CEIT 390 – Database Management Systems

Database Design (ER Diagram) Submission Guide

1. Project Title

Educational Science Course Management System

2. Introduction

The **Educational Science Course Management System** aims to provide a centralized, semester-based platform for managing and accessing course-related information in the Educational Sciences Department.

Purpose of the System:

- Enable students to easily view and search available courses.
- Allow teachers to create, update, and manage course content.
- Facilitate semester-based course organization and access.

Main Features:

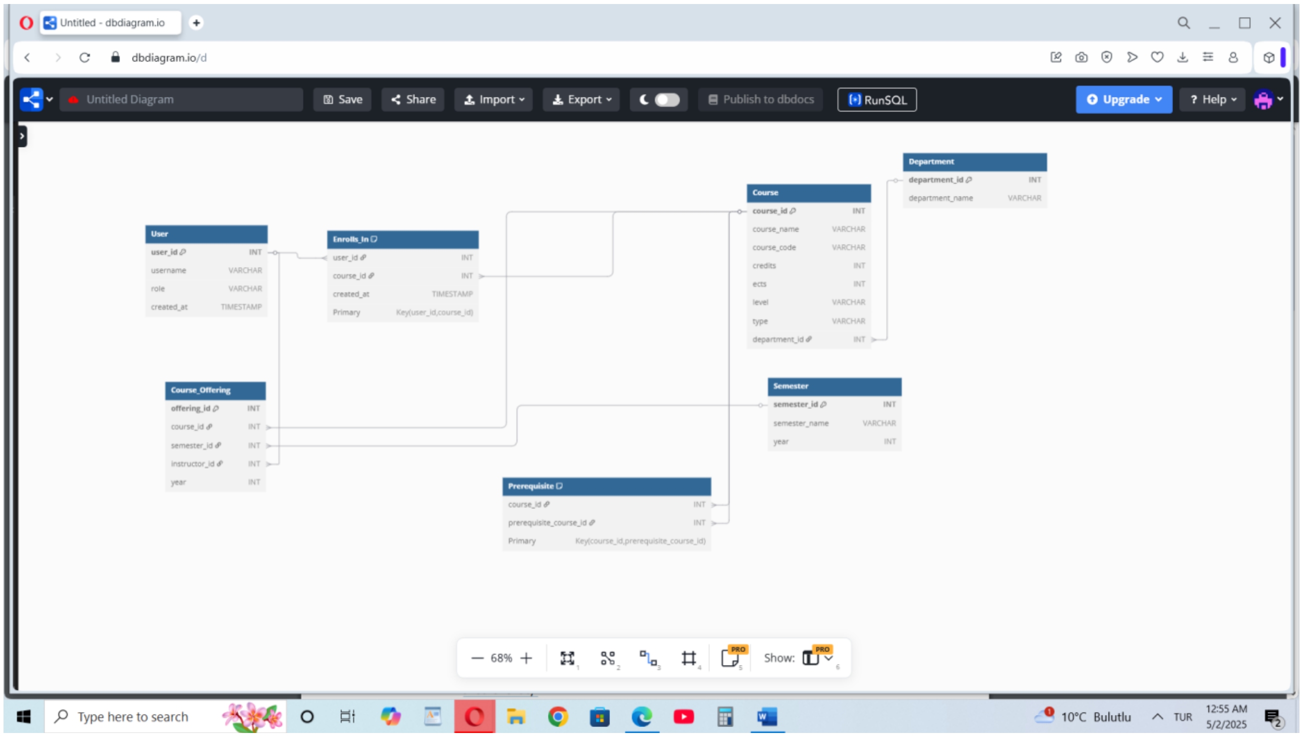
- User authentication and role-based access
- Course listings with advanced search/filter capabilities
- Semester-based filtering
- Course detail management with prerequisite tracking
- Clean UI with role-based dashboards

3. Entity-Relationship (ER) Diagram

Create a detailed **ER Diagram** that represents the database structure for your project. The diagram should clearly show:

- **Entities**: These represent the objects or things that will be stored in the database (e.g., Student, Course, Event, etc.).
- **Attributes**: These represent the data fields for each entity (e.g., Student might have attributes like Name, ID, Age, etc.).
- **Primary Keys**: Identify the unique identifier for each entity (e.g., StudentID for Student).
- **Relationships**: Show how entities are related (e.g., a Student enrolls in a Course).
  - **One-to-One (1:1)**: One instance of an entity is related to only one instance of another entity.
  - **One-to-Many (1:M)**: One instance of an entity is related to multiple instances of another entity.
  - **Many-to-Many (M:M)**: Multiple instances of one entity are related to multiple instances of another entity.

The ER Diagram should be clear, well-organized, and readable. You can use various tools to create it, ensuring that the diagram effectively represents entities, relationships, and constraints.



User	
user_id	INT
username	VARCHAR
role	VARCHAR
created_at	TIMESTAMP

Enrolls_In	
user_id	INT
course_id	INT
created_at	TIMESTAMP
Primary	Key(user_id,course_id)

Course	
course_id	INT
course_name	VARCHAR
course_code	VARCHAR
credits	INT
ects	INT
level	VARCHAR
type	VARCHAR
department_id	INT

Department	
department_id	INT
department_name	VARCHAR

Course_Offering	
offering_id	INT
course_id	INT
semester_id	INT
instructor_id	INT
year	INT

Semester	
semester_id	INT
semester_name	VARCHAR
year	INT

Prerequisite	
course_id	INT
prerequisite_course_id	INT
Primary	Key(course_id,prerequisite_course_id)

4. Description of Entities and Relationships

Entities:

- **User**
  Attributes: user_id (PK), username, role, created_at
- **Course**
  Attributes: course_id (PK), course_name, course_code, credits, ects, level, type, department_id (FK)
- **Department**
  Attributes: department_id (PK), department_name
- **Semester**
  Attributes: semester_id (PK), semester_name, year
- **Course_Offering**
  Attributes: offering_id (PK), course_id (FK), semester_id (FK), instructor_id (FK), year
- **Enrolls_In**
  Attributes: user_id (FK), course_id (FK), created_at
  (Composite Primary Key: user_id + course_id)
- **Prerequisite**
  Attributes: course_id (FK), prerequisite_course_id (FK)
  (Composite Primary Key: course_id + prerequisite_course_id)

Relationships:

- One user (teacher) can teach many course offerings
- One student can enroll in many courses
- A course can be offered in multiple semesters
- A course may have multiple prerequisites
- A department can have multiple courses

5. Database Schema

Table Name	user	
Column Name	**Data Type**	**Constraints**
user_id	SERIAL	PRIMARY KEY
username	VARCHAR	NOT NULL, UNIQUE
role	VARCHAR	CHECK (role IN (‘student’,teacher’,’admin’))
create_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

Table Name	department	
Column Name	**Data Type**	**Constraints**
Department_id	SERIAL	PRIMARY KEY
Department_name	VARCHAR	NOT NULL, UNIQUE

Table Name	course	
Column Name	**Data Type**	**Constraints**
Course_id	SERIAL	PRIMARY KEY
Course_name	VARCHAR	NOT NULL
Course_code	VARCHAR	NOT NULL, UNIQUE
credits	INT	CHECK (credits > 0)
ects	INT	CHECK (ects > 0)
level	VARCHAR	
type	VARCHAR	
Department_id	INT	FOREIGN KEY REFERENCES department(department_id)

Table Name	**semester**	
Column Name	**Data Type**	**Constraints**
Semester_id	SERIAL	PRIMARY KEY
Semester_name	VARCHAR	NOT NULL
year	INT	CHECK (year >= 2000)

Table Name	**Course_offering**	
Column Name	**Data Type**	**Constraints**
Offering_id	SERIAL	PRIMARY KEY
Course_id	INT	FOREIGN KEY REFERENCES course(course_id)
Semester_id	INT	FOREIGN KEY REFERENCES semester(semester_id)
Instructor_id	INT	FOREIGN KEY REFRENCES user(user_id)
year	INT	CHECK (year >= 2000)

Table Name	**Enrolls_in**	
Column Name	**Data Type**	**Constraints**
User_id	INT	FOREIGN KEY REFERNECES user(user_id)
Course_id	INT	FOREIGN KEY REFERENCES course(course_id)
Created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
		PRIMARY KEY (user_id, course_id)

Table Name	**prerequisite**	
Column Name	**Data Type**	**Constraints**
Course_id	INT	FOREIGN KEY REFERENCES course(course_id)
Prerequisite_course_id	INT	FOREIGN KEY REFERENCES course(course_id)
		PRIMARY KEY (course_id, prerequisite_course_id)

7. Submission Details

- **File Format**: PDF
- **Length**: 3-5 pages
- **Naming Convention**: GroupName_ER_Diagram.pdf (One group member's submission is enough)

Evaluation Criteria (5 Points Total)

Criteria	Weight	Description
ER Diagram & Relationship	**2.5 pts**	- Clear, complete, and accurate ER diagram with correct notation. - Well-defined entities with primary keys. - Correct representation of relationships (1:M, M:M) and foreign keys.
Database Schema	**1.5 pts**	- Well-structured tables with appropriate columns, data types, and constraints.
Entities & Relationships Descriptions	**0.75 pts**	- Clear explanations of entities, attributes, and relationships.
Documentation	**0.25 pts**	- Organized, neat, and clear report with minimal errors.