Βάσεις Δεδομένων Ι

**Εξέταση:** Βάση δεδομένων Ταινιών

Χ. Σκουρλάς, Α. Τσολακίδης, Τμήμα Μηχανικών Πληροφορικής ΤΕ

Έστω βάση δεδομένων ταινιών. Για κάθε ταινία (film) ενημερώνουμε σε ποιά κατηγορία (category) ή σε ποιές κατηγορίες ανήκει. Κάθε ταινία μπορεί να ανήκει σε πολλές κατηγορίες και σε μία κατηγορία μπορεί να έχουμε πολλές ταινίες. Κατά συνέπεια θεωρούμε ότι έχουμε στη βάση τον πίνακα film (ταινίες), τον πίνακα category (κατηγορίες ταινιών) και τον πίνακα film\_category (κατηγορίες στις οποίες ανήκει κάθε ταινία) . Ένας συνεργάτης (user) ενημερώνει τις κατηγορίες και τις κατηγορίες στις οποίες ανήκει μια ταινία. Η ημερομηνία (last\_update) και το όνομα του συνεργάτη (user) που ενημέρωσε τελευταίος μια κατηγορία καταχωρίζονται στον πίνακα category. Επίσης, η ημερομηνία (last\_update) και το όνομα του συνεργάτη (user) που ενημέρωσε τελευταίος μια κατηγορία μιας ταινίας στον πίνακα film\_category καταχωρίζονται στον πίνακα.



Οι στήλες των πινάκων είναι οι εξής:

Πίνακας film\_category

|  |  |  |  |
| --- | --- | --- | --- |
| film\_id | category\_id | last\_update | user |
| Κωδικός ταινίας  | Κωδικός Κατηγορίας | Η τελευταία ημερομηνία που άλλαξε η κατηγορία της ταινίας | Συνεργάτης που έκανε τη μεταβολή |

Πίνακας category

|  |  |  |  |
| --- | --- | --- | --- |
| category\_id | Name | last\_update | user |
| Κωδικός Κατηγορίας | Κατηγορία  | Η τελευταία ημερομηνία που άλλαξε η κατηγορία  | Συνεργάτης που έκανε τη μεταβολή |

Πίνακας film

|  |  |  |
| --- | --- | --- |
| film\_id | Title | Length |
| Κωδικός ταινίας  | Τίτλος | Διάρκεια |

1. Γράψτε δηλώσεις SQL δημιουργίας της βάσης δεδομένων **films\_db** και των πινάκων της. Να ορίσετε κύρια και ξένα κλειδιά στους πίνακες.

**Χωρίς κύρια και ξένα κλειδιά**

DROP DATABASE films\_db;

CREATE DATABASE films\_db;

USE films\_db;

CREATE TABLE film\_category (

 film\_id INT NOT NULL,

 category\_id INT NOT NULL,

 last\_update DATE,

user VARCHAR(25)

);

CREATE TABLE category (

 category\_id INT NOT NULL,

 name VARCHAR(25) NOT NULL,

 last\_update DATE,

 user VARCHAR(25)

);

CREATE TABLE film (

 film\_id INT NOT NULL,

 title VARCHAR(255) NOT NULL,

 length INT

 );

SHOW TABLES;

**Με κύρια και ξένα κλειδιά**

DROP DATABASE films\_db;

CREATE DATABASE films\_db;

USE films\_db;

CREATE TABLE film (

 film\_id INT NOT NULL,

 title VARCHAR(255) NOT NULL,

 length INT,

PRIMARY KEY(film\_id)

 );

CREATE TABLE category (

 category\_id INT NOT NULL,

 name VARCHAR(25) NOT NULL,

 last\_update DATE,

 user VARCHAR(25),

PRIMARY KEY(category\_id)

);

CREATE TABLE film\_category (

 film\_id INT NOT NULL,

 category\_id INT NOT NULL,

 last\_update DATE,

user VARCHAR(25),

PRIMARY KEY(film\_id, category\_id),

FOREIGN KEY(film\_id) REFERENCES film(film\_id),

FOREIGN KEY(category\_id) REFERENCES category(category\_id)

);



1. Γράψτε τις απαραίτητες δηλώσεις SQL INSERT για να εισάγετε τα στοιχεία της ταινίας 1 (ACADEMY DINOSAUR) που ανήκει στις κατηγορίες Documentary, Family και την ταινία 3 (ADAPTATION HOLES) που ανήκει στην κατηγορία Documentary.

Προσοχή! Η σειρά των δηλώσεων INSERT δεν είναι τυχαία. Στη συνέχεια παραθέτουμε όλες τις δηλώσεις.



INSERT INTO category VALUES (1,'Action','2006-02-15', 'Codd'),

(2,'Animation','2008-01-15', 'Codd'),

(3,'Children','2010-12-15', 'Codd'),

(4,'Classics','2011-09-23', 'Codd'),

(5,'Comedy','2011-09-23', 'Codd'),

(6,'Documentary','2014-01-16', 'Codd'),

(7,'Drama','2014-12-15', 'Codd'),

(8,'Family','2014-12-15', 'Codd'),

(9,'Foreign','2015-01-15', 'Date'),

(10,'Games','2015-02-15', 'Codd'),

(11,'Horror','2015-02-15', 'Date'),

(12,'Music','2015-02-15', 'Codd'),

(13,'New','2016-02-15', 'Date'),

(14,'Sci-Fi','2016-02-15', 'Codd'),

(15,'Sports','2016-02-15', 'Date'),

(16,'Travel','2016-02-15', 'Date');

INSERT INTO film VALUES

(1,'ACADEMY DINOSAUR', 86),

(2,'ACE GOLDFINGER', 48),

(3,'ADAPTATION HOLES', 50),

(4,'AFFAIR PREJUDICE', 117),

(5,'AFRICAN EGG', 130),

(6,'AGENT TRUMAN', 169);

INSERT INTO film\_category VALUES (1,6,'2006-02-15', 'Date'),

(1,8,'2006-02-15', 'Codd'),

(2,11,'2015-02-15', 'Date'),

(3,6,'2015-02-15', 'Codd'),

(4,11,'2016-02-15', 'Date'),

(5,8,'2016-02-15', 'Date'),

(6,9,'2016-02-15', 'Codd');

Select \* from film\_category;

Select \* from category;

Select \* from film;

1. Γράψτε μια δήλωση SQL που δείχνει τον κωδικό (film\_id), τον τίτλο (title), και τη διάρκεια (length) για τις ταινίες με διάρκεια από 50 λεπτά έως και 130 λεπτά.

 (με δύο τρόπους).

SELECT film.film\_id, title, length

FROM film

WHERE length BETWEEN 50 AND 130;

SELECT film.film\_id, title, length

FROM film

WHERE length >= 50 AND length <= 130;



1. Γράψτε μια δήλωση SQL που δείχνει τον κωδικό ταινίας (film\_id), τον τίτλο (title), και τον κωδικό κατηγορίας (category\_id) στην οποία ανήκει, για όλες τις ταινίες.

SELECT film.film\_id, title, film\_category.category\_id

FROM film, film\_category

WHERE film.film\_id=film\_category.film\_id;

Προσοχή! Αν γράψετε SELECT film\_id, ... είναι λάθος! Αν παραλείψετε τη σύνδεση είναι, επίσης, λάθος!



1. Γράψτε μια δήλωση SQL που δείχνει κωδικό ταινίας (film\_id), τίτλο (title), κωδικό κατηγορίας (category\_id) και κατηγορία (name) στην οποία ανήκει, για όλες τις ταινίες.

SELECT film.film\_id, title, film\_category.category\_id, name

FROM film, film\_category, category

WHERE film.film\_id=film\_category.film\_id

AND film\_category.category\_id=category.category\_id;

Προσοχή! Αν γράψετε SELECT film\_id, title, category\_id, name ... είναι λάθος! Αν παραλείψετε συνδέσεις είναι, επίσης, λάθος!



1. Εμφανίστε το πλήθος των ταινιών ανά κατηγορία

SELECT category\_id, count(\*)

FROM film\_category

GROUP BY category\_id;

Προσοχή! Αν γράψετε SELECT film\_id, category\_id, count(\*) … είναι λάθος!



1. Εμφανίστε film\_id, category\_id, last\_update, user για όλες τις ταινίες που ανήκουν στις κατηγορίες 6 (Documentary) ή 8 (Family), αν τις τελευταίες αλλαγές στον πίνακα film\_category για τις κατηγορίες των ταινιών αυτών τις έκανε ο Codd.

Δεχθήκαμε ως ορθή και την πρώτη λύση παρακάτω έστω και αν δεν είναι λογικό όταν γράφουμε δηλώσεις SELECT να θυμόμαστε απ’ έξω τους κωδικούς των κατηγοριών.

Προτιμήστε τις άλλες δύο λύσεις.

SELECT film\_id, category\_id, last\_update, user

FROM film\_category

WHERE category\_id IN (6, 8)

AND user='Codd';

SELECT film\_id, category\_id, last\_update, user

FROM film\_category

WHERE category\_id IN (SELECT category\_id FROM category

 WHERE name IN ('Documentary', 'Family'))

AND user='Codd';

SELECT film\_category.film\_id, film\_category.category\_id, film\_category.last\_update,

Film\_category.user

FROM film\_category, category

WHERE film\_category.category\_id=category.category\_id

AND name IN ('Documentary', 'Family')

AND film\_category.user='Codd';



1. Εμφανίστε film\_id, category\_id, last\_update, user για όλες τις ταινίες. Ταξινομήστε τις ταινίες ανά συνεργάτη (user) που καταχωρίζει τελευταίος τις κατηγορίες της ταινίας. Οι ταινίες κάθε συνεργάτη καταχωρίζονται ανά ημερομηνία (last\_update) (2 επίπεδα ταξινόμησης).

SELECT film\_id, category\_id, last\_update, user FROM film\_category ORDER BY user, last\_update;



1. Γράψτε μια δήλωση SQL που υπολογίζει τη μέση διάρκεια των ταινιών που έχουμε καταχωρισμένες και ο τίτλος τους αρχίζει από το λατινικό γράμμα A.

SELECT avg(length) FROM film

WHERE substr(title, 1, 1)='A' ;

Προσοχή! Αν γράψετε SELECT title, avg(length) FROM film ... είναι λάθος!



1. Γράψτε δηλώσεις SQL διαγραφής των παραπάνω πινάκων.

DROP TABLE FILM\_CATEGORY;

DROP TABLE CATEGORY;

DROP TABLE FILM;

Προσοχή! Η σειρά κατάργησης των πινάκων δεν είναι τυχαία.