

Η Γλώσσα SQL

(Μέρος 3:Γλώσσα Ορισμού, Γλώσσα Τροποποίησης, Ενσωματωμένη SQL)

Η γλώσσα SQL

Η SQL έχει διάφορα τμήματα:

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

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

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

όπου R είναι το όνομα της σχέσης, A_i τα ονόματα των γνωρισμάτων, και D_i οι τύποι των αντίστοιχων πεδίων τιμών.

Τύποι Πεδίου Ορισμού

Για τον ορισμό του πεδίου ορισμού, οι διαθέσιμοι built-in τύποι περιλαμβάνουν:

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)

Ο ορισμός πεδίου μπορεί να περιέχει τον προσδιορισμό **not null**

Επίσης, επιτρέπεται δημιουργία πεδίου:

```
create domain <name> as <type-description>
```

```
create domain Όνομα-Προσώπου char(20)
```

Ορισμός Σχήματος

create table R($A_1 D_1, A_2 D_2, \dots, A_n D_n$),
<περιορισμός-ακεραιότητας₁>,
...,
<περιορισμός-ακεραιότητας_k>

όπου R είναι το όνομα της σχέσης, A_i τα ονόματα των γνωρισμάτων, και D_i οι τύποι των αντίστοιχων πεδίων τιμών.

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)
- **unique** $A_{j_1}, A_{j_2}, \dots, A_{j_n}$, (δεν επιτρέπονται επαναλαμβανόμενες τιμές; NULL τιμές επιτρέπονται)
- **check** P
- **foreign key** (A_i) **references** A_j

Παραδείγματα

(1)

```
create table Πελάτης  
  (Όνομα-Πελάτη char(20) not null,  
   Οδός char(30),  
   Πόλη char(30),  
   primary key (Όνομα-Πελάτη))
```

(2)

```
create table Λογαριασμός  
  (Αριθμός-Λογαριασμού char(10) not null,  
   Όνομα-Υποκαταστήματος char(15),  
   Ποσό int,  
   primary key (Αριθμός-Λογαριασμού)  
   check (Ποσό >= 0))
```

Ορισμός Σχήματος

Επίσης, πιο περίπλοκες συνθήκες, π.χ., για ξένα κλειδιά:

check (Όνομα-Υποκαταστήματος **in select** Όνομα-Υποκαταστήματος
from Υποκατάστημα)

Περιορισμοί Ακεραιότητας

Περιορισμοί Αναφοράς

Σύνταξη:

foreign key (A_i) **references** A_j

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

on delete cascade
on update cascade

Περιορισμοί Ακεραιότητας

Παράδειγμα

create table

..

foreign key (Όνομα-Υποκαταστήματος) **references** Υποκατάστημα

on delete cascade

on update cascade

...

Περιορισμοί Ακεραιότητας

Περιορισμοί Ακεραιότητας

Πεδίου ορισμού

Χρησιμοποιώντας την εντολή **check**:

Παραδείγματα

(1) Ελάχιστος ωρομίσθιο

create domain Ωρομίσθιο **numeric**(5, 2)

constraint Έλεγχος-Ωρομισθίου **check**(Ποσό >= 4.00)

Περιορισμοί Ακεραιότητας

(2) Να μην περιέχει την τιμή null

create domain Πεδίο-Αριθμός-Λογαριασμού **char(10)**
constraint Έλεγχος-Αριθμός-Λογαριασμού **check(value not null)**

(3) Να παίρνει συγκεκριμένες τιμές

create domain Τύπος-Λογαριασμού **char(10)**
constraint Έλεγχος-Τύπος-Λογαριασμού **check value in ('Όψεως', 'Ταμειευτηρίου')**

Διαγραφή Σχήματος

Διαγραφή Σχήματος

Μια καινούργια σχέση είναι αρχικά άδεια.

Για να σβηστεί ένα σχήμα:

drop table R

Διαφορά από

delete from R

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

- Τροποποίηση Βάσης Δεδομένων
Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)

Τροποποιήσεις

1. Διαγραφή
2. Εισαγωγή
3. Ενημέρωση

Εισαγωγή

Για να εισάγουμε δεδομένα σε μια σχέση είτε

(α) προσδιορίζουμε την πλειάδα, είτε

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

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

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

Παράδειγμα για το (α)

```
insert into Λογαριασμός  
values ("Ψηλά-Αλώνια", "A--9732", 1200)
```

Όταν με οποιαδήποτε σειρά, π.χ.:

```
insert into Λογαριασμός (Αριθμός-Λογαριασμού, Όνομα-  
Υποκαταστήματος, Ποσό)  
values ("A--9732", "Ψηλά-Αλώνια", 1200)
```

Παράδειγμα για το (β):

Για κάθε πελάτη που έχει πάρει δάνειο από το υποκατάστημα Ψηλά Αλώνια προστίθεται ως δώρο ένας λογαριασμός των \$200

```
insert into Λογαριασμός  
  select Όνομα-Υποκαταστήματος, Αριθμός-Δανείου, 200  
  from Δάνειο  
  where Όνομα-Υποκαταστήματος = "Ψηλά Αλώνια"
```

Πρέπει πρώτα να υπολογιστεί το **select** πλήρως και μετά να γίνει η εισαγωγή.

Τι αποτέλεσμα έχει η παρακάτω εντολή αν αυτό δε συμβαίνει;

```
insert into Λογαριασμός  
  select *  
  from Λογαριασμός
```

Επίσης, εισαγωγή null τιμών:

```
insert into Λογαριασμός  
  values (null, "A--9732", 1200)
```

Διαγραφή

Μπορούμε να σβήσουμε 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) Όλους τους λογαριασμούς του Παπαδόπουλου

```
delete from Καταθέτης  
where Όνομα-Πελάτη = "Παπαδόπουλος"
```

(2) Όλους τους λογαριασμούς στα υποκαταστήματα της Πάτρας

```
delete from Λογαριασμός  
where Όνομα-Υποκαταστήματος in (select Όνομα-Υποκαταστήματος  
from Υποκατάστημα  
where Πόλη = "Πάτρα")
```

Παρατήρηση: δεν υπάρχει τρόπος να διαγράψουμε τη μία από δυο ίδιες πλειάδες

Αν και μπορούμε να σβήσουμε πλειάδες μόνο από μία σχέση τη φορά μπορούμε να αναφερθούμε σε περισσότερες από μια σχέσεις στην υποερώτηση του **where**

(3) Όλους τους λογαριασμούς μιας τράπεζας με ποσό μικρότερο από το μέσο ποσό στην τράπεζα.

```
delete from Λογαριασμός  
where Ποσό > (select avg(Ποσό)  
from Λογαριασμός)
```

Πρώτα γίνεται ο έλεγχος σε όλες τις πλειάδες και μετά αυτές που ικανοποιούν τη συνθήκη διαγράφονται.

Παράδειγμα: μια τράπεζα θέλει να κλείσει όλα τα υποκαταστήματά της που βρίσκονται στην Καστοριά

delete from Υποκατάστημα

where Όνομα-Υποκαταστήματος **in** (**select** Όνομα-Υποκαταστήματος
from Υποκατάστημα
where Πόλη = "Καστοριά')

Πρέπει να διαγράψουμε και όλους τους λογαριασμούς:

delete from Λογαριασμός

where Όνομα-Υποκαταστήματος **in** (**select** Όνομα-Υποκαταστήματος
from Υποκατάστημα
where Πόλη = "Καστοριά')

ΠΡΟΣΟΧΗ: όταν θέλουμε να διαγράψουμε κάποια δεδομένα, πρέπει να διαγράψουμε *όλα* τα δεδομένα που συσχετίζονται με αυτά. Επίσης πρέπει να προσέξουμε την σειρά με την οποία θα γίνουν οι διαγραφές.

Διαγραφή

<u>υποκατάστημα</u>		<u>λογαριασμός</u>		
Πόλη	Όνομα_Υποκ.	Όνομα_Υποκ.	Όνομα-Πελάτη	Υπόλοιπο
Καστοριά	K1	K1	ΚΩΤΣΗΣ	350.000
Καστοριά	K3	K2	ΑΠΟΣΤΟΛΙΔΗΣ	230.000
Θεσσαλονίκη	Θ1	Θ1	ΣΤΕΦΑΝΟΥ	670.000
Θεσσαλονίκη	Θ2	Θ2	ΠΑΠΑΝΙΚΟΛΑΟΥ	256.000
Αθήνα	A1	K3	ΧΑΤΖΟΠΟΥΛΟΣ	410.000
...		...		

• αν διαγράψουμε από τον πίνακα υποκατάστημα όλα τα υποκαταστήματα της Καστοριάς, θα έχουμε πρόβλημα ορθότητας στον πίνακα λογαριασμός.

• πρώτα πρέπει να διαγράψουμε τους λογαριασμούς και μετά τα υποκαταστήματα.

Ενημερώσεις

Ενημερώσεις

Παράδειγμα: Αύξηση όλων των καταθέσεων που είναι μεγαλύτερες των \$100 κατά 5% λόγω τοκισμού

```
update Λογαριασμός
set Ποσό = Ποσό * 1.05
where Ποσό > 100
```

Παράδειγμα:

στους πελάτες που έχουν υπόλοιπο < 1.000.000 η τράπεζα δίνει 5% και στους πελάτες που έχουν υπόλοιπο > 1.000.000 δίνει 9%:

```
update Λογαριασμός  
set Ποσό = Ποσό * 1.05  
where Ποσό < 1.000.000
```

```
update Λογαριασμός  
set Ποσό = Ποσό * 1.09  
where Ποσό > 1.000.000
```

Ποιο update πρέπει να τρέξουμε πρώτα;

Παράδειγμα: Αύξηση όλων των υπολοίπων που είναι μεγαλύτερα από τον μέσο όρο κατά 5%

```
update Λογαριασμός  
set Υπόλοιπο = Υπόλοιπο * 1.05  
where Υπόλοιπο > select avg(Υπόλοιπο)  
from Λογαριασμός
```

Ενσωματωμένη SQL

Προσπέλαση μιας ΒΔ μέσω μια γλώσσας προγραμματισμού γενικού σκοπού απαιτείται τουλάχιστον γιατί:

- υπάρχουν ερωτήσεις που δε μπορούν να διατυπωθούν σε SQL, γιατί η SQL δεν έχει όλες τις δυνατότητες μιας γλώσσας προγραμματισμού γενικού σκοπού
- μη-δηλωτικές εντολές (π.χ., εκτύπωση, επικοινωνία με το χρήστη) δε μπορούν να γίνουν μέσω της SQL

Ενσωμάτωση της SQL σε μια γλώσσα που καλείται *φιλόξενη* (host)

Σε αυτήν την περίπτωση, η επεξεργασία των ερωτήσεων γίνεται από τη ΒΔ, και το αποτέλεσμα γίνεται διαθέσιμο στο πρόγραμμα **μια πλειάδα τη φορά**

Ένας ειδικός προ-επεξεργαστής (preprocessor) αντικαθιστά τον ενσωματωμένο κώδικα της SQL με δηλώσεις και κλήσεις συναρτήσεων στη host γλώσσα και μεταφράζεται το πρόγραμμα

Σύνταξη της μορφής:

```
EXEC SQL < embedded SQL statement > END-EXEC
```

Η ακριβής σύνταξη εξαρτάται από τη host γλώσσα

Ενσωματωμένη SQL

Χρησιμοποιούμε την εντολή: **SQL INCLUDE**, για να δηλώσουμε στον προ-επεξεργαστή που πρέπει να εισάγει τις δηλώσεις των μεταβλητών της SQL

Μεταβλητές της γλώσσας μπορεί να χρησιμοποιηθούν στην εντολή της SQL αν το σύμβολο : προηγείται του ονόματός τους

Για να γράψουμε μια ερώτηση σε SQL:

EXEC SQL

```
declare c cursor for
select Όνομα-Πελάτη, Πόλη
from Κατάθεση, Πελάτης
where Κατάθεση.Όνομα-Πελάτη = Πελάτης.Όνομα-Πελάτη
and Κατάθεση.Ποσό > :amount
```

END-EXEC

Ενσωματωμένη SQL

Η παραπάνω εντολή δεν προκαλεί την εκτέλεση της ερώτησης, για να εκτελεστεί:

```
EXEC SQL open c END-EXEC
```

Το αποτέλεσμα σώζεται σε μια προσωρινή σχέση, αν υπάρχει λάθος το διαγνωστικό μήνυμα σώζεται σε μια ειδική μεταβλητή (SQLCA)

Οι πλειάδες του αποτελέσματος γίνονται διαθέσιμες στο πρόγραμμα μέσω μιας σειράς από **fetch** εντολές, χρειάζεται μια μεταβλητή της host γλώσσας για κάθε γνώρισμα

```
EXEC SQL fetch c into :cn$, :cc END-EXEC
```

Ενσωματωμένη SQL

Για να σβήσουμε τη προσωρινή σχέση:

```
EXEC SQL close c END-EXEC
```

Επίσης:

```
EXEC SQL < update, insert ή delete έκφραση > END-EXEC
```

Ενσωματωμένη SQL

Δυναμική SQL

Τα προγράμματα μπορούν να δημιουργούν ερωτήσεις σε SQL ως συμβολοσειρές δυναμικά κατά την εκτέλεση και είτε να τα εκτελούν αμέσως είτε να τα προετοιμάζουν (τα μεταφράζουν) για να χρησιμοποιηθούν αργότερα

Παράδειγμα:

```
char * sqlprog = 'update Λογαριασμός set Ποσό = Ποσό * 1.05  
                where Αριθμός-Λογαριασμού = ?'  
EXEC SQL prepare dynprog from :sqlprog;  
char account[10] = "A-101";  
EXEC SQL execute dynprog using :account;
```

Άλλα Χαρακτηριστικά

- Γλώσσες 4ης Γενιάς
- Έννοια του *session* μεταξύ ενός *client* και του *server* του ΣΒΔ
- Δημιουργία σχήματος: `create schema` και
- Σβήσιμο σχήματος : `drop schema`