

Ενδεικτικές Λύσεις 2^{ου} Συνόλου Ασκήσεων

Ακαδημαϊκό Έτος 2021-2022

Απορίες/σχόλια στο μάθημα της Δευτέρας 10/1/2022

Άσκηση 1 – σχεσιακή άλγεβρα

(α) Τις επιχειρήσεις (το id και το όνομα τους) που έχουν περισσότερες από 30 αξιολογήσεις (σε αυτό το ερώτημα χρησιμοποιείτε το πεδίο review_count του πίνακα business), μέσο αριθμό αστεριών μεγαλύτερο του 4 (σε αυτό το ερώτημα χρησιμοποιείτε το πεδίο stars του πίνακα business), και όλες τους οι αξιολογήσεις του στον πίνακα review έχουν τουλάχιστον 3 αστέρια.

$Q \leftarrow \pi_{id} \left(\sigma_{\text{review_count} > 30 \text{ and stars} > 4} (\text{business}) \right) -$

$\pi_{\text{business_id}} \left(\sigma_{\text{stars} < 3} (\text{review}) \right)$

Προσοχή στις προβολές

Για να έχουμε και το όνομα τους

$\pi_{id, name} (Q * \text{business})$

↑ φυσική ένωση

(β) Ζεύγη φίλων (ζεύγη από `user_id`) που έχουν αξιολογήσει τουλάχιστον μια ίδια επιχείρηση. Αν εμφανίζεται το ζεύγος (`id1`, `id2`) να μην εμφανίζεται το συμμετρικό του (`id2`, `id1`) και επίσης να μην εμφανίζονται ταυτοτικά ζεύγη δηλαδή ζεύγη (`id1`, `id2`) όπου $id2 = id1$.

Σε αυτό και σε κάποια από τα επόμενα ερωτήματα έχει σημασία το είδος της σχέσης `friend`.

Θα λέμε ότι είναι συμμετρική αν για κάθε πλειάδα (`id1`, `id2`) που υπάρχει στο `friend` υπάρχει πάντα και η συμμετρική πλειάδα (`id2`, `id1`).

Αλλιώς, θα λέμε ότι δεν είναι συμμετρική, δηλαδή μπορεί να υπάρχει η (`id2`, `id1`) ή να μην υπάρχει.

(β) Ζεύγη φίλων (ζεύγη από user_id) που έχουν αξιολογήσει τουλάχιστον μια ίδια επιχείρηση. Αν εμφανίζεται το ζεύγος (id1, id2) να μην εμφανίζεται το συμμετρικό του (id2, id1) και επίσης να μην εμφανίζονται ταυτοτικά ζεύγη δηλαδή ζεύγη (id1, id2) όπου id2 = id1. *συμμετρική ή όχι η σχέση της φίλιας?*

review1 () ←
 review2 () ←

$Q(u_1, u_2) \leftarrow \pi \left(\left(\text{review1} \bowtie \text{review2} \right) \bowtie \text{friend} \right)$
 business_id1 = business_id2
 user_id1 = user_id2
 and friend-user = user_id2

(αν συμμετρική, αρκεί η παρουσία συνδέου)

(αν όχι) and user_id1 < user_id2

$R \leftarrow \pi \left(Q(u_1, u_2) \bowtie Q(u_3, u_4) \right)$
 $u_1 = u_4 \text{ and } u_2 = u_3$

• το R έχει αυτά που έχουν και (1,2) και (2,1), δηλαδή περιέχει συμμετρικά ζεύγη

$Q - \sigma_{u_1 > u_2} (R)$

(γ) Τις επιχειρήσεις (το id τους) που έχουν δεχθεί και θετικές αξιολογήσεις (δηλαδή, αξιολογήσεις με 4 ή 5 αστέρια) και αρνητικές αξιολογήσεις (δηλαδή, αξιολογήσεις με 1 ή 2 αστέρια).

\cap business-id $(\sigma_{stars=5 \vee stars=4} (review)) \cap$
 $\vee stars \geq 4$

\cap business-id $(\sigma_{stars=1 \vee stars=2} (review))$

(δ) Για το χρήστη με id = 59181, τις επιχειρήσεις (το id τους) που αυτός έχει αξιολογήσει με 5 αστέρια και κάθε φίλος του που τις έχει αξιολογήσει τους έχει δώσει λιγότερα από 3 αστέρια.

B(b)

$\leftarrow \pi_{\text{business-id}} \left(\sigma_{\text{review}} \left(\text{user-id} = 59181 \text{ and stars} = 5 \right) \right)$

B: επιχειρήσεις που
έχω αξιολογήσει
ο 59181

$\rightarrow F(u)$
 $\pi_{\text{friend-user-id}}$

$\left(\sigma_{\text{friend}} \left(\text{user-id} \neq 59181 \right) \right) \cup$

$\pi_{\text{user-id}} \left(\sigma_{\text{friend}} \left(\text{friend-user-id} = 59181 \right) \right)$

F: φίλοι

του 59181

$F' \leftarrow \pi_{\text{business-id}} \left(F \bowtie_{\text{review}} \right)$
 $u = \text{user-id}$
and stars ≥ 3

F': επιχειρήσεις
που φίλοι του 59181
έχουν αξιολογήσει με
3 ή περισσότερα
αστέρια

B - F'

(ε) Τη μικρότερη ημερομηνία που κάποιος χρήστης εγγράφηκε στο σύστημα.

$$R \leftarrow \pi_{\text{yelping_since}} \left(\text{user 1} \bowtie \text{user 2} \right)$$
$$\text{yelping_since 1} > \text{yelping_since 2}$$

$$\pi_{\text{yelping_since}}(\text{user}) - R$$

Άσκηση 2 – σχεσιακός λογισμός

(α) Τις επιχειρήσεις (το id και το όνομα τους) που έχουν περισσότερες από 30 αξιολογήσεις (σε αυτό το ερώτημα χρησιμοποιείτε το πεδίο review_count του πίνακα business), μέσο αριθμό αστεριών μεγαλύτερο του 4 (σε αυτό το ερώτημα χρησιμοποιείτε το πεδίο stars του πίνακα business), και όλες τους οι αξιολογήσεις του στον πίνακα review έχουν τουλάχιστον 3 αστέρια.

$$\left\{ t.id, t.name \mid business(t) \text{ and } t.review_count > 30 \text{ and } t.stars > 4 \text{ and } (\text{not } (\exists r) (review(r) \text{ and } t.id = r.business_id \text{ and } r.stars < 3)) \right\}$$

(β) Για το χρήστη με id = 59181, τις επιχειρήσεις (το id τους) που αυτός έχει αξιολογήσει με 5 αστέρια και κάθε φίλος του που τις έχει αξιολογήσει τους έχει δώσει λιγότερα από 3 αστέρια. Δώστε δύο διαφορετικές ερωτήσεις μία με χρήση του υπαρξιακού και μία με χρήση του καθολικού ποσοδείκτη.

$\{ t.business_id \mid review(t) \text{ and } t.user_id = 59181 \text{ and } t.stars = 5$
 $\text{and not } ((\exists r) (review(r) \text{ and } r.stars < 3 \text{ and}$
 $(\exists d) (friend(d) \text{ and } (d.user_id = r.user_id$
 $\text{and } d.friend_user_id = 59181) \text{ or}$
 $(d.user_id = 59181 \text{ and } d.friend_user_id = r.user_id))) \}$

$\{ t.business_id \mid review(t) \text{ and } t.user_id = 59181 \text{ and } t.stars = 5$
 $\text{and } ((\forall r) (not review(r) \text{ or } r.stars > 3 \text{ or}$
 $not (\exists d) \text{ όπως πριν}$

Άσκηση 3 – SQL

(β) – (i) Τις επιχειρήσεις (το id και το όνομα τους) που έχουν περισσότερες από 30 αξιολογήσεις (σε αυτό το ερώτημα χρησιμοποιείτε το πεδίο review_count του πίνακα business), μέσο αριθμό αστεριών μεγαλύτερο του 4 (σε αυτό το ερώτημα χρησιμοποιείτε το πεδίο stars του πίνακα business), και όλες τους οι αξιολογήσεις του στον πίνακα review έχουν τουλάχιστον 3 αστέρια.

```
SELECT id, name
FROM business
WHERE review-count > 30 AND stars > 4 and
      id NOT IN (SELECT business.id
                 FROM review
                 WHERE stars < 3)
```

β – (ii) Για κάθε επιχείρηση (το id της) που έχει λάβει τουλάχιστον 3 αξιολογήσεις με 5 ή 4 αστέρια, το πλήθος (αθροιστικά) αυτών των αξιολογήσεων.

```
SELECT business_id, count(id)
FROM review
WHERE stars = 4 OR stars = 5
GROUP BY business_id
HAVING count(id) >= 3
```

β - (iii) Το ποσοστό των επιχειρήσεων για τις οποίες οι θετικές αξιολογήσεις (αξιολογήσεις με 4 ή 5 αστέρια) είναι περισσότερες από τις αρνητικές αξιολογήσεις (αξιολογήσεις με 1 ή 2 αστέρια).

```
SELECT count(DISTINCT posrev.business_id)/buscnt * 100
```

```
FROM (SELECT business_id, count(id) as poscnt FROM review WHERE stars = 5 OR stars = 4 GROUP BY business_id) as posrev, SELECT business_id, count(id) as negcnt FROM review WHERE stars = 1 OR stars = 2 GROUP BY business_id) AS negrev, SELECT count(*) as buscnt FROM business
```

```
WHERE posrev.business_id = negrev.business_id AND posrev.poscnt > negrev.negcnt;
```

ή με WITH

```
WITH posrev(posbus, poscnt) as SELECT business_id, count(id) FROM review WHERE stars = 5 OR stars = 4 GROUP BY business_id), negrev(negbus, negcnt) AS SELECT business_id, count(id) FROM review WHERE stars = 5 OR stars = 4 GROUP BY business_id), busrev(buscnt) as SELECT count(*) as buscnt FROM business
```

```
SELECT count(DISTINCT posrev.business_id)/busrev.buscnt * 100
```

```
FROM posrev, negrev, busrev
```

```
WHERE posrev.posbus = negrev.negbus_id AND posrev.poscnt > negrev.negcnt;
```

β - (iv) Για κάθε επιχείρηση (το id της), το μέσο αριθμό αστεριών των αξιολογήσεων που έχει δεχθεί λαμβάνοντας υπόψη μόνο τις αξιολογήσεις που έχουν τουλάχιστον 2 ψήφους ως χρήσιμες. Για τις επιχειρήσεις, για τις οποίες δεν υπάρχουν τέτοιες αξιολογήσεις να τυπώνεται ο αριθμός 0.

```
(SELECT business_id, AVG(stars)
FROM review
WHERE useful >= 2
GROUP BY business_id)
UNION
(SELECT id, 0
FROM business
WHERE id NOT IN (SELECT business_id FROM review where useful >= 2))
```


β - (ν) Το πλήθος των χρηστών που έχουν δώσει περισσότερες αξιολογήσεις με 5 αστέρια από ό,τι αθροιστικά οι φίλοι τους.

```
SELECT COUNT(DISTINCT frel.user-id)
```

```
FROM
```

```
(SELECT user_id, COUNT(id) as cnt FROM review WHERE stars = 5 GROUP BY user_id) as urel,
```

```
(SELECT U.user_id, COUNT(id) as cnt FROM user as U, friend as F, review as R
```

```
WHERE F.user_id = U.user_id AND R.user_id = F.friend_user_id AND stars = 5
```

```
GROUP BY U.user_id) as frel
```

```
WHERE frel.user_id = urel.user_id AND urel.cnt > frel.cnt;
```

* όπως και πριν τε κτλ

* το παραπάνω δεν εξετάζει την περίπτωση που οι φίλοι δεν έχουν αξιολογήσει με 5 αστέρια οπότε δίνει για να κάλυψει και αυτήν την περίπτωση ένα UNION

(γ) (i) Εισάγετε την τιμή “MA” στο πεδίο state του πίνακα business.

```
UPDATE business SET state = 'MA';
```

ή

```
INSERT INTO business(state) VALUE('MA');
```

(γ) (ii) Εισάγετε ένα νέο χρήστη στον πίνακα user. Στη συνέχεια εισάγετε μια αξιολόγηση αυτού του χρήστη με 3 αστέρια για κάθε επιχείρηση που είναι ανοικτή.

```
INSERT INTO user VALUES(xid, ....)
```

```
INSERT INTO review  
SELECT xid, id, 3, 0, 0, 0, "", null  
FROM business  
WHERE is_open = 1;
```

Εδώ οποιοδήποτε συμβατικό τιμές

