

Σύνολα Ασκήσεων

1^ο Σύνολο Ασκήσεων

Άσκηση 1(α)

Έστω ένας τύπος συσχέτιση R μεταξύ δύο τύπων οντοτήτων $E1$ και $E2$.

Υποθέστε ότι σε κάποιο έγκυρο στιγμιότυπο η R περιέχει **6** συσχετίσεις.

Ποιος είναι ο μικρότερος και ποιος ο μεγαλύτερος αριθμός οντοτήτων που μπορεί να έχει σε αυτό το στιγμιότυπο η $E1$ και ποιος η $E2$ σε κάθε μια από τις παρακάτω περιπτώσεις (εξηγείστε την απάντησή σας).

- (i) Η συσχέτιση είναι 1-1 και η συμμετοχή των $E1$ και $E2$ ολική.
- (ii) Η συσχέτιση είναι 1-1, η συμμετοχή της $E1$ ολική και της $E2$ μερική.
- (iii) Η συσχέτιση είναι 1-N (από την $E1$ στην $E2$) και η συμμετοχή των $E1$ και $E2$ ολική.
- (iv) Η συσχέτιση είναι N-M και η συμμετοχή των $E1$ και $E2$ ολική.

Ολική συμμετοχή: Δε μπορεί να υπάρχει οντότητα που δε συμμετέχει (καθορίζει το **μέγιστο** αριθμό οντοτήτων)

Συσχέτιση 1: Μια οντότητα συμμετέχει **το πολύ** μια φορά (καθορίζει τον **ελάχιστο** αριθμό οντοτήτων)

Άσκηση 1(β)

Έστω ένας τύπος συσχέτιση R μεταξύ δύο τύπων οντοτήτων $E1$ και $E2$.

Υποθέστε ότι σε κάποιο έγκυρο στιγμιότυπο, η $E1$ έχει 3 οντότητες και η $E2$ έχει 7 οντότητες.

Ποιος είναι ο μικρότερος και ποιος ο μεγαλύτερος αριθμός συσχετίσεων που μπορεί να περιέχει η R σε κάθε μια από τις παρακάτω περιπτώσεις (εξηγήστε την απάντησή σας).

(i) Η συμμετοχή της $E1$ είναι μερική, της $E2$ ολική και η συσχέτιση $N-M$.

(ii) Η συμμετοχή της $E1$ είναι μερική, της $E2$ ολική και η συσχέτιση $1-N$ (από την $E1$ στην $E2$).

(iii) Η συμμετοχή της $E1$ είναι μερική, της $E2$ μερική και η συσχέτιση $1-1$;

Ολική συμμετοχή: Πρέπει να υπάρχει τουλάχιστον 1 συσχέτιση για κάθε οντότητα (ελάχιστο)

Συσχέτιση 1: Μια οντότητα συμμετέχει το πολύ σε μία συσχέτιση (μέγιστο)

Άσκηση 2

Έστω μια σχέση $R(A, B, C, D)$ και το παρακάτω στιγμιότυπο της.

A	B	C	D
1	2	3	1
5	3	3	2
5	4	6	3
1	3	3	1

Υποθέτοντας, ότι το στιγμιότυπο είναι έγκυρο, δώστε όλα τους πιθανούς συνδυασμούς υποψηφίων κλειδιών που μπορεί να ισχύουν και εξηγήστε την απάντησή σας.

Άσκηση 3

Θέλουμε να κατασκευάσουμε μια βάση δεδομένων στην οποία θα αποθηκεύουμε αποτελέσματα μετρήσεων από αισθητήρες που έχουμε εγκαταστήσει στα δωμάτια ενός κτιρίου.

- Σε κάθε δωμάτιο έχουμε εγκαταστήσει **τουλάχιστον έναν** αισθητήρα.
- Οι αισθητήρες μετρούν θερμοκρασία και ποσοστό υγρασίας.
- Για κάθε δωμάτιο έχουμε έναν **μοναδικό** αριθμό-δωματίου, τον όροφο στον οποίο βρίσκεται και τα τετραγωνικά του μέτρα.
- Ένας αισθητήρας χαρακτηρίζεται από τον αριθμό-δωματίου στον οποίο έχει εγκατασταθεί και από έναν αριθμό-αισθητήρα που είναι **μοναδικός ανά δωμάτιο** (δηλαδή, δεν υπάρχουν αισθητήρες με τον ίδιο αριθμό-αισθητήρα στο ίδιο δωμάτιο). Για κάθε αισθητήρα καταγράφουμε τον κατασκευαστή του, τη μέγιστη διάρκεια ζωής της μπαταρίας του (σε ώρες), την ακρίβεια του (σε ποσοστό) ως αναφορά τη θερμοκρασία, και την ακρίβεια του (σε ποσοστό) ως αναφορά τη υγρασία.
- Για κάθε μέτρηση, αναφέρουμε τη χρονική στιγμή της καταγραφή της, τον αισθητήρα που την κατέγραψε και τις δύο τιμές (θερμοκρασία, υγρασία) της μέτρησης. Επίσης σε κάθε μέτρηση καταγράφουμε και το ποσοστό της διαθέσιμης μπαταρίας του αισθητήρα.
- Κάθε χρονική στιγμή, έχουμε **το πολύ μία μέτρηση** ανά αισθητήρα.

Άσκηση 3 (συνέχεια)

(α) Σχεδιάστε ένα κατάλληλο **μοντέλο οντοτήτων/συσχετίσεων**.

(β) Σχεδιάστε ένα κατάλληλο **σχεσιακό μοντέλο**.

(γ) Εξηγείστε ποιοι είναι οι **περιορισμοί ακεραιότητας** για το συγκεκριμένο πρόβλημα και πως ικανοποιούνται από καθένα από τα μοντέλα που σχεδιάσατε.

(δ) Υλοποιήστε τη σχεσιακή βάση που σχεδιάσατε στην SQLite. Συγκεκριμένα:

(i) Ορίστε τους πίνακες του σχεσιακού σχήματος. Για κάθε γνώρισμα, ορίστε ένα κατάλληλο πεδίο ορισμού. Ορίστε όλα τα κλειδιά, ξένα κλειδιά και όσους περιορισμούς ακεραιότητας θεωρείτε απαραίτητους. Για τα ξένα κλειδιά, προσδιορίστε τη λειτουργία που πρέπει να ακολουθηθεί αν αυτά παραβιαστούν. Συγκεκριμένα, όταν ενημερώνουμε (UPDATE) ένα δωμάτιο, θέλουμε να ενημερώνονται (cascade) και οι αντίστοιχοι αισθητήρες και μετρήσεις. Ορίστε κατάλληλες λειτουργίες και για τη διαγραφή (DELETE) και εξηγήστε την επιλογή σας.

(ii) Εισάγετε 5 πλειάδες σε καθέναν από τους πίνακες του σχήματός σας. Δώστε τις σχετικές εντολές INSERT.

Πρέπει ποσοστά κλπ

(iii) Για κάθε πίνακα, δώστε μια εντολή που να οδηγεί σε παραβίαση του πρωτεύοντος κλειδιού του.

(iii) Για κάθε ξένο κλειδί, δώστε μια εντολή DELETE που να οδηγεί σε παραβίαση ξένου κλειδιού. Δώστε το περιεχόμενο των αντίστοιχων πινάκων πριν και μετά τη διαγραφή.

(iv) Για κάθε ξένο κλειδί, δώστε μια εντολή UPDATE που να οδηγεί σε παραβίαση ξένου κλειδιού. Δώστε το περιεχόμενο των αντίστοιχων πινάκων πριν και μετά τη διαγραφή.

Άσκηση 4

Σας ζητούν να σχεδιάσετε και να υλοποιήσετε μια βάση δεδομένων για ένα online κατάστημα πώλησης ρούχων. Θέλουμε να διατηρήσουμε πληροφορία για τα διαθέσιμα ρούχα, τους πελάτες και τα καλάθια με τις αγορές τους.

- Για κάθε ρούχο έχουμε ένα μοναδικό κωδικό, μια σύντομη περιγραφή (π.χ., “mid rise flared jean”) και την τιμή του.
- Ένα ρούχο με τον ίδιο κωδικό μπορεί να διατίθεται σε διαφορετικά χρώματα και μεγέθη. Διατηρούμε επίσης τη διαθεσιμότητα (τον αριθμό κομματιών) που υπάρχουν στο κατάστημα για κάθε συνδυασμό μεγέθους-χρώματος για κάθε ρούχου (δηλαδή, κωδικό ρούχο).
- Για κάθε πελάτη έχουμε έναν μοναδικό κωδικό, το όνομα του, το email του και τη διεύθυνσή του.
- Κάθε καλάθι-αγοράς ενός πελάτη έχει έναν μοναδικό κωδικό και περιλαμβάνει έναν ή περισσότερους κωδικούς ρούχου (σε συγκεκριμένο μέγεθος και χρώμα ο καθένας). Διατηρούμε επίσης την ημερομηνία που δημιουργήθηκε το καλάθι.
- Ένας πελάτης μπορεί να έχει πολλά καλάθια, αλλά κάθε καλάθι αφορά μόνο ένα πελάτη

Άσκηση 4 (συνέχεια)

- (α) Σχεδιάστε ένα κατάλληλο μοντέλο οντοτήτων/συσχετίσεων.
- (β) Σχεδιάστε ένα κατάλληλο σχεσιακό μοντέλο.
- (γ) Εξηγείστε ποιοι είναι οι περιορισμοί ακεραιότητας για το συγκεκριμένο πρόβλημα και πως ικανοποιούνται από καθένα από τα μοντέλα που σχεδιάσατε.
- (δ) Υλοποιήστε τη σχεσιακή βάση που σχεδιάσατε στη SQLite. Συγκεκριμένα:
- (i) Ορίστε τους πίνακες του σχεσιακού σχήματος. Για κάθε γνώρισμα, ορίστε ένα κατάλληλο πεδίο ορισμού. Ορίστε όλα τα κλειδιά, ξένα κλειδιά και όσους περιορισμούς θεωρείτε απαραίτητους. Για τα ξένα κλειδιά, ορίστε τη λειτουργία που πρέπει να ακολουθηθεί αν αυτά παραβιαστούν. και εξηγείστε την επιλογή σας.
 - (ii) Εισάγετε 5 πλειάδες σε καθέναν από τους πίνακες του σχήματός σας. Δώστε τις σχετικές εντολές INSERT.

2^ο Σύνολο Ασκήσεων

Άσκηση 1

course(course_id, courseLevel)

student (s_id, inPhase, yearsinProgram)

professor(p_id, hasPosition)

taughtby(course_id, u_id)

advisedby(s_id, p_id)

(α) Τα μαθήματα (course_id) που διδάσκονται *είτε* από καθηγητές *είτε* από φοιτητές *αλλά όχι* και από καθηγητές και από φοιτητές.

β) Μια δυάδα (t1, t2) όπου t1 είναι η *μικρότερη διάρκεια* και t2 η *μεγαλύτερη διάρκεια* σπουδών.

(γ) Τα μαθήματα (course_id) που έχουν ως διδάσκοντα τουλάχιστον έναν φοιτητή και έναν καθηγητή που είναι σύμβουλος αυτού του φοιτητή.

(δ) Τα μαθήματα (course_id) που *δεν* διδάσκονται.

(ε) Τους φοιτητές (s_id) που έχουν *ακριβώς τους ίδιους* συμβούλους καθηγητές με τον φοιτητή με s_id = 81.

Άσκηση 2

course(course_id, courseLevel)

student (s_id, inPhase, yearsinProgram)

professor(p_id, hasPosition)

taughtby(course_id, u_id)

advisedby(s_id, p_id)

(β) Μια δυάδα (t1, t2) όπου t1 είναι η *μικρότερη διάρκεια* και t2 η *μεγαλύτερη διάρκεια* σπουδών.

(ε) Τους φοιτητές (s_id) που έχουν *ακριβώς τους ίδιους συμβούλους καθηγητές* με τον φοιτητή με s_id = 81.

Άσκηση 3(α)

course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

(i) Μια δυάδα (t1, t2) όπου t1 είναι η *μικρότερη διάρκεια* και t2 η *μεγαλύτερη διάρκεια* σπουδών.

```
SELECT MIN(yearsinProgram),  
MAX(yearsinProgram)  
FROM student;
```

Άσκηση 3(α)

course(course_id, courseLevel)

student (s_id, inPhase, yearsinProgram)

professor(p_id, hasPosition)

taughtby(course_id, u_id)

advisedby(s_id, p_id)

(ii) Τα μαθήματα (course_id) που έχουν ως διδάσκοντα τουλάχιστον έναν φοιτητή και έναν καθηγητή που είναι σύμβουλος αυτού του φοιτητή.

SELECT DISTINCT T1.course_id

FROM taughtby **AS** T1, taughtby **AS** T2. advisedby **AS** Ad

WHERE T1.course_id = T2.course_id **AND** T1.u_id = Ad.s-id **AND** T2.u_id = Ad.p_id;

Άσκηση 3(α)

course(course_id, courseLevel)

student (s_id, inPhase, yearsinProgram)

professor(p_id, hasPosition)

taughtby(course_id, u_id)

advisedby(s_id, p_id)

(iii) Τους φοιτητές (s_id) που έχουν *ακριβώς τους ίδιους* συμβούλους καθηγητές με τον φοιτητή με s_id = 81.

```
SELECT [DISTINCT] T.s_id
FROM advisedby AS T /* T is the student we are looking for */
WHERE /* all advisors of T are advisors of 81 */
      NOT EXISTS (SELECT T1.p_id FROM advisedby AS T1
                  WHERE T1.s_id = T.s_id AND T1.p_id NOT IN (SELECT p_id FROM advisedby
                  WHERE s_id = 81) AND
/* all advisors of 81 are advisors of T */
      NOT EXISTS (SELECT Ad.p_id FROM advisedby AS Ad
                  WHERE A.s_id = 81 AND A.p_id NOT IN (SELECT p_id FROM advisedby
                  WHERE s_id = T.s_id);
```

Άσκηση 3(α)

course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

(iv) Τη θέση (hasPosition) του καθηγητή που επιβλέπει το μεγαλύτερο αριθμό φοιτητών. Σε περίπτωση ισοβαθμίας, δώστε όλες τις θέσεις.

```
SELECT DISTINCT P.hasPosition
FROM professor as P,
/* max number of students per advisor */
  (SELECT MAX(AC1.Stcount) AS Stmax FROM (SELECT p_id, COUNT(s_id) AS
  Stcount FROM advisedby GROUPBY p_id) AS AC1) as MaxC,
/* (p_id, Stcount) pairs, where Stcount is the number of students advised by p_id */
  ((SELECT p_id, COUNT(s_id) AS Stcount FROM advisedby GROUPBY p_id) AS AC2)
WHERE P.p_id = AC2.p_id AND AC2.Stcount = MaxC.Stmax;
```


Άσκηση 3(α)

course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

(ν) Για κάθε μάθημα, τον αριθμό των καθηγητών και των φοιτητών που τα διδάσκουν. Συγκεκριμένα, το αποτέλεσμα πρέπει να είναι τριάδες της μορφής (x, y, z) όπου x το course_id του μαθήματος, y ο αριθμός των καθηγητών, και z ο αριθμός των φοιτητών που το διδάσκουν

```
SELECT CP.course_id, CP.Ucount, SP.Ucount,  
FROM  
/* number of professors */  
(SELECT course_id, COUNT(u_id) AS Ucount FROM taughtby WHERE u_id IN  
(SELECT p_id FROM professor) GROUPBY course_id) AS CP,  
FULL OUTER JOIN  
/* number of students */  
(SELECT course_id, COUNT(u_id) AS Ucount FROM taughtby WHERE u_id IN  
(SELECT s_id FROM student) GROUPBY course_id) AS SP,  
ON CP.course_id = SP.Course_id;
```

Άσκηση 3(α)

course(course_id, courseLevel)

student (s_id, inPhase, yearsinProgram)

professor(p_id, hasPosition)

taughtby(course_id, u_id)

advisedby(s_id, p_id)

(vi) Για κάθε έτος σπουδών, το ποσοστό των φοιτητών που είναι σε αυτό και διδάσκουν κάποιο μάθημα. Συγκεκριμένα, το αποτέλεσμα πρέπει να είναι δυάδες της μορφής (x, y) όπου x το yearinProgram και y% το ποσοστό των φοιτητών στο έτος x που διδάσκουν κάποιο μάθημα.

(Μπορεί να χρειαστεί cast)

```
SELECT S1.YearinProgram, (T1.SCount * 100)/(S1.Scount)
```

```
FROM
```

```
/* number of students per year */
```

```
(SELECT yearsinProgram, COUNT(s_id) AS Scount FROM student GROUPBY  
yearsinProgram) AS S1,
```

```
/* number of students teaching per year*/
```

```
(SELECT yearsinProgram, COUNT(DISTINCT T.u_id) as Scount FROM taughtby JOIN  
student ON taughtby.u_id =student s_id GROUPBY yearsinProgram) AS T1
```

```
WHERE S1.yearinProgram = T1.yearinProgram;
```

Άσκηση 3(β)

course(course_id, courseLevel)
student (s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

Δώστε μια SQL έκφραση που να αναθέτει ως επιβλέποντα τον καθηγητή με p_id = 5 σε κάθε φοιτητή με περισσότερα από 3 έτη σπουδών που δεν έχει ήδη επιβλέποντα.

```
INSERT INTO advisedby  
SELECT S.s_id, 5  
FROM student as S  
WHERE S.yearsinProgram > 3 and S.s_id NOT IN (SELECT s_id FROM advisedby);
```

Άσκηση 3(δ)

course(course_id, courseLevel)
student(s_id, inPhase, yearsinProgram)
professor(p_id, hasPosition)
taughtby(course_id, u_id)
advisedby(s_id, p_id)

Ορίστε μια όψη professor-count(p_id, s_count) που να περιέχει για κάθε καθηγητή που επιβλέπει φοιτητές, τον αριθμό των φοιτητών που επιβλέπει.

```
CREATE VIEW professor-count(p_id, s_count) AS  
SELECT p_id, COUNT(s_id)  
FROM advisedby  
GROUPBY p_id;
```

Ερωτήσεις;