



ΣΧΟΛΗ
ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ ΣΤΙΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Ακ. έτος 2018-2018

6ο Εξάμηνο, Σχολή ΗΜ&ΜΥ

Ομάδα

Βεργοπούλου Έλενα
(Α.Μ: 03116723)

Λαμπρογεώργου Φωτεινή
(Α.Μ: 03116027)

Παπαγεωργίου Ιάσων- Λάζαρος
(Α.Μ: 03114034)

A. ΕΙΣΑΓΩΓΗ

Στα πλαίσια του παρόντος Project αναπτύξαμε σε σχεσιακό σύστημα τη βάση της βιβλιοθήκης του Πολυτεχνείου σύμφωνα με όσα περιγράφονται στην Άσκηση 1 και με βάση την προτεινόμενη λύση που δόθηκε στη σελίδα του μαθήματος. Η εφαρμογή μας αναπτύχθηκε μέσω της πλατφόρμας phpMyAdmin.

B. ΥΛΟΠΟΙΗΣΗ

1. Δημιουργία Πινάκων

Δημιουργούμε πίνακες για κάθε στοιχείο που θέλουμε. Κατά την εισαγωγή ή την ενημέρωση κάποιων στοιχείων δεν επιτρέπεται η εισαγωγή οποιουδήποτε εύρους τιμών. Για κάθε πεδίο έχουμε συγκεκριμένο Data Type char, varchar, text, int το οποίο έχει συγκεκριμένο περιορισμό χαρακτήρων ώστε τα δεδομένα που εισάγονται να είναι πραγματικά και να μην υπερβαίνουν τα λογικά όρια. Έτσι όταν δημιουργούμε ή επεξεργαζόμαστε μία νέα εγγραφή δεν μπορούμε να εισάγουμε άλλο τύπο, ενώ ορίζοντας κάποια ιδιότητα με περιορισμό not null (όπως οι περισσότερες στη βάση μας) δεν επιτρέπουμε την εισαγωγή ή την ενημέρωση χωρίς τη συμπλήρωσή τους. Τέλος, σε κάθε περίπτωση ξεχωριστά ελέγχουμε κάτι διαφορετικό όπως θετικούς-αρνητικούς, ημερομηνίες, ηλικίες κλπ το οποίο αναγράφεται παρακάτω. Αυτά εξασφαλίστηκαν με την εντολή check, όπως φαίνεται στους παρακάτω κώδικες.

Πίνακας Μελών

Κώδικας:

```
CREATE table member (  
memberID int AUTO_INCREMENT NOT NULL,  
PRIMARY KEY (memberID),  
MFirst varchar(20) NOT NULL,  
MLast varchar(20) NOT NULL,  
Street varchar(30),  
Snumber int CHECK (Snumber > 0),  
PostalCode int CHECK (PostalCode >= 10000 AND PostalCode <=100000),  
MBirthDate date  
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	memberID	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	MFirst	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	3	MLast	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	4	Street	varchar(30)	latin1_swedish_ci	Yes	NULL			Change Drop More
<input type="checkbox"/>	5	Snumber	int(11)		Yes	NULL			Change Drop More
<input type="checkbox"/>	6	PostalCode	int(11)		Yes	NULL			Change Drop More
<input type="checkbox"/>	7	MBirthDate	date		Yes	NULL			Change Drop More

Ορίσαμε σαν primary key το memberID το οποίο είναι AUTO_INCREMENT άρα κάθε φορά που κάνουμε insert παίρνει αυτόματα την επόμενη τιμή και δεν μπορεί να είναι NULL. Έτσι κάθε φορά αυξάνεται και είναι μοναδικό. Ακόμη ορίσαμε σαν περιορισμούς να μην είναι NULL το όνομα και επίθετο, να έχουμε θετικό αριθμό για διεύθυνση και ταχυδρομικό κώδικα αυστηρά 5 ψηφίων.

Πίνακας Εκδοτών:

Κώδικας:

```
CREATE table publisher (  
pubName varchar (20) NOT NULL,  
PRIMARY KEY (pubname),  
estYear int CHECK (estYear >= 1685),  
Street varchar(20),  
Snumber int CHECK (Snumber > 0),  
PostalCode int CHECK (PostalCode >= 10000 AND PostalCode <=100000)  
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	pubName	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	2	estYear	int(11)		Yes	NULL			Change Drop More
<input type="checkbox"/>	3	Street	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop More
<input type="checkbox"/>	4	Snumber	int(11)		Yes	NULL			Change Drop More
<input type="checkbox"/>	5	PostalCode	int(11)		Yes	NULL			Change Drop More

Με primary key το pubName και περιορισμούς ξανά αντίστοιχα σε διεύθυνση και TK.

Πίνακας Βιβλίων

Κώδικας:

```
CREATE table book(  
ISBN varchar(17) NOT NULL,  
PRIMARY KEY (ISBN),  
Title varchar(40) NOT NULL,  
pubyear int CHECK (pubyear >= 1465),  
numpages int CHECK (numpages > 0 AND numpages <10000),  
pubname varchar(50),  
FOREIGN KEY (pubname) REFERENCES publisher(pubname)  
ON DELETE CASCADE  
ON UPDATE CASCADE  
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	ISBN	varchar(17)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 2	Title	varchar(40)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	pubyear	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/> 4	numpages	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/> 5	pubname	varchar(50)	latin1_swedish_ci		Yes	NULL			Change Drop More

Ορίσαμε σαν primary key το ISBN που δεν μπορεί να είναι NULL, όπως και ο τίτλος. Ακόμη σαν Foreign Key ορίστηκε το pubname που συνδέει το books με το publisher. Ακόμη ορίσαμε σαν περιορισμούς τη χρονολογία και τον αριθμό των σελίδων.

Πίνακας Συγγραφέων

Κώδικας:

```
CREATE table author (
authorid int PRIMARY KEY NOT NULL AUTO_INCREMENT,
AFirst varchar(20) not null,
ALast varchar(20) not null,
ABirthDate int
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	authorid	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	AFirst	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	ALast	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 4	ABirthDate	int(11)			Yes	NULL			Change Drop More

Ορίσαμε σαν primary key το authorID το οποίο είναι AUTO_INCREMENT και όχι NULL. Έτσι και αυτό κάθε φορά αυξάνεται και είναι μοναδικό.

Πίνακας Κατηγοριών:

Κώδικας:

```
CREATE table category (
CategoryName varchar(20) NOT NULL,
PRIMARY KEY (CategoryName),
SuperCatName varchar(20),
FOREIGN KEY (SuperCatName) REFERENCES category(CategoryName)
ON DELETE CASCADE
ON UPDATE CASCADE
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	CategoryName	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	2	SuperCatName	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop More

Ορίσαμε σαν primary key το CategoryName που δεν μπορεί να είναι NULL. Ακόμη σαν Foreign Key ορίστηκε το SuperCatName.

Πίνακας Αντιγράφων:

Κώδικας:

```
CREATE table copies (
ISBN varchar(17),
FOREIGN KEY (ISBN) REFERENCES book(ISBN)
ON DELETE CASCADE
ON UPDATE CASCADE,
copyNr int CHECK (copyNr <=100),
shelf int NOT NULL CHECK (shelf > 0 AND shelf < 10000),
PRIMARY KEY (ISBN,copyNr)
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	ISBN	varchar(17)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	2	copyNr	int(11)		No	None			Change Drop More
<input type="checkbox"/>	3	shelf	int(11)		No	None			Change Drop More

Ορίσαμε σαν primary key το ISBN και το copyNr για να ορίσουν μοναδικά το αντίγραφο.

Πίνακας Υπαλλήλων:

Κώδικας:

```
CREATE table employee (
empID int AUTO_INCREMENT NOT NULL,
PRIMARY KEY (empID),
EFirst varchar(20) not null,
ELast varchar(20) not null,
Salary float,
CHECK (salary >=200,00 AND salary <= 100000,00)
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	empID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	ELast	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	EFirst	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 4	salary	float			Yes	NULL			Change Drop More

Με primary key το empID που είναι AUTO_INCREMENT και περιορισμούς στον μισθό ο οποίος είναι δεκαδικός αριθμός μεταξύ κάποιων ορίων.

Πίνακας Μόνιμων Υπαλλήλων:

Κώδικας:

```
CREATE TABLE permanent_employee (
empID int,
HiringDate int,
FOREIGN KEY (empID) REFERENCES employee(empID)
ON DELETE CASCADE
ON UPDATE CASCADE,
PRIMARY KEY (empID, HiringDate),
CHECK (HiringDate>=1950)
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	empID	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	HiringDate	int(11)			No	None			Change Drop More

Ορίσαμε σαν primary key το empID και το HiringDate για να ορίσουν μοναδικά τον μόνιμο υπάλληλο.

Πίνακας Προσωρινών Υπαλλήλων:

Κώδικας:

```
CREATE TABLE temporary_employee (
empID int,
ContractNr int,
PRIMARY KEY (empID, ContractNr),
FOREIGN KEY (empID) REFERENCES employee(empID)
ON DELETE CASCADE
ON UPDATE CASCADE
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 empID 🔑	int(11)			No	None			Change Drop More
<input type="checkbox"/>	2 ContractNr 🔑	int(11)			No	None			Change Drop More

Ορίσαμε σαν primary key το empID και το ContractNr για να ορίσουν μοναδικά τον προσωρινό υπάλληλο

Πίνακας σχέσης 'δανείζεται':

Κώδικας:

```
CREATE TABLE borrows (
memberID int,
FOREIGN KEY (memberID) REFERENCES member(memberID)
ON DELETE CASCADE
ON UPDATE CASCADE,
ISBN varchar(17),
FOREIGN KEY (ISBN) REFERENCES book(ISBN)
ON DELETE CASCADE
ON UPDATE CASCADE,
copyNr int,
FOREIGN KEY (ISBN,CopyNr) REFERENCES copies(ISBN,CopyNr)
ON DELETE CASCADE
ON UPDATE CASCADE,
date_of_borrowing date,
date_of_return date,
PRIMARY KEY (memberID,ISBN,CopyNr,date_of_borrowing)
);
```

```
CREATE INDEX date_of_borrowing
ON borrows (date_of_borrowing);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 memberID 🔑	int(11)			No	None			Change Drop More
<input type="checkbox"/>	2 ISBN 🔑	varchar(17)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	3 copyNr 🔑	int(11)			No	None			Change Drop More
<input type="checkbox"/>	4 date_of_borrowing 🔑	date			No	None			Change Drop More
<input type="checkbox"/>	5 date_of_return	date			Yes	NULL			Change Drop More
<input type="checkbox"/>	6 Rating	int(11)			No	None			Change Drop More

Ορίσαμε σαν primary key το memberID, ISBN, copyNr και date_of_borrowing για να ορίσουν μοναδικά έναν δανεισμό από ένα μέλος μια συγκεκριμένη μέρα για ένα συγκεκριμένο αντίγραφο. Ακόμη το ISBN, copyNr λειτουργεί σαν Foreign Key για τον πίνακα βιβλίων και αντιγράφων. Επίσης για τη σωστή λειτουργία του query ορίσαμε το date_of_borrowing ως index.

Πίνακας σχέσης 'ανήκει':

Κώδικας:

```
CREATE TABLE belongs_to (  
  ISBN varchar(17),  
  FOREIGN KEY (ISBN) REFERENCES book (ISBN)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  categoryName varchar(20),  
  PRIMARY KEY (ISBN, categoryName),  
  FOREIGN KEY (categoryName) REFERENCES category(categoryName)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE  
);
```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 ISBN	varchar(17)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	2 categoryName	varchar(20)	latin1_swedish_ci		No	None			Change Drop More

Ορίσαμε σαν primary key το memberID, categoryName για να ορίσουν μοναδικά το ανήκει.
Ακόμη το categoryName λειτουργεί σαν Foreign Key για τον πίνακα κατηγοριών.

Πίνακας υπενθύμισης:

Κώδικας:

```
CREATE TABLE reminder (  
  empID int ,  
  FOREIGN KEY (empID) REFERENCES employee(empID)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  memberID int,  
  FOREIGN KEY (memberID) REFERENCES member(memberID)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  ISBN varchar(17),  
  FOREIGN KEY (ISBN) REFERENCES book (ISBN)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  copyNr int,  
  FOREIGN KEY (ISBN, CopyNr) REFERENCES copies (ISBN, CopyNr)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  date_of_borrowing date,  
  FOREIGN KEY (date_of_borrowing) REFERENCES borrows (date_of_borrowing)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE,
```



```

date_of_reminder date,
CHECK (copyNr <=100 AND date_of_return>=date_of_borrowing),
PRIMARY KEY (empID, memberID, ISBN, copyNr, date_of_borrowing,
date_of_reminder)
);

```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 empID	int(11)			No	None			Change Drop More
<input type="checkbox"/>	2 memberID	int(11)			No	None			Change Drop More
<input type="checkbox"/>	3 ISBN	varchar(17)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	4 copyNr	int(11)			No	None			Change Drop More
<input type="checkbox"/>	5 date_of_borrowing	date			No	None			Change Drop More
<input type="checkbox"/>	6 date_of_reminder	date			No	None			Change Drop More

Ορίσαμε σαν primary key το memberID, empID ISBN, copyNr και date_of_borrowing για να ορίσουν μοναδικά μια υπενθύμιση από έναν υπάλληλο σε ένα μέλος μια συγκεκριμένη μέρα για ένα συγκεκριμένο αντίγραφο. Ακόμη το ISBN, copyNr, memberID και date_of_borrowing λειτουργούν σαν Foreign Key για τον πίνακα μελών, δανεισμού και αντιγράφων.

Πίνακας σχέσης 'γράφτηκε από':

Κώδικας:

```

CREATE TABLE written_by (
ISBN varchar(17) ,
FOREIGN KEY (ISBN) REFERENCES book (ISBN)
ON DELETE CASCADE
ON UPDATE CASCADE,
authorid int AUTO_INCREMENT,
PRIMARY KEY (ISBN, authorID),
FOREIGN KEY (authorid) REFERENCES author (authorid)
ON DELETE CASCADE
ON UPDATE CASCADE
);

```

PhpMyAdmin:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 ISBN	varchar(17)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	2 authorid	int(11)			No	None	AUTO_INCREMENT		Change Drop More

Ορίσαμε σαν primary key το ISBN, authorID για να ορίσουν μοναδικά το γράφτηκε. Ακόμη το authorID λειτουργεί σαν Foreign Key για τον πίνακα συγγραφέων.

Τέλος, οι αναφορικές ακεραιότητες (referential integrities) έχουν ιδιαίτερη σημασία κατά τη διαγραφή μιας εγγραφής. Όταν για παράδειγμα διαγραφεί ένα μέλος διαγράφονται και όλοι οι δανεισμοί. Αυτό το εξασφαλίζουμε μέσω των foreign keys που συνδέει τις 2 σχέσεις να είναι on_delete cascade μέσα στο PhpMyAdmin. Αυτό εξασφαλίζεται για όλα τα ξένα κλειδιά.

2. Εισαγωγή – Ενημέρωση – Διαγραφή

Υλοποιήσαμε τις δυνατότητες εισαγωγής, ενημέρωσης και διαγραφής στοιχείων από τη βάση μέσω των συναρτήσεων (και αντίστοιχων αρχείων) save, update και delete για κάθε πίνακα της βάσης μέσα στις οποίες υπάρχουν τα sql ερωτήματα insert, update και delete αντίστοιχα. Για να υποστηρίξουμε τους περιορισμούς αναφορικής ακεραιότητας κατά τις αλλαγές, όπως είπαμε παραπάνω τα ξένα κλειδιά τα ορίζουμε ως on delete cascade έτσι ώστε όταν διαγράφεται η εγγραφή μιας σχέσης, να μην έχει μείνει σε καμία άλλη σχέση η οποία είχε εξάρτηση απ' αυτό.

Ιδιαίτερη περίπτωση ενημέρωσης αποτέλεσε η επιλογή για δανεισμό ενός βιβλίου. Συγκεκριμένα για να το υλοποιήσουμε πρέπει αρχικά να γνωρίζουμε σε κάποιον πίνακα την διαθεσιμότητα κάθε βιβλίου που έχουμε.

Ακολουθήσαμε τα εξής βήματα:

1. Δημιουργία πίνακα διαθεσιμότητας κάθε βιβλίου με στήλες τα ISBN και τον αριθμό αντιγράφου και τον αριθμό των αντιτύπων που είναι διαθέσιμα για δανεισμό:

```
CREATE table sumbooks
AS SELECT DISTINCT ISBN, COUNT(CopyNr) as CopyNum
FROM copies
GROUP BY ISBN
```

2. Δημιουργία όψης που αποθηκεύει πόσα από τα δανεισμένα βιβλία δεν έχουν επιστραφεί:

```
Creat vew b as
SELECT ISBN
From borrows
Where borrows.date_of_return IS NULL
```

3. Δημιουργία όψης που αποθηκεύει όλα τα δανεισμένα βιβλία μειώνοντας κατά ένα το αριθμό των αντιτύπων σε σχέση με πόσα έχουν:

```
Create table borrowed as
SELECT s.ISBN,
CopyNum - 1 as leftovers
FROM sumbooks AS s
JOIN b ON b.ISBN= s.ISBN
```

4. Ανανέωση του πίνακα sumbooks με βάση αυτό τον αριθμό βιβλίων που μένουν εν τέλει και είναι διαθέσιμα:

```
UPDATE sumbooks
JOIN borrowed ON sumbooks.ISBN=borrowed.isbn
SET sumbooks.copynum= borrowed.leftovers
```

Κάθε φορά που έχουμε έναν δανεισμό λοιπόν γίνονται UPDATE στους πίνακες borrows (με τα στοιχεία βιβλίων και μελών), sumbooks (με τα αντίτυπα που μένουν), borrowed (με τα δανεισμένα αντίτυπα) καθώς και στην ανανεώσιμη όψη b.

Ενδεικτικά παραθέτουμε την ανανέωση του πίνακα δανεισμού σύμφωνα με τις πληροφορίες που παίρνει από τους χρήστες που επιλέγουν να δανειστούν:

```
INSERT INTO borrows(memberID, ISBN, copyNr, date_of_borrowing,  
date_of_return)  
VALUES ($memberID, $ISBN, $copyNr, $date_of_borrowing, NULL);
```

Καθώς και την ανανέωση του πίνακα με τα sumbooks που κρατάει τα αντίτυπα που μένουν εν τέλει στη βάση και είναι διαθέσιμα:

```
UPDATE sumbooks  
SET sumbooks.copyNum = sumbooks.copyNum - 1
```

Περισσότερα για το δανεισμό θα αναλυθούν και παρακάτω στο κομμάτι των triggers.

3. Όψεις

Στη συνέχεια πέρα από τις παραπάνω υλοποιήσαμε αρκετές όψεις, είτε ενημερώσιμες είτε μη ενημερώσιμες. Ουσιαστικά, οι όψεις αποτελούν virtual tables που χρησιμοποιούν τις εγγραφές από έναν ή περισσότερους πίνακες και ορίζονται όπως επιθυμούμε εμείς. Χρησιμοποιούνται για τον περιορισμό δεδομένων, καθώς ορίζουμε εμείς τι θα φαίνεται και τι όχι, την ασφάλεια, καθώς υπάρχουν πίνακες που δεν είναι επεξεργάσιμοι, τη δημιουργία νέων πινάκων - κανόνων που αυξάνουν την αποδοτικότητα και την απλότητα. Γενικά είναι μια μορφή που καταλαμβάνει λιγότερο χώρο και προσφέρει μια πιο εύκολη μορφή για υπολογισμούς και δημιουργία queries.

Ενημερώσιμη Όψη:

```
CREATE VIEW permanent_view AS  
SELECT employee.ELast,employee.EFirst, permanent_employee.HiringDate  
FROM employee JOIN permanent_employee ON employee.EmpID =  
permanent_employee.EmpID
```

Είναι ενημερώσιμη καθώς ο όρος from περιέχει μόνο μία σχέση βάσης δεδομένων, ο όρος select περιέχει μόνο τα ονόματα των ιδιοτήτων της σχέσης και δεν έχει οποιεσδήποτε εκφράσεις, συνοπτικές συναρτήσεις ή προδιαγραφή distinct, οποιαδήποτε ιδιότητα δεν αναφέρεται στο select δεν έχει περιορισμό not null και δεν είναι μέρος ενός πρωτεύοντος κλειδιού και τέλος το ερώτημα δεν έχει κανέναν όρο group by ή having. Χρησιμοποιούμε αυτή την όψη ώστε να μπορεί ένας υπάλληλος να δει όλους τους μόνιμους υπαλλήλους και την ημερομηνία πρόσληψής τους χωρίς να μπορεί να δει το μισθό τους. Έτσι συνδέει μέσω του join αυτούς τους δύο πίνακες και δείχνει τις πληροφορίες που επιθυμούμε. Το view αυτό είναι επεξεργάσιμο, δηλαδή δίνει όλες τις δυνατότητες (delete, update, insert) που δίνει και ένα κανονικό table.

Μη Ενημερώσιμη Όψη:

```
CREATE VIEW temporary_view AS
SELECT employee.ELast, employee.EFirst, temporary_employee.ContractNr
FROM employee JOIN temporary_employee ON employee.EmpID =
temporary_employee.EmpID
ORDER BY Elast ASC
```

Προβάλλει σε λίστα το όνομα και επίθετο κάθε προσωρινού υπαλλήλου με αλφαβητική σειρά καθώς και το Contract Number αλλά όχι το μισθό πάλι. Αυτή δεν είναι ενημερώσιμη διότι ο όρος from δεν περιέχει μόνο μία σχέση αλλά ένωση (join) 2 σχέσεων. Με αυτήν την όψη εξυπηρετούνται μόνο σκοπούς προβολής καθώς δεν είναι επεξεργάσιμη.

Χρησιμοποιήσαμε τις 2 παραπάνω όψεις, και τις παραθέσαμε ενδεικτικά, καθώς τις θεωρήσαμε χρήσιμες για τον διαχειριστή της βάσης και δεν είχαμε πουθενά αλλού συγκεντρωμένα τα στοιχεία όλων των υπαλλήλων σύμφωνα και με τις θέσεις τους. Φτιάξαμε όμως και άλλες για τα μέλη αλλά και τα βιβλία.

4. Triggers

Δημιουργήσαμε triggers, δηλαδή ενέργειες που θα πρέπει να πραγματοποιούνται αυτόματα όταν γίνονται κάποιες αλλαγές στη βάση.

Το πρώτο είναι το παρακάτω:

```
DELIMITER $$
```

```
CREATE TRIGGER overdue
BEFORE INSERT ON borrows
FOR EACH ROW
BEGIN
IF ( ((SELECT COUNT(*) FROM borrows WHERE memberID = new.memberID AND
date_of_return IS NULL AND
DATEDIFF(CURRENT_TIMESTAMP, date_of_borrowing) > 30) > 0) OR ((SELECT
COUNT(*) FROM borrows WHERE memberID = new.memberID AND date_of_return
IS NULL) >= 5) OR ((SELECT copyNum FROM sumbooks WHERE ISBN = new.ISBN)
< 1))
THEN SIGNAL SQLSTATE "45000"
      SET MESSAGE_TEXT = "ERROR! You can't borrow book either because
you are overdue or because it's unavailable";
END IF;
END $$
DELIMITER ;
```

Με το trigger αυτό δημιουργούμε ουσιαστικά 3 διαφορετικές συνθήκες που δεν επιτρέπουν στο μέλος να δανειστεί το βιβλίο που επέλεξε:

1. Να έχει δανειστεί από ένα και πάνω βιβλία και να μην τα έχει επιστρέψει σε περισσότερες από 30 ημέρες
2. Να έχει δανειστεί 5 βιβλία
3. Να μην υπάρχουν άλλα διαθέσιμα αντίγραφα (το οποίο το ελέγχουμε μέσω ενός πίνακα που έχουμε φτιάξει ο οποίος κρατάει κάθε φορά τα διαθέσιμα αντίγραφα)

Το δεύτερο είναι το εξής:

```
DELIMITER $$
```

```
CREATE TRIGGER diagrafimelous
BEFORE DELETE ON member
FOR EACH ROW
BEGIN
IF (SELECT COUNT(*) FROM borrows WHERE memberID = old.memberID AND
date_of_return IS NULL) > 0
THEN SIGNAL SQLSTATE "45000"
      SET MESSAGE_TEXT = "You can't delete this member because he/she
has to return a book";
END IF;
END $$
```

Μέσω αυτού του trigger εξασφαλίζεις ότι δεν θα μπορεί να διαγραφεί ένα μέλος το οποίο έχει δανεισμένο κάποιο βιβλίο. Ελέγχεις για το συγκεκριμένο memberID αν υπάρχει στη βάση με τους δανεισμούς κάποιο βιβλίο χωρίς ημερομηνία επιστροφής και έτσι δεν επιτρέπεται η διαγραφή του.

5. Queries

Έχουμε φτιάξει τα εξής queries τα οποία έχουν συνδεθεί με το UI:

1. **Ταξινόμηση (order by)** είτε κατά αλφαβητική σειρά (ονομάτων και τίτλων) είτε κατά αριθμητική σειρά (έτους, αριθμό σελίδων). Ενδεικτικά παρατίθενται τα εξής:

```
SELECT Title
FROM book
ORDER BY title ASC
```

```
SELECT Title, pubYear
FROM book
ORDER BY pubYear DESC
```

2. **Συνένωση δύο πινάκων(join)** ώστε να μας εμφανίζει τα "Active Members" και τα στοιχεία τους, δηλαδή τα μέλη που έχουν δανειστεί κάποιο βιβλίο, το ονοματεπώνυμό τους, το ISBN του βιβλίου και την ημερομηνία δανεισμού αυτού του βιβλίου.

```

Select member.MFirst,member.MLast, borrows.date_of_borrowing,
borrows.ISBN
FROM member
join borrows on member.memberID = borrows.memberID

```

Ακόμη, ενώσαμε τους πίνακες book, borrows, category ώστε να εμφανίζουμε τα top 5 βιβλία αλλά και τις κατηγορίες τους και τίτλο και ISBN καθώς και πόσες φορές έχουν δανειστεί ως εξής:

```

SELECT book.ISBN, book.title, C.categoryName, COUNT(book.ISBN)
FROM book
LEFT JOIN borrows ON borrows.ISBN = book.ISBN
LEFT JOIN belongs_to AS C ON C.ISBN=book.ISBN
GROUP BY (book.ISBN), (C.categoryName)
ORDER BY COUNT(book.ISBN) DESC
LIMIT 5;

```

3. **Ομαδοποίηση πινάκων (group by)** ώστε να μας εμφανίζει το κάθε μέλος πόσα βιβλία έχει δανειστεί μέσω των πινάκων μελών και δανεισμού ώστε να διαμορφώσουμε το κουμπί "Borroed Books" για κάθε μέλος.

```

Select MFirst,MLast,count(*) BookCount
from members
join borrows on member.memberId = borrows.memberId
group by member.memberId,MFirst,MLast
order by BookCount

```

4. **Ομαδοποίησης με περιορισμό (group by με having)** ώστε να μας εμφανίζει πόσες φορές έχει δανειστεί ένα συγκεκριμένο βιβλίο ώστε να διαμορφώσουμε το κουμπί popularity.

```

Select title, count(*)
from borrows as bo,book as b
where bo.ISBN=b.ISBN
group by bo.ISBN
HAVING bo.ISBN=' '

```

5. **Χρήση συναθροιστικών συναρτήσεων (aggregate query)** ώστε να μας εμφανίζει πόσα βιβλία έχουμε σε κάθε κατηγορία μετρώντας το ISBN τους.

```

SELECT
    categoryName,
    COUNT (ISBN)
FROM belongs_to
GROUP BY
    categoryName;

```

6. Χρήση εμφωλευμένων ερωτημάτων (nested query) ώστε να μας εμφανίζει ποια βιβλία δεν έχει επιστρέψει το κάθε μέλος.

```
SELECT Title
FROM book
WHERE book.ISBN= (
    SELECT ISBN
    FROM borrows
    WHERE memberID=' ' AND date_of_return IS NULL
);
```

6. Σχόλια

1. Πλεονεκτήματα

MySQL:

- Εύκολη και δωρεάν εγκατάσταση και χρήση
- Χρήση διαφορετικών γλωσσών προγραμματισμούς
- Υποστήριξη του συντακτικού της γλώσσας SQL
- Υποστήριξη ενσωματωμένων συναρτήσεων και πράξεων
- Υποστήριξη εργαλείων όπως η διανομή XAMPP

XAMPP:

- Εύκολη και δωρεάν εγκατάσταση και χρήση
- Ευκολία διασύνδεσης με την πλατφόρμα MySQL
- Υποστήριξη της γλώσσας PHP

2. Μειονεκτήματα

MySQL

- Περιορισμένο εύρος προσφερόμενων τύπων δεδομένων
- Αδυναμία υποστήριξης δυναμικών ελέγχων στα CHECKS (πχ χρήση συναρτήσεων)
- Αδυναμία χειρισμού απροσδιόριστων εξαιρέσεων

XAMPP

- Διεπαφή χρήστη με αρκετές ελλείψεις
- Περιέχει πολλά πακέτα λογισμικού τα οποία δεν συνέβαλαν στον σκοπό της εργασίας μας
- Δυσκολία στην χρήση και την εκμάθηση σε μικρό χρονικό διάστημα

3. Προκλήσεις

Χρησιμοποιήσαμε το περιβάλλον αυτό γιατί είχαμε πολλές δυνατότητες ανάπτυξης καθώς και εκμάθησης της PHP διαδικτυακά. Οι προκλήσεις προέρχονται από την έλλειψη εξοικείωσης με τη γλώσσα αυτή. Ακόμη, ήταν σύνθετο να συνδεθεί όλο το backend κομμάτι που είχαμε υλοποιήσει σε sql με το UI. Έπρεπε κάθε ενέργεια που είχαμε επιλέξει να υλοποιείται να συνδεθεί με κάποιο αντίστοιχο button στο interface και να μπορεί να οδηγείται σε κάποιο διαφορετικό tab ώστε να γίνεται το αποτέλεσμα. Ακόμη, πρόκληση αποτελεί η χρήση πληροφοριών που δίνει ο χρήστης σαν input και πρέπει να χρησιμοποιηθούν για κάποια ενέργεια. Τέλος, πρόκληση αποτελεί η χρήση triggers που είναι σημαντικό να υπάρχουν καθώς η MySQL δεν υποστηρίζει την χρήση συναρτήσεων στους δυναμικούς ελέγχους (CHECK CONSTRAINTS). Έτσι αφού αναλύσαμε τι πληροφορίες δεν θέλουμε να αποδέχεται η βάση μας έπρεπε να υλοποιηθούν τα αντίστοιχα triggers για να αποτρέπουν από αυτές τις εισαγωγές. Ακόμη, triggers είναι απαραίτητο να δημιουργηθούν για να μπορούμε να ελέγχουμε ενέργειες όπως ο δανεισμός, η διαγραφές και τα updates καθώς πρέπει κάθε φορά να ανανεώνονται διαφορετικοί πίνακες, να γίνονται έλεγχοι επίτρεψης και οι κατάλληλες ειδοποιήσεις στους χρήστες.