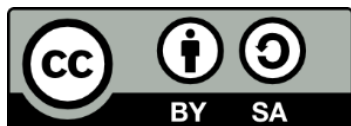


ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ Ι

Ενότητα 6: SQL (Συζεύξεις, Εμφώλευση, Ομαδοποίηση)

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

σύζευξη υπό συνθήκη, εξωτερικές συζεύξεις,
συναθροιστικά αιτήματα, ομαδοποίηση

Query 01

-- σύζευξη υπό συνθήκη (εδώ φυσική σύζευξη)
select distinct *
from performer, track
where performer.pid = track.pid;

Query 02

-- εναλλακτικός τρόπος γραφής (δουλεύει και
-- χωρίς το inner)
select distinct name, cid
from performer inner join track
on performer.pid = track.pid;

Query 03

```
-- η επιπλέον συνθήκη μπορεί να μπει και στο on
select distinct *
from performer inner join track on
    performer.pid = track.pid and cid < 5;
```

Query 04

- τίτλοι cd με ονόματα ερμηνευτών
- ΠΡΟΣΟΧΗ: δεν δουλεύει στην Oracle -
- η σύζευξη είναι δυαδική πράξη

```
select distinct ctitle, name
from (cd c join track t on c.cid=t.cid) join
     performer p on t.pid=p.pid;
```

```
select distinct ctitle, name
from performer p, cd, track t
where p.pid=t.pid and cd.cid=t.cid;
```


Query 05

- φυσική σύζευξη η οποία κρατά μόνο ένα
- αντίγραφο της κοινής στήλης (όπως στη
- σχεσιακή άλγεβρα!)

```
select distinct *  
from performer natural join track;
```

Query 06

-- φυσική σύζευξη με χρήση του using
-- (καλύτερη πρακτική)
select distinct *
from performer join track using(pid);

Query 07

-- ζεύγη cd που κυκλοφόρησαν την ίδια χρονιά
-- (από το προηγούμενο μάθημα)
select c1.ctitle, c1.year, c2.ctitle, c2.year
from cd c1, cd c2
where c1.year = c2.year and c1.cid < c2.cid;

Query 08

-- σύζευξη πίνακα με τον εαυτό του με using
select c1.ctitle, c1.year, c2.ctitle, c2.year
from cd c1 join cd c2 using (year)
where c1.cid > c2.cid;

Query 09

-- φυσική σύζευξη πίνακα με τον εαυτό του
select *
from cd c1 natural join cd c2;

Query 10

-- η φυσική σύζευξη επιστρέφει μόνο τις
-- ταιριαστές εγγραφές
select ctitle, year, name
from cd inner join company using(comid);

Query 11

- left outer join για να πάρουμε όλες τις
- εγγραφές των cd
- μπορούμε να παραλείψουμε το outer
- μπορούμε να χρησιμοποιήσουμε
- natural left join

```
select ctitle, year, name
from cd left outer join company using(comid);
```

Query 12

-- τρόπος υλοποίησης left join αν δεν
-- υποστηρίζεται από το DBMS

```
select ctitle, year, name  
from cd join company using(comid)  
union  
select ctitle, year, NULL  
from cd  
where comid not in (select comid from company);
```


Query 13

-- right outer join για να πάρουμε όλες τις
-- εγγραφές των εταιριών
select ctitle, year, name
from cd right join company using(comid);

Query 14

-- τρόπος υλοποίησης right join αν δεν
-- υποστηρίζεται από το DBMS
select ctitle, year, name
from company join cd using(comid)
union
select null, null, name
from company
where not comid in (select comid from cd);

Query 15

```
-- full outer join  
-- ΠΡΟΣΟΧΗ: δεν υποστηρίζεται από την MySQL  
select ctitle, year, name  
from cd full outer join company using(comid);
```

Query 16

-- τρόπος υλοποίησης full outer join αν δεν
-- υποστηρίζεται από το DBMS
select ctitle, year, name
from cd left join company using(comid)
union
select ctitle, year, name
from cd right join company using(comid);

Query 17

-- τρόπος υλοποίησης full outer join χωρίς

-- συζεύξεις

```
select ctitle, year, name from cd, company  
where cd.comid = company.comid
```

union

```
select ctitle, year, NULL from cd  
where comid not in (select comid from company)
```

union

```
select null, null, name from company  
where not comid in (select comid from cd);
```

Query 18

```
-- αιτήματα συνάθροισης  
-- (avg, max, min, sum, count)  
select avg(year)  
from cd;
```

Query 19

-- μέση χρονιά κυκλοφορίας των cd της Adele

```
select avg(year)
from (cd join track using(cid)) join
     performer using(pid)
where name = 'Adele';
```

Query 20

-- η σωστή εκδοχή του Query 19

```
select avg(year)
```

```
from cd
```

```
where cid in
```

```
  (select cid
```

```
   from track join performer using(pid)
```

```
   where name = 'Adele');
```


Query 21

-- πλήθος εγγραφών που ικανοποιούν συνθήκη
select count(*)
from cd
where ctitle > 'k';

Query 22

-- πλήθος ερμηνευτών που ερμηνεύουν κομμάτια
select count(pid)
from track;

Query 23

```
-- ενδιαφέρουσα χρήση του count στη where
-- (δοκιμάστε και με >=, >)
select *
from performer p1, performer p2
where
    (select count(pid) from track
     where pid = p1.pid) =
    (select count(pid) from track
     where pid = p2.pid)
and p1.pid > p2.pid;
```

Query 24

-- έλεγχος πλήθους δυο ομάδων που
-- δημιουργούνται στο from
select AtoJ.tracks >= KtoZ.tracks
from

```
(select count(pid) as tracks  
from performer join track using(pid)  
where name < 'k') AtoJ,  
(select count(pid) as tracks  
from performer join track using(pid)  
where name >= 'k') KtoZ;
```

Query 25

-- το ίδιο με το Query 24 αλλά
-- με τα υπο-αιτήματα στο select
select

(select count(pid) as tracks
from performer join track using(pid)
where name < 'k') >=

(select count(pid) as tracks
from performer join track using(pid)
where name >= 'k') as result

from cd;

Query 26

```
-- ομαδοποίηση  
select cid, count(*)  
from track  
group by cid;
```

Query 27

-- άλλη μια ομαδοποίηση με χρήση
-- συναρτήσεων αλφαριθμητικών
select substring(stitle,1,1), count(*) cnt
from song
group by substring(stitle,1,1)
order by cnt desc;

Query 28

```
-- πιο πολύπλοκη ομαδοποίηση  
select pid, name, count(*)  
from performer p join track t using(pid)  
where name > 'K'  
group by pid;
```


Query 29

-- ομαδοποίηση δυο επιπέδων
select cid, substring(stitle,1,1) letter, count(*) cnt
from track join song using(sid)
group by cid, substring(stitle,1,1);

Query 30

-- προσοχή στο τί μετράμε
select year, count(distinct cid)
from cd join track using(cid)
group by year;

Query 31

-- πως εμφανίζουμε αυτά που δεν
-- συμμετέχουν σε μια σύζευξη

```
select comid, count(*)  
from cd join company using (comid)  
group by comid  
union  
select comid, 0  
from company  
where comid not in (select comid from cd);
```

Query 32

```
-- συνθήκη σε επίπεδο ομάδας (having)
select cid
from track
group by cid
having count(*) > 13;
```

Query 33

```
-- εναλλακτική σύνταξη χωρίς group by ... having
select distinct cid
from track t1
where 13 <
    (select count(*)
     from track t2
     where t1.cid = t2.cid);
```

Query 34

-- τα cd που έχουν πλήθος κομματιών μεγα-
-- λύτερο από τον μέσο όρο κομματιών ανά cd

```
select cid, count(*)  
from track  
group by cid  
having count(*) >  
    (select avg(trackcount)  
    from  
        (select count(*) as trackcount from track  
        group by cid) A);
```

Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ