

## Ορισμοί Σχεσιακού Μοντέλου και Τροποποιήσεις Σχέσεων σε SQL

Στα προηγούμενα μαθήματα:

### Μοντελοποίηση

- Εννοιολογικός Σχεδιασμός Βάσεων Δεδομένων (με χρήση του Μοντέλου Οντοτήτων/Συσχετίσεων)
- Λογικός Σχεδιασμός Βάσεων Δεδομένων (με χρήση του Σχεσιακού Μοντέλου)
- Μετατροπή ανάμεσα στα μοντέλα

Θα δούμε κάποιες βασικές εντολές για τον ορισμό σχήματος και τη δημιουργία/τροποποίηση στιγμιότυπου

Πως θα υλοποιήσουμε (προγραμματίσουμε) την εφαρμογή μας χρησιμοποιώντας ένα σχεσιακό ΣΔΒΔ:

- Γλώσσα Ορισμού Δεδομένων (ΔΟΧ) (του σχήματος) <Data Definition Language (DDL)>
- Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ) <Data Manipulation Language (DML)>
  - Γλώσσα Τροποποίησης (εισαγωγή, διαγραφή πλειάδων)
  - Γλώσσα Ερωτήσεων (Επερωτήσεις)

Οι γλώσσες ερωτήσεων απαντούν σε **ερωτήσεις** για τον παρόν στιγμιότυπο της βάσης δεδομένων (*querying*) - ανάκτηση δεδομένων (θα τις δούμε αναλυτικά σε επόμενα μαθήματα)

### Βήματα Δημιουργίας και Χρήσης μιας (Σχεσιακής) Βάσης Δεδομένων

Σχεδιασμός Σχήματος

Δημιουργία Σχήματος χρησιμοποιώντας τη ΓΟΔ (DDL)

Μαζική Φόρτωση των αρχικών δεδομένων  
⇒ Η βάση δεδομένων έχει δεδομένα

Repeat: εκτέλεση ερωτήσεων (*select-from-where*) και τροποποιήσεων (*insert-delete-update*) στη βάση δεδομένων

Η SQL είναι η γλώσσα για όλα τα εμπορικά σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων αρχικά Sequel στην IBM ως μέρος του System R, τώρα SQL (Structured Query Language)

SQL-89, SQL-92, SQL-99 +...

Η SQL έχει διάφορα τμήματα:

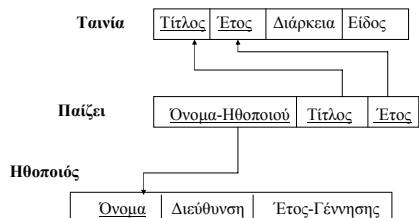
- Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)
- Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)
- Ενσωματωμένη Γλώσσα Χειρισμού Δεδομένων
- Ορισμό Όψεων
- Εξουσιοδότηση (authentication)
- Ακεραιότητα
- Έλεγχο Συναλλαγών

Συγκεκριμένα:

**DDL** (Data Definition Language) - ορισμός, δημιουργία, τροποποίηση και διαγραφή *σχήματος*.

**DML** (Data Manipulation Language) - ορισμός, δημιουργία, τροποποίηση, διαγραφή και *επιλογή δεδομένων* (γλώσσα ερωτήσεων).

**Προδιαγραφές ασφάλειας** - χρήστες και δικαιώματα.



Μερικές Γενικές Παρατηρήσεις

Oracle SQL και η MySQL μερικές φορές δεν ακολουθούν ακριβώς τα standards - μερικές εντολές στις διαφάνειες μπορεί να μη «τρέχουν»

Κάποιες αποκλίσεις περιγράφονται στη web σελίδα του μαθήματος

Επίσης, "interactive" SQL - εντολές που πληκτρολογούνται μετά από το prompt και οι απαντήσεις εμφανίζονται στην οθόνη ως πίνακες

"Embedded" και "dynamic" SQL: θα τη δούμε στην (επόμενη) προγραμματιστική άσκηση

Ορισμός Σχήματος

Για κάθε σχεσιακό σχήμα μια γλώσσα ορισμού δεδομένων

1. Ορισμός σχήματος (όνομα στη σχεσιακή βάση δεδομένων)
2. Ορισμός των (σχημάτων) σχέσεων που αποτελούν τη βάση

Όνομα σχέσης, ονόματα και πεδία ορισμού των γνωρισμάτων, περιορισμοί ορθότητας

3. Ορισμοί πεδίων ορισμού

Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)

Σχετικά με το λογικό σχήμα, η ΓΟΔ SQL υποστηρίζει τους ορισμούς:

- του σχήματος κάθε σχέσης
- του πεδίου τιμών κάθε γνωρίσματος
- των περιορισμών ακεραιότητας

Γενική Δομή Ορισμού

```
create table R(A1 D1, A2 D2, ..., An Dn),
<περιορισμός-ακεραιότητας1>,
...,
<περιορισμός-ακεραιότηταςk>
```

όπου R είναι το όνομα της σχέσης, A<sub>i</sub> τα ονόματα των γνωρισμάτων, και D<sub>i</sub> οι τύποι των αντίστοιχων πεδίων τιμών.

Τύποι Πεδίου Ορισμού

Για τον ορισμό του πεδίου ορισμού, οι διαθέσιμοι built-in τύποι περιλαμβάνουν - περισσότερα στο βιβλίο και στη σελίδα του μαθήματος:

- char(n) (σταθερού μήκους)
- varchar(n)
- int
- smallint
- numeric(p, d) (d από τα p ψηφία είναι στα δεξιά της υποδιαστολής)
- real, double precision
- float(n)
- date (ημερομηνία)
- time (ώρα)

## Πεδία Τιμών

Ο ορισμός πεδίου μπορεί να περιέχει τον προσδιορισμό **not null** και **default** τιμή

## Ορισμός Σχήματος

### Παράδειγμα

```
CREATE TABLE Ταβία
(Τίτλος varchar(20) not null,
 Έτος int not null,
 Διάρκεια int,
 Είδος varchar(20),
 primary key (Τίτλος, Έτος));
```

```
CREATE TABLE Ηθοποιός
(Όνομα varchar(20) not null,
 Διεύθυνση varchar(15),
 Έτος-Γέννησης int,
 primary key (Όνομα),
 check (Έτος-Γέννησης <= 1800));
```

```
CREATE TABLE Παιζει
(Όνομα varchar(20) not null,
 Τίτλος varchar(20) not null,
 Έτος int not null,
 primary key (Όνομα, Τίτλος, Έτος),
 foreign key (Όνομα) references Ηθοποιός(Όνομα),
 foreign key (Τίτλος, Έτος) references Ταβία(Τίτλος, Έτος));
```

Ορισμός σχήματος σχέσης  
Όνομα σχέσης + γνωρίσματα

## Ορισμός Σχήματος: Περιορισμοί Ακεραιότητας

Επιτρεπτοί περιορισμοί ακεραιότητας είναι της μορφής:

- **primary key** ( $A_{j1}, A_{j2}, \dots, A_{jn}$ ), (δεν επιτρέπονται επαναλαμβανόμενες τιμές και NULL τιμές)  
*για τον ορισμό του πρωτεύοντος κλειδιού*
- **unique** ( $A_{j1}, A_{j2}, \dots, A_{jn}$ ), (δεν επιτρέπονται επαναλαμβανόμενες τιμές; NULL τιμές επιτρέπονται (μόνο μία))  
*για τον ορισμό υποψηφίων κλειδιών*
- **check P**  
*για τον ορισμό σημασιολογικών περιορισμών*
- **foreign key ( $A_i$ ) references  $A_j$**   
*για τον ορισμό ξένου κλειδιού*

## Ορισμός Σχήματος

### Παράδειγμα

```
CREATE TABLE Ταβία
(Τίτλος varchar(20) not null,
 Έτος int not null,
 Διάρκεια int,
 Είδος varchar(20),
 primary key (Τίτλος, Έτος));
```

```
CREATE TABLE Ηθοποιός
(Όνομα varchar(20) not null,
 Διεύθυνση varchar(15),
 Έτος-Γέννησης int,
 primary key (Όνομα),
 check (Έτος-Γέννησης <= 1800));
```

```
CREATE TABLE Παιζει
(Όνομα varchar(20) not null,
 Τίτλος varchar(20) not null,
 Έτος int not null,
 primary key (Όνομα, Τίτλος, Έτος),
 foreign key (Όνομα) references Ηθοποιός(Όνομα),
 foreign key (Τίτλος, Έτος) references Ταβία(Τίτλος, Έτος));
```

Ορισμός πρωτεύοντος κλειδιού

## Ορισμός Σχήματος

### Παράδειγμα

```
CREATE TABLE Ταβία
(Τίτλος varchar(20) not null,
 Έτος int not null,
 Διάρκεια int,
 Είδος varchar(20),
 primary key (Τίτλος, Έτος));
```

```
CREATE TABLE Ηθοποιός
(Όνομα varchar(20) not null,
 Διεύθυνση varchar(15),
 Έτος-Γέννησης int,
 primary key (Όνομα),
 check (Έτος-Γέννησης <= 1800));
```

```
CREATE TABLE Παιζει
(Όνομα varchar(20) not null,
 Τίτλος varchar(20) not null,
 Έτος int not null,
 primary key (Όνομα, Τίτλος, Έτος),
 foreign key (Όνομα) references Ηθοποιός(Όνομα),
 foreign key (Τίτλος, Έτος) references Ταβία(Τίτλος, Έτος));
```

Ορισμός ξένου κλειδιού  
Προφανώς, ο ορισμός του πίνακα στον οποίο αναφέρεται, πρέπει να προηγείται

## Ορισμός Σχήματος

### Παράδειγμα

```
CREATE TABLE Ταβία
(Τίτλος varchar(20) not null,
 Έτος int not null,
 Διάρκεια int,
 Είδος varchar(20),
 primary key (Τίτλος, Έτος));
```

```
CREATE TABLE Ηθοποιός
(Όνομα varchar(20) not null,
 Διεύθυνση varchar(15),
 Έτος-Γέννησης int,
 primary key (Όνομα),
 check (Έτος-Γέννησης <= 1800));
```

```
CREATE TABLE Παιζει
(Όνομα varchar(20) not null,
 Τίτλος varchar(20) not null,
 Έτος int not null,
 primary key (Όνομα, Τίτλος, Έτος),
 foreign key (Όνομα) references Ηθοποιός(Όνομα),
 foreign key (Τίτλος, Έτος) references Ταβία(Τίτλος, Έτος));
```

Απλό παράδειγμα  
σημασιολογικού περιορισμού

Οι περιορισμοί ορίζονται μια φορά στο σχήμα και ελέγχονται κάθε φορά που γίνεται μια τροποποίηση του στιγμιότυπου



## Τροποποίηση Σχήματος

**ALTER TABLE** όνομα πίνακα

- **ADD** - προσθέτει καινούργια στήλη
- **DROP** - διαγράφει μια στήλη
- **MODIFY** - τροποποιεί μια στήλη



Προσθήκη νέου γνωρίσματος:

**alter table R add A D**

προσθήκη σε μια σχέση R που ήδη υπάρχει του γνωρίσματος A με πεδίο τιμών D, η τιμή των πλειάδων της R στο καινούργιο γνώρισμα είναι null.

Διαγραφή γνωρίσματος:

**alter table R drop A**



**alter table R modify** (όνομα\_στήλης new\_datatype)

**modify** μπορεί να τροποποιήσει μόνο τον τύπο δεδομένων, όχι το όνομα της στήλης



## Διαγραφή Σχήματος

Μια καινούργια σχέση είναι αρχικά άδεια.

Για να σβηστεί ένα σχήμα:

**drop table R**



Τροποποίηση Βάσης Δεδομένων: Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)

## Τροποποιήσεις

1. Εισαγωγή πλειάδας
2. Διαγραφή Πλειάδας
3. Τροποποίηση (Ενημέρωση) Πλειάδας

Οι εντολές αυτές ΤΡΟΠΟΠΟΙΟΥΝ το στιγμιότυπο της βάσης δεδομένων (δηλαδή, το περιεχόμενο των πινάκων)



**Εισαγωγή:** Παρέχει μια λίστα από τιμές γνωρισμάτων για μια νέα πλειάδα που πρέπει να εισαχθεί στη σχέση



Εισαγωγή πλειάδας σε SQL

Για να εισάγουμε δεδομένα σε μια σχέση είτε

(α) προσδιορίζουμε την πλειάδα,

```
insert into R(A1, ..., An) values (v1, ..., vn)
```

*σε το λίστα τιμών*

είτε

(β) γράφουμε μια ερώτηση που το αποτέλεσμα της εισάγεται στη σχέση.

```
insert into R(A1, ..., An) select-from-where
```



Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)
Παιζεί(Όνομα, Τίτλος, Έτος)
Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης)

Παράδειγμα

```
insert into Ταινία
values ('The Big Blue', 1988, 132, 'Έγχρωμη')
```

Όταν με οποιαδήποτε σειρά, π.χ.:

```
insert into Ταινία (Τίτλος, Είδος, Διάρκεια, Έτος)
values ('The Big Blue', 'Έγχρωμη', 132, 1988)
```



Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)
Παιζεί(Όνομα, Τίτλος, Έτος)
Ηθοποιός(Όνομα, Διεύθυνση, Έτος-Γέννησης)

Επίσης, εισαγωγή null τιμών:

```
insert into Ταινία
values ('The Big Blue', 1988, null, 'Έγχρωμη')
```

ή αν *δε δίνω τιμές για όλα* τα γνωρίσματα

```
insert into Ταινία (Τίτλος, Έτος, Είδος)
values ('The Big Blue', 1988, 'Έγχρωμη')
```



Ποιους από τους περιορισμούς (πεδίου ορισμού, κλειδιού, ακεραιότητας οντοτήτων και αναφορικής ακεραιότητας) μπορεί να παραβιάζει μια τέτοια λίστα τιμών;

Σε περίπτωση παραβίασης:

Απόρριψη εισαγωγής



**Διαγραφή:** Προσδιορίζεται μια συνθήκη πάνω στα γνωρίσματα της σχέσης και διαγράφονται οι πλειάδες που την ικανοποιούν



Διαγραφή σε SQL

Μπορούμε να σρήσουμε μόνο *ολόκληρες* πλειάδες και όχι συγκεκριμένα γνωρίσματα.

```
delete from R where P
```

Σβήνει όλες τις πλειάδες της R για τις οποίες ισχύει το P.

Όταν λείπει το **where** σβήνονται όλες οι πλειάδες μιας σχέσης.



Παραδείγματα

(1) Όλες οι ηθοποιοί με το όνομα Kidman

**delete from** Ηθοποιοί  
**where** Όνομα = 'Kidman'

(2) Όλες τις ταινίες που έχουν γυριστεί πριν το 1950

**delete from** Ταινία  
**where** Έτος < 1950



Συνθήκη του where

<Όνομα\_Γνωρίσματος> <τελεστής> <Όνομα\_Γνωρίσματος> ή <Τιμή>

Τελεστές σύγκρισης: <, <=, >, >=, =, <>, κλπ

Λογικοί τελεστές: **and, or, not**

- Στο **from** μόνο μια σχέση, *αλλά στη συνθήκη του where μπορεί να εμφανίζονται και άλλες (θα το δούμε στη συνέχεια)*
- Σβήνονται «ολόκληρες» πλειάδες
- Αν υπάρχουν παραπάνω από μια πλειάδες που ικανοποιούν τη συνθήκη, δεν υπάρχει τρόπος να διακρίνουμε τις πλειάδες, δηλαδή να σβήσουμε κάποιες
- Πρώτα, υπολογίζεται η συνθήκη του **where** και μετά διαγράφονται οι πλειάδες που ικανοποιούν τη συνθήκη



Ποιους από τους περιορισμούς (πεδίου ορισμού, κλειδιού, ακεραιότητας οντοτήτων και αναφορικής ακεραιότητας) μπορεί να παραβιάζει το αποτέλεσμα μια διαγραφής;



Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)
Παιζεί(Όνομα, Τίτλος, Έτος)
Ηθοποιοί(Όνομα, Διεύθυνση, Έτος-Γέννησης)

Παράδειγμα: διαγραφή της ταινίας "The Big Blue" που γυρίστηκε το 1988

**delete from** Ταινία

**where** Τίτλος = 'The Big Blue' and Έτος = 1988



Σε περίπτωση παραβίασης (αναφορικής ακεραιότητας), έχουμε τις επιλογές:

- απόρριψη της διαγραφής
- διάδοση της διαγραφής (αυτόματη διαγραφή όλων των πλειάδων που αναφέρονται σε αυτήν)
- τροποποίηση των τιμών των αναφορικών γνωρισμάτων Πως:  
    - μια ειδική τιμή ή  
    - την τιμή NULL (αν επιτρέπεται)

Η SQL μας επιτρέπει να προσδιορίσουμε ποια από τις παραπάνω επιλογές θα πραγματοποιείται σε περίπτωση παραβίασης

Πότε: όταν ορίζουμε στο σχήμα τους περιορισμούς ξένου κλειδιού



Περισσότερα για τους Περιορισμούς Αναφοράς

Σύνταξη:

Μετά τον ορισμό του:

**foreign key (A<sub>i</sub>) references A<sub>j</sub>**

**cascade, set null, set default (no action)** είναι το ίδιο με το να μην προσδιορίσουμε τίποτα

**on delete**

Όταν μια πράξη παραβιάζει έναν περιορισμό αναφοράς απορρίπτεται εκτός αν έχει οριστεί κάτι άλλο

## Διαγραφή Πλειάδας

Σε περίπτωση παραβίασης (αναφορικής ακεραιότητας):

- απόρριψη της διαγραφής (αν δεν υπάρχει προσδιορισμός) ή **on delete no action**
- διάδοση της διαγραφής (αυτόματη διαγραφή όλων των πλειάδων που αναφέρονται σε αυτήν) **on delete cascade**
- τροποποίηση των τιμών των αναφορικών γνωρισμάτων Πως:  
    • μια ειδική τιμή **on delete set default** ή  
    • την τιμή NULL **on delete set null**

## Ορισμός Σχήματος

```
CREATE TABLE Ταβία
(Τίτλος varchar(20) not null,
Έτος int not null,
Διάρκεια int,
Είδος varchar(20),
primary key (Τίτλος, Έτος));

CREATE TABLE Ηθοποιοί
(Όνομα varchar(20) not null,
Διεύθυνση varchar(15),
Έτος-Γέννησης int,
primary key (Όνομα),
check (Έτος-Γέννησης >= 1800));

CREATE TABLE Παιζει
(Όνομα varchar(20) not null,
Τίτλος varchar(20) not null,
Έτος int not null,
primary key (Όνομα, Τίτλος, Έτος),
foreign key (Όνομα) references Ηθοποιοί(Όνομα),
on delete set NULL,
foreign key (Τίτλος, Έτος) references Ταβία(Τίτλος, Έτος),
on delete cascade);

CREATE TABLE Παιζει
(Όνομα varchar(20) not null,
Τίτλος varchar(20) not null,
Έτος int not null,
primary key (Όνομα, Τίτλος, Έτος),
foreign key (Όνομα) references Ηθοποιοί(Όνομα),
on delete cascade,
foreign key (Τίτλος, Έτος) references Ταβία(Τίτλος, Έτος));
```

## Διαγραφή Πλειάδας

```
Ταβία (Τίτλος, Έτος, Διάρκεια, Είδος)
Παιζει(Όνομα, Τίτλος, Έτος)
Ηθοποιοί(Όνομα, Διεύθυνση, Έτος-Γέννησης)
```

*Παράδειγμα: διαγραφή της ταινίας "The Big Blue" που γυρίστηκε το 1988*

```
delete from Ταβία
where Τίτλος = 'The Big Blue' and Έτος = 1988
```

*Το αποτέλεσμα εξαρτάται από το είδος περιορισμού αναφοράς που έχουμε ορίσει*

## Διαγραφή Σχήματος και Πλειάδων

### Διαγραφή Σχήματος

Μια καινούργια σχέση είναι αρχικά άδεια.

Για να σβηστεί ένα σχήμα:

```
drop table R
```

Διαφορά από

```
delete from R
```

## Τροποποίηση Πλειάδας

**Τροποποίηση:** Προσδιορίζεται μια συνθήκη πάνω στα γνωρίσματα της σχέσης και τροποποιούνται οι πλειάδες που την ικανοποιούν

## Τροποποίηση Πλειάδας

### Τροποποίηση Πλειάδας σε SQL

```
update R
set Attr = New_Value
where P
```

Παράδειγμα: Αύξηση τις διάρκειας κάθε ταινίας κατά 10 λεπτά για όλες τις ταινίες με διάρκεια < 100

```
update Ταβία
set Διάρκεια = Διάρκεια + 10
where Διάρκεια < 100
```

## Τροποποίηση Πλειιάδας



Όπως και για τη διαγραφή:

- Στο **update** μόνο μια σχέση, αλλά στη συνθήκη του **where** μπορεί να εμφανίζονται και άλλες
- Αν υπάρχουν παραπάνω από μια πλειιάδες που ικανοποιούν τη συνθήκη, δεν υπάρχει τρόπος να διακρίνουμε τις πλειιάδες, δηλαδή να ενημερώσουμε κάποιες
- Πρώτα, υπολογίζεται η συνθήκη του **where** και μετά τροποποιούνται οι πλειιάδες που ικανοποιούν τη συνθήκη - δηλαδή, η συνθήκη υπολογίζεται στο τρέχων στιγμιότυπο - όχι στο τροποποιημένο

## Τροποποίηση Πλειιάδας



Ποιους από τους περιορισμούς (πεδίου ορισμού, κλειδιού, ακεραιότητας οντοτήτων και αναφορικής ακεραιότητας) μπορεί να παραβιάζει το αποτέλεσμα μιας τροποποίησης:

Όταν το γνώρισμα που τροποποιείται είναι ξένο κλειδί ή κλειδί;

## Ορισμός Σχήματος: Περιορισμοί Ακεραιότητας



Όπως και στη διαγραφή, κατά τον ορισμό του σχήματος ορίζουμε την κατάλληλη πράξη

**cascade, set null, set default** (no action είναι το ίδιο με το να μην προσδιορίσουμε τίποτα)

**on update**

## Ορισμός Σχήματος



```
CREATE TABLE Ταγία
(Τίτλος varchar(20) not null,
 Έτος int not null,
 Διάκριση int,
 Είδος varchar(20),
 primary key (Τίτλος, Έτος));
CREATE TABLE Ηθοποιός
(Όνομα varchar(20) not null,
 Διεύθυνση varchar(15),
 Έτος-Γέννησης int,
 primary key (Όνομα),
 check (Έτος-Γέννησης >= 1800));
CREATE TABLE Παιζει
(Όνομα varchar(20) not null,
 Τίτλος varchar(20) not null,
 Έτος int not null,
 primary key (Όνομα, Τίτλος, Έτος),
 foreign key (Όνομα) references Ηθοποιός(Όνομα),
 on delete cascade,
 on update set NULL,
 foreign key (Τίτλος, Έτος) references Ταγία(Τίτλος, Έτος);
```

## Τροποποίηση Πλειιάδας



Σε περίπτωση παραβίασης (αναφορικής ακεραιότητας):

- απόρριψη της τροποποίησης (αν δεν υπάρχει προσδιορισμός ή **on update no action**)
- διάδοση της τροποποίησης (αυτόματη τροποποίηση όλων των πλειιάδων που αναφέρονται σε αυτήν) **on update cascade**
- τροποποίηση των τιμών των αναφορικών γνωρισμάτων Πως:  
μια ειδική τιμή **on update set default** ή  
την τιμή NULL **on update set null**

## Γλώσσα Χειρισμού Δεδομένων (επανάληψη)



### 1. Εισαγωγές

```
insert into R(A1, ..., An) values (v1, ..., vn)
```

### 2. Διαγραφές

```
delete from R where P
```

### 3. Ενημερώσεις/Τροποποιήσεις

```
update R
set Attr = New_Value
where P
```