



Εργαστηριακό μάθημα 4

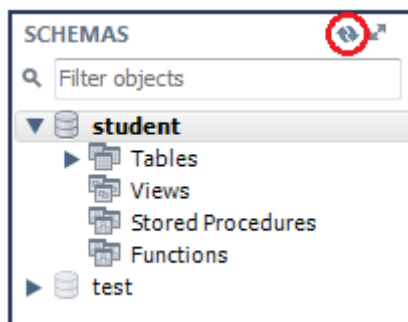
Βελτιστοποίηση SQL ερωτημάτων στη βάση SongsDB

Δημιουργία της βάσης SongsDB

1. Ανοίξτε την εφαρμογή **MySQL Workbench** και συνδεθείτε στη βάση (όπως στο πρώτο εργαστήριο).
2. Αν το schema student υπάρχει ήδη διαγράψτε το. Κάνετε δεξί κλικ πάνω του, επιλέξτε “**Drop schema...**” και στο μενού που θα εμφανιστεί επιλέξτε “**Drop now**”.
3. Επιλέξτε «**File -> Open SQL Script...**» για να ανοίξετε το αρχείο “**Lab4Dump.sql**” και στη συνέχεια πιάστε το κουμπί «**Execute**» (βρίσκεται στην γραμμή εργαλείων). Εναλλακτικά, μπορείτε να εκτελέσετε το script επιλέγοντας «**Query -> Execute (All or Selection)**» ή πιέζοντας Ctrl+Shift+Enter.



4. Ελέγξτε στο «**SCHEMAS**» (στα αριστερά του GUI) αν κάτω από το student έχουν όντως δημιουργηθεί όλοι οι πίνακες με τα ζητούμενα κλειδιά και δεδομένα. Θα χρειαστεί να κάνετε “refresh” τα SCHEMAS για να εμφανιστεί το σχήμα της ΒΔ “student”.

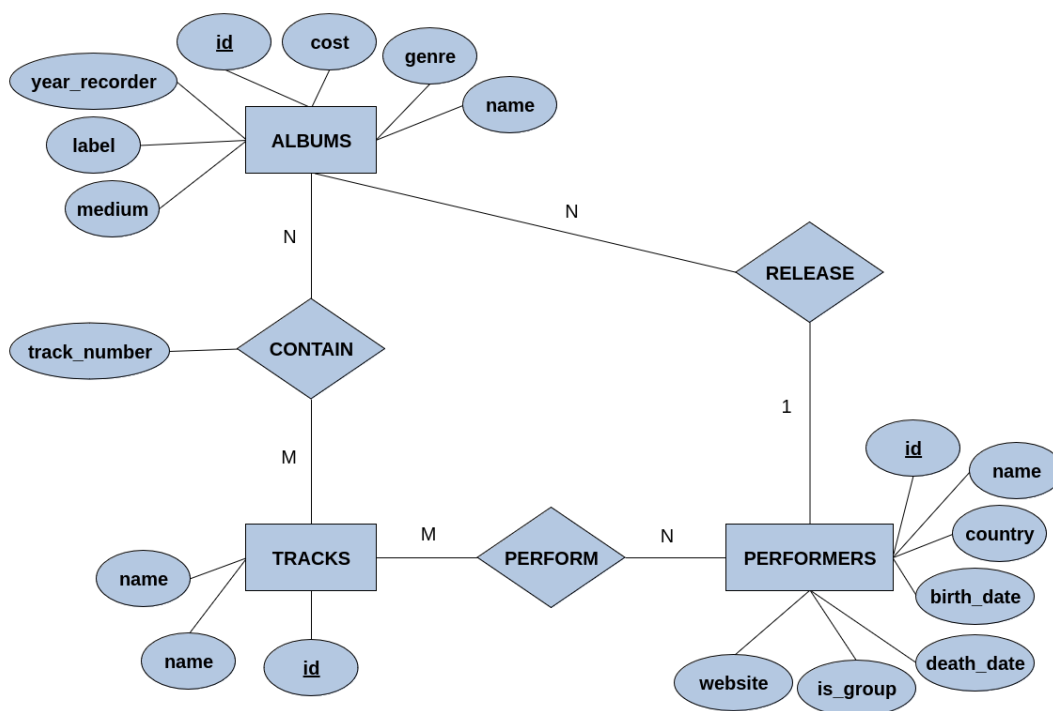


5. Επιλέξτε «**File -> New Query Tab**» (εναλλακτικά μπορείτε να πατήσετε το κουμπί «**New Query Tab**» στην γραμμή εργαλείων Standard ή τον συνδυασμό πλήκτρων Ctrl+N).





Απλοποιημένο διάγραμμα Ο/Σ της βάσης SongsDB



SQL Ερωτήματα προς Υλοποίηση

1. Να βρεθούν όλα τα στοιχεία των album της βάσης που ανήκουν στο είδος rock ή γενικά σε κάποια παραλλαγή της rock μουσικής (η λέξη rock να περιέχεται μέσα στο πεδίο genre).

```
SELECT * FROM albums  
WHERE genre LIKE '%rock%'
```

Η συνθήκη **WHERE** χρησιμοποιείται για να φιλτράρει τις γραμμές ώστε να επιστρέφονται μόνο αυτές που περιέχουν τη λέξη "rock" στη στήλη genre. Η λέξη-κλειδί **LIKE** χρησιμοποιείται για την αναζήτηση κειμένου μέσα σε μια στήλη. Έχουμε 2 φορές το "%":

- **% πριν από το "rock"**: μπορεί να υπάρχει οποιοδήποτε κείμενο πριν από τη λέξη "rock".
- **% μετά από το "rock"**: μπορεί να υπάρχει οποιοδήποτε κείμενο μετά από τη λέξη "rock".

2. Να βρεθούν όλα τα στοιχεία των τραγουδιών (track) της βάσης που το 6ο γράμμα του ονόματος τους είναι το e και το 8ο γράμμα είναι το t (τα υπόλοιπα γράμματα



μπορούν να είναι οποιαδήποτε, ενώ επιτρέπεται επίσης να ακολουθούν γράμματα μετά το t).

```
SELECT * FROM tracks
WHERE name LIKE '%_a_%'
```

Οι κάτω παύλες () αντιπροσωπεύουν **οποιονδήποτε ENAN** χαρακτήρα, εδώ έχουμε 2 “_”, άρα θέλουμε οποιουσδήποτε 2 χαρακτήρες, πριν και μετά το a. Το τι άλλο θα υπάρχει δε μας αφορά, εφόσον έχουμε εκατέρωθεν το “%”.

- Αν το αριστερά “%” δεν υπήρχε, το κείμενο που θα επέστρεφε το query θα έπρεπε να **ξεκινάει** με 2 χαρακτήρες, μετά “a” και μετά 2 ακόμα χαρακτήρες.
- Αν το δεξιά “%” δεν υπήρχε, το κείμενο που θα επέστρεφε το query θα έπρεπε να **τελειώνει** με 2 χαρακτήρες, μετά “a” και μετά 2 ακόμα χαρακτήρες.

3. Να βρεθούν όλα τα στοιχεία των τραγουδιών (track) της βάσης που το προ-προ-τελευταίο γράμμα του ονόματος τους (τρίτο από το τέλος) είναι το l και το τελευταίο γράμμα είναι το w (τα υπόλοιπα γράμματα μπορούν να είναι οποιαδήποτε, ενώ επιτρέπεται επίσης να μην υπάρχουν γράμματα πριν το l).

4. Να βρεθούν τα ονόματα και οι χώρες των καλλιτεχνών που δεν προέρχονται από τις χώρες UK και USA.

```
SELECT name, country FROM performers
WHERE country NOT IN ('USA', 'UK');
```

Σημειώστε πως μπορεί εναλλακτικά το τμήμα **WHERE** του ερωτήματος να γραφεί ως:
WHERE country <> 'USA' **AND** country <> 'UK'

WHERE country **NOT IN** ('USA', 'UK'): Αυτή η συνθήκη φιλτράρει τις γραμμές για να περιλάβει μόνο εκείνες όπου η στήλη country δεν είναι ούτε USA ούτε UK.

Το σύμβολο <> σημαίνει "όχι ίσο με".

Εδώ, η χρήση του **AND** με το <> για δύο τιμές ('USA' και 'UK') έχει το ίδιο αποτέλεσμα με τη χρήση του **NOT IN** ('USA', 'UK')

5. Να βρείτε τη δεύτερη μεγαλύτερη διάρκεια τραγουδιού της βάσης και να την εμφανίσετε.

```
SELECT DISTINCT duration FROM tracks t1
WHERE 2 = (SELECT COUNT(DISTINCT duration) FROM tracks t2 WHERE
          t1.duration <= t2.duration);
```



SELECT DISTINCT duration **FROM** tracks t1: Επιλέγεται η μοναδική τιμή duration από τον πίνακα tracks, τον οποίο ονομάζουμε t1.

WHERE 2 = (SELECT COUNT(DISTINCT duration) FROM tracks t2 WHERE t1.duration <= t2.duration);

- **SELECT COUNT(DISTINCT duration) FROM tracks t2 WHERE t1.duration <= t2.duration;** Μετράει πόσες διαφορετικές τιμές duration στον πίνακα tracks (τον αναφέρουμε εδώ ως t2) είναι μεγαλύτερες ή ίσες από την duration του t1.
- **WHERE 2 = (...):** Όπου η συνθήκη είναι αληθής, σημαίνει ότι η τιμή duration στο t1 είναι η δεύτερη μεγαλύτερη μοναδική τιμή στον πίνακα. Σκεφτείτε ότι ο πίνακας ξεκινάει από την 1η καταχώρηση, όπου αυτή είναι το μεγαλύτερο τραγούδι σε διάρκεια και κατεβαίνει, οπότε το “2” είναι το δεύτερο μεγαλύτερο, το “5” είναι το 5ο μεγαλύτερο κ.ο.κ..

Ενας άλλος τρόπος που θα μπορούσατε να εκτελέσετε το παραπάνω είναι ο ακόλουθος:

SELECT MAX (duration) AS second_largest_duration
FROM tracks
WHERE duration < (SELECT MAX(duration) FROM tracks);

6. Να βρεθούν τα ονόματα και τα έτη ηχογράφησης των album που ηχογραφήθηκαν μεταξύ του 1990 και του 2000 (συμπεριλαμβανομένων των ετών 1990 και 2000). Χρησιμοποιήστε τη δομή **BETWEEN ... AND**.

SELECT name, year_recorded **FROM** albums
WHERE year_recorded **BETWEEN** 1990 **AND** 2000

Σημειώστε πως μπορεί εναλλακτικά το τμήμα **WHERE** του ερωτήματος να γραφεί ως:
WHERE year_recorded >= 1990 **AND** year_recorded <= 2000

BETWEEN 1990 AND 2000: φιλτράρει τις εγγραφές για να επιστρέψει μόνο εκείνες όπου η τιμή στο γνώρισμα year_recorded βρίσκεται ανάμεσα στα έτη 1990 και 2000, συμπεριλαμβανομένων και των δύο ορίων.

Αν δε θέλετε τα έτη 1990 και 2000, πρέπει να γράψετε **BETWEEN 1991 AND 1999** (ή **WHERE** year_recorded > 1990 **AND** year_recorded < 2000)

7. Να βρεθούν τα ονόματα και τα έτη ηχογράφησης των album που ηχογραφήθηκαν το 2000. Επιπλέον, να ελέγξετε το πλάνο εκτέλεσης του ερωτήματος και να αναφέρετε πόσες εγγραφές αναμένεται να εξεταστούν.

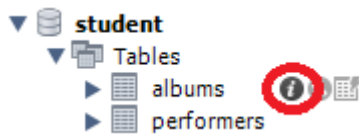
SELECT name, year_recorded **FROM** albums
WHERE year_recorded = 2000



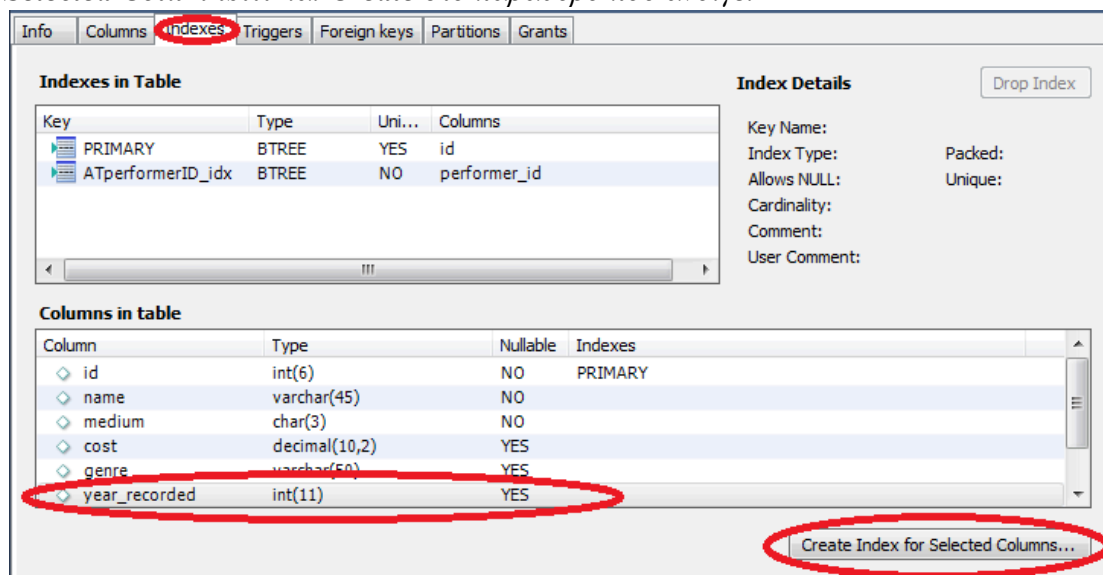
Μπορούμε να ελέγξουμε το πλάνο εκτέλεσης του ερωτήματος προσθέτοντας το keyword **EXPLAIN** πριν από το ερώτημα. Ο αριθμός των εγγραφών που αναμένεται να εξεταστούν (από το πεδίο rows) είναι 10.

8. Κατασκευάστε ένα **index** στο `year_recorded` και εκτελέστε το προηγούμενο ερώτημα. Ελέγξτε εκ νέου το πλάνο εκτέλεσης του ερωτήματος και αναφέρετε πόσες εγγραφές αναμένεται να εξεταστούν αυτή τη φορά.

Για να κατασκευάσουμε το *index* ανοίγουμε τον πίνακα *albums* με το *table inspector*:



Στη συνέχεια, επιλέγουμε τη στήλη `year_recorded` και πατάμε **Create Index for Selected Columns...** και **Create** στο παράθυρο που ανοίγει



Όπως και προηγουμένως, μπορούμε να ελέγξουμε το πλάνο εκτέλεσης του ερωτήματος προσθέτοντας το keyword **EXPLAIN** πριν από το ερώτημα. Ο αριθμός των εγγραφών που αναμένεται να εξεταστούν (από το πεδίο rows) είναι 1.

(Σημείωση: Αν δε θέλετε πλέον να χρησιμοποιείται το *index* που φτιάξατε, στην ίδια οθόνη επιλέγете το *index* από τον πίνακα *indexes in table* και πατάτε **Drop Index**)

9. Απαντήστε εκ νέου στα δύο προηγούμενα ερωτήματα αυτή τη φορά αν θέλουμε να βρούμε τα ονόματα και τα έτη ηχογράφησης των *album* που ηχογραφήθηκαν μετά το 2000. Δείτε τα πλάνα εκτέλεσης και αναφέρετε πόσες εγγραφές αναμένεται να εξεταστούν σε κάθε περίπτωση (δηλαδή χωρίς και με *index*).

Κατασκευάζουμε το ερώτημα όμοια με παραπάνω (όπου `year_recorded = 2000` βάζουμε `year_recorded > 2000`).



```
SELECT name, year_recorded FROM albums
WHERE year_recorded > 2000
```

Στη συνέχεια, μπορούμε να ελέγχουμε το πλάνο εκτέλεσης του ερωτήματος προσθέτοντας το keyword **EXPLAIN** πριν από το ερώτημα. Χωρίς το index αναμένεται να εξεταστούν 10 εγγραφές, ενώ με το index αναμένεται να εξεταστούν 2. Αυτό γιατί το index οδηγεί την αναζήτηση κατευθείαν στις χρονιές που είναι > 2000.

10. Να βρεθούν ο τίτλος και το είδος των albums των Depeche Mode (ο καλλιτέχνης που μας αφορά) που είναι καταχωρημένα στη βάση. Μετονομάστε τις στήλες στο αποτέλεσμα του ερωτήματος σε Album, Genre, Artist. Εκτελέστε το ερώτημα δύο φορές, μία διαβάζοντας πρώτα τον πίνακα albums και μία διαβάζοντας πρώτα τον πίνακα performers. Δείτε τα πλάνα εκτέλεσης και αναφέρετε πόσες εγγραφές αναμένεται να εξεταστούν σε κάθε περίπτωση.

```
SELECT albums.name AS Album, genre AS Genre, performers.name AS Artist FROM
albums STRAIGHT_JOIN performers ON albums.performer_id = performers.id
WHERE performers.name = 'Depeche Mode'
```

Εκτελώντας το albums **STRAIGHT_JOIN** performers (και προσθέτοντας το keyword **EXPLAIN**) αναμένεται να εξεταστούν 10 εγγραφές για τον πίνακα albums και 9 εγγραφές για τον πίνακα performers. Εκτελώντας το performers **STRAIGHT_JOIN** albums αναμένεται να εξεταστούν 9 εγγραφές για τον πίνακα performers και 1 εγγραφή για τον albums.

11. Επαναλάβετε το προηγούμενο ερώτημα. Εκτελέστε το ερώτημα δύο φορές, μία ως έχει και μία αφού προσθέσετε ένα index στη στήλη name των performers. Δείτε τα πλάνα εκτέλεσης και αναφέρετε πόσες εγγραφές αναμένεται να εξεταστούν σε κάθε περίπτωση (δηλαδή χωρίς και με index).

```
SELECT albums.name AS Album, genre AS Genre, performers.name AS Artist FROM
albums JOIN performers ON albums.performer_id = performers.id
WHERE performers.name = 'Depeche Mode'
```

Μπορούμε να ελέγχουμε το πλάνο εκτέλεσης του ερωτήματος προσθέτοντας το keyword **EXPLAIN** πριν από το ερώτημα. Χωρίς το index αναμένεται να εξεταστούν 9 εγγραφές για τον πίνακα performers και 1 εγγραφή για τον πίνακα albums, ενώ με το index αναμένεται να εξεταστούν 1 εγγραφή για τον πίνακα performers και 1 εγγραφή για τον πίνακα albums.

12. Να βρεθούν όλοι οι τίτλοι τραγουδιών για την καλλιτέχνη Kylie Minogue. Αναγράψτε τις συνενώσεις δύο φορές, μία με τη σειρά performers, performers_tracks, tracks, και μία με τη σειρά tracks, performers_tracks, performers και συγκρίνετε τον αριθμό των εγγραφών που αναμένεται να εξεταστούν σε κάθε περίπτωση.



Σειρά 1: *performers, performers_tracks, tracks*

```
SELECT tracks.name
FROM performers
  STRAIGHT_JOIN performers_tracks
                ON performers.id = performers_tracks.performer_id
  STRAIGHT_JOIN tracks
                ON tracks.id = performers_tracks.track_id
WHERE performers.name = 'Kylie Minogue';
```

Σειρά 2: *tracks, performers_tracks, performers*

```
SELECT tracks.name
FROM tracks
  STRAIGHT_JOIN performers_tracks
                ON tracks.id = performers_tracks.track_id
  STRAIGHT_JOIN performers
                ON performers.id = performers_tracks.performer_id
WHERE performers.name = 'Kylie Minogue';
```

Μπορούμε να ελέγξουμε το πλάνο εκτέλεσης του ερωτήματος προσθέτοντας το keyword **EXPLAIN** πριν από το ερώτημα. Βλέπουμε ότι σε κάθε περίπτωση αναμένεται να εξεταστεί διαφορετικό πλήθος εγγραφών.

Πιο συγκεκριμένα, στην πρώτη περίπτωση θα συναντήσουμε 9 προσπελάσεις στον πίνακα *performers* και από 1 στους πίνακες *performers_tracks* και *tracks*. Στη δεύτερη περίπτωση, θα παρατηρήσετε ότι έχουμε 16 προσπελάσεις στον πίνακα *tracks* και από 1 στους πίνακες *performers_tracks* και *performers*.

Μπορούμε να βελτιστοποιήσουμε τις προσπελάσεις ακόμα περισσότερο, δημιουργώντας ένα *index* στον πίνακα *performers* στη στήλη *name*. Κατ' αυτό τον τρόπο, θα έχουμε από 1 προσπέλαση σε κάθε πίνακα, τρέχοντας την πρώτη εντολή (με ή χωρίς **STRAIGHT_JOIN**).