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/αντιστοίχηση 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 (Stuctured 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

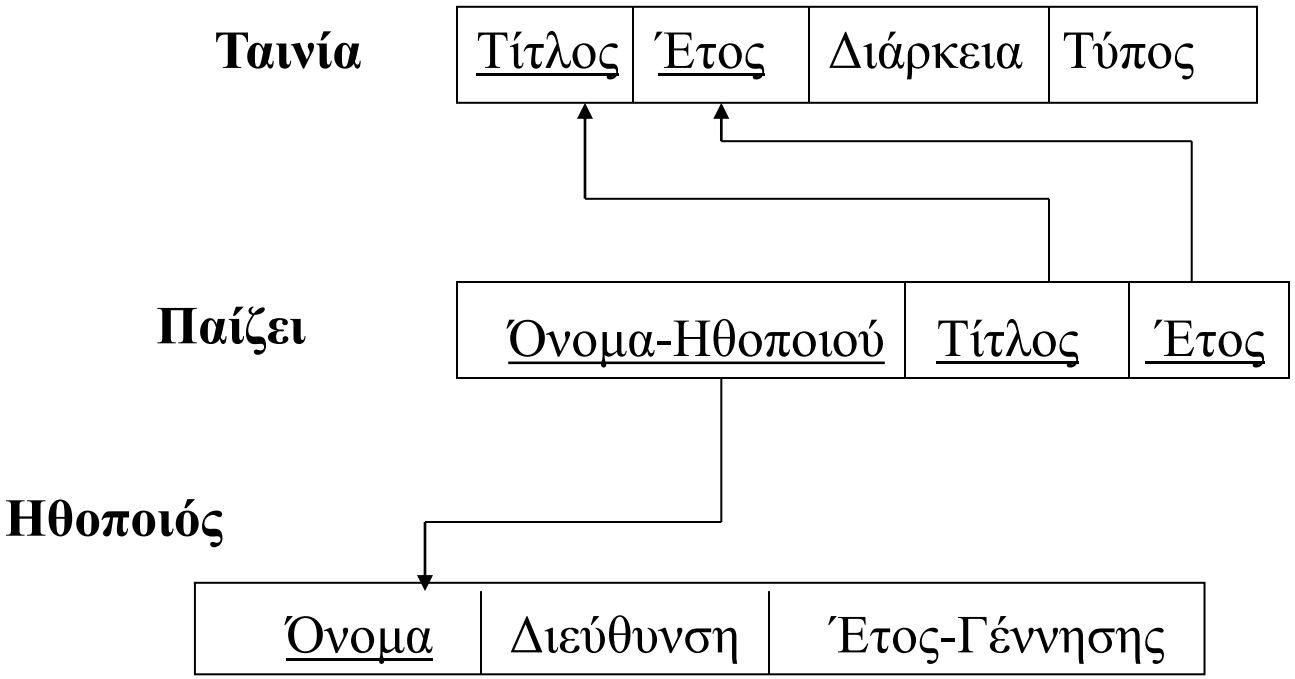
Σχεδιασμός Σχήματος

Δημιουργία Σχήματος χρησιμοποιώντας τη ΓΟΔ (DDL)

Μαζική φόρτωση των αρχικών δεδομένων
$\Rightarrow$ Η βάση δεδομένων έχει δεδομένα

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

Παράδειγμα


Ταινία
Τίτλος
Έτος
Διάρκεια
Τύπος
Παίζει
Όνομα-Ηθοποιού
Τίτλος
Έτος
Ηθοποιός
Όνομα
Διεύθυνση
Έτος-Γέννησης

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)

CREATE DATABASE <database-name>;

USE DATABASE <database-name>;

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)

Γενική Δομή Ορισμού

CREATE TABLE R(

A_1 D_1,

A_2 D_2,

…,

A_n D_n,

<περιορισμός-ακεραιότητας$_1$>,

…,

<περιορισμός-ακεραιότητας$_k$>);

όπου **R** είναι το όνομα της σχέσης, $\mathbf{A_i}$ τα ονόματα των γνωρισμάτων, και $\mathbf{D_i}$ 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,
             Τύπος 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);
```

Ορισμός σχήματος σχέσης

Όνομα σχέσης + γνωρίσματα

Πεδίο Ορισμού: περιορισμοί ακεραιότητας

Επιτρεπτοί περιορισμοί ακεραιότητας είναι της μορφής:

- **PRIMARY KEY**(A_{j1}, A_{j2}, ..., A_{jn}), (δεν επιτρέπονται επαναλαμβανόμενες τιμές και NULL τιμές)
  - *για τον ορισμό του πρωτεύοντος κλειδιού*

- **UNIQUE**(A_{j1}, A_{j2}, ..., 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*

Πεδίο Ορισμού

Παράδειγμα

```
CREATE TABLE Ταινία
            (Τίτλος varchar(20),
             Έτος int,
             Διάρκεια int,
             Τύπος 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,
             Τύπος 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);
```

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

Πεδίο Ορισμού

Παράδειγμα

```
CREATE TABLE Ταινία
            (Τίτλος varchar(20),
             Έτος int,
             Διάρκεια int,
             Τύπος 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

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(A_1, …, A_n) VALUES (v_1, …, v_n);

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

INSERT INTO R(A_1, …, A_n) select-from-where

Θα το δούμε αργότερα

Εισαγωγή Πλειάδας

Ταινία (Τίτλος, Έτος, Διάρκεια, Είδος)
Παίζει(Όνομα, Τίτλος, Έτος)
Ηθοποιός(Όνομα, 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 (Τίτλος, Έτος, Διάρκεια, Είδος)
Παίζει(Όνομα, Τίτλος, Έτος)
Ηθοποιός(Όνομα, 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 Περιεχομένου

```
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> <‘Ονομα_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 (Τίτλος, Έτος, Διάρκεια, Είδος)
Παίζει(Όνομα, Τίτλος, Έτος)
Ηθοποιός(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;

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_i) **REFERENCES** A_j

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,
         Τύπος 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,
         Τύπος 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

INSET INTO R(A_1, …, A_n) VALUES (v_1, …, v_n);

2. Διαγραφές

DELETE FROM R **WHERE** P;

3. Ενημερώσεις/Τροποποιήσεις

UPDATE R
SET Attr = New_Value
WHERE P;

SQLite

Για την πρώτη άσκηση θα χρησιμοποιήσουμε την SQLite3

Μπορείτε να την κατεβάσετε από
https://www.sqlite.org/index.html

Public domain, open source

Σε επόμενες ασκήσεις, μπορεί να χρησιμοποιήσουμε την MySQL

SQLite: serverless

- Most SQL database engines implemented as a separate *server process* (including MySQL).
  - Programs that want to access the database communicate with the server using some kind of interprocess communication (typically TCP/IP)
  - Requires connect to database
- Serveless
  - The database engine runs *within the same process*, thread, and address space as the application.
  - No message passing or network activity.
  - Reads and writes directly from the database files on disk

(+) zero configuration (no separate server process to install, setup, configure, initialize, manage, and troubleshoot)
(-) protection, security, access writes, finer-grained locking and concurrency
(-) memory, disk storage

SQLiteStudio

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

Command line shell
Για κάποιο user interface SQLiteStudio
https://sqlitestudio.pl/index.rvt

Συχνά οι εντολές μέσα από μια γλώσσα προγραμματισμού

"Embedded" και "dynamic" SQL

SQLite: Διαφορές από SQL

Για να δημιουργήσετε μια νέα βάση δεδομένων (αντί για CREATE DATABASE testdb)

.open testdb.db

Για να δείτε το schema (τον ορισμό) ενός πίνακα
.schema <table-name>
Για να δείτε τους πίνακες μιας βάσης δεδομένων
.tables

. εντολές – δείτε το .help
Χωρίς ερωτηματικό στο τέλος

SQLite: Διαφορές από SQL

Υποστηρίζει μόνο
VARCHAR (text)
INT, REAL

SQLite: Υποστήριξη ξένων κλειδιών

Για υποστήριξη ξένων κλειδιών

```
PRAGMA foreign_keys = ON;
```

Επίσης:
The parent key of a foreign key constraint is the primary key of the parent table.
If they are not the primary key, then the parent key columns must be collectively subject to a UNIQUE constraint or have a UNIQUE index.

Ερωτήσεις;

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

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 Α μιας σχέσης R *που δεν είναι κλειδί* θα πρέπει στον ορισμό της R να ορίσετε ένα ευρετήριο στο γνώρισμα Α. Αυτό γίνεται με τη χρήση της εντολής INDEX.

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

Παράδειγμα

Consider a database to store information for a social networking website. The database has the following properties:

- Every user has a unique user ID (integer) along with a full name, age and phone number.
- Every group has a unique group ID (integer) and a name. Every group must have at least one user that serves as moderator of the group.
- A user may be a member of zero or more groups; groups may contain zero or more members (and one or more moderators).
- Users are allowed to create zero or more albums. An album has a unique album ID (integer), a creation date, and a name. An album is owned by exactly one user: the user that created it.
- An album can contain zero or more media files. For every media file, we record its unique URL , the date the file was added to the album, and a caption (if one exists).
- Users can zero or more photos to albums. Photos are a type of media file, but we also track the encoding (e.g., JPEG, PNG, etc.) and the size of the photo (in bytes).
- Users may add zero or more videos to albums. Videos are a type of media file, and we track the codec used to encode the video (e.g., MPEG-4), the length of the video (in seconds), and the video's bitrate.
- A media file may belong to at most one album.