

**CPE-344**

**DATABASE MANAGEMENT SYSTEMS AND PROGRAMMING II**

**Term Project**

**Prepared By:**

Yasser El Kabbout - 20131196

**Advisor**:

Lecturer Salahi Halil Altıncı

June 2016,

Nicosia

**TABLE OF CONTENTS**

[**1. TABLES CREATION** 3](#_Toc452748502)

[**2.** **CONSTRAINTS ON TABLES** 3](#_Toc452748503)

[**2.1 PRIMARY KEYS & FOREIGN KEYS** 3](#_Toc452748504)

[**2.2 NULL CONSTRATINTS** 3](#_Toc452748505)

[**3.** **FUNCTIONS** 3](#_Toc452748506)

[**3.1 IS\_BOOK\_AVAILABLE** 3](#_Toc452748507)

[**3.2 MEMBER\_HAS\_OVERDUE\_BOOK** 4](#_Toc452748508)

[**3.3 MAXIMUM\_BOOK\_LIMIT** 4](#_Toc452748509)

[**3.4 GET\_TOTAL\_BORROW\_COUNT** 5](#_Toc452748510)

[**4.TRIGGERS** 5](#_Toc452748511)

[**5. PROCEDURES** 6](#_Toc452748512)

[**5.1 BORROW\_A\_BOOK** 6](#_Toc452748513)

[**5.2 RETURN\_A\_BOOK** 6](#_Toc452748514)

[**6. JAVA APPLICATION** 7](#_Toc452748515)

# **1. TABLES CREATION**

The DDL regarding the tables creation was available on the Moodle’s course page. To access them, the exported database is available within the project submission.

# **CONSTRAINTS ON TABLES**

## **2.1 PRIMARY KEYS & FOREIGN KEYS**

All the primary keys and the foreign keys were implemented on the tables. They can be checked within the exported database.

## **2.2 NULL CONSTRATINTS**

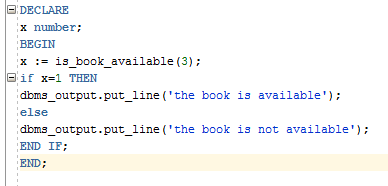
All the required NOT NULL constraints were implemented on the tables. They can be checked within the exported database.

# **FUNCTIONS**

## **3.1 IS\_BOOK\_AVAILABLE**

The required function was implemented and it is checking whether a book is available to be borrowed or not.

Books should be checked if they are borrowed by another members, or available.



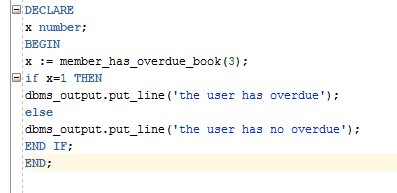
Snip 1 | is\_book\_available function in action

The function’s body is available within the exported database.

## **3.2 MEMBER\_HAS\_OVERDUE\_BOOK**

The required function was implemented and it is checking whether the member is having a due or not.

Dues should be checked before borrowing a new book.

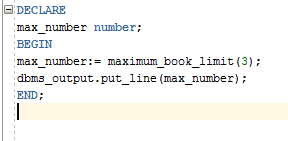


Snip 2 | member\_has\_overdue function in action

The function body is available within the exported database.

## **3.3 MAXIMUM\_BOOK\_LIMIT**

The required function was implemented and it is getting the maximum number of books a member can borrow.

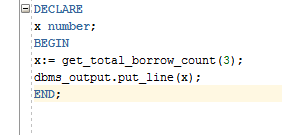


Snip 3 | maximum\_book\_limit function in action

The function’s body is available within the exported database.

## **3.4 GET\_TOTAL\_BORROW\_COUNT**

The required function was implemented and it is getting the total number of books a user is currently borrowing.



Snip 4 | get\_total\_borrow\_count function in action

The function’s body is available within the exported database.

# **4.TRIGGERS**

A trigger is a special kind of stored procedure that automatically executes when an event occurs in the database server. In this example, a trigger was implemented to fulfill the following requirements:

1. Check if a **book is available** on selected date (Use the stored function is\_book\_available). If it is not available, raise an error and prevent borrowing the book.
2. Check if the member has an **overdue book** (Use the stored function member\_has\_overdue). In case of a overdue or non-availability raise an error and prevent borrowing the book.
3. Check if the member reached the **maximum number of books** that he/she can borrow. If so, raise an error and prevent borrowing the book.

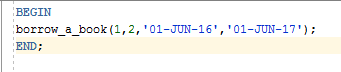
This trigger will be triggered before inserting a new row to the table transactions.

The Trigger’s body is available within the exported database.

# **5. PROCEDURES**

## **5.1 BORROW\_A\_BOOK**

The required procedure was implemented and it is inserting a new row to the transactions’ table. By adding a new row, the implemented trigger will be triggered as well.

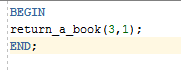


Snip 5 | borrow\_a\_book function in action

The procedure’s body is available within the exported database.

## **5.2 RETURN\_A\_BOOK**

The required procedure was implemented and it is performing an update on the transactions table. It is updating the return\_date column when a member is returning the book he withdrawed.



Snip 6 | return\_a\_book in action

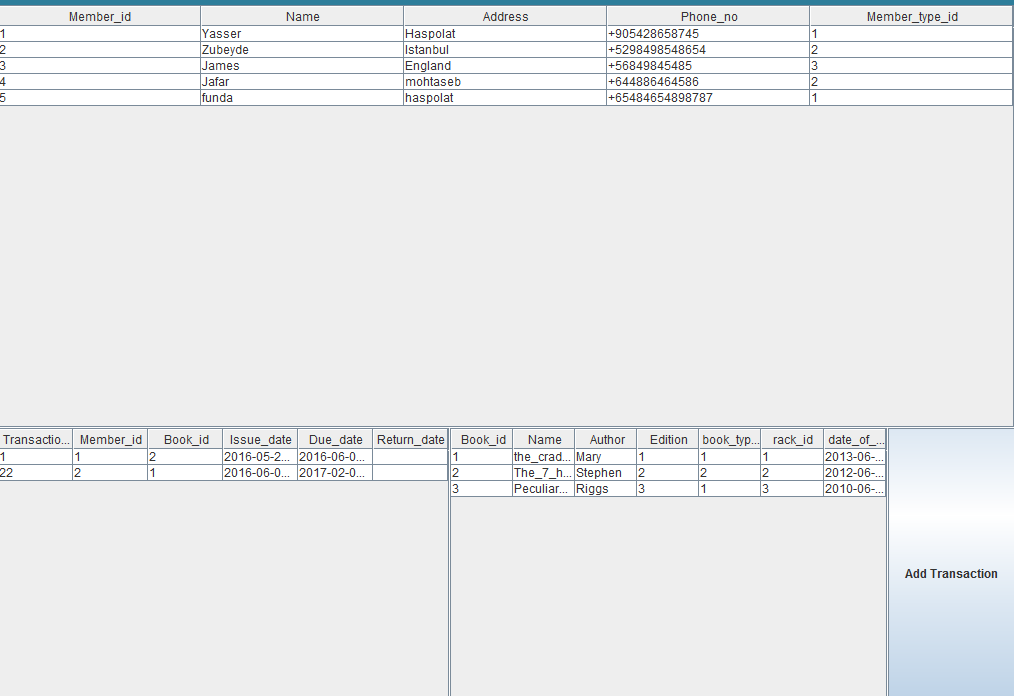
The procedure’s body is available within the exported database.

# **6. JAVA APPLICATION**

A simple application was implemented by using the JAVA programming language. The NetBeans IDE was used as the development environment.

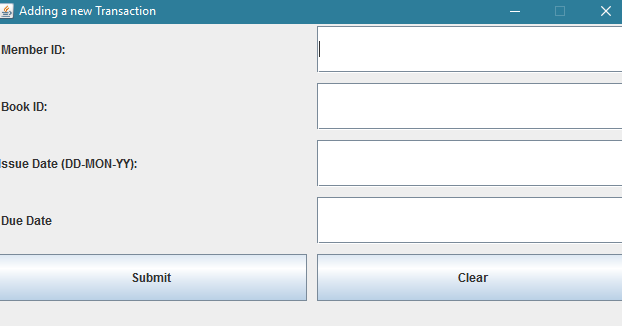
The graphical user interface was implemented by using JFrames.

The main interface is as follows:



Snip 7 | Main user Interface

By clicking on the *Add Transaction* button, the following interface will be available:



Snip 8 | Adding a transaction

For this program, three classes where used and they are as follow:

1. *mainProgram* Class used as the main user interface.
2. *javaConnectDb* Class used to establish connections with the oracle database.
3. *addTransaction* Class used to add a new transaction.

To connect to the database, the JDBC driver was used.

The full source code of this application is attached as well.