



## **INDIVIDUAL ASSIGNMENT 1**

**Shuya Chen**

**CIS467: DATA MANAGEMENT, WAREHOUSING, AND VISUALIZATION**

**Professor: Mikhail Lysyakov**

**Spring A 2025**

**Janu Due Tuesday, January 28, 2025 at 11:59 PM.**

### **Homework 1 (total 100 points)**

**Please paste diagrams and SQL code scripts into this Word document and upload this Word file with your answers to the HW1 assignment folder on Blackboard. Check the file after uploading it.**

1. (40 points) Create a database diagram in MySQL Workbench based on the following rules (choose the appropriate data type for each attribute). **Paste the diagram after this question.**

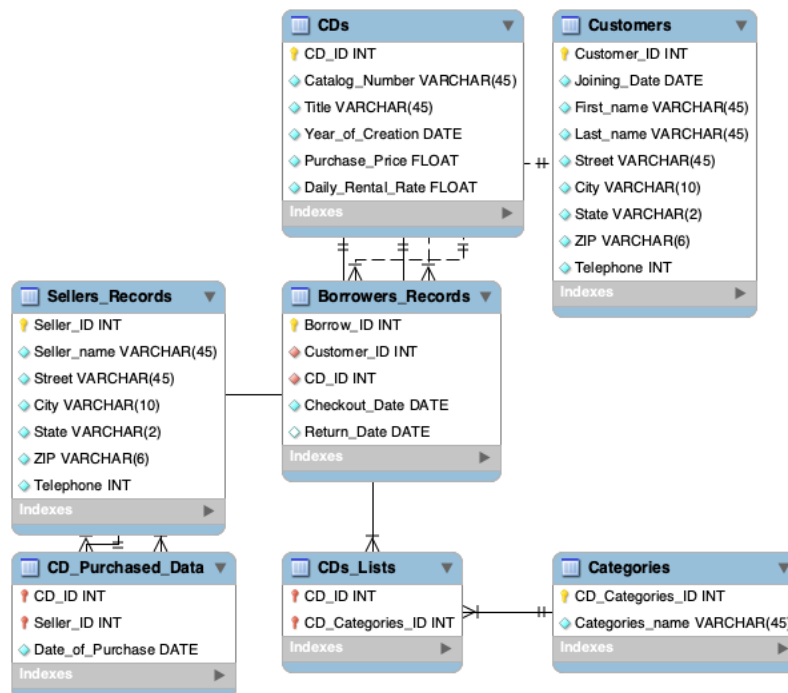
#### **AI Podcasts CD Rental Database**

The AI CD store asks you to create a database application system to assist with the administration of the store. The requirements collection and analysis phase has provided the following requirements for the database.

- The store stocks AI CDs that fall into the following categories - comedy, drama, education, artificial intelligence, nature and science fiction. Each category is identified by a unique identifier. A CD can belong to many categories while each category has many CDs. For this HW, please consider each CD as a unique physical CD with a unique CD identifier. For example, the same CD (for instance, with the same podcast titled “AI education for all”), can be printed 99 times or more and each copy of the CD is uniquely identified. Since customers will borrow each physical copy of the CD, each CD copy should be tracked even though it could be the same podcast with the same title.
- The data stored on the CDs for rent includes the CD number, catalog number, title, year of the store manager and uniquely identifies each CD. Think about how to show that each CD can belong to multiple categories.
- The data stored on each customer renting CD from the store includes customer number, date of joining the store, customer name (consists of first and last name as two separate attributes), address (consists of street, city, state, and zip code as 4 separate attributes), and one telephone number.

- Each customer may borrow an unlimited number of CDs, with the checkout date and the return date on each CD. Each physical copy of the CD can be borrowed by many customers over time. Each physical copy of the CD may be borrowed more than once by the same customer. Think about how to reflect this in the ER diagram.
- Data on sellers of CDs (from whom a store buys CDs) should also be stored, such as seller number, name, address (consists of street, city, state, and zip code as 4 separate attributes), and one telephone number. Date of CD purchase by the store should also be tracked for each CD (think about where to put date of purchase as an attribute). Each CD has only one seller, but a seller can sell many CDs to the store.
- You may specify all the relationships as non-identifying (except for any many-to-many relationship)
- Save your MySQL diagram as image (File – Export - Export as png) and paste it here

## Diagrams

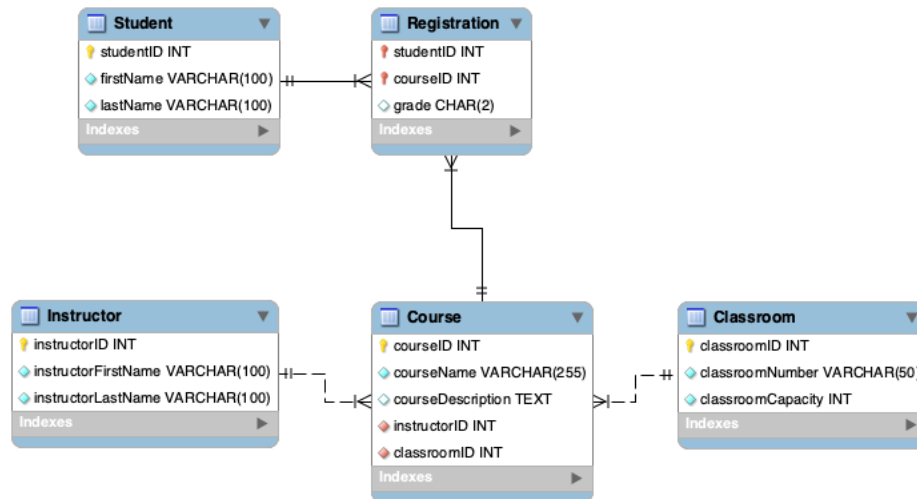


2. (30 points) Create a database design diagram for student courses in MySQL workbench with the following rules (choose the appropriate data type for each attribute). **Paste the diagram after this question.**

### **Student Courses**

- A student should register for at least one course (and can register for many courses) and a course should have one or more students
- A student table should have the following attributes: studentID as a primary key, and such attributes as firstName, lastName
- A course table should have the following attributes: courseID as a primary key, and such attributes as courseName, courseDescription
- Since the relationship between a student and a course is many-to-many, you need to add an intermediate table between them titled 'Registration' that would hold the appropriate foreign keys and an attribute Grade. For this 'Registration' table you need to correctly specify a composite primary key that consists of several foreign keys
- An instructor can teach one or many courses and a course is taught by at least one and at most one instructor
- An instructor table should have the following attributes: instructorID as a primary key, and such attributes as instructorFirstName, instructorLastName
- A classroom can be used by one or many courses, and each course is held in at least one and at most one classroom
- A classroom table should have the following attributes: classroomID as a primary key, and such attributes as classroomNumber, classroomCapacity
- Make sure all the foreign keys are in the correct tables
- You may specify all the relationships as non-identifying (except for any many-to-many relationship)
- Save your MySQL diagram as image (File – Export - Export as png) and paste it here

## Diagrams



3. (30 points) Create a code from scratch to create tables based on your previous question design (**only for student courses, question 2 database**) in MySQL Workbench, and run the code to create those tables (including the correct data types for each attribute). There is no need to insert any data into the tables but make sure all foreign keys and primary keys are correctly specified (similar to the lecture in week 1 and to the week2 self-paced Panopto video 2 – look at scripts `create_university.sql` and `insert_university.sql`)

## SQL code scripts

-- Student Courses Database ER Diagram Design

-- Table: Student

```
CREATE TABLE Student (  
    studentID INT AUTO_INCREMENT PRIMARY KEY,  
    firstName VARCHAR(100) NOT NULL,  
    lastName VARCHAR(100) NOT NULL  
);
```

-- Table: Course

```
CREATE TABLE Course (  
    courseID INT AUTO_INCREMENT PRIMARY KEY,  
    courseName VARCHAR(255) NOT NULL,  
    courseDescription TEXT  
);
```

-- Table: Instructor

```
CREATE TABLE Instructor (  
    instructorID INT AUTO_INCREMENT PRIMARY KEY,  
    instructorFirstName VARCHAR(100) NOT NULL,  
    instructorLastName VARCHAR(100) NOT NULL  
);
```

-- Table: Classroom

```
CREATE TABLE Classroom (  
    classroomID INT AUTO_INCREMENT PRIMARY KEY,  
    classroomNumber VARCHAR(50) NOT NULL,  
    classroomCapacity INT NOT NULL  
);
```

-- Table: Registration (junction table for Student-Course relationship)

```
CREATE TABLE Registration (  
    studentID INT NOT NULL,  
    courseID INT NOT NULL,  
    grade CHAR(2),  
    PRIMARY KEY (studentID, courseID),  
    FOREIGN KEY (studentID) REFERENCES Student(studentID),  
    FOREIGN KEY (courseID) REFERENCES Course(courseID)  
);
```

-- Adding foreign keys for Course table to connect with Instructor and Classroom

```
ALTER TABLE Course  
  
ADD COLUMN instructorID INT NOT NULL,  
  
ADD COLUMN classroomID INT NOT NULL,  
  
ADD FOREIGN KEY (instructorID) REFERENCES Instructor(instructorID),  
  
ADD FOREIGN KEY (classroomID) REFERENCES Classroom(classroomID);
```

**Please copy the whole SQL script into this Word doc and upload your answers to all 3 HW1 questions as a Word document to the Blackboard HW1 folder**