

## Project Βάσεις δεδομένων 2018-2019

Ομάδα 19

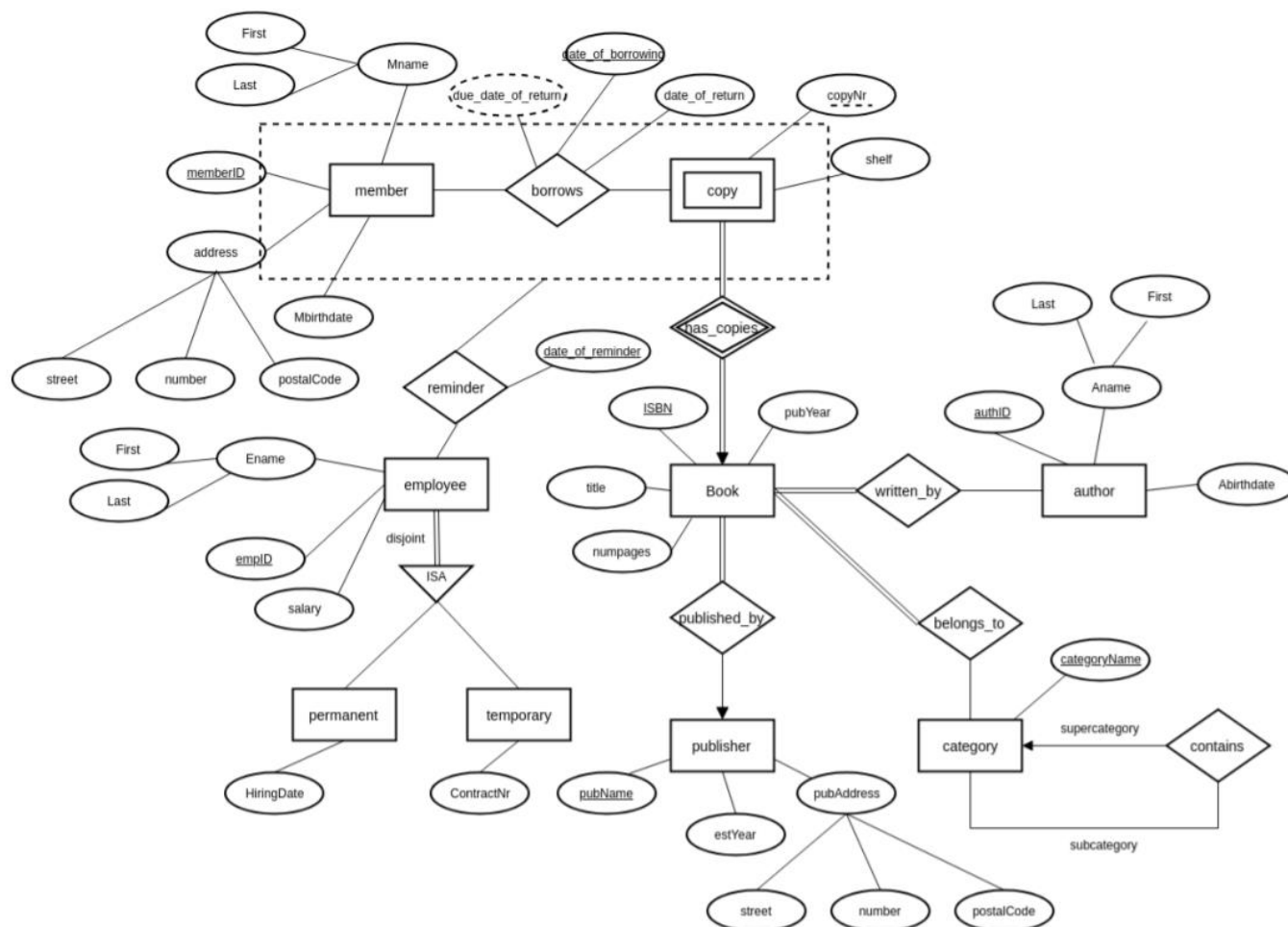
Ονοματεπώνυμο: Γεζεκελιάν Βικέν

AM: el16112

Ονοματεπώνυμο: Λυμπεράτος Βασίλειος

AM: el15034

Αρχικά, για το σχεδιασμό της βάσης μας, χρησιμοποιήσαμε το παρακάτω σχεσιακό μοντέλο, το οποίο πάρθηκε απο την ενδεικτική λύση της πρώτης άσκησης στο mycourses:



Έχει γίνει όμως μια προσθήκη στις στήλες των copies, όπου προστέθηκε η στήλη "Availability", για να ορίζει εάν είναι διαθέσιμο κάθε βιβλίο.

Σχεδιάσαμε τη βάση μας στο περιβάλλον rhrmyadmin, χρησιμοποιώντας Mysql καθώς και τις λειτουργίες που παρείχε το περιβάλλον. Το περιβάλλον ήταν φιλικό καθώς και παρείχε αρκετές διευκολύνσεις. Ο κώδικας για τη δημιουργία ολόκληρης της βάσης βρίσκεται στο αρχείο basic.sql. Όσον αφορά τους περιορισμούς που έχουμε πάρει:

Primary keys:

Σε αντικείμενα:

memberID στο member

ISBN στο book

authorID στο author

CatName στο category

(ISBN, copyNr) στο copies

EmpID στο employee, perm\_employee, temp\_employee

PubName στο publisher

Σε σχέσεις:

(ISBN, CatName) στο belongs\_to

(MemberID, ISBN, CopyNr, Date\_of\_Borrowing) στο borrows

(EmpID, ISBN, memberID, CopyNr, Date\_of\_Borrowing, Date\_of\_reminder) στο reminder

(ISBN, AuthorID) στο written\_by

Τα πεδία memberID, EmpID, ISBN είναι όλα auto-increment(Εδώ έχει γίνει ένα λάθος με το ISBN, καθώς δεν θα έπρεπε να είναι auto-increment, όμως νομίζαμε πως είναι ένας μοναδικός κωδικός τον οποίο χρησιμοποιεί μόνο η βιβλιοθήκη).

Foreign keys:

CatName στο belongs\_to από το category

PubName στο book απο το publisher

memberID στο member από το borrows

ISBN στο book από το borrows

(ISBN, CopyNr) στο copies από το borrows

SuperCatName στο category από το category

EmpID στο employee από το reminder

memberID στο member από το reminder

ISBN στο book από το reminder

(memberID, ISBN, CopyNr, Date\_of\_borrowing) στο borrows από το reminder

(ISBN, CopyNr) στο copies από το reminder

AuthorID στο author από το written\_by

Όλες οι συσχετίσεις με foreign keys είναι CASCADE, το οποίο σημαίνει ότι όποιες αλλαγές ή διαγραφές γίνουν σε ένα αντικείμενο στο οποίο αναφέρονται και άλλα αντικείμενα, οι αλλαγές αυτές επηρεάζουν και τα άλλα αντικείμενα.

Στη βάση έχουμε δημιουργήσει 2 όψεις. Η μία, "see\_all\_books" μας δείχνει όλα τα βιβλία τα οποία διαθέτει η βιβλιοθήκη, ενώ η δεύτερη, "copies\_available", μας δείχνει πόσα αντίτυπα από το κάθε βιβλίο είναι διαθέσιμα.

View 1:

```
CREATE ALGORITHM = UNDEFINED DEFINER = `root`@`localhost` SQL SECURITY DEFINER VIEW
```

```
`see_all_books` AS SELECT
```

```
`b`.`Title` AS `Title`,
```

```
`b`.`PubName` AS `PubName`,
```

```
`b`.`PubYear` AS `PubYear`,
```

```
`a`.`AFirst` AS `Afirst`,
```

```
`a`.`ALast` AS `Alast`
```

```
FROM
```

```
(
```

```
(`book` `b`
```

```
JOIN `author` `a`)
```

```

JOIN `written_by`
)
WHERE
(
(`b`.`ISBN` = `written_by`.`ISBN`) AND(
`a`.`AuthorID` = `written_by`.`AuthorID`
)
);

```

View 2:

```

CREATE ALGORITHM = UNDEFINED DEFINER = `root`@`localhost` SQL SECURITY DEFINER VIEW
`copies_available` AS SELECT
`b`.`Title` AS `Title`,
`b`.`ISBN` AS `ISBN`,
(
SELECT
COUNT(0)
FROM
`copies` `c`
WHERE
(
(`c`.`ISBN` = `b`.`ISBN`) AND(`c`.`Availability` = 1)
)
) AS `Available_Number`
FROM
`book` `b`;

```

Επίσης στη βάση έχουμε δύο triggers το “Book\_Borrowed”, όπου ανανεώνει την τιμή “Availability” του table copies απο 1 σε 0 όταν δανειστεί κάποιος το βιβλίο και το “Book\_Returned”, όπου ανανεώνει την τιμή “Availability” του βιβλίου από 0 σε 1 όταν επιστρέψει κάποιος το βιβλίο.

Trigger 1:

```

CREATE TRIGGER `Book_Borrowed` AFTER INSERT ON `borrows`
FOR EACH ROW UPDATE copies
SET copies.Availability = 0
WHERE copies.CopyNr=new.CopyNr AND copies.ISBN = new.ISBN

```

Trigger 2:

```

CREATE TRIGGER `Book_Returned` AFTER UPDATE ON `borrows`
FOR EACH ROW UPDATE copies
SET copies.Availability = 1
WHERE copies.CopyNr=new.CopyNr AND copies.ISBN=new.ISBN

```

Κάτι το οποίο δεν έχουμε υλοποιήσει στη βάση μας, είναι ο περιορισμός πως κάποιος μπορεί να δανειστεί έως 5 βιβλία. Για να υλοποιήσουμε αυτόν τον περιορισμό, θα έπρεπε να έχουμε προσθέσει μια στήλη `Books_not_returned` καθώς και ένα `trigger`, το οποίο εάν προσπαθούσε κάποιος ο οποίος έχει ήδη δανειστεί 5 βιβλία να δανειστεί και άλλο, θα τον ανάγκαζε να επιστρέψει πρώτα ένα άλλο βιβλίο το οποίο έχει ήδη δανειστεί. Ο κώδικας για την στήλη `Books_borrowed` υπάρχει παρακάτω στα queries μας.

Για τα queries που ζητήθηκαν, έχουμε φτιάξει 9 και μπορούν να υλοποιηθούν και μέσω του UI μας και είναι τα εξής:

**Query 1: Σε ποια υπερκατηγορία ανήκει το κάθε βιβλίο:**

```
SELECT
    b.Title,
    c.SuperCatName
FROM
    book b
INNER JOIN belongs_to be ON
    b.ISBN = be.ISBN
INNER JOIN category c ON
    be.CatName = c.CatName
```

**Query 2: Πόσα βιβλία χρωστάει το κάθε μέλος**

```
SELECT
    m.memberID,
    m.Mfirst,
    m.Mlast,
    COUNT(bo.MemberID) as Number_owed
FROM
    member AS m,
    borrows AS bo
WHERE
    m.MemberID = bo.MemberID AND bo.Date_of_return IS NULL
GROUP BY
    m.MemberID
```

**Query 3: Δείχνει όλα τα βιβλία με πάνω απο ένα αντίτυπο**

```
SELECT
    b.Title,
    COUNT(b.ISBN = c.ISBN) AS NumberOfCopies
FROM
    book b
INNER JOIN copies c ON
    b.ISBN = c.ISBN
GROUP BY
    b.ISBN
HAVING
    COUNT(b.ISBN = c.ISBN) > 1
```

**Query4: Ποιοι είναι οι 5 πιο σύγχρονοι συγγραφείς**

```
SELECT
    a.AFirst,
    a.ALast,
    a.ABirthDate
FROM
    author a
ORDER BY
    a.ABirthDate
DESC
LIMIT 5
```

**Query 5: Πότε στάλθηκαν οι υπενθυμίσεις, απο ποιον υπάλληλο και σε ποια μέλη**

```
SELECT
    e.EFirst,
    e.ELast,
    m.Mfirst,
    m.Mlast,
    r.Date_of_reminder
FROM
    employee e
INNER JOIN reminder r ON
    e.EmpID = r.EmpID
INNER JOIN member m ON
    m.MemberID = r.MemberID
ORDER BY
    r.Date_of_reminder DESC
```

**Query 6: Ποιοι έχουν δανειστεί τουλάχιστον ένα βιβλίο σε κάποια στιγμή**

```
SELECT
    m.Mfirst,
    m.Mlast,
    m.Mlast
FROM
    member AS m
WHERE EXISTS
    (
        SELECT
            *
        FROM
            borrows AS b
        WHERE
            m.MemberID = b.MemberID
    )
```

**Query 7: Τα ονόματα των μελών που τελειώνουν σε 'η'**

```
SELECT
    m.Mfirst,
    m.Mlast
FROM
    member AS m
WHERE
    (m.Mfirst LIKE '%η') OR (m.Mlast LIKE '%η')
```

Το 8<sup>ο</sup> και το 9<sup>ο</sup> query, προσφέρουν στον user τη δυνατότητα αναζήτησης βιβλίων με βάση τον συγγραφέα και τον εκδοτικό οίκο αντίστοιχα.

**Query 8:**

```
SELECT
    b.Title,
    b.PubYear,
    b.PubName
FROM
    book as b,
    written_by as wr
WHERE
    b.ISBN=wr.ISBN AND wr.AuthorID='$AuthorID'
ORDER BY
    b.PubYear
```

**Query 9:.**

```
SELECT
    b.Title,
    b.PubYear,
    b.Numpages
FROM
    book as b
WHERE
    b.PubName='$PubName'
ORDER BY
    b.PubYear
```

Όπου τα \$AuthorID και \$PubName είναι είσοδος που έχει δώσει προηγουμένως ο χρήστης μέσω ενός drop-down list.,

Για το User Interface που δημιουργήσαμε χρησιμοποιήσαμε αρχικά html για τον σχεδιασμό της ιστοσελίδας μας, καθώς και css για το γραφικό κομμάτι. Η σύνδεση μεταξύ βάσης και html έγινε με τη χρήση της php. Ολόκληρος ο κώδικας για html και php βρίσκεται στο αρχείο που παραδίδουμε.