



# PREMIER UNIVERSITY CHATTOGRAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

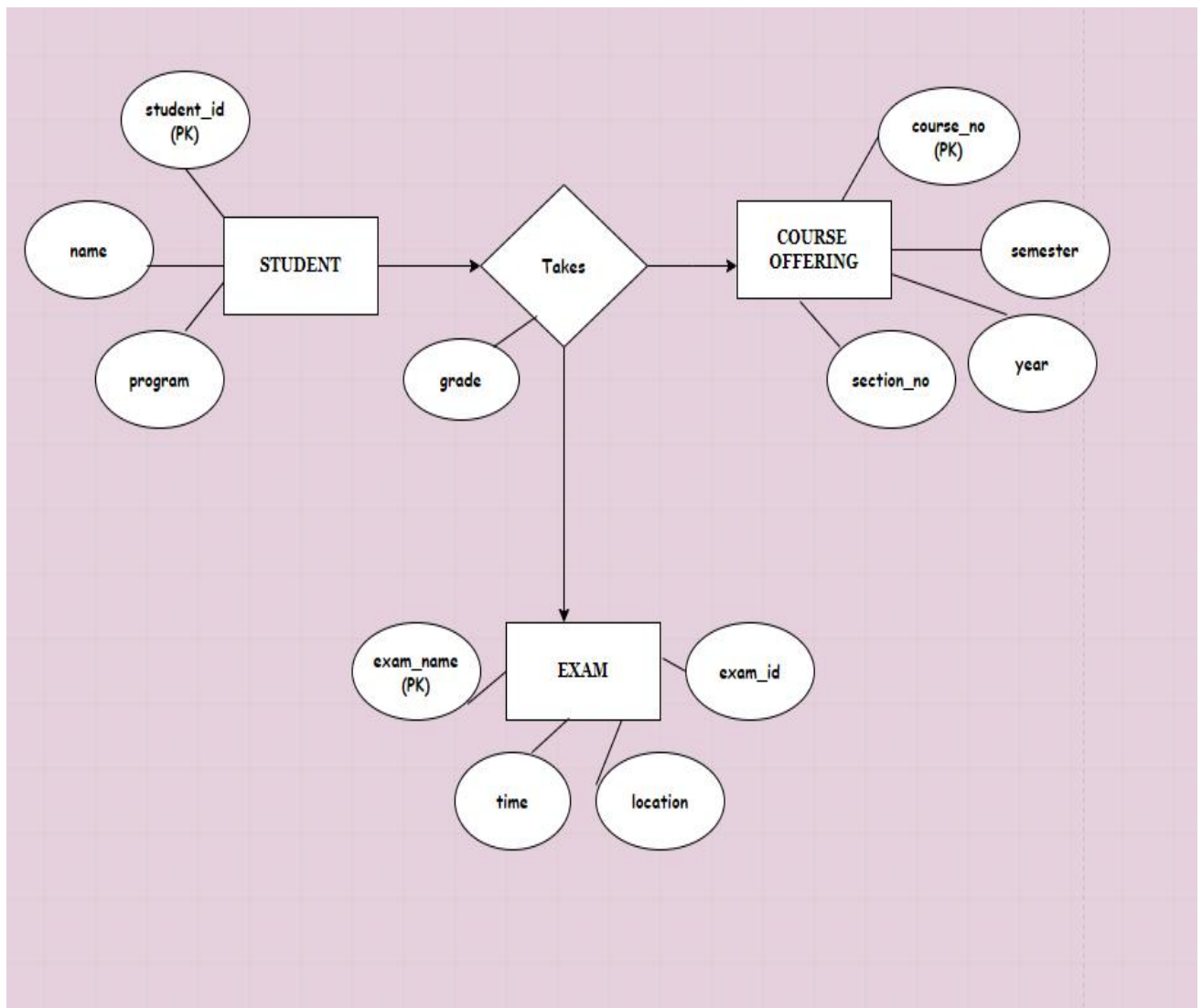
## ASSIGNMENT

COURSE NAME	Database Management System	
COURSE CODE	CSE 2221	
ASSIGNMENT TOPIC	Database Design for Tracking Student Exam Scores: E-R Diagram Modeling with Ternary and Binary Relationships.	
DATE OF ASSIGNMENT	3-09-2024	
DATE OF SUBMISSION	7-09-2024	
SUBMITTED TO		
ASIF SAAD LECTURER DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING		
REMARKS	SUBMITTED BY	
	NAME	Rimjhim Dey
	ID	0222220005101039
	SEMESTER	4th
	BATCH	42
	SESSION	Spring 2024
	SECTION	A

## Answer to Question No. 1

### Constructing an E-R Diagram that Models Exams as Entities Using a Ternary Relationship.

The objective is to create an E-R (Entity-Relationship) diagram that effectively models the relationship between students, course offerings, and exams. In this model, exams will be treated as separate entities, and a ternary relationship will be used to capture the interaction between these three entities.



## **Explanation of the E-R Diagram:**

### **Entities:**

**1.Student:** This entity represents students who take courses.

Attributes include:

**student\_id (Primary Key):** A unique identifier for each student.

**name:** The name of the student.

**program:** The academic program the student is enrolled in (e.g., Computer Science, Electrical Engineering).

### **2.Course Offering:**

This entity represents specific instances of courses being offered at a particular time and location.

Attributes include:

**course\_no (Primary Key):** A unique identifier for the course.

**semester:** The semester in which the course is offered.

**year:** The year in which the course is offered.

**section\_no:** The section number of the course offering.

### **3.Exam:**

This entity represents exams that students take in a course offering.

Attributes include:

**exam\_name (Primary Key):** The name of the exam (e.g., Midterm, Final).

**location:** The location where the exam is held.

**time:** The time of the exam.

**exam\_id:** An optional unique identifier for each exam.

### **Ternary Relationship: "Takes"**

- \* The "Takes" relationship connects the three entities: Student, Course Offering, and Exam.
- \* It captures the fact that a student takes a particular exam as part of a specific course offering.
- \* The grade attribute is associated with this relationship to track the score or grade the student received on the exam.

This means for each combination of Student, Course Offering, and Exam, you can store a different grade.

### **Grade Attribute**

- \* The grade is an attribute of the ternary relationship "Takes".
- \* It represents the score or grade a student receives in a specific exam for a particular course offering.

The grade is stored as part of the relationship between the Student, Course Offering, and Exam, rather than being an attribute of the entities themselves. This is because the grade is specific to the combination of a student, a course offering, and an exam.

### **How the Diagram Works**

- \* A Student is linked to a Course Offering through the Takes relationship, meaning they are enrolled in the course.
- \* The Takes relationship also connects the Exam entity, allowing the model to track which exams the student took as part of the course offering.
- \* The grade attribute within the "Takes" relationship stores the specific marks a student received on each exam for that course offering.

## **Example Scenario**

Imagine the following situation:

John Doe (student) is enrolled in the "Database Systems" course offering in Fall 2024 (course\_no: CS101, section\_no: 1).

John takes two exams for this course: the Midterm and the Final.

The "Takes" relationship would track that John took both exams as part of the Fall 2024 offering of Database Systems.

The grade attribute would store John's marks on the Midterm and Final (e.g., 85 for the Midterm, 90 for the Final).

### **In this setup:**

There is a single relationship between John, the Database Systems course offering, and each exam.

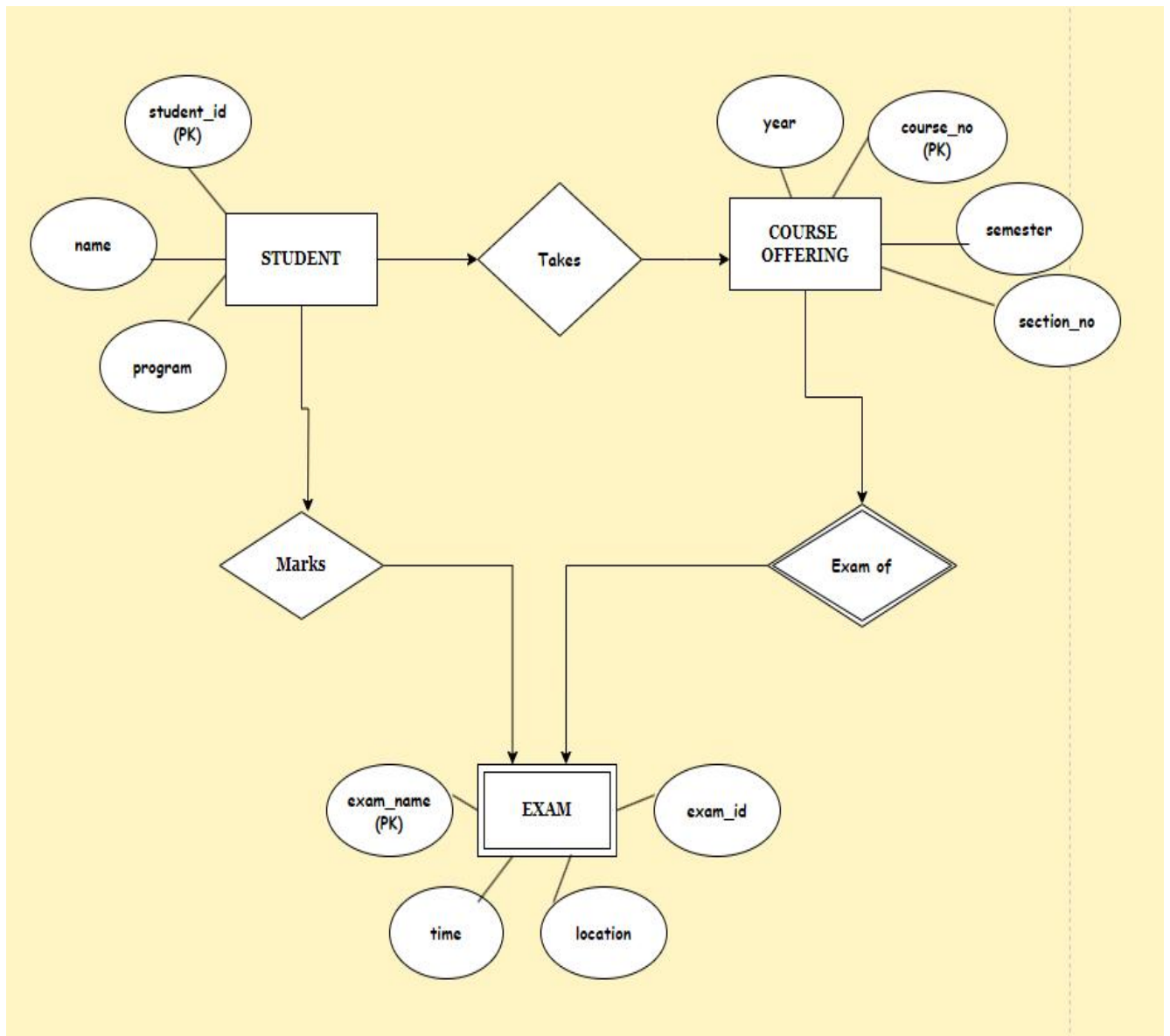
Each exam can have a different grade for the student, allowing you to track the performance on multiple exams for the same course offering.

## **Answer to Question No. 2**

**Constructing an alternative E-R diagram that uses only a binary relationship between students and course-offerings.**

The objective is to create an alternative E-R diagram for a database that tracks students' scores on various exams within different course offerings. This diagram models a binary relationship between Students and Course Offerings, yet still ensures that student grades on specific exams are captured.

This model simplifies the schema by connecting students directly to course offerings and using a weak entity to capture the relationship between students and course offerings.



### Explanation of the E-R Diagram:

In this alternative E-R diagram, the entities are Student, Course Offering, and Exam.

**Student** : Represents students identified by attributes like student\_id, name, and program.

**Course Offering** : Represents various course offerings identified by course\_no, semester, year, and section\_no.

**Exam** : Represents different exams identified by exam\_name, location, time, and exam\_id.

## Explanation of Relationships

**Takes:** This relationship links Student to Course Offering. It establishes which student is enrolled in which course offering.

**Marks:** The Marks relationship records the student's score for an exam within a particular course offering.

**Exam of:** This is a weak relationship that connects Exam with Course Offering. It shows that an exam is associated with a specific course offering, allowing us to track which exams are administered for each course offering.

## How the Diagram Works

- \* Student participates in a Course Offering through the Takes relationship. This tells us which students are enrolled in which courses.

- \* Marks records the performance of a student in a specific exam for the course offering they are enrolled in. It shows how well a student did in a particular exam.

- \* Exam of indicates that an exam is associated with a specific course offering. This relationship helps in organizing exams under their respective course offerings.

## Example Scenario

Imagine the following situation:

Student: John Doe (ID: 123)

Course Offering: CS101, Fall 2024, Section 1

Exam: Midterm Exam, Room 101, 10:00 AM

John Doe is enrolled in the CS101 course offering for Fall 2024, Section 1. There is a Midterm Exam associated with this course offering.

The Takes relationship shows that John Doe is taking the CS101 course offering.

The Marks relationship records John's score in the Midterm Exam.

The Exam\_of relationship indicates that the Midterm Exam is part of the CS101 course offering for Fall 2024, Section 1.

**In this setup:**

John Doe is linked to CS101 through the Takes relationship. This shows his enrollment in the course offering.

Marks captures John's score in the Midterm Exam. The relationship is established through the Takes relationship, indicating that the mark is relevant to his enrollment in the course.

Exam\_of ensures that the Midterm Exam is properly associated with the CS101 course offering, allowing tracking of which exams are given for which course.

Here,

The **Exam** entity and the **Exam\_of** relationship are classified as weak because they depend on the context of course offerings and student enrollments. The Exam entity's identity is dependent on its association with a specific course offering, and the Exam\_of relationship shows how exams are linked to course offerings. This design ensures that the database accurately represents the context in which exams are administered and the scores are recorded.