


Το Σχεσιακό Μοντέλο

Κεφάλαιο 3



Σχεσιακό Μοντέλο: Αναγκαιότητα εξέτασης

- Το πιο διαδεδομένο μοντέλο.
 - Χρησιμοποιείται: IBM, Informix, Microsoft, Oracle, Sybase, etc.
- Πρόσφατος ανταγωνιστής:
αντικειμενοστραφές μοντέλο

Σχεσιακή ΒΔ: Ορισμοί

- Σχεσιακή ΒΔ: ένα σύνολο σχέσεων (πινάκων)
- Σχέση: αποτελείται από δύο τμήματα:
 - Στιγμότυπο : πίνακας, με γραμμές και στήλες.
#Γραμμές = πληθυσμός, #πεδία = βαθμός.
 - Σχήμα : καθορίζει την ονομασία της σχέσης, και δίνει ονομασίες και τύπους στα πεδία.
 - π.χ. Students(*sid*: string, *name*: string, *login*: string, *age*: integer, *gpa*: real).
- Κάθε σχέση είναι σύνολο γραμμών ή πλειάδων (όλες οι γραμμές είναι διακριτές).

Παράδειγμα Στιγμιότυπου

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

🚶 Πληθυσμός = 3, βαθμός = 5, διακριτές γραμμές

🚶 Οι στήλες πρέπει να είναι επίσης διακριτές;



Γλώσσες Σχεσιακών Αιτημάτων

- Πλεονέκτημα του Σχεσιακού Μοντέλου:
ευκολία υλοποίησης και ταχύτητα
επεξεργασίας αιτημάτων.
- Τα αιτήματα διατυπώνονται διαισθητικά, και
το ΣΔΒΔ τα βελτιστοποιεί.
 - Κλειδί: ακριβής σημασιολογία των σχεσιακών
αιτημάτων.
 - Επιτρέπει στο βελτιστοποιητή να αλλάξει τη σειρά
εκτέλεσης των λειτουργιών χωρίς να μεταβληθεί το
αποτέλεσμα.



Η Γλώσσα Αιτημάτων SQL

- ☞ Αναπτύχθηκε από IBM τη δεκαετία '70
- ☞ Ανάγκη ύπαρξης προτύπου, χρησιμοποιείται από πολλά ΣΔΒΔ.
- ☞ Standards:
 - SQL-86
 - SQL-89
 - SQL-92
 - SQL-99
 - SQL-02



Η Γλώσσα Αιτημάτων SQL

🚗 Αναζήτηση φοιτητών ηλικίας 18 ετών:

```
SELECT *  
FROM Students S  
WHERE S.age=18
```

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2

Η 1η γραμμή για εμφάνιση name και login ως εξής:

```
SELECT S.name, S.login
```

Αιτήματα επί Πολλαπλών Σχέσεων

☞ Τι υπολογίζει το επόμενο Αίτημα;

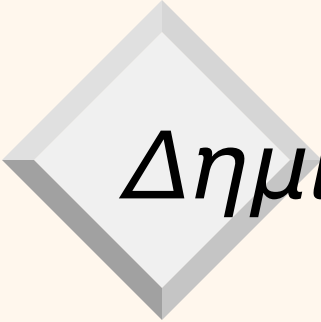
```
SELECT S.name, E.cid  
FROM Students S, Enrolled E  
WHERE S.sid=E.sid AND E.grade='A'
```

Με βάση το επόμενο
στιγμιότυπο του Enrolled:

sid	cid	grade
53831	Carnatic101	C
53831	Reggae203	B
53650	Topology112	A
53666	History105	B

λαμβάνουμε:

S.name	E.cid
Smith	Topology112



Δημιουργία Σχέσεων στην DDL

- ❏ Δημιουργεί τη σχέση Students. Σημειώστε ότι οι περιορισμοί του τύπου δεδομένων για κάθε πεδίο πρέπει να ικανοποιούνται κατά την εισαγωγή ή τη μεταβολή των πλειάδων.
- ❏ Ο πίνακας Enrolled αποθηκεύει την πληροφορία για τα μαθήματα που παρακολουθούν οι φοιτητές.

```
CREATE TABLE Students  
(sid: CHAR(20),  
name: CHAR(20),  
login: CHAR(10),  
age: INTEGER,  
gpa: REAL)
```

```
CREATE TABLE Enrolled  
(sid: CHAR(20),  
cid: CHAR(20),  
grade: CHAR(2))
```



Διαγραφή και Μεταβολή Σχέσεων

DROP TABLE Students

- ☛ Διαγράφει τη Σχέση Students. Διαγράφεται τόσο το Σχήμα όσο και οι πλειάδες.

ALTER TABLE Students

ADD COLUMN firstYear: integer

- ☛ Μεταβολή του Σχήματος της Σχέσης Students. Προσθήκη νέου πεδίου. Κάθε πλειάδα του υπάρχοντος Στιγμιότυπου έχει τιμή null για το νέο πεδίο.



Εισαγωγή και Διαγραφή Πλειάδων

- Εισαγωγή πλειάδας στη Σχέση:

```
INSERT INTO Students (sid, name, login, age, gpa)  
VALUES (53688, 'Smith', 'smith@ee', 18, 3.2)
```

- Διαγραφή πλειάδων που ικανοποιούν συνθήκες (π.χ., name = Smith):

```
DELETE  
FROM Students S  
WHERE S.name = 'Smith'
```




Περιορισμοί Ακεραιότητας (ΠΑ)

- **ΠΑ:** συνθήκη που πρέπει να ισχύει για κάθε στιγμότυπο μιας ΒΔ; π.χ., **Περιορισμοί Πεδίων Ορισμού.**
 - ΠΑ: καθορίζονται μαζί με το Σχήμα.
 - ΠΑ: ελέγχονται κατά τη μεταβολή των Σχέσεων.
- Έγκυρο Στιγμότυπο ικανοποιεί όλους τους ΠΑ.
 - Τα ΣΔΒΔ πρέπει να αποθηκεύουν μόνο Έγκυρα Στιγμότυπα

Περιορισμοί Κύριου Κλειδιού

- Ένα σύνολο πεδίων είναι κλειδί μιας Σχέσης αν:
 1. Δεν υπάρχουν πλειάδες με ταυτόσημες τιμές στα πεδία κλειδιού, και
 2. Το 1 δεν ισχύει για τα υποσύνολα του κλειδιού.
 - Αν ισχύει; Έχουμε **εμπλουτισμένο κλειδί**.
 - Αν υπάρχουν περισσότερα του ενός κλειδιά σε μια Σχέση επιλέγεται ένα ως το **κύριο κλειδί**.
- Πχ, sid είναι κλειδί του Students. (To name?) Το σύνολο {sid, gra} είναι εμπλουτισμένο κλειδί.



Κύρια και Υποψήφια κλειδιά στην DDL

- ☛ Πολλά υποψήφια κλειδιά (ορίζονται με το **UNIQUE**), ένα επιλέγεται ως κύριο κλειδί.

```
CREATE TABLE Enrolled  
(sid CHAR(20)  
  cid CHAR(20),  
  grade CHAR(2),  
  PRIMARY KEY (sid,cid) )
```

- ☛ Ποιες οι σημασιολογικές διαφορές ανάμεσα στις Σχέσεις που δηλώνονται στις CREATE TABLE του παραδείγματος;

```
CREATE TABLE Enrolled  
(sid CHAR(20)  
  cid CHAR(20),  
  grade CHAR(2),  
  PRIMARY KEY (sid),  
  UNIQUE (cid, grade) )
```

Ξένα κλειδιά, Αναφορική Ακεραιότητα

- **Ξένο Κλειδί** : Σύνολο πεδίων σε μια Σχέση που χρησιμοποιείται ως «αναφορά» σε πλειάδα άλλης Σχέσης. (Αντιστοιχεί το κύριο κλειδί της δεύτερης Σχέσης.) Κάτι σαν «λογικός δείκτης».
- Πχ. το **sid** ως ξένο κλειδί αναφέρεται στο **Students**:
 - Enrolled(sid: string, cid: string, grade: string)
 - Αν υλοποιηθούν όλοι οι περιορισμοί ξένου κλειδιού επιτυγχάνεται η **αναφορική ακεραιότητα**.

Ξένα Κλειδιά στην DDL

- Μόνο όσοι φοιτητές είναι καταχωρημένοι στον Students μπορούν να εγγραφούν σε μαθήματα.

```
CREATE TABLE Enrolled  
(sid CHAR(20), cid CHAR(20), grade CHAR(2),  
PRIMARY KEY (sid,cid),  
FOREIGN KEY (sid) REFERENCES Students )
```

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8



Υλοποίηση της Αναφορικής Ακεραιότητας


- Έστω οι Students και Enrolled; Το sid στον Enrolled είναι ξένο κλειδί με αναφορά στον Students.
- Τι πρέπει να γίνει αν μια πλειάδα με ανύπαρκτο sid στον Student εισαχθεί στον Enrolled? (Απόρριψη!)
- Τι πρέπει να γίνει αν διαγραφεί ένας Student;
 - Πρέπει να διαγραφούν οι αντίστοιχες εγγραφές Enrolled.
 - Πρέπει να απορρίπτονται οι διαγραφές με αναφορές.
 - Αλλαγή της τιμής του sid στον Enrolled ως εξ' ορισμού.
 - (Στην SQL: Τιμή null στο sid του Enrolled.)
- Ομοίως σε περίπτωση μεταβολής κάποιου Student.



Αναφορική Ακεραιότητα στην DDL

- Η SQL / 92 υποστηρίζει όλες τις προαναφερθείσες δυνατότητες.
 - Εξ' ορισμού ισχύει **NO ACTION** (απόρριψη)
 - **CASCADE** (διαγραφή ή μεταβολή των αναφερόμενων)
 - **SET NULL / SET DEFAULT** (μεταβολή της τιμής της αναφερόμενης πλειάδας)

```
CREATE TABLE Enrolled
(sid CHAR(20),
cid CHAR(20),
grade CHAR(2),
PRIMARY KEY (sid,cid),
FOREIGN KEY (sid)
REFERENCES Students
ON DELETE CASCADE
ON UPDATE SET DEFAULT )
```



Αναγκαιότητα των ΠΑ

- ❑ Οι ΠΑ προκύπτουν από τη σημασιολογία των πραγματικών οντοτήτων και συσχετίσεων που εκφράζουν οι Σχέσεις.
- ❑ Οι περιορισμοί κλειδιού και ξένου κλειδιού είναι οι δημοφιλέστεροι από το σύνολο όλων των διαθέσιμων ΠΑ.

Όψεις

- Μια **όψη** είναι μια Σχέση, απλά αποθηκεύουμε τον **ορισμό** της, όχι πλειάδες.

```
CREATE VIEW YoungActiveStudents (name, grade)
AS SELECT S.name, E.grade
FROM Students S, Enrolled E
WHERE S.sid = E.sid and S.age<21
```

- Οι όψεις διαγράφονται με την εντολή **DROP VIEW**.
 - Τι γίνεται με τη **DROP TABLE** αν υπάρχει όψη στον πίνακα;
 - Η εντολή DROP TABLE έχει επιλογές για το χειρισμό τέτοιων περιπτώσεων.



Όψεις και Ασφάλεια

- ❏ Οι όψεις είναι χρήσιμες αν θέλουμε να αποκρύψουμε λεπτομέρειες των Σχέσεων από τις οποίες προέρχονται.