SQL: Αιτήματα
(*Συνέχεια...*)

Κεφάλαιο 5

Ενημέρωση: 23/12/2008

Τελεστές Συνάθροισης

- Σημαντική προέκταση της σχεσιακής άλγεβρας.

```
COUNT (*)
COUNT ( [DISTINCT] A)
SUM ( [DISTINCT] A)
AVG ( [DISTINCT] A)
MAX (A)
MIN (A)
```

μονό πεδίο

Τελεστές Συνάθροισης

Υπολογισμός πλήθους ναυτών

```
SELECT  COUNT(*) FROM  Sailors S
```

Εύρεση μέσου όρου ηλικίας ναυτών με τιμή διατίμησης (rating) 10

```
SELECT  AVG(S.age)
FROM  Sailors S
WHERE  S.rating=10
```

Υπολογισμός πλήθους διακριτών τιμών διατίμησης για ναύτες με όνομα Yiannis

```
SELECT  COUNT(DISTINCT S.rating)
FROM  Sailors S
WHERE S.sname='Yiannis'
```

Τελεστές Συνάθροισης

Εύρεση ονομάτων ναυτικών με τη μέγιστη τιμή διατίμησης

```
SELECT  S.sname
FROM  Sailors S
WHERE  S.rating= (SELECT  MAX(S2.rating)
                  FROM  Sailors S2)
```

Υπολογισμός του μέσου όρου των διακριτών τιμών ηλικίας για ναύτες με τιμή διατίμησης 10

```
SELECT  AVG(DISTINCT S.age)
FROM  Sailors S
WHERE  S.rating=10
```

Η Διαίρεση (1 από 2)

Ναυτικοί οι οποίοι έχουν κάνει κράτηση σε ΟΛΑ τα σκάφη

```
SELECT  S.sname
FROM  Sailors S
WHERE  NOT EXISTS
          ((SELECT  B.bid
            FROM  Boats B)
           EXCEPT
            (SELECT  R.bid
             FROM  Reserves R
             WHERE  R.sid=S.sid))
```

Η Διαίρεση (2 από 2)

Ναυτικοί οι οποίοι έχουν κάνει κράτηση σε ΟΛΑ τα σκάφη

```
SELECT  S.sname                                  ....Ναυτικοί για τους οποίους....
FROM  Sailors S
WHERE  NOT EXISTS  (SELECT  B.bid
                    FROM  Boats B                ....δεν υπάρχει σκάφος που....
                    WHERE  NOT EXISTS  (SELECT  R.bid
                                        FROM  Reserves R
                                        WHERE  R.bid=B.bid
                                          AND R.sid=S.sid))
```

....να μην το έχουν κάνει κράτηση....

Εύρεση ονόματος και ηλικίας του αρχαιότερου ναύτη

1) SELECT S.sname, MAX(S.age)
FROM Sailors S

Πρόβλημα σύνταξης! (Θα εξηγήσουμε γιατί, όταν ασχοληθούμε με την GROUP BY.)

---

2) SELECT S.sname, S.age
FROM Sailors S
WHERE S.age =
(SELECT MAX(S2.age)
FROM Sailors S2)

---

3) SELECT S.sname, S.age
FROM Sailors S
WHERE (SELECT MAX(S2.age)
FROM Sailors S2)
= S.age

Το τρίτο είναι ισοδύναμο με το δεύτερο και προβλέπεται από το πρότυπο SQL/92, αλλά δεν υποστηρίζεται από όλα τα συστήματα.

GROUP BY **και** *HAVING*

- Μέχρι στιγμής εφαρμόσαμε τους τελεστές συνάθροισης σε όλες τις επιστρεφόμενες πλειάδες. Μερικές φορές είναι χρήσιμη η εφαρμογή τους σε ομάδες συνόλων πλειάδων.
- Π.χ.: *Η ηλικία του νεαρότερου ναύτη σε κάθε επίπεδο διατίμησης.*
  - Δεν είναι γνωστό ούτε το πλήθος των επιπέδων, ούτε οι τιμές διατίμησης κάθε επιπέδου!
  - Ας υποθέσουμε ότι το πεδίο των τιμών διατίμησης ειναι από 1 έως 10. Άρα χρειαζόμαστε 10 αιτήματα:

For $i = 1, 2, \ldots, 10$:

```
SELECT  MIN(S.age)
FROM  Sailors S
WHERE  S.rating = i
```

Αιτήματα με GROUP BY και HAVING

SELECT	[DISTINCT] *λίστα-επιλογής*
FROM	*λίστα-σχέσεων*
WHERE	*καταλληλότητα*
GROUP BY	*λίστα-ομαδοποίησης*
HAVING	*καταλληλότητα-ομάδας*

- Η λίστα-επιλογής περιέχει (i) ονομασίες γνωρισμάτων (ii) όρους με τελεστές συνάθροισης (π.χ. MIN(S.age)).
  - Οι ονομασίες γνωρισμάτων (i) πρέπει να είναι υποσύνολο της λίστας-ομαδοποίησης. Διαισθητικά, κάθε επιστρεφόμενη πλειάδα αντιστοιχεί σε μια ομάδα και τα γνωρίσματά της πρέπει να είναι μοναδικά για την ομάδα. (Μια ομάδα είναι ένα σύνολο πλειάδων με την ίδια τιμή για κάθε γνώρισμα της λίστας-ομαδοποίησης).

Ιδεατή Στρατηγική Υπολογισμού

- Υπολογίζεται το καρτεσιανό γινόμενο της λίστας-σχέσεων, οι ακατάλληλες πλειάδες απορρίπτονται, τα περιττά πεδία διαγράφονται και οι εναπομένουσες πλειάδες χωρίζονται σε ομάδες ανάλογα με τις τιμές των γνωρισμάτων της λίστας-ομαδοποίησης.
- Η καταλληλότητα ομάδας εφαρμόζεται στη συνέχεια για τη διαγραφή ομάδων. Οι παραστάσεις στην καταλληλότητα ομάδας πρέπει να έχουν μοναδική τιμή ανά ομάδα!
  - Κατά συνέπεια, ένα γνώρισμα το οποίο αναφέρεται στην καταλληλότητα-ομάδας οφείλει επίσης να εμφανίζεται στη λίστα-ομαδοποίησης ή να αποτελεί ονοματισμό του αποτελέσματος ενός τελεστή συνάθροισης.
- Μια πλειάδα επιστρέφεται για κάθε κατάλληλη ομάδα.

Να βρεθεί η ηλικία του νεότερου ενήλικα ναυτικού για κάθε τιμή διατίμησης με τουλάχιστο 2 ενήλικες ναυτικούς

```
SELECT  S.rating,  MIN(S.age)
FROM  Sailors S
WHERE  S.age >= 18
GROUP BY  S.rating
HAVING  COUNT(*) > 1
```

- Μόνο τα S.rating και S.age αναφέρονται στις συνιστώσες SELECT, GROUP BY or HAVING. Περιττά τα άλλα γνωρίσματα.
- Η δεύτερη στήλη μπορεί να ονοματιστεί χρησιμοποιώντας το AS.

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

rating	age
1	33.0
7	45.0
7	35.0
8	55.5
10	35.0

rating	
7	35.0

Σχέση αποτέλεσμα

Για κάθε κόκκινη βάρκα να βρεθεί ο συνολικός αριθμός κρατήσεών της

```
SELECT  B.bid,  COUNT(*) AS scount
FROM  Sailors S, Boats B, Reserves R
WHERE  S.sid=R.sid AND R.bid=B.bid AND B.color='red'
GROUP BY  B.bid
```

- Ομαδοποίηση επί σύζευξης τριών σχέσεων.
- Τι θα συμβεί αν αφαιρέσουμε τη B.color='red' από τη συνιστώσα WHERE και προσθέσουμε μια συνιστώσα HAVING με την ίδια συνθήκη;
- Τι θα συμβεί αν αφαιρέσουμε τη Sailors και τη συνθήκη του S.sid;

Να βρεθεί η ηλικία του νεότερου ενήλικα ναυτικού για κάθε τιμή διατίμησης με τουλάχιστο 2 ναυτικούς

```
SELECT  S.rating,  MIN(S.age)
FROM  Sailors S
WHERE  S.age > 18
GROUP BY  S.rating
HAVING  1  <  (SELECT  COUNT(*)
               FROM  Sailors S2
               WHERE  S.rating=S2.rating)
```

- Η HAVING μπορεί να περιέχει υποαίτημα.
- Σύγκριση αυτού του αιτήματος και του αιτήματος με τους δύο ενήλικες ναύτες!
- Τι θα συμβεί αν η HAVING αντικατασταθεί με:
  - HAVING COUNT(*) >1

Να βρεθούν οι τιμές διατίμησης με τη νεότερη μέση τιμής ηλικίας ναύτη

Οι λειτουργίες συνάθροισης δεν εμφωλεύονται! ΛΑΘΟΣ:

```
SELECT S.rating
FROM Sailors S
WHERE S.age = (SELECT MIN(AVG(S2.age)) FROM Sailors S2)
```

Σωστή λύση (στην SQL/92):

```
SELECT Temp.rating, Temp.avgage
FROM (SELECT S.rating, AVG(S.age) AS avgage
      FROM Sailors S
      GROUP BY S.rating) AS Temp
WHERE Temp.avgage = (SELECT MIN(Temp.avgage)
                     FROM Temp)
```

Τιμές NULL

- Οι τιμές των πεδίων των στηλών μπορεί να είναι άγνωστες (π.χ. η τιμή κάποιας διατίμισης δεν έχει οριστεί) ή ανύπαρκτες (π.χ. όνομα συζύγου).
  - Η SQL προβλέπει την ειδική τιμή null γι' αυτές τις περιπτώσεις.
- Περιπλέκονται κάποια ζητήματα:
  - Ειδικοί τελεστές ελέγχου τιμών null και not null.
  - Η rating>8 είναι αληθής ή ψευδής για null τιμές του rating; Τι συμβαίνει τότε με AND, OR και NOT;
  - Χρειαζόμαστε λογική τριών τιμών (true, false και άγνωστη).
  - Προσεκτικός ορισμός του νοήματος των δομικών στοιχείων. (π.χ. συνιστώσα WHERE: απορρίπτει ό,τι δεν είναι true).
  - Νέοι τελεστές (συγκεκριμένα, στις εξωτερικές συζεύξεις).

Εξάσκηση

Θεωρήστε τους παρακάτω πίνακες

Student(snum: integer, sname: sting, major: string, level: string, age: integer)

Class(name: string, meets_at: time, room: string, fid: integer)

Enrolled(snum: string, cname: string)

Faculty(fid: integer, fname: string, deptid: integer)

Ο πίνακας Enrolled καταχωρεί μία γραμμή για κάθε ζεύγος φοιτητή-τάξης και αυτό αντιστοιχεί στην πληροφορία ότι ο συγκεκριμένος φοιτητής έχει εγγραφεί και παρακολουθεί τη συγκεκριμένη τάξη (μάθημα). Ο πίνακας Faculty αποθηκεύει πληροφορίες για το διδακτικό προσωπικό

Εξάσκηση

Να εμφανιστούν τα καταχωρημένα εκπαιδευτικά επίπεδα (level) φοιτητών και δίπλα στην κάθε τιμή επιπέδου η μέση τιμή ηλικίας των φοιτητών οι οποίοι ανήκουν σε αυτό, εκτός από το επίπεδο level='JR'

```
SELECT S.level, AVG(S.age)
FROM  Student S
WHERE S.level <> 'JR'
GROUP BY S.level
```

Εξάσκηση

Να βρεθούν τα ονόματα όλων των τάξεων οι οποίες είτε διδάσκονται στην αίθουσα room='R128', είτε έχουν εγγραφεί σε αυτές για να τις παρακολουθούν τουλάχιστον πέντε φοιτητές

```
SELECT  C.name
FROM Class C
WHERE C.room='R128'
        OR C.name IN (SELECT E.cname
                      FROM Enrolled E
                      GROUP BY E.cname
                      HAVING COUNT(*) > 5)
```

Εξάσκηση

Να βρεθούν τα ονόματα των μελών του εκπαιδευτικού προσωπικού ο κάθε ένας από τους οποίους διδάσκει (σε όλες τις τάξεις του) σε ακροατήριο μικρότερο των πέντε φοιτητών

```
SELECT DISTINCT F.fname
FROM Faculty F
WHERE 5 > ALL (SELECT COUNT(*)
               FROM Class C, Enrolled E
               WHERE C.name = E.cname
               AND C.fid = F.fid
               GROUP BY E.cname)
```

Περίληψη

- Η SQL αποτέλεσε σημαντικό παράγοντα για την αποδοχη του σχεσιακού μοντέλου. Πιο φυσική από προηγούμενες, διαδικαστικές γλώσσες αιτημάτων.
- Χαρακτηρίζεται από σχεσιακή πληρότητα. Έχει περισσότερες δυνατότητες από τη σχεσιακή άλγεβρα.
- Αιτήματα που εκφράζονται σε ΣΑ εκφράζονται με φυσικότερο τρόπο σε SQL.
- Εναλλακτικοί τρόποι διατύπωσης αιτημάτων. Ο βελτιστοποιητής ανακαλύπτει τον αποδοτικότερο.
  - Στην πράξη ωστόσο οι χρήστες πρέπει να έχουν γνώση των σχεδίων βελτιστοποίησης.