

Faculty of Computers and Artificial Intelligence

Today

IS211 Database Systems I - 2023

University Library Management System

|  |  |
| --- | --- |
| Name | ID |
| Aya Ali Hassan Ali | 20210083 |
| Khadija Ayman Elsayed Eshra | 20210540 |
| Sara Tamer Bihery | 20210155 |
| Sara Walid Farouk Mohamed | 20210495 |
| Sohaila Abdelazim Khalifa | 20210492 |

**Table of contents**

Section: 29,30

Eng. Nancy Saeed

[Introduction 2](#_Toc135753440)

[Assumptions 2](#_Toc135753441)

[ERD 3](#_Toc135753442)

[Conceptual Diagram 4](#_Toc135753443)

[Physical Diagram 4](#_Toc135753444)

[SQL Queries 5](#_Toc135753445)

[System requirements & Functions Proposed 9](#_Toc135753446)

[Bonus 10](#_Toc135753447)

[Tools 10](#_Toc135753448)

[Program reference 10](#_Toc135753449)

# Introduction

In this system we are trying to simulate a University Library System by providing a system database with different functionalities such as add, delete, update and display information, about the system entities: admin, student, book. Also, the user can search for any book that satisfy certain criteria, such as search for certain author to see all his books. The system also generates reports when used by an admin to view several statistics such as the number of students using the system, number of the books available in the library. Moreover, it shows the number of students in each department and number of books grouped and ordered by publishing year. Our project provides these functions in friendly interface which allows the user to act easily with.

# Assumptions

1. A Book is written by only one author.
2. An author writes many books.
3. A Book is published by only one publisher.
4. A Publisher can publish many books.
5. An Admin can own only one Account.
6. A Student can own only one Account.
7. Student can borrow many Books.
8. A Book could be borrowed by many Students.
9. Student & Admin login with their ID provided by University.

# C:\Users\Lenovo\Downloads\ER - F_.pngERD

For better Visualization please refer to:

<https://shorturl.at/pHU04>

# C:\Users\Lenovo\Downloads\Database ER diagram - Physical (1).pngConceptual Diagram

# Physical Diagram

For better Visualization please refer to:

<https://shorturl.at/kWZ47>

# SQL Queries

Attached as: UniveristyLibarary.sql



CREATE TABLE Account (

ACC\_ID int AUTO\_INCREMENT primary key,

Email varchar(50),

Password varchar(8)

);

CREATE TABLE Admin (

A\_ID int,

FirstName varchar(50),

LastName varchar(50),

City varchar(50),

Street varchar(50),

PhoneNumber varchar(50),

Gender varchar(8),

Role varchar(50),

ACC\_ID int,

PRIMARY KEY (A\_ID),

CONSTRAINT FK\_Account

FOREIGN KEY (ACC\_ID)

REFERENCES Account(ACC\_ID)

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE Book (

ISBN int,

Title varchar(70),

Category varchar(70),

Languge varchar(70),

Edition varchar(70),

PublisingYear varchar(70),

PRIMARY KEY (ISBN)

);

CREATE TABLE Student (

S\_ID int,

FirstName varchar(50),

LastName varchar(50),

City varchar(50),

Street varchar(50),

PhoneNumber varchar(50),

Gender varchar(8),

Department varchar(50),

ACC\_ID int,

PRIMARY KEY (S\_ID),

CONSTRAINT FK\_SAccount

FOREIGN KEY (ACC\_ID)

REFERENCES Account(ACC\_ID)

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE Publisher (

Name varchar(50),

City varchar(50),

Country varchar(50),

Street varchar(50),

ISBN int,

CONSTRAINT P\_Book

FOREIGN KEY (ISBN)

REFERENCES Book(ISBN)

ON DELETE CASCADE

ON UPDATE CASCADE,

PRIMARY KEY (Name, ISBN)

);

CREATE TABLE Author (

Name varchar(50),

Nationality varchar(50),

ISBN int,

CONSTRAINT A\_Book

FOREIGN KEY (ISBN)

REFERENCES Book(ISBN)

ON DELETE CASCADE

ON UPDATE CASCADE,

PRIMARY KEY (Name, ISBN)

);

CREATE TABLE Borrow (

S\_ID int,

ISBN int,

IssueDate Date,

ReturnDate Date,

CONSTRAINT S\_Borrow

FOREIGN KEY (S\_ID)

REFERENCES Student(S\_ID)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT B\_Borrow

FOREIGN KEY (ISBN)

REFERENCES Book(ISBN)

ON DELETE CASCADE

ON UPDATE CASCADE,

PRIMARY KEY (S\_ID, ISBN)

);SQL quarries to test the program:

File has attached as: SQL insert.txt



# System requirements & Functions Proposed

1. 2 insert statements on 2 different tables:

* Insert new student in Student table.
* Insert new book in Book Table.
* Insert new admin in Admin Table.
* Insert new Account in Account Table.
* Insert new Author in Author Table.

1. 2 delete statements on 2 different tables (with conditions).

* Delete student from Student table.
* Delete book from Book table. (Using on delete cascade)

1. 2 update statements on 2 different tables (with condition).

* Update Student data.
* Update Admin data.
* Update book information**.**

1. Select data from any table(s) of the database.

* Select books and display them.

1. Select data that involves more than one table of the database (using joins).

* Displays the author with all books he wrote.
* When someone makes login need to check two different tables to know whose account belongs to.

1. Meaningful report - by using aggregate function.
2. Number of Books.
3. Number of Admins.
4. Number of Students.
5. Number of Students in each department.
6. Number of Books published in each year.

# Bonus

* Included Weak Entity.
* Generated 1 meaningful report.
* Implemented GUI.

# Tools

* Drawio: [draw.io (drawio.com)](https://www.drawio.com/)
* Lucid: https: [Documents (lucid.app)](https://lucid.app/documents#/documents?folder_id=recent)
* My SQL
* IntelliJ idea (java)

# Program reference

File has attached as: 20210492\_20210083\_20210540\_20210155\_20210495.file

