue, specify the location where you want the BACPAC file to be created, and then click Next. To specify a subset of tables to export, use the Advanced option.

Settings Advanced

☒ Save to local disk

C:\Users\A8ana\OneDrive\Databases\Lec1DB.bacpac Browse...

☐ Save to Microsoft Azure

Storage account: Connect...

Container: ▾

File name: Lec1DB.bacpac

Temporary file name: C:\Users\A8ana\AppData\Local\Temp\Lec1DB-20231101182205.bacpac Browse...

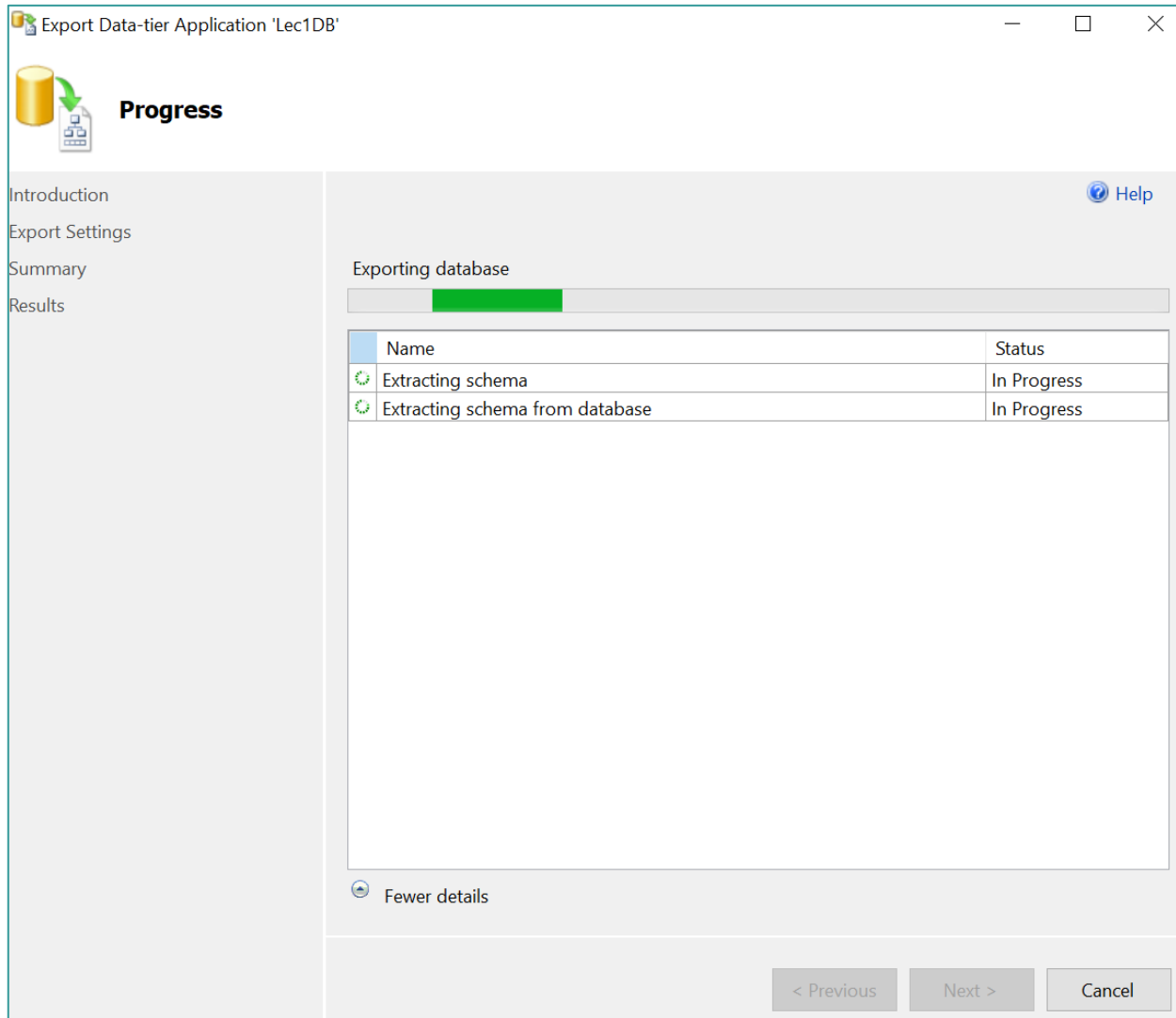
< Previous Next > Cancel

- Αποθηκεύουμε σε ένα αρχείο με κατάληξη .bacpac



# Export Data-tier (3)

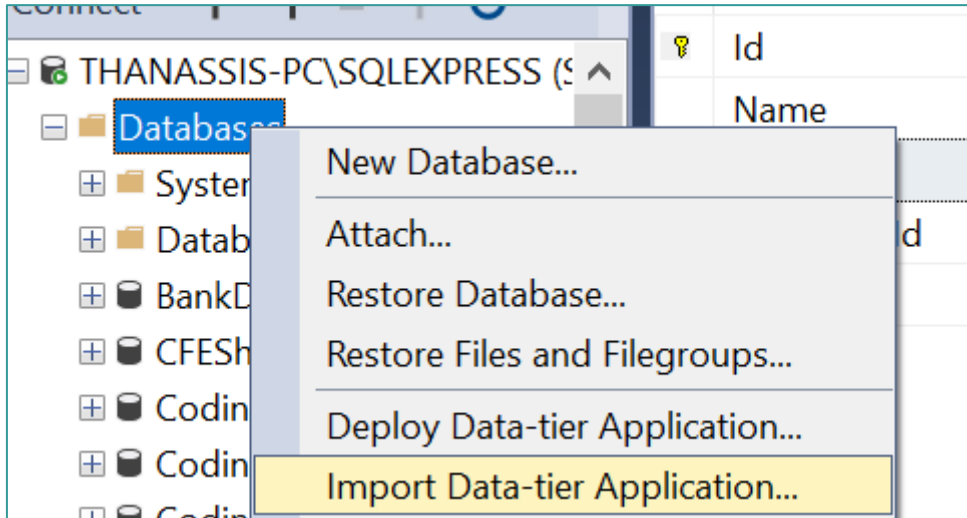
Βάσεις Δεδομένων



- Τελικά γίνεται το export στο αρχείο που ορίσαμε



# Restore DB (1)



- Επιλέγουμε με δεξί κλικ στο Databases το Import Data-tier Application





# Restore DB (2)

Import Data-tier Application

**Import Settings**

Introduction  
**Import Settings**  
Database Settings  
Summary  
Results

[Help](#)

**Specify the BACPAC to import.**  
This operation will create a database from a BACPAC file. To continue, specify the location of the BACPAC. Optionally, specify settings for the new database. Click Next to continue.

☒ Import from local disk

C:\Users\A8ana\Downloads\Lec1DB.bacpac [Browse...](#)

☐ Import from Windows Azure

Storage account:  [Connect...](#)

Container:  [Browse...](#)

File name:  [Browse...](#)

Temporary file name: C:\Users\A8ana\AppData\Local\Temp\temp\_bacpac-20231101184357.bacpac [Browse...](#)

< Previous   Next >   Cancel

- Ως import επιλέγουμε το αρχείο που είχαμε κάνει export



# Restore DB (3)

**Database Settings**

roduction  
port Settings  
**Database Settings**  
mmmary  
sults

Help

**Specify settings for the new database.**

This operation will create a database from a BACPAC file. To continue, specify the settings for the new database and click Next.

THANASSIS-PC\SQLEXPRESS (THANASSIS-PC\sa8ana) Connect...

New database name:  
Lec1DBRestored

SQL Server Settings

Data file path:  
Browse... C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSD

Log file path:  
Browse... C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSD

< Previous Next > Cancel


- Δίνουμε ένα (νέο) όνομα στη ΒΔ και προχωράμε στο Restore



# Restore DB (4)

Βάσεις Δεδομένων


Import Data-tier Application




 **Progress**


Introduction  
Import Settings  
Database Settings  
Summary  
Results

Help

Importing database



Name	Status
 Creating database on target	Success
 Creating deployment plan	In Progress
 Initializing deployment	In Progress

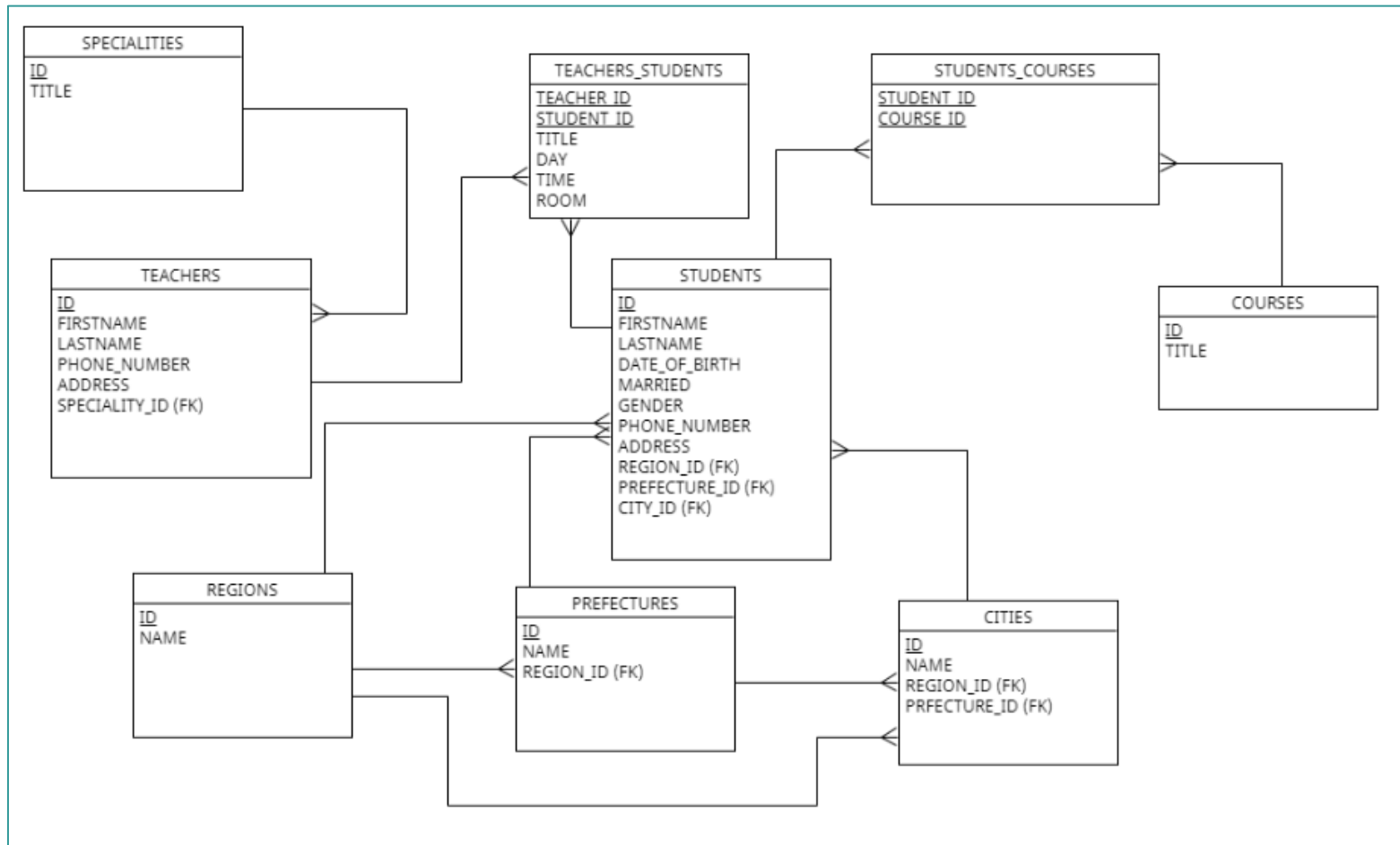
 Fewer details

< Previous   Next >   Cancel



# Εργασία

## Βάσεις Δεδομένων



- Σχεδιάστε στο draw.io και υλοποιήστε σε SQL Express το παραπάνω DB Schema