

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ Ι

Ενότητα 9β: SQL (Διαχείριση εγγρα- φών, Όψεις, Περιορισμοί, Εναύσματα)

Ευαγγελίδης Γεώργιος
Τμήμα Εφαρμοσμένης Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Σκοποί ενότητας

εισαγωγή-διαγραφή-τροποποίηση εγγραφών, όψεις, περιορισμοί ακεραιότητας (non-null, κλειδιού, αναφορικής ακεραιότητας, πεδίου, εγγραφής, γενικοί ισχυρισμοί), εναύσματα

Αιτήματα διαχείρισης δεδομένων

- insert into Table values (VA1, VA2, ... , VAn)
- insert into Table Select-Statement
- delete from Table where Condition
- update Table
- set A1 = Expr1, A2 = Expr2, ..., An = Exprn
- where Condition

Query 01

-- εισαγωγή μιας εγγραφής στον πίνακα cd

insert into cd

values (100, 'My first album', 2013, 1);

Query 02

```
-- μαζική εισαγωγή από άλλη select  
insert into performer  
select pid+1000, concat(name, ' Junior')  
from performer;
```

Query 03

-- μια πιο περιπετειώδης εισαγωγή

```
insert into cd
```

```
select 1000+sid, stitle, 2013, 1
```

```
from song s join track t using (sid)
```

```
where pos = 13
```


Query 04

-- διαγραφή εγγραφών βάσει συνθήκης

delete from performer

where pid not in (select pid from track);

Query 05

-- αυτό όμως δεν δουλεύει σε όλα τα DBMS

delete from performer

where pid in

(select pid

from performer

where name like '%Junior%');

Query 06

-- λύση: δημιουργούμε προσωρινό πίνακα και
-- μετά εκτελούμε τη διαγραφή

```
create temporary table tt as  
select pid from performer  
where name like '%Junior%';  
  
delete from performer  
where pid in (select pid from tt);
```

Query 07

-- τροποποίηση πεδίων βάσει συνθήκης - τι
-- πετυχαίνει η εντολή αυτή;

```
update track  
set pid=pid+1000  
where pid+1000 in (select pid from performer);
```

Query 08

-- να και μια περίεργη τροποποίηση...

```
update cd
set year = 2000+comid,
    comid = (select min(comid) from company);
```

Όψεις

- View = Query
- σχήμα της όψης είναι το σχήμα του αιτήματος
- θα μπορούσαμε να ορίσουμε την όψη ως
- ιδεατό (virtual) πίνακα

Query 09

-- δημιουργία όψης

```
create view 21centurycd as  
select * from cd where year > 1999;
```

Query 10

-- χρήση όψης

```
select cc.ctitle, count(*)  
from 21centurycd cc join track t using (cid)  
group by cc.cid;
```


Query 11

-- ιδεατή εκτέλεση του παραπάνω

```
create temporary table temp as
select * from cd where year > 1999;

select cc.ctitle, count(*)
from temp cc join track t using (cid)
group by cc.cid;

drop table temp;
```

Query 12

-- στην όμως πράξη το DBMS
-- κάνει αυτό (rewriting):

```
select cc.ctitle, count(*)  
from (select * from cd where year > 1999) cc  
     join track t using (cid)  
group by cc.cid, cc.ctitle;
```

Query 13

-- ...ή για την ακρίβεια αυτό! (αν έχει καλό
-- βελτιστοποιητή)

```
select cc.ctitle, count(*)  
from cd cc join track t using (cid)  
where year > 1999  
group by cc.cid, cc.ctitle;
```

Query 14

```
-- ορισμός όψης πάνω σε όψη  
create view 21centuryperf as  
select name  
from (21centurycd join track using(cid))  
join performer using(pid);
```

Query 15

-- τί γίνεται αν διαγράψω την όψη;

```
drop view 21centurycd;
```

Περιορισμοί

- περιορισμοί non-null,
 - κλειδιού (primary, unique), ξένου κλειδιού
 - καθυστερημένος έλεγχος όταν έχουμε
 - μπλοκ εντολών τύπου
 - begin...commit ή begin...rollback (συναλλαγή)
 - το παρακάτω χρειάζεται στην sqlite γιατί ο
 - περιορισμός ξένου κλειδιού δεν ελέγχεται
 - by default
- ```
PRAGMA foreign_keys = ON;
```

# Query 16

```
-- τί σημαίνει unique(year, ctitle);

create table cd2 (
 cid integer primary key,
 ctitle varchar(200),
 year integer not null,
 comid integer references company2(comid)
 deferrable initially deferred,
 unique(year, ctitle));
```

# Query 17

---

-- έχει ενδιαφέρον η περίπτωση unique όταν το  
-- name είναι null

```
create table company2 (
 comid integer primary key,
 name varchar(200) unique);
```



# Query 18

-- εξαιτίας του "deferrable initially deferred" τα  
-- παρακάτω είναι OK

```
begin;
```

```
insert into cd2 values
```

```
 (1, 'My first cd', 2013, 3);
```

```
insert into company2 values
```

```
 (3, 'My record company');
```

```
commit;
```

# Query 19

- περιορισμός ξένου κλειδιού: αντιμετώπιση
- ενημερώσεων/διαγραφών

```
create table cd2 (
 cid integer primary key,
 ctitle varchar(200),
 year integer not null,
 comid integer references company2(comid)
on
 update cascade on delete set null);
```

# Query 20

- περιορισμοί επιπέδου πεδίου/εγγραφής
- ΠΡΟΣΟΧΗ: η MySQL δέχεται τέτοιους
- περιορισμούς, αλλά δεν τους ελέγχει...

```
create table cd2 (cid integer primary key,
 ctitle varchar(200) check(ctitle not like '0%'),
 year integer not null check(year > 1900),
 comid integer references company2(comid)
 deferrable initially deferred,
 unique(year, ctitle),
 check(ctitle like 'a%'
 or year between 1990 and 2000));
```

# Query 21

---

- παρόλο που το standard της SQL προβλέπει
- εμφωλευμένα αιτήματα μέσα σε περιορισμούς
- τύπου check, κανένα DBMS δεν έχει
- υλοποιήσει κάτι τέτοιο

# Query 21 (συνέχεια)

- Επίσης, κανένα DBMS δεν έχει υλοποιήσει
- γενικούς ισχυρισμούς (general assertions)
- παρόλο που υποστηρίζονται από το standard
- της SQL

```
create assertion cdsize
check (not exists
(select cid
from track join cd using(cid)
group by cid
having count(*)>99));
```

# Query 22

```
-- Εναύσματα - δοκιμάστε τα στην MySQL
-- τί κάνει αυτό το έναυσμα;

create trigger cascade_delete
after delete on company2
for each row
begin
delete from cd2 where comid = old.comid;
end;
```

# Query 23

```
-- αυτόματη εισαγωγή νέας εταιρίας

create trigger auto_insert_company
after insert on cd2
for each row
begin
insert into company2 values (new.comid,
concat('new company for cd: ', new.ctitle));
end;
```

# Query 24

---

-- ενημέρωση μετά από εισαγωγή

```
create trigger update_counter
after insert on cd2
for each row
begin
update counter set val=val+1;
end;
```



# Query 25

-- Εναύσματα - δοκιμάστε τα στην sqlite  
-- έναυσμα υπό συνθήκη

```
create trigger tracks_less_than_4
after insert on track2
for each row
when 3 <
 (select count(*) from track2 where cid =
new.cid)
begin
delete from track2 where cid=new.cid and pos=new.pos;
end;
```

# Query 26

```
-- ή ακόμα καλύτερα χρησιμοποιώντας το
-- raise(ignore) της sqlite:
create trigger tracks_less_than_4
before insert on track2
for each row
when 2 < (select count(*) from track2 where cid =
new.cid)
begin
select raise(ignore);
end;
```

# Query 27

```
-- έλεγχος σειριακής αρίθμησης των track
create trigger track_numbering
before insert on track2
for each row
when (new.pos=1 and exists
 (select cid from track2 where cid=new.cid)) or
 (new.pos<>1 and new.pos-1 not in
 (select pos from track2 where cid=new.cid))
begin
select raise(ignore);
end;
```

# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

