

ΔΕΝ ΕΠΙΤΡΕΠΕΤΑΙ Η ΧΡΗΣΗ ΚΙΝΗΤΩΝ ΤΗΛΕΦΩΝΩΝ - ΚΑΛΗ ΕΠΙΤΥΧΙΑ!

ΟΝΟΜΑ ΦΟΙΤΗΤΗ: _____

1. Συναρτησιακές Εξαρτήσεις και Κανονικοποίηση (18 μονάδες)

Έστω η σχέση $R(A, B, C, D)$ για την οποία σας δίνεται ότι ισχύουν οι ακόλουθες συναρτησιακές εξαρτήσεις:
 $F = \{AB \rightarrow CD, C \rightarrow AD, D \rightarrow A\}$.

- 1) **(9 μονάδες)** Δείξτε ότι το F δεν είναι minimal cover. Βρείτε το minimal cover για το παραπάνω σύνολο συναρτησιακών εξαρτήσεων (βοήθεια: θα χρειαστεί να απλοποιήσετε 2 συναρτησιακές εξαρτήσεις – πρέπει να εξηγήσετε αναλυτικά γιατί είναι επιτρεπτές οι απλοποιήσεις αυτές).
- 2) **(9 μονάδες)** Εφαρμόστε τον αλγόριθμο σύνθεσης σε 3NF για να λάβετε μία χωρίς απώλειες αποσύνθεση της R σε ένα σύνολο από σχέσεις που είναι σε 3NF. Δείξτε αναλυτικά τη διαδικασία που ακολουθήσατε, δικαιολογώντας σε κάθε στάδιο/βήμα της διαδικασίας την επιλογή σας για την αποσύνθεση.
Η απάντησή σας πρέπει να περιλαμβάνει τους πίνακες, με τα χαρακτηριστικά τους και μία ένδειξη για τα κλειδιά κάθε πίνακα (με υπογράμμιση), καθώς και αιτιολόγηση για το πώς προέκυψε το κλειδί του κάθε πίνακα.

ΣΗΜΕΙΩΣΗ: Ζητείται συγκεκριμένος αλγόριθμος παραπάνω. Η εφαρμογή άλλου αλγόριθμου (πχ, αποσύνθεσης σε BCNF) οδηγεί αυτόματα σε μηδενισμό του συγκεκριμένου κομματιού.

2. Ανάνηψη με Undo Logging (12 μονάδες)

- Ποιος είναι ο αλγόριθμος για ανάνηψη όταν χρησιμοποιούμε για καταλογογράφηση τον αλγόριθμο του Undo Logging;
- Εξηγήστε γιατί ο συγκεκριμένος αλγόριθμος είναι σωστός.
- Αν υπάρξει πτώση του ρεύματος κατά τη διάρκεια της ανάνηψης, εξηγήστε αν θα μπορέσει να γίνει σωστά η ανάνηψη κατά τη 2^η προσπάθεια (όταν τελειώσει η διακοπή ρεύματος).

3. Βελτιστοποίηση Ερωτήσεων (16 μονάδες: 12+4)

Έστω οι σχέσεις **RelA(idA, name)** και **RelB(AMKA, refID, age)**

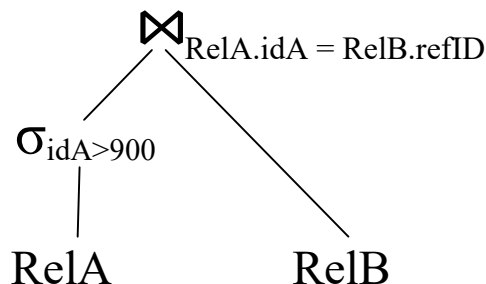
Έστω ότι το idA έχει 1000 διαφορετικές πιθανές τιμές (1, 2, 3, ..., 999, 1000).

Έστω ότι η RelA καταλαμβάνει 5000 σελίδες, με 80 tuples/σελίδα \Rightarrow 400,000 tuples

Έστω ότι η RelB καταλαμβάνει 500 σελίδες, με 100 tuples/σελίδα \Rightarrow 50,000 tuples

Έστω ότι υπάρχει συσταδοποιημένο ευρετήριο στο γνώρισμα refID της RelB.

Θεωρήστε το ακόλουθο ερώτημα:



(α) Υπολογίστε το κόστος σε IO για το παραπάνω ερώτημα όταν το join είναι index nested loop join. Είναι πολύ σημαντικό να εξηγήσετε πώς υπολογίσατε το κόστος αυτό. Θα χρειαστεί σίγουρα να εκτιμήσετε, για κάθε πλειάδα που βγαίνει από το selection, με πόσα tuples της RelB θα κάνει join.

(β) Ποιο είναι το κόστος σε IO αν επιπλέον υπάρχει μη συσταδοποιημένο ευρετήριο στο γνώρισμα idA της RelA.

4. ER διάγραμμα (20 μονάδες)

Κατασκευάστε ένα **διάγραμμα ER** το οποίο να αναπαριστά την πληροφορία που πρόκειται να αποθηκευτεί σε μια βάση δεδομένων για την ακόλουθη εφαρμογή μιας φαρμακευτικής εταιρείας:

- Κάθε ασθενής (patient) χαρακτηρίζεται από το ΑΦΜ του, και επίσης έχει όνομα (name) του και ηλικία (age).
- Κάθε ιατρός (doctor) χαρακτηρίζεται από το ΑΦΜ του, και επίσης έχει όνομα (name) του και ειδικότητα (specialty).
- Κάθε ασθενής έχει ακριβώς ένα κύριο ιατρό. Κάθε ιατρός είναι κύριος ιατρός για τουλάχιστον 1 ασθενή.
- Κάθε φαρμακευτική εταιρεία χαρακτηρίζεται από το όνομά της (name), και επίσης έχει ένα τηλέφωνο.
- Για κάθε φάρμακο καταγράφουμε το εμπορικό του όνομα (trade name) και τις οδηγίες χρήσης του. Κάθε φάρμακο πωλείται από μία συγκεκριμένη φαρμακευτική εταιρεία, και το εμπορικό όνομα χαρακτηρίζει μοναδικά το φάρμακο ανάμεσα στα προϊόντα της εταιρείας αυτής (και μόνο αυτής). Μία φαρμακευτική εταιρεία πουλάει 0 ή περισσότερα φάρμακα.
- Αν διαγράψουμε μία φαρμακευτική εταιρεία, δεν έχει νόημα να διατηρήσουμε πληροφορία για τα φάρμακα που αυτή πουλάει.
- Κάθε φαρμακείο χαρακτηρίζεται από το συνδυασμό της ονομασίας του (naming) και του ονόματος του ιδιοκτήτη του (ownerName), ενώ επίσης έχει και ένα αριθμό τετραγωνικών μέτρων (ως την επιφάνειά του).
- Κάθε φαρμακείο πουλάει 1 ή περισσότερα φάρμακα και έχει μία (διαφορετική) τιμή για το καθένα. Ένα φάρμακο μπορεί να πωλείται σε 0 ή πολλά φαρμακεία, και η τιμή του μπορεί να είναι διαφορετική σε κάθε φαρμακείο.
- Ιατροί συνταγογραφούν φάρμακα σε ασθενείς. Ένας ιατρός μπορεί να γράψει (ή όχι) ένα ή περισσότερα φάρμακα για αρκετούς ασθενείς, και σε ένα ασθενή μπορούν να υπάρχουν συνταγογραφήσεις φαρμάκων από πολλούς ιατρούς. Δε χρειάζεται να κρατήσουμε κάποια επιπλέον πληροφορία για κάθε συνταγογράφηση, παρά μόνο να καταγράφονται οι μοναδικοί συνδυασμοί <ιατρός, ασθενής, φάρμακο>.

5. SQL (34 μονάδες: 5 + 9 + 10 + 10)

Θεωρήστε το ακόλουθο σχεσιακό σχήμα που αφορά λέξεις που εμφανίζονται σε urls:

- Σε κάθε url μπορούν να εμφανίζονται πολλές λέξεις, οι οποίες να ανήκουν πιθανώς σε διαφορετικές γλώσσες.
- Μία λέξη στο λεξικό μπορεί να ανήκει σε μία μόνο γλώσσα.

Occurs(url, word)

Dictionary(language, word)

Δώστε τα ακόλουθα ερωτήματα σε SQL:

1. Τυπώστε όλα τα μοναδικά ζευγάρια <url,γλώσσα>, όπου θα εμφανίσετε για κάθε ιστοσελίδα (url) τις γλώσσες που εμφανίζονται σε αυτήν. Μία γλώσσα εμφανίζεται σε ένα url αν στο url περιέχεται έστω και μία λέξη αυτής της γλώσσας.
2. Βρείτε όλες τις γλώσσες που εμφανίζονται σε 5 ή περισσότερα url. Σκεφτείτε αν σας βοηθάει το προηγούμενο ερώτημα κάπως στην απάντησή σας.
3. Τυπώστε ανά url το μέγιστο αριθμό λέξεων που ανήκουν σε κάποια γλώσσα. Πχ, αν σε ένα url υπάρχουν 40 λέξεις στα Γερμανικά, 60 στα Αγγλικά και 20 στα Ελληνικά, θα πρέπει για αυτό το url να τυπωθεί μία εγγραφή <url, 60> (χωρίς τα '<', '>').
4. Τυπώστε όλα τα url τα οποία περιέχουν τουλάχιστον μία λέξη, και στα οποία όλες οι λέξεις ανήκουν στην ίδια γλώσσα (δηλαδή δεν υπάρχουν 2 λέξεις που να ανήκουν σε διαφορετικές γλώσσες).