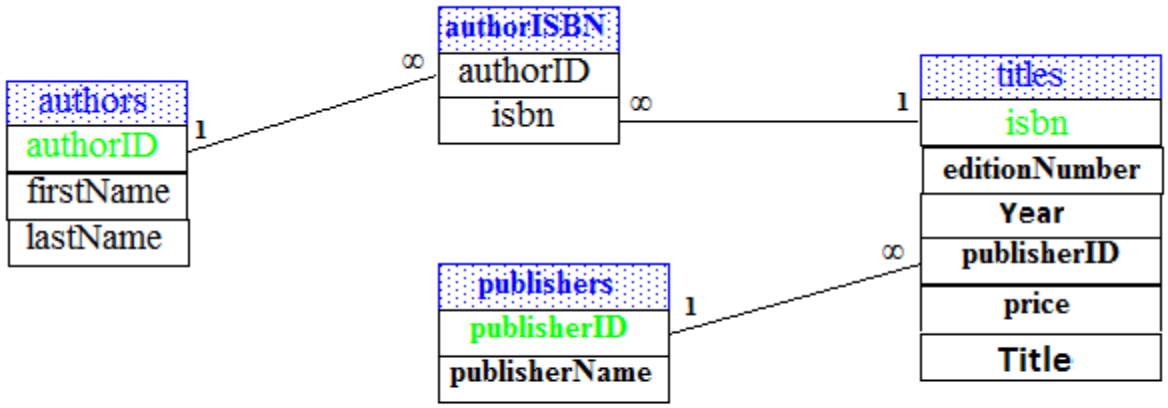
Northeastern University
College *of* Computer and Information Science

Department of Computer Science
Database Management Systems (CS 5200)
Instructor: Ahmed Ezzat

Project #1 (5 pts)

Project Overview:

The goal of this project is to use JDBC to create a Books database, populate it, and then execute different SQL statements to query or manipulate the Books database.



The Books Database Schema

In the above schema (E-R diagram), blue represents the name of the table, and **green** represents the primary key. The line between authors and authorISBN is one-to-many relationship (author can have many ISBNs). Line between authorISBN and titles tables is many-to-one (one title can have many authors). The sole purpose of the authorISBN table is to represent a many-to-many relationship between the authors and titles tables; an author can have many books and a book can have many authors.

Overview:

In this project, using JDBC application that access an installed RDBMS (you need to have an account), you will need to:

- Create the Books database tables as specified in the schema definition below.

- Initialize the different tables (at least 15 entries per table) appropriately: all fields cannot be null.
- Issue the following SQL statements. For queries print the results from java into your console:
- ~~Select all authors from the authors table. Order the Order the information alphabetically by the author's last name and first name~~.
- Select all authors from the authors table. Order the information alphabetically by the author's last name and first name.
- Select all publishers from the publishers table.
- Select a specific publisher and list all books published by that publisher. Include the title, year and ISBN number. Order the information alphabetically by title
- Add new Author
- Edit/Update the existing information about an author
- Add a new title for an author
- Add new publisher
- Edit/Update the existing information about a publisher

Schema Definition:

1. authors Table:

Field	SQL Type	Java Type	Description
authorID	INTEGER	integer	This is an *autoincremented* field. For each new record inserted in this table, the database automatically increments the **authorID** value to ensure that each record has a unique **authorID**. This field is the primary key for this table.
firstName	CHAR(20)	String	Author's first name.
lastName	CHAR(20)	String	Author's last name.

2. authorISBN Table:

Field	SQL Type	Java Type	Description
authorID	INTEGER	integer	The integer ID in this field must appear also in the **authors** table
isbn	CHAR(10)	String	The ISBN number for a book

3. titles Table:

Field	SQL Type	Java Type	Description
isbn	CHAR(10)	String	ISBN number of the book
title	VARCHAR2(500)	String	Title of the book
editionNumber	INTEGER	integer	Edition number of the book
Year	CHAR(4)	String	Year for this Edition.
publisherID	INTEGER	integer	Publisher's ID number. This value corresponds to an ID number in the **publishers** table.
price	NUMBER(8,-2)	float	Suggested retail price of the book

4. publishers Table:

Field	SQL Type	Java Type	Description
publisherID	INTEGER	integer	Publisher's ID number. This *autoincremented* integer is the table's primary-key field.
publisher-Name	CHAR(100)	String	The name of the publisher.

Programming Hints:

1. Creating the database is typically done in a SQL script (i.e., books.sql) which would be executed using SQL*PLUS tool from Oracle. For simplicity we will use JDBC to create the tables to avoid learning SQL*PLUS.

2. Be careful when adding a new title for an author. Remember that the book must have an entry in the **authorISBN** table.

3. Remember that no fields/columns are allowed to be NULL. This is specified in the create Table statement as follows:

   CREATE TABLE **student** (name CHAR(20) NOT NULL, birth year (CHAR(4), nationality CHAR(5));

4. Include one/two pages description of your project implementation. Remember to have good comments in your code. Also, include a hard copy of the java printout of the select statement results (cut-and paste from the screen). Finally, write clearly on the Front page the **absolute UNIX pathname** for your application to the grader to try (making sure RWX permissions for others).

5. Bring to class two complete copy sets of your project (one for the grader and the second for myself).