University of Belize

Education Empowers a Nation

Faculty of Science and Technology

CMPS1171 INTRODUCTION TO DATABASES
PROJECT II
Name: Tysha Daniels

Instructions:

1. Make a Google Folder named Your Name Database Project II. Share the folder with me.
2. Make a copy of this file and place it in the folder. Rename the File to Your Name Project II.
3. Show each of the five steps in creating your ER diagram.
4. Write the SQL code to create the database. Upload the text file with the SQL code to the folder.
5. Create the .csv files with the data that you need for your database.
6. On your local virtual machine run your SQL code to build and populate the database.
7. Recording of your database running on your virtual machine with interesting queries.

In a university, a Student (represented by the USERS table) enrolls in Courses. A Student must be assigned to at least one or more Courses. Each Course is taught by a single Professor (not represented in the current ER diagram). To maintain instruction quality, a Professor can deliver only one Course. Additionally, each Student is assigned to a single Advisor who provides academic guidance and support. An Advisor can be assigned to multiple Students, establishing a one-to-many relationship between Advisors and Students.

The university maintains a database system to manage student enrollments, course offerings, and advisor assignments. The database consists of the following tables:

1.) USERS (Students) Table:

- Contains information about each student, including their unique identifier (id), first name, last name, email address, enrollment date, and assigned advisor (advisor_id).

2.) COURSES Table:

- Stores details about each course, including the unique identifier (id), description, course outline, pass rate, and average grade.

3.) SEMESTER Table:

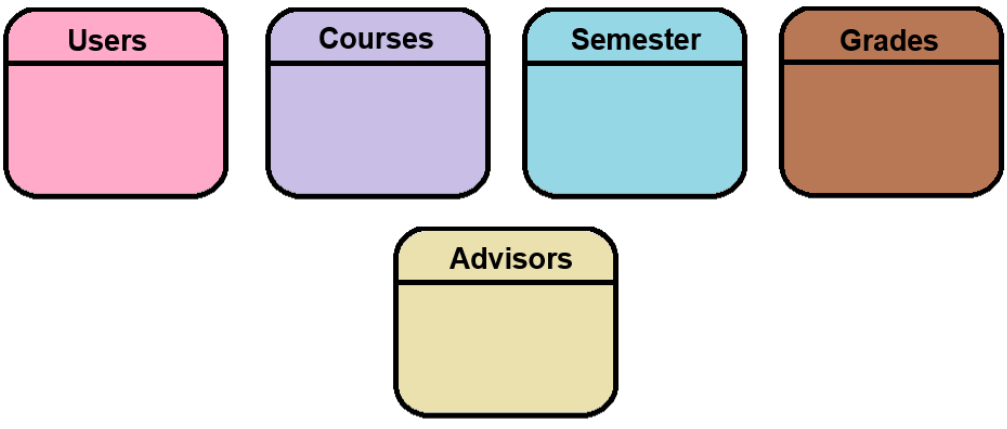
- Holds information about each semester, including the unique identifier (id) and name.

4.) GRADES Table:

- Acts as a junction table connecting USERS (Students), COURSES, and SEMESTER.
- Each record represents a student's grade for a specific course in a particular semester.
- Includes a unique identifier (id) as the primary key.

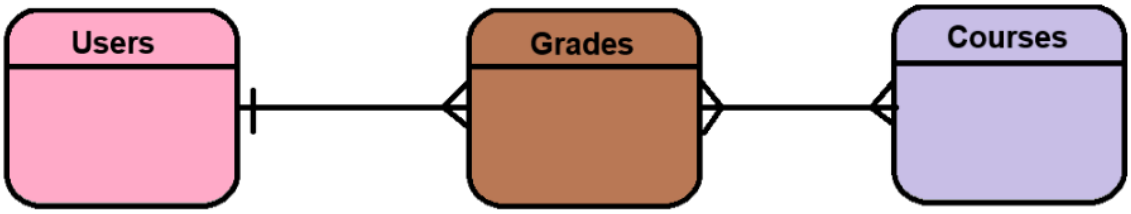
5.) ADVISORS Table:

- Contains information about academic advisors, including their unique identifier (id), name, phone number, email address, office location, and office hours.
- Each student is associated with an advisor who provides academic guidance and support.

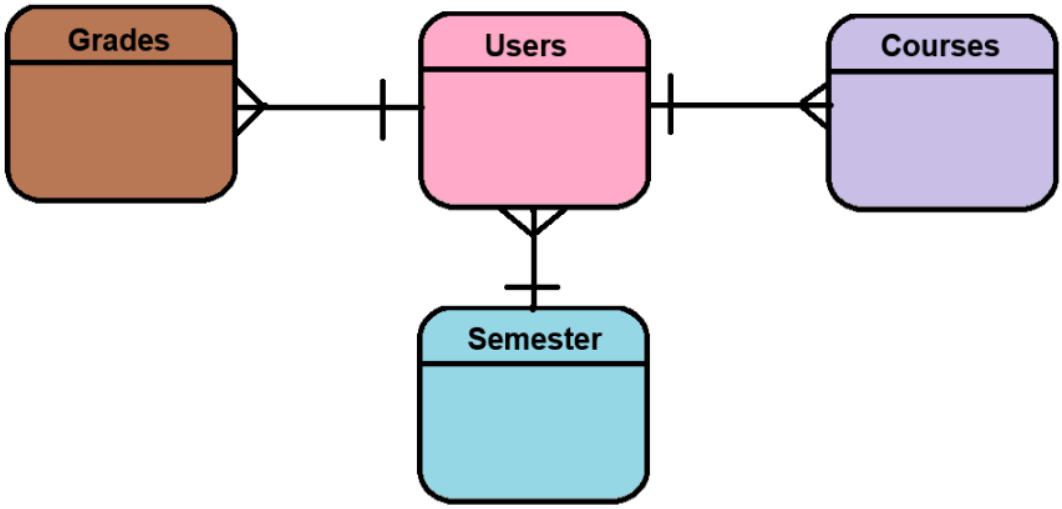


The relationships between these tables are as follows:

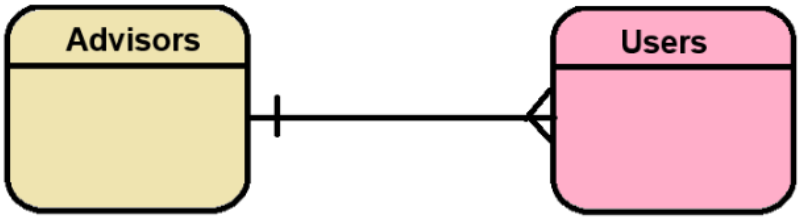
A Student (USERS) can earn Grades in one or more Courses (many-to-many relationship) through the GRADES table.



Each Grade is associated with a single Student (USERS), a single Course, and a single Semester.



Each Student (USERS) is advised by a single Advisor (one-to-many relationship).



Attributes:

1.) USERS (Students):

- user_id (PK): int
- user_first_name: varchar(50)
- user_last_name: varchar(50)
- user_email: varchar(100)
- user_enrollment_date: date
- advisor_id (FK): int

Users			
🔑	user_id	int	---
	user_first_name	varchar	50
	user_last_name	varchar	50
	user_email	varchar	100
	user_enrollment_date	date	---
🔶	advisor_id	int	---

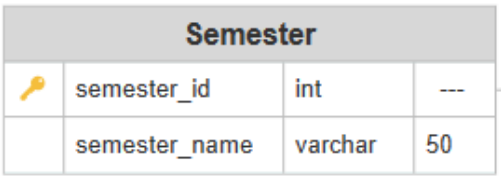
2.) COURSES:

- course_id (PK): int
- course_description: varchar(255)
- course_outline: text
- course_pass_rate: decimal(5,2)
- course_average: decimal(5,2)

Courses			
🔑	course_id	int	---
	course_description	varchar	255
	course_outline	text	---
	course_pass_rate	decimal	5,2
	course_average	decimal	5,2

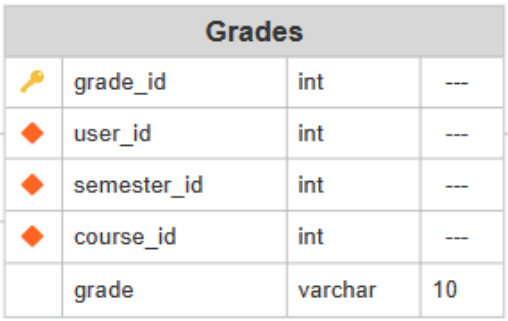
3.) SEMESTER:

- semester_id (PK): int
- semester_name: varchar(50)



4.) GRADES:

- grade_id (PK): int
- semester_id (FK): int
- user_id (FK): int
- course_id (FK): int
- grade: varchar(10)



5.) ADVISORS:

- advisor_id (PK): int
- advisor_first_name: varchar(50)
- advisor_last_name: varchar(50)
- advisor_phone_number: varchar(20)
- advisor_email: varchar(100)
- advisor_office_location: varchar(100)
- advisor_office_hours: varchar(100)

Advisors			
🔑	advisor_id	int	---
	advisor_first_name	varchar	50
	advisor_last_name	varchar	50
	advisor_phone_number	varchar	20
	advisor_email	varchar	100
	advisor_office_location	varchar	100
	advisor_office_hours	varchar	100

ER Diagram: