**Due Friday, Sept 16 at 11:59 PM.**

**Homework 1 (total 100 points)**

**Please paste diagrams and 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).

**Video Rental**

The manager of the StayHome Videos shop requires you to create a database application system to assist with the administration of the shop. The requirements collection and analysis phase has provided the following requirements for the database.

* The video shop stocks videos that fall into the following categories - comedy, drama, horror, musical, science fiction, and children. Each category is identified by a unique identifier. A video can belong to many categories. For this HW, please consider each video as a unique DVD disk with a DVD disk video identifier. For example, one movie can have many DVD video disks with a separate unique identifier.
* The data stored on the videos for rent includes the video number, catalog number, title, certification, year of release, main actor/actress, purchase price, daily rental rate. The video number is created by the manager and uniquely identifies each video.
* The data stored on each member of the video shop includes member number, date joined, name (consists of first and last name), address (consists of street, city, state, zip code), and one telephone number.
* Each member may borrow an unlimited number of videos, with the checkout date and the return date on each video. Each video can be borrowed by many members over time.
* Data on suppliers of videos should also be stored, such as supplier number, name, address (consists of street, city, state, zip code), and one telephone number. Date of supply should also be tracked for each video (think about where to put date of supply as an attribute). Each video has only one supplier, but a supplier supplies many videos.
* You may specify all the relationships as non-identifying (except for many-to-many relationship)
* Save your MySQL diagram as image (File – Export - Export as png) and paste it here

**图示

描述已自动生成**

**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: instructorD 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 many-to-many relationship)
* Save your MySQL diagram as image (File – Export - Export as png) and paste it here

图示

描述已自动生成

3. (30 points) Create a code from scratch to create tables based on your previous question design (only for student courses, question 2) 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 2 and to the week2 self-paced Panopto video – look at scripts create\_university.sql and insert\_university.sql)

-- MySQL Script generated by MySQL Workbench

-- Thu Sep 15 19:30:19 2022

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

USE `mydb` ;

-- -----------------------------------------------------

-- Table `mydb`.`Student`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`Student` (

`StudentID` INT NOT NULL,

`FirstName` VARCHAR(45) NOT NULL,

`LastName` VARCHAR(45) NOT NULL,

PRIMARY KEY (`StudentID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Instructor`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`Instructor` (

`InstructorID` INT NOT NULL,

`InstructorFirstName` VARCHAR(45) NOT NULL,

`InstructorLastName` VARCHAR(45) NOT NULL,

PRIMARY KEY (`InstructorID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Classroom`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`Classroom` (

`ClassroomID` INT NOT NULL,

`ClassroomNumber` VARCHAR(45) NOT NULL,

`ClassroomCapacity` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ClassroomID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Course`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`Course` (

`CourseID` INT NOT NULL,

`CourseName` VARCHAR(45) NOT NULL,

`CourseDescription` VARCHAR(45) NOT NULL,

`Instructor\_InstructorID` INT NOT NULL,

`Classroom\_ClassroomID` INT NOT NULL,

PRIMARY KEY (`CourseID`),

INDEX `fk\_course\_instructor1\_idx` (`Instructor\_InstructorID` ASC) VISIBLE,

INDEX `fk\_course\_classroom1\_idx` (`Classroom\_ClassroomID` ASC) VISIBLE,

CONSTRAINT `fk\_course\_instructor1`

FOREIGN KEY (`Instructor\_InstructorID`)

REFERENCES `mydb`.`Instructor` (`InstructorID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_course\_classroom1`

FOREIGN KEY (`Classroom\_ClassroomID`)

REFERENCES `mydb`.`Classroom` (`ClassroomID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Registration`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`Registration` (

`Student\_StudentID` INT NOT NULL,

`Course\_CourseID` INT NOT NULL,

`Grade` VARCHAR(10) NOT NULL,

INDEX `fk\_Registration\_student\_idx` (`Student\_StudentID` ASC) VISIBLE,

INDEX `fk\_Registration\_course1\_idx` (`Course\_CourseID` ASC) VISIBLE,

PRIMARY KEY (`Student\_StudentID`, `Course\_CourseID`),

CONSTRAINT `fk\_Registration\_student`

FOREIGN KEY (`Student\_StudentID`)

REFERENCES `mydb`.`Student` (`StudentID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Registration\_course1`

FOREIGN KEY (`Course\_CourseID`)

REFERENCES `mydb`.`Course` (`CourseID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**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**