

# Ορισμοί Σχεσιακού Μοντέλου και (απλές) Τροποποιήσεις Σχέσεων στην SQL

# Τι έχουμε δει

## Μοντελοποίηση

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

# Τι θα δούμε σήμερα

## Βασικές εντολές

- Για τον ορισμό και τροποποίηση σχήματος
- Για τη δημιουργία και τροποποίηση στιγμιότυπου (εισαγωγή, διαγραφή, ενημέρωση δεδομένων)

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

# Εισαγωγή

**Γλώσσα Ορισμού Δεδομένων (ΔΟΧ)** (του σχήματος) (**Data Definition Language (DDL)**) - ορισμός, δημιουργία, τροποποίηση και διαγραφή σχήματος.

**Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)** (**Data Manipulation Language (DML)**)

- **Γλώσσα Τροποποίησης** (εισαγωγή, διαγραφή, τροποποίηση πλειάδων)
- **Γλώσσα Ερωτήσεων (Επερωτήσεων) Query Languages:** **διατυπώνουν ερωτήσεις** στο τρέχων στιγμιότυπο της βάσης δεδομένων για την ανάκτηση/επιλογή δεδομένων (θα τις δούμε αναλυτικά σε επόμενα μαθήματα)

# Η γλώσσα SQL

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

# Βήματα Δημιουργίας και Χρήσης μιας ΒΔ

Σχεδιασμός Σχήματος

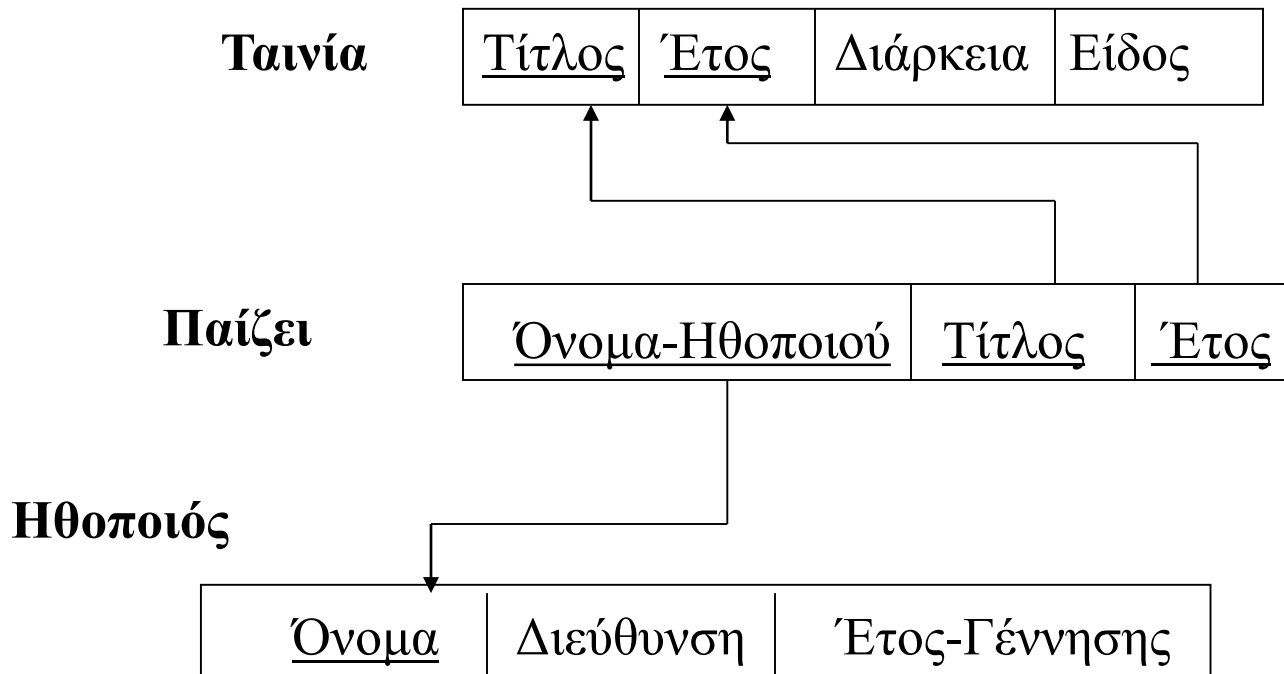
Δημιουργία Σχήματος χρησιμοποιώντας τη ΓΟΔ (DDL)

Μαζική Φόρτωση των αρχικών δεδομένων

⇒ Η βάση δεδομένων έχει δεδομένα

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

# Παράδειγμα



# Ορισμός σχήματος

Με χρήση μιας γλώσσας ορισμού δεδομένων προσδιορίζεται

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*

# Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)

```
CREATE DATABASE <database-name>;
```

```
USE DATABASE <database-name>;
```

# Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)

## 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.

# Πεδίο Ορισμού

## 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)

# Πεδίο Ορισμού

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

# Πεδίο Ορισμού

## Παράδειγμα

### CREATE TABLE Ταινία

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

Ορισμός σχήματος σχέσης

Όνομα σχέσης + γνωρίσματα

### CREATE TABLE Ηθοποιός

```
(Όνομα varchar(20),  
Διεύθυνση varchar(15),  
Έτος-Γέννησης int,  
primary key (Όνομα),  
check (Έτος-Γέννησης >= 1800));
```

### CREATE TABLE Παίζει

```
(Όνομα varchar(20),  
Τίτλος varchar(20),  
Έτος int,  
primary key (Όνομα, Τίτλος, Έτος),  
foreign key (Όνομα) references Ηθοποιός(Όνομα),  
foreign key (Τίτλος, Έτος) references Ταινία(Τίτλος, Έτος);
```

# Πεδίο Ορισμού: περιορισμοί ακεραιότητας

Επιτρεπτοί περιορισμοί ακεραιότητας είναι της μορφής:

- **PRIMARY KEY**( $A_{j_1}, A_{j_2}, \dots, A_{j_n}$ ), (δεν επιτρέπονται επαναλαμβανόμενες τιμές και NULL τιμές)
  - για τον ορισμό του πρωτεύοντος κλειδιού
- **UNIQUE**( $A_{j_1}, A_{j_2}, \dots, A_{j_n}$ ), (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

# Πεδίο Ορισμού

## Παράδειγμα

### CREATE TABLE Ταινία

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

### CREATE TABLE Ηθοποιός

```
(Όνομα varchar(20),  
Διεύθυνση varchar(15),  
Έτος-Γέννησης int,  
primary key (Όνομα),  
check (Έτος-Γέννησης >= 1800));
```

### CREATE TABLE Παίζει

```
(Όνομα varchar(20),  
Τίτλος varchar(20),  
Έτος int,  
primary key (Όνομα, Τίτλος, Έτος),  
foreign key (Όνομα) references Ηθοποιός(Όνομα),  
foreign key (Τίτλος, Έτος) references Ταινία(Τίτλος, Έτος);
```



# Πεδίο Ορισμού

## Παράδειγμα

### CREATE TABLE Ταινία

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

### CREATE TABLE Ηθοποιός

```
(Όνομα varchar(20),  
Διεύθυνση varchar(15),  
Έτος-Γέννησης int,  
primary key (Όνομα),  
check (Έτος-Γέννησης >= 1800));
```

### CREATE TABLE Παίζει

```
(Όνομα varchar(20),  
Τίτλος varchar(20),  
Έτος int,  
primary key (Όνομα, Τίτλος, Έτος),  
foreign key (Όνομα) references Ηθοποιός(Όνομα),  
foreign key (Τίτλος, Έτος) references Ταινία(Τίτλος, Έτος);
```

# Πεδίο Ορισμού

## Παράδειγμα

### CREATE TABLE Ταινία

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

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

### CREATE TABLE Ηθοποιός

```
(Όνομα varchar(20),  
Διεύθυνση varchar(15),  
Έτος-Γέννησης int,  
primary key (Όνομα),  
check (Έτος-Γέννησης >= 1800));
```

### CREATE TABLE Παίζει

```
(Όνομα varchar(20),  
Τίτλος varchar(20),  
Έτος int,  
primary key (Όνομα, Τίτλος, Έτος),  
foreign key (Όνομα) references Ηθοποιός(Όνομα),  
foreign key (Τίτλος, Έτος) references Ταινία(Τίτλος, Έτος);
```

# Πεδίο Ορισμού

## Παράδειγμα

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

```
CREATE TABLE Ηθοποιός  
  (Όνομα varchar(20),  
   Διεύθυνση varchar(15),  
   Έτος-Γέννησης int,  
   primary key (Όνομα),  
   check (Έτος-Γέννησης >= 1800));
```

```
CREATE TABLE Παίζει  
  (Όνομα varchar(20),  
   Τίτλος varchar(20),  
   Έτος int,  
   primary key (Όνομα, Τίτλος, Έτος),  
   foreign key (Όνομα) references Ηθοποιός(Όνομα),  
   foreign key (Τίτλος, Έτος) references Ταινία(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

# Τροποποίηση Σχήματος

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
```

# Τροποποίηση Σχήματος

**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**

# Σχήμα

**SHOW DATABASES | SHEMAS**

**SHOW TABLES**



# Ξένα κλειδιά στη MySQL

Για να ορίσετε ξένα κλειδιά θα πρέπει να ορίσετε σε μηχανή αποθήκευσης την INNODB σε κάθε εντολή δημιουργίας πίνακα,

```
CREATE TABLE R ( ... ) ENGINE=INNODB;
```

# Ξένα κλειδιά στη MySQL

Αν θέλετε να ορίσετε ξένο κλειδί το οποίο να αναφέρεται σε κάποιο γνώρισμα A μιας σχέσης R *που δεν είναι κλειδί* θα πρέπει στον ορισμό της R να ορίσετε ένα ευρετήριο στο γνώρισμα A. Αυτό γίνεται με τη χρήση της εντολής INDEX.

```
CREATE TABLE R (  
  ... ,  
  INT A,  
  ... ,  
  INDEX (A),  
  ... ,  
) ENGINE=INNODB;
```

# 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 Πλειάδας

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

# Εισαγωγή Πλειάδας

Για να εισάγουμε δεδομένα σε μια σχέση είτε

(α) προσδιορίζουμε την πλειάδα,

```
INSERT INTO R(A1, ..., An) VALUES (v1, ..., vn);
```

είτε

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

```
INSERT INTO R(A1, ..., An) select-from-where
```

Θα το δούμε  
αργότερα

# Εισαγωγή Πλειάδας

Ταινία ( <u>Τίτλος</u> , <u>Έτος</u> , Διάρκεια, Είδος)
Παίζει( <u>Όνομα</u> , <u>Τίτλος</u> , <u>Έτος</u> )
Ηθοποιός( <u>Όνομα</u> , Διεύθυνση, Έτος-Γέννησης)

Παράδειγμα

**INSERT INTO** Ταινία

**VALUES** ('The Big Blue', 1988, 132, 'Εγχρωμη');

Όταν με οποιαδήποτε σειρά, π.χ.,:

**INSERT INTO** Ταινία (Τίτλος, Είδος, Διάρκεια, Έτος)

**VALUES** ('The Big Blue', 'Εγχρωμη', 132, 1988);

# Εισαγωγή Πλειάδας

Ταινία ( <u>Τίτλος</u> , <u>Έτος</u> , Διάρκεια, Είδος)
Παίζει( <u>Όνομα</u> , <u>Τίτλος</u> , <u>Έτος</u> )
Ηθοποιός( <u>Όνομα</u> , Διεύθυνση, Έτος-Γέννησης)

Επίσης, εισαγωγή *null* τιμών:

```
INSERT INTO Ταινία  
VALUES ('The Big Blue', 1988, null, 'Εγχρωμη')
```

ή αν δε δίνω τιμές για όλα τα γνωρίσματα

```
INSERT INTO Ταινία (Τίτλος, Έτος, Είδος)  
VALUES ('The Big Blue', 1988, 'Εγχρωμη')
```

# Εισαγωγή Πλειάδας

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

Σε περίπτωση παραβίασης:

Απόρριψη εισαγωγής



# Εμφάνιση Περιοχομένου

```
SELECT * FROM R
```

# Διαγραφή Πλειάδας

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

# Διαγραφή Πλειάδας

Μπορούμε να σβήσουμε μόνο ολόκληρες πλειάδες (γραμμές) 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.

# Διαγραφή Πλειάδας

## Παραδείγματα

(1) Όλες οι ηθοποιοί με το όνομα Kidman

**DELETE FROM** Ηθοποιός  
**WHERE** Όνομα = 'Kidman';

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

**DELETE FROM** Ταινία  
**WHERE** Έτος < 1950;

# Διαγραφή Πλειάδας

## Συνθήκη του where

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

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

Λογικοί τελεστές: **and**, **or**, **not**

- Πρώτα, υπολογίζεται η συνθήκη του **where** και μετά διαγράφονται οι πλειάδες που ικανοποιούν τη συνθήκη

# Διαγραφή Πλειάδας

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

# Διαγραφή Πλειάδας

Ταινία ( <u>Τίτλος</u> , <u>Έτος</u> , Διάρκεια, Είδος)
Παίζει( <u>Όνομα</u> , <u>Τίτλος</u> , <u>Έτος</u> )
Ηθοποιός( <u>Όνομα</u> , 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;

*Ποιοι περιορισμοί ελέγχονται;*

# Διαγραφή Πλειάδας

Σε περίπτωση παραβίασης (αναφορικής ακεραιότητας - ξένου κλειδιού), έχουμε τις παρακάτω 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)



# Διαγραφή Πλειάδας

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

Πότε: όταν ορίζουμε στο σχήμα τους περιορισμούς ξένου κλειδιού

# Ορισμοί Σχήματος: περιορισμοί ακεραιότητας

Όταν μια πράξη παραβιάζει έναν περιορισμό αναφοράς απορρίπτεται εκτός αν έχει οριστεί κάποια άλλη δράση – Πως?

Μετά τον ορισμό του:

**FOREIGN KEY (A<sub>i</sub>) REFERENCES A<sub>j</sub>**

Μπορούμε να προσδιορίσουμε

**ON DELETE**

- (1) **CASCADE,**
- (2) **SET NULL,**
- (3) **SET DEFAULT,**
- (4) **NO ACTION**

# Ορισμοί Σχήματος: περιορισμοί ακεραιότητας

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

**ON DELETE SET NULL**

# Ορισμοί Σχήματος: περιορισμοί ακεραιότητας

```
CREATE TABLE Ταινία
(Τίτλος varchar(20),
Έτος int,
Διάρκεια int,
Είδος varchar(20),
primary key (Τίτλος, Έτος));

CREATE TABLE Ηθοποιός
(Όνομα varchar(20),
Διεύθυνση varchar(15),
Έτος-Γέννησης int,
primary key (Όνομα),
check (Έτος-Γέννησης >= 1800));
```

```
CREATE TABLE Παίζει
(Όνομα varchar(20),
Τίτλος varchar(20),
Έτος int,
primary key (Όνομα, Τίτλος, Έτος),
foreign key (Όνομα) references Ηθοποιός(Όνομα),
on delete cascade,
foreign key (Τίτλος, Έτος) references Ταινία(Τίτλος, Έτος),
on delete cascade
```

```
CREATE TABLE Παίζει
(Όνομα varchar(20),
Τίτλος varchar(20),
Έτος int,
primary key (Όνομα, Τίτλος, Έτος),
foreign key (Όνομα) references Ηθοποιός(Όνομα),
on delete cascade,
foreign key (Τίτλος, Έτος) references Ταινία(Τίτλος, Έτος);
```

# Διαγραφή Σχήματος και Πλειάδων

## Διαγραφή Σχήματος

Μια καινούργια σχέση είναι αρχικά άδεια.

Για να σβηστεί ένα σχήμα:

**DROP TABLE R**

Διαφορετικό από

**DELETE FROM R**

# Τροποποίηση Πλειάδας

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

# Τροποποίηση Πλειάδας

```
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 Ταινία  
SET Διάρκεια = Διάρκεια + 10  
WHERE Διάρκεια < 100;
```

# Τροποποίηση Πλειάδας

*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;



# Ορισμοί 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**

# Ορισμοί Σχήματος: περιορισμοί ακεραιότητας

Σε περίπτωση παραβίασης (αναφορικής ακεραιότητας):

- απόρριψη της τροποποίησης (αν δεν υπάρχει προσδιορισμός ή

**ON UPDATE NO ACTION**

- διάδοση της τροποποίησης (αυτόματη τροποποίηση όλων των πλειάδων 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**

# Ορισμοί Σχήματος: περιορισμοί ακεραιότητας

```
CREATE TABLE Ταινία
  (Τίτλος varchar(20),
   Έτος int,
   Διάρκεια int,
   Είδος varchar(20),
   primary key (Τίτλος, Έτος));

CREATE TABLE Ηθοποιός
  (Όνομα varchar(20),
   Διεύθυνση varchar(15),
   Έτος-Γέννησης int,
   primary key (Όνομα),
   check (Έτος-Γέννησης >= 1800));

CREATE TABLE Παίζει
  (Όνομα varchar(20) default `John Doe`,
   Τίτλος varchar(20),
   Έτος int,
   primary key (Όνομα, Τίτλος, Έτος),
   foreign key (Όνομα) references Ηθοποιός(Όνομα),
   on update cascade,
   on delete set default,
   foreign key (Τίτλος, Έτος) references Ταινία(Τίτλος, Έτος));
```

# Γλώσσα Χειρισμού Δεδομένων

## 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;
```

# Παρατηρήσεις

Η 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*

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

"Embedded" και "dynamic" SQL

# Ερωτήσεις;