

2^ο Σύνολο Ασκήσεων

Καταληκτική Ημερομηνία Παράδοσης: 29 Νοεμβρίου, πριν το μάθημα

Θεματική Ενότητα: Γλώσσες ερωτήσεων (Σχεσιακή Άλγεβρα, Σχεσιακός Λογισμός, SQL)

Για γενικές οδηγίες σχετικά με τις ασκήσεις, συμβουλευτείτε την ιστοσελίδα του μαθήματος. Εξηγήστε τις απαντήσεις σας με σαφήνεια. Αν κάνετε οποιαδήποτε υπόθεση, καταγράψτε την.

Άσκηση 1

(α) Έστω δυο σχέσεις (σχήματα σχέσεων) $R(A, B)$ και $S(A, B)$. Το γνώρισμα A είναι κλειδί και για τις δύο σχέσεις, ενώ το B δεν είναι κλειδί για καμία από αυτές. Κανένα από τα γνωρίσματα δεν επιτρέπεται να πάρει την τιμή null. Για τα παρακάτω ζεύγη ερωτήσεων δείξτε αν είναι η όχι ισοδύναμα, αν είναι επιχειρηματολογήστε για αυτό, αν όχι δώστε ένα αντιπαράδειγμα. Δυο ερωτήσεις θεωρούνται ισοδύναμες όταν επιστρέφουν το ίδιο αποτέλεσμα σε όλες τις περιπτώσεις.

- | | |
|---|---|
| (1) $\pi_A(R - S)$ | $\pi_A(R) - \pi_A(S)$ |
| (2) $\pi_B(R - S)$ | $\pi_B(R) - \pi_B(S)$ |
| (3) $\pi_B(R \cup S)$ | $\pi_B(R) \cup \pi_B(S)$ |
| (4) $\pi_{R.A, S.A}(R \times S)$ | $\pi_A(R) \times \pi_A(S)$ |
| (5) select B from R
where A in (select A from S) | select R.B from R, S where R.A = S.A |
| (6) select B from R
where A not in (select A from S) | select R.B from R, S where R.A <> S.A |
| (7) select A from R | select A from R group by A |
| (8) select B from R | select B from R group by B |
| (9) select A from S
where not A >= any (select A from S) | select A from S
where A < (select max(A) from S) |

(β) Έστω δυο σχέσεις $R(A, B)$ που περιέχει r πλειάδες και $S(B, C)$ που περιέχει s πλειάδες. Για τις παρακάτω εκφράσεις της σχεσιακής άλγεβρας δώστε το μικρότερο και μεγαλύτερο δυνατό αριθμό πλειάδων.

- $\pi_{A,C}(R \times S)$
- $\pi_B(R) - (\pi_B(R) - \pi_B(S))$
- $(R \times R) \times R$
- $\sigma_{A > B}(R) \cup \sigma_{A < B}(R)$

Άσκηση 2

Θεωρείστε το παρακάτω σχεσιακό σχήμα για μια βάση δεδομένων με ταινίες:

MOVIES(mid, title, year)
COUNTRIES(mid, country)
GENRES(mid, genre)
USER(uid, age, nationality)
RATINGS(mid, uid, rating, timestamp)

Η σχέση MOVIES έχει πληροφορίες για ταινίες, συγκεκριμένα: ένα μοναδικό κωδικό (mid), τον τίτλο (title) και το έτος (year) της ταινίας. Η σχέση COUNTRIES έχει τις χώρες (country) παραγωγής της ταινίας mid, ενώ η σχέση GENRES τις κατηγορίες (genre), π.χ., δράμα, περιπέτεια κλπ, της ταινίας mid. Η σχέση USER έχει πληροφορίες για χρήστες, συγκεκριμένα: έναν μοναδικό κωδικό (uid), την ηλικία (age) και την εθνικότητα (nationality) του χρήστη. Τέλος, η σχέση RATINGS έχει την αξιολόγηση του χρήστη uid για την ταινία mid,

συγκεκριμένα: την ημερομηνία (timestamp) που αυτή έγινε και ένα βαθμό αξιολόγησης (rating) που είναι ένας ακέραιος από το 1 έως το 5.

(α) Δώστε ένα κατάλληλο μοντέλο Οντοτήτων/Συσχετίσεων για αυτή τη βάση δεδομένων.

- (β) Διατυπώστε σε σχεσιακή άλγεβρα τις ερωτήσεις (χωρίς χρήση συναθροιστικών συναρτήσεων) που δίνουν:
- E1. Όλες τις Ελληνικές δραματικές ταινίες (το γνώρισμα mid).
 - E2. Όλους τους χρήστες που έχουν αξιολογήσει μόνο Ελληνικές ταινίες (το γνώρισμα uid).
 - E3. Τους χρήστες που έχουν βαθμολογήσει ταινίες όλων των χωρών (το γνώρισμα uid).
 - E4. Για κάθε χρήστη την ταινία στην οποία έχει δώσει το μεγαλύτερο βαθμό αξιολόγησης από αυτές που έχει αξιολογήσει (τα ζεύγη uid, mid). Σε περίπτωση ισοβαθμίας, δώστε όλες τις ταινίες που ισοβαθμούν.

(γ) Διατυπώστε σε σχεσιακό λογισμό πλειάδων όλες τις ερωτήσεις του ερωτήματος (β). Για τις ερωτήσεις E2 και E3 δώστε δυο εκδοχές, μία με καθολικό (\forall) και μια με υπαρξιακό (\exists) ποσοδείκτη.

Άσκηση 3

Τα παρακάτω ερωτήματα αφορούν τη βάση δεδομένων με ταινίες της Άσκησης 2 και πρέπει να υλοποιηθούν στη MySQL.

(α) Δημιουργείστε τους 5 πίνακες. Ορίστε κατάλληλα πεδία ορισμού για τα γνωρίσματα τους, καθώς και τα κλειδιά, ξένα κλειδιά και όποιους άλλους περιορισμούς θεωρείστε απαραίτητους. Στη συνέχεια, κάντε εισαγωγή των δεδομένων που θα βρείτε στη σελίδα του μαθήματος. Για το σκοπό αυτό μπορείτε να χρησιμοποιήσετε την εντολή LOAD.

(β) Δώστε SQL ερωτήσεις που να δίνουν ως αποτέλεσμα τα παρακάτω.

- E1. Τους χρήστες που έχουν βαθμολογήσει ταινίες όλων των χωρών (το γνώρισμα uid).
- E2. Την ταινία (ταινίες) που έχει (έχουν) λάβει τις περισσότερες αξιολογήσεις. Σε περίπτωση ισοβαθμίας, δώστε όλες τις ταινίες σε φθίνουσα διάταξη με βάση το έτος. Δώστε και τα τρία γνωρίσματα (mid, title, year).
- E3. Την ταινία (ταινίες) με το μεγαλύτερο μέσο βαθμό αξιολόγησης από τα «νεανικά κοινά», δηλαδή από χρήστες ηλικίας από 15 έως 44. Σε περίπτωση ισοβαθμίας, δώστε όλες τις ταινίες σε διάταξη με βάση το mid. Δώστε τα γνωρίσματα mid και title.
- E4. Για κάθε βαθμό αξιολόγησης από το 1 έως το 5, δώστε τον αριθμό ταινιών που έχουν λάβει αυτό το βαθμό σε αύξουσα διάταξη με βάση το βαθμό.
- E5. Την ταινία (ταινίες) που έχει (έχουν) λάβει την πιο πρόσφατη αξιολόγηση (το γνώρισμα mid).

(γ) Κάντε τις παρακάτω τροποποιήσεις στα δεδομένα:

- T1. Για κάθε Ελληνική ταινία, εισάγετε μια πλειάδα στη σχέση RATINGS με κωδικό χρήστη uid = 1021, timestamp την τρέχουσα ημερομηνία και βαθμό ίσο με το μεγαλύτερο βαθμό από τις αξιολογήσεις που έχει λάβει, αν αυτή έχει λάβει τουλάχιστον μια αξιολόγηση και με βαθμό 3 αλλιώς.
- T2. Για κάθε ταινία διαγράψτε την εγγραφή της στο RATINGS με το μικρότερο βαθμό αξιολόγησης.
- T3. Αυξήστε το βαθμό αξιολόγησης κάθε κωμωδίας κατά 1 αν αυτός είναι μικρότερος του 5.

Άσκηση 4 [προαιρετικό ερώτημα, bonus + 10]

Θεωρείστε έναν πίνακα T(A) στην SQL με ένα μόνο γνώρισμα A με πεδίο ορισμού τους θετικούς ακέραιους και που πιθανόν περιέχει διπλότυπα. Στόχος είναι να γράψετε μια SQL ερώτηση που να δίνει ως αποτέλεσμα το μικρότερο θετικό ακέραιο που δεν υπάρχει στον πίνακα.

Υπόδειξη: Μπορείτε να προχωρήσετε σε δύο βήματα:

- (1) Λύστε το πρόβλημα υποθέτοντας ότι ο πίνακας περιέχει τουλάχιστον μια πλειάδα με την τιμή 1
 - (2) Λύστε το πρόβλημα υποθέτοντας ότι ο πίνακας μπορεί ή όχι να περιέχει μια πλειάδα με την τιμή 1
- Δοκιμάστε την ερώτησή σας στη MySQL.