



**ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ**

Αρχεία και Βάσεις Δεδομένων

Διάλεξη 10η: SQL Μέρος 3ο

Δημήτρης Πλεξουσάκης

Τμήμα Επιστήμης Υπολογιστών

SQL Data Manipulation Language

- Τελεστής **union**
 - συνδυάζει **subselects** τα οποία παράγουν συμβατές σχέσεις
 - γενική μορφή: *subselect* {**union** [**all**] *subselect*}
 - περιορισμός: τα *subselects* δεν μπορούν να περιέχουν τον τελεστή **union**
 - **Παράδειγμα:** Βρείτε τις πόλεις στις οποίες διαμένουν είτε πελάτες είτε πράκτορες

```
select city from customers union  
select city from agents;
```

ή για να επιτρέψουμε επαναλαμβανόμενες πλειάδες

```
select city from customers union all  
select city from agents;
```

SQL: Διαίρεση

Orders(orderno, month,cid,aid,pid,qty,amt)

Agents(aid, aname,city,percent)

Customers(cid, cname,city,discont)

- Διαίρεση: η SQL δεν παρέχει τελεστή για διαίρεση
- Παράδειγμα: Βρείτε τα ids των πελατών που **κάνουν παραγγελίες** μέσω **όλων των πρακτόρων της Νέας Υόρκης**.
 - Ή ισοδύναμα: Βρείτε τα ids των πελατών, έτσι ώστε **να μην υπάρχει πράκτορας στη Νέα Υόρκη** που δεν κάνει παραγγελίες για κάθε έναν από αυτούς τους πελάτες.

```
select cid from customers where not exists
(select * from agents where city="New York"
and not exists
(select * from orders where
orders.cid=customers.cid and
orders.aid=agents.aid) );
```

SQL: Διαίρεση

Orders(orderno, month,cid,aid,pid,qty,amt)

Agents(aid, aname,city,percent)

Customers(cid, cname,city,discont)

- **Παράδειγμα:** Βρείτε τα ids των πρακτόρων στη Νέα Υόρκη ή το Dallas που **κάνουν παραγγελίες για όλα τα προϊόντα που κοστίζουν πάνω από 1\$**.
- **Ή ισοδύναμα:** Βρείτε τα ids των πρακτόρων στη Νέα Υόρκη ή το Dallas, έτσι ώστε **να μην υπάρχει προϊόν που να κοστίζει πάνω από 1\$ και να μην το παραγγέλνουν.**

```
select aid from agents where (city="New York"
or city = "Dallas") and not exists
(select pid from products where price > 1.00
and not exists
(select * from orders where
orders.pid=products.pid and
orders.aid=agents.aid) );
```

SQL: Διαίρεση

- Παράδειγμα: Βρείτε τα ids των προϊόντων που παραγγέλνονται από όλους τους πελάτες στο Dallas.
 - Ή ισοδύναμα: Βρείτε τα ids των προϊόντων που είναι τέτοια ώστε να μην υπάρχει πελάτης στο Dallas που να μην τα παραγγέλνει.

```
select pid from products where not exists  
(select cid from customers where city="Dallas"  
and not exists  
  
  (select * from orders where  
   orders.pid=products.pid and  
   orders.cid=customers.cid) ) ;
```
- Η έκφραση της διαίρεσης με αυτό τον τρόπο βασίζεται στην ισοδυναμία:

$$\forall z \exists y p(z,y) \equiv \neg \exists z \neg \exists y p(z,y)$$

SQL: Συναρτήσεις Συνάθροισης

- Συναρτήσεις Συνάθροισης (Aggregate Functions)
- Εφαρμόζονται πάνω σε σύνολα τιμών γνωρισμάτων.
- **count, max, min, avg, sum**
- Περιορισμοί:
 - η συνάρτηση **count** μπορεί να εφαρμοστεί σε γνωρίσματα οποιουδήποτε τύπου
 - οι συναρτήσεις **avg** και **sum** εφαρμόζονται μόνο σε γνωρίσματα αριθμητικών τύπων
 - οι συναρτήσεις **min** και **max** εφαρμόζονται σε γνωρίσματα αριθμητικών ή αλφαριθμητικών τύπων

SQL: Συναρτήσεις Συνάθροισης

Orders(orderno, month,cid,aid,pid,qty,amt)

Agents(aid, aname,city,percent)

Customers(cid, cname,city,discnt)

- Παραδείγματα:

1. Υπολογίστε το συνολικό ποσό όλων των παραγγελιών

```
select sum(amt) from orders;
```

2. Υπολογίστε τη συνολική ποσότητα του προϊόντος p03 που έχει παραγγελθεί.

```
select sum(qty) as TOTAL from orders where  
pid= 'p03' ;
```

3. Βρείτε το συνολικό αριθμό πελατών.

```
select count(cid) from customers;
```

ή ισοδύναμα: **select count(*) from customers;**

Οι κενές τιμές δεν μετρούνται. Οι δύο εκφράσεις δίνουν την ίδια απάντηση γιατί δεν επιτρέπονται κενές τιμές στο γνώρισμα cid.

SQL: Συναρτήσεις Συνάθροισης

Orders(orderno, month,cid,aid,pid,qty,amt)

Agents(aid, aname,city,percent)

Customers(cid, cname,city,discnt)

4. Βρείτε το συνολικό αριθμό πόλεων όπου υπάρχουν πελάτες.

```
select count (distinct city) from customers;
```

5. Βρείτε τα ids των πελατών των οποίων η έκπτωση είναι μικρότερη από τη μέγιστη έκπτωση.

```
select cid from customers where discnt <  
max(discnt) ;
```

Η έκφραση είναι λανθασμένη! Συγκρίσεις με συναρτήσεις συνάθροισης επιτρέπονται μόνο όταν η συνάρτηση επιστρέφεται από subselect.

Η ορθή έκφραση είναι:

```
select cid from customers where discnt <  
(select max(discnt)from customers) ;
```


SQL: Κενές τιμές

- Κενές τιμές στην SQL:
 - Μια κενή τιμή είναι μια ειδική σταθερά η οποία αναπαριστά μια τιμή η οποία είτε δεν είναι γνωστή είτε δεν έχει νόημα για ένα συγκεκριμένο στιγμιότυπο
 - Τα περισσότερα ΣΔΒΔ δε διαφοροποιούν τις δύο ερμηνείες των κενών τιμών
 - Κενές τιμές μπορούν να εισαχθούν με την εντολή **insert**
 - **Παράδειγμα:** Ένας νέος πελάτης εισάγεται στη σχέση customers αλλά δεν είναι γνωστή η τιμή του γνωρίσματος `discnt`.

```
insert into customers (cid, cname, city) values  
( 'c007' , 'James Bond' , 'London' );
```

Η τιμή στο γνώρισμα `discnt` θα είναι `null`.

Customers(cid, cname,city,discnt)

SQL: Κενές τιμές

- Κενές τιμές στην SQL:
 - Το αποτέλεσμα μιας σύγκρισης με μια κενή τιμή είναι `unknown` (ούτε `true`, ούτε `false`).
 - Παράδειγμα: η ερώτηση
`select * from customers where discnt <=10 or discnt >10;`
δε θα επιστρέψει την πλειάδα με `cid='c007'` του προηγούμενου παραδείγματος
 - Κενές τιμές μπορούν να ανακτηθούν με χρήση του κατηγορήματος `is null`
 - Παράδειγμα: `select * from customers where discnt is null;`

SQL: Κενές τιμές

- Κενές τιμές στην SQL:
 - Οι κενές τιμές **δε** συμμετέχουν στον υπολογισμό συναρτήσεων συνάθροισης
 - **Παράδειγμα:** Βρείτε τη μέση έκπτωση των πελατών
Η ερώτηση **`select avg(discnt) from customers;`**
δε θα συμπεριλάβει την πλειάδα με **`cid='c007'`**
 - Κενές τιμές μπορούν να επιστραφούν σαν το αποτέλεσμα συναρτήσεων συνάθροισης αν αυτές υπολογιστούν πάνω στο κενό σύνολο:
 - Οι συναρτήσεις **`avg`**, **`sum`**, **`max`**, **`min`** επιστρέφουν **`null`** για το κενό σύνολο. Η συνάρτηση **`count`** επιστρέφει **`0`**.

SQL: Ομαδοποίηση πλειάδων

- Ομαδοποίηση πλειάδων: `Orders(orderno, month,cid,aid,pid,qty,amt)`
 - Παρέχεται η δυνατότητα ομαδοποίησης των πλειάδων που αποτελούν την απάντηση σε μια ερώτηση σύμφωνα με τις κοινές τιμές κάποιων γνωρισμάτων.
 - Μπορούν επίσης να εφαρμοστούν συναρτήσεις συνάθροισης στις ομαδοποιημένες πλειάδες.
 - Παράδειγμα: η ερώτηση
**`select pid, sum(qty) from orders
group by pid;`**
θα επιστρέψει τα διακριτά pids μαζί με τη συνολική ποσότητα για την οποία έχουν γίνει παραγγελίες.

SQL: Ομαδοποίηση πλειάδων

Orders(orderno, month,cid,aid,pid,qty,amt)

- Όταν μια συνάρτηση συνάθροισης εμφανίζεται σε μια εντολή select η οποία περιέχει group-by, η συνάρτηση εφαρμόζεται σε όλες τις πλειάδες μιας ομάδας (δηλαδή όλες τις πλειάδες οι οποίες έχουν την ίδια τιμή στα γνωρίσματα για τα οποία γίνεται η ομαδοποίηση) και επιστρέφεται μια τιμή για κάθε ομάδα.
- Όλα τα γνωρίσματα τα οποία επιστρέφονται ως απάντηση πρέπει να έχουν μοναδική τιμή για κάθε συνδυασμό τιμών των γνωρισμάτων σύμφωνα με τα οποία γίνεται η ομαδοποίηση.
- Παράδειγμα: η ερώτηση
`select pid, cid, sum(qty) from orders
group by pid;`
είναι λανθασμένη γιατί για ένα προϊόν δίνονται παραγγελίες από έναν ή περισσότερους πελάτες.

SQL: Ομαδοποίηση πλειάδων

Orders(orderno, month,cid,aid,pid,qty,amt)

- Η ομαδοποίηση μπορεί να γίνεται με περισσότερα από ένα γνωρίσματα
- **Παράδειγμα:** Υπολογίστε τη συνολική ποσότητα που παραγγέλνεται για κάθε προϊόν από κάθε πράκτορα.

```
select pid, aid, sum(qty) as TOTAL from orders  
group by pid, aid;
```

pid	aid	TOTAL
p01	a01	3000
p01	a06	1800
p02	a02	400
p03	a03	1000
p03	a05	800

SQL: Ομαδοποίηση πλειάδων

Orders(orderno, month,cid,aid,pid,qty,amt)

Agents(aid, aname,city,percent)

Customers(cid, cname,city,discont)

- Ομάδες πλειάδων μπορούν να σχηματιστούν με συνδυασμό σχέσεων
- **Παράδειγμα:** Βρείτε τα ονόματα και ids πρακτόρων, τα ονόματα και ids προϊόντων και τη συνολική ποσότητα που παραγγέλνει κάθε πράκτορας για τους πελάτες c02 και c03.

```
select aname, a.aid, pname, p.pid, sum(qty)
as TOTAL from orders o, products p, agents a
where o.pid=p.pid and o.aid=a.aid and o.cid
in ('c02','c03') group by a.aid, aname,
p.pid, pname;
```

aname	aid	pname	pid	TOTAL
Brown	a03	pencil	p05	2400
Brown	a03	razor	p03	1000

SQL: Ομαδοποίηση πλειάδων

- Υπολογισμός ερωτήσεων που περιέχουν `group-by`
 1. Υπολογίζεται το Καρτεσιανό γινόμενο των σχέσεων στο `from`
 2. Οι πλειάδες που δεν ικανοποιούν τις συνθήκες στο `where` αφαιρούνται
 3. Οι υπόλοιπες πλειάδες ομαδοποιούνται σύμφωνα με το `group-by`
 4. Υπολογίζονται οι εκφράσεις που επιστρέφονται ως απάντηση

- Παράδειγμα: η ερώτηση

```
select pid, sum(qty) from orders where  
sum(qty) >1000 group by pid;
```

είναι λανθασμένη γιατί η συνθήκη στο `where` δε μπορεί να υπολογιστεί πριν γίνει η ομαδοποίηση των πλειάδων.

SQL: having

- Συνθήκες πάνω στις ομάδες των πλειάδων μπορούν να εκφραστούν με το `having`. Ο υπολογισμός αυτής της συνθήκης γίνεται μετά την ομαδοποίηση.
 - **Παράδειγμα:** Βρείτε τα ids των προϊόντων και τη συνολική ποσότητα που έχει παραγγελθεί, όταν η ποσότητα αυτή είναι μεγαλύτερη από 1000.

```
select pid, sum(qty) from orders  
group by pid  
having sum(qty)>1000;
```

- Αν δεν υπάρχει `group-by` αλλά υπάρχει `having`, τότε το σύνολο των πλειάδων θεωρείται ως μια ομάδα.

SQL: having

- Οι συνθήκες που εκφράζονται στο **having** μπορούν να περιλαμβάνουν μόνο γνωρίσματα τα οποία έχουν μοναδική τιμή για κάθε ομάδα.
 - **Παράδειγμα:** Βρείτε τα ids προϊόντων που έχουν παραγγελθεί από τουλάχιστον δύο πελάτες
**select pid from orders
group by pid
having count(distinct cid) >= 2;**
 - Το γνώρισμα **cid**, ως έχει, δε μπορεί να συμμετέχει σε συνθήκη του **having** (δηλαδή η συνθήκη **having cid=3;** είναι λανθασμένη) γιατί για ένα προϊόν δίνονται παραγγελίες από έναν ή περισσότερους πελάτες.
 - Για τον ίδιο λόγο, το γνώρισμα **cid** δεν μπορεί να επιστρέφεται από την ερώτηση.

Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης**» 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.



Σημειώματα

Σημείωμα αδειοδότησης

•Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0 Διεθνές [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

•Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

•Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Κρήτης, Δημήτρης Πλεξουσάκης. «**Αρχεία και Βάσεις Δεδομένων. Διάλεξη 10η: SQL Μέρος 3ο**». Έκδοση: 1.0. Ηράκλειο/Ρέθυμνο 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://www.csd.uoc.gr/~hy360/>