

Database Systems

Department of Computer Engineering

Spring 2021

Design Report

Online Course Platform - Sapientia

Project Group 4:

* Ömer Ünlüsoy - Section 01 - 21702136
* Elif Gamze Güliter - Section 01 - 21802870
* İrem Tekin - Section 01 - 21803267
* Ece Ünal - Section 01 - 21703149

Supervisor:

* Uğur Güdükbay

Table of Contents

[Revised E/R Diagram 5](#_Toc68287823)

[Relation Schemas and SQL 6](#_Toc68287824)

[Notes 6](#_Toc68287825)

[1. Student 6](#_Toc68287826)

[Relational Model: 6](#_Toc68287827)

[SQL Definition: 6](#_Toc68287828)

[2. Instructor 6](#_Toc68287829)

[Relational Model: 6](#_Toc68287830)

[SQL Definition: 7](#_Toc68287831)

[3. Admin 7](#_Toc68287832)

[Relational Model: 7](#_Toc68287833)

[SQL Definition: 7](#_Toc68287834)

[4. Course 7](#_Toc68287835)

[Relational Model: 7](#_Toc68287836)

[SQL Definition: 7](#_Toc68287837)

[5. Section 8](#_Toc68287838)

[Relational Model: 8](#_Toc68287839)

[SQL Definition: 8](#_Toc68287840)

[6. Lecture 8](#_Toc68287841)

[Relational Model: 8](#_Toc68287842)

[SQL Definition: 8](#_Toc68287843)

[7. Quiz 9](#_Toc68287844)

[Relational Model: 9](#_Toc68287845)

[SQL Definition: 9](#_Toc68287846)

[8. Quiz\_Question 9](#_Toc68287847)

[Relational Model: 9](#_Toc68287848)

[SQL Definition: 10](#_Toc68287849)

[9. Note 10](#_Toc68287850)

[Relational Model: 10](#_Toc68287851)

[SQL Definition: 10](#_Toc68287852)

[10. Course\_Announcement 10](#_Toc68287853)

[Relational Model: 10](#_Toc68287854)

[SQL Definition: 11](#_Toc68287855)

[11. QnA\_Entry\_Student 11](#_Toc68287856)

[Relational Model: 11](#_Toc68287857)

[SQL Definition: 11](#_Toc68287858)

[12. QnA\_Entry\_Instructor 11](#_Toc68287859)

[Relational Model: 11](#_Toc68287860)

[SQL Definition: 11](#_Toc68287861)

[13. Complaint\_Entry\_Student 12](#_Toc68287862)

[Relational Model: 12](#_Toc68287863)

[SQL Definition: 12](#_Toc68287864)

[14. Complaint\_Entry\_Instructor 12](#_Toc68287865)

[Relational Model: 12](#_Toc68287866)

[SQL Definition: 12](#_Toc68287867)

[15. Wishlist 12](#_Toc68287868)

[Relational Model: 12](#_Toc68287869)

[SQL Definition: 13](#_Toc68287870)

[16. Enrolls 13](#_Toc68287871)

[Relational Model: 13](#_Toc68287872)

[SQL Definition: 13](#_Toc68287873)

[17. Certificate 13](#_Toc68287874)

[Relational Model: 13](#_Toc68287875)

[SQL Definition: 13](#_Toc68287876)

[18. Teaches 14](#_Toc68287877)

[Relational Model: 14](#_Toc68287878)

[SQL Definition: 14](#_Toc68287879)

[19. Evaluates\_Complaint\_Entry\_Student 14](#_Toc68287880)

[Relational Model: 14](#_Toc68287881)

[SQL Definition: 14](#_Toc68287882)

[20. Evaluates\_Complaint\_Entry\_Instructor 15](#_Toc68287883)

[Relational Model: 15](#_Toc68287884)

[SQL Definition: 15](#_Toc68287885)

[21. Discount 15](#_Toc68287886)

[Relational Model: 15](#_Toc68287887)

[SQL Definition: 15](#_Toc68287888)

[22. Zoom\_Session 16](#_Toc68287889)

[Relational Model: 16](#_Toc68287890)

[SQL Definition: 16](#_Toc68287891)

[23. Take\_Quiz 16](#_Toc68287892)

[Relational Model: 16](#_Toc68287893)

[SQL Definition: 16](#_Toc68287894)

[24. Take\_Quiz\_Question 17](#_Toc68287895)

[Relational Model: 17](#_Toc68287896)

[SQL Definition: 17](#_Toc68287897)

[25. Take\_Lecture 17](#_Toc68287898)

[Relational Model: 17](#_Toc68287899)

[SQL Definition: 17](#_Toc68287900)

[26. Evaluations 18](#_Toc68287901)

[Relational Model: 18](#_Toc68287902)

[SQL Definition: 18](#_Toc68287903)

# Revised E/R Diagram

# Relation Schemas and SQL

## Notes

General Assumptions:

ID -> INT AUTO\_INCREMENT

referencing ID -> INT NOT NULL

short string -> VARCHAR(30)

mid string -> VARCHAR(50)

long string -> VARCHAR(70)

description -> VARCHAR(300)

money -> NUMERIC(12, 2) OR NUMERIC(8, 2)

% -> NUMERIC(5, 2)

date -> DATE

URL -> VARCHAR(1024)

Special Cases:

membership\_type -> ENUM('BRZ', 'SLV', 'GLD')

refund\_request -> BOOLEAN

duration -> TIME

lecture\_content -> BLOB NOT NULL

membership\_type -> ENUM('BRZ', 'SLV', 'GLD') NOT NULL DEFAULT 'BRZ'

answer -> ENUM('choice1', 'choice2', 'choice3') NOT NULL DEFAULT 'choice1'

## Student

### Relational Model:

Student(SID, name, e\_mail, password, membership\_type)

### SQL Definition:

CREATE TABLE Student(

SID INT AUTO\_INCREMENT,

name VARCHAR(50) NOT NULL,

e\_mail VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(30) NOT NULL,

membership\_type ENUM('BRZ', 'SLV', 'GLD') NOT NULL DEFAULT 'BRZ',

PRIMARY KEY (SID)

);

## Instructor

### Relational Model:

Instructor(IID, name, e\_mail, password)

### SQL Definition:

CREATE TABLE Instructor(

IID INT AUTO\_INCREMENT,

name VARCHAR(50) NOT NULL,

e\_mail VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(30) NOT NULL,

PRIMARY KEY (IID)

);

## Admin

### Relational Model:

Admin(AID, name, e\_mail, password, salary)

### SQL Definition:

CREATE TABLE Admin(

AID INT AUTO\_INCREMENT,

name VARCHAR(50) NOT NULL,

e\_mail VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(30) NOT NULL,

salary NUMERIC(12, 2) NOT NULL DEFAULT 0.00,

PRIMARY KEY (AID)

);

## Course

### Relational Model:

Course(CID, IID, course\_name, course\_photo, description, rating, category, level, cost, discount\_allowed, quiz\_threshold, lecture\_completed\_threshold)

### SQL Definition:

/\* Assumptions:

include functions [rating(), student\_count(), lecture\_count()]

constraints for category and level

0 <= cost <= 999,999.99

discount\_allowed = True OR False

lecture\_completed\_threshold = %70

0 <= certificate\_threshold <= 100.00

attention: FOREIGN KEY (creator\_IID) REFERENCES Instructor(IID) ON DELETE NO ACTION ON UPDATE CASCADE

UNIQUE (creator\_IID, course\_name)

\*/

CREATE TABLE Course(

CID INT AUTO\_INCREMENT,

IID INT NOT NULL,

course\_name VARCHAR(70) NOT NULL,

course\_photo BLOB,

description VARCHAR(300),

rating NUMERIC(5, 2) DEFAULT 0.00,

category VARCHAR(30),

level VARCHAR(30),

cost NUMERIC(8, 2) NOT NULL DEFAULT 0.00,

discount\_allowed BOOLEAN,

quiz\_threshold NUMERIC(5, 2),

lecture\_completed\_threshold NUMERIC(5, 2),

PRIMARY KEY (CID),

FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE NO ACTION ON UPDATE CASCADE,

UNIQUE (IID, course\_name)

);

### Section

### Relational Model:

Section(CID, section, title)

### SQL Definition:

CREATE TABLE Section(

CID INT NOT NULL,

section INT NOT NULL,

title VARCHAR(300),

PRIMARY KEY (CID, section),

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE,

INDEX (section)

);

## Lecture

### Relational Model:

Lecture(CID, content\_num, IID, section, title, lecture\_content, duration, date)

### SQL Definition:

/\* Assumptions:

lecture\_content is big data

lecture\_content is blob (not recommended, try to store videos in file directories and hold URLs)

Attention: UNIQUE (CID, title)

\*/

CREATE TABLE Lecture(

CID INT NOT NULL,

content\_num INT NOT NULL,

IID INT NOT NULL,

section INT NOT NULL,

title VARCHAR(300) NOT NULL,

lecture\_content BLOB NOT NULL,

duration TIME NOT NULL,

date DATE,

PRIMARY KEY (CID, content\_num),

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (section) REFERENCES Section(section) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE NO ACTION ON UPDATE CASCADE,

UNIQUE (CID, title),

INDEX (content\_num)

);

## Quiz

### Relational Model:

Quiz(CID, content\_num, IID, section, title, date)

### SQL Definition:

CREATE TABLE Quiz(

CID INT NOT NULL,

content\_num INT NOT NULL,

IID INT NOT NULL,

section INT NOT NULL,

title VARCHAR(300) NOT NULL,

date DATE,

PRIMARY KEY (CID, content\_num),

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (section) REFERENCES Section(section) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE NO ACTION ON UPDATE CASCADE,

UNIQUE (CID, title),

INDEX (content\_num)

);

## Quiz\_Question

### Relational Model:

Quiz\_Question(CID, content\_num, question\_num, question\_text, choice1, choice2, choice3, answer)

### SQL Definition:

/\*

Attention: answer ENUM('choice1', 'choice2', 'choice3') NOT NULL DEFAULT 'choice1'

answer = 1 => answer = choice1

\*/

CREATE TABLE Quiz\_Question(

CID INT NOT NULL,

content\_num INT NOT NULL,

question\_num INT NOT NULL,

question\_text VARCHAR(300) NOT NULL,

choice1 VARCHAR(50) NOT NULL DEFAULT '',

choice2 VARCHAR(50),

choice3 VARCHAR(50),

answer ENUM('choice1', 'choice2', 'choice3') NOT NULL DEFAULT 'choice1',

PRIMARY KEY (CID, content\_num, question\_num),

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (content\_num) REFERENCES Quiz(content\_num) ON DELETE CASCADE ON UPDATE CASCADE

);

## Note

### Relational Model:

Note(Note\_ID, SID, content\_num, time, text)

### SQL Definition:

CREATE TABLE Note(

Note\_ID INT AUTO\_INCREMENT,

SID INT NOT NULL,

content\_num INT NOT NULL,

time TIME,

text VARCHAR(300) NOT NULL,

PRIMARY KEY (Note\_ID),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (content\_num) REFERENCES Lecture(content\_num) ON DELETE CASCADE ON UPDATE CASCADE

);

## Course\_Announcement

### Relational Model:

Course\_Announcement(Ann\_ID, IID, CID, date, subject\_name, text)

### SQL Definition:

CREATE TABLE Course\_Announcement(

Ann\_ID INT AUTO\_INCREMENT,

IID INT NOT NULL,

CID INT NOT NULL,

date DATE,

subject\_name VARCHAR(30) NOT NULL,

text VARCHAR(300) NOT NULL,

PRIMARY KEY (Ann\_ID),

FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## QnA\_Entry\_Student

### Relational Model:

QnA\_Entry\_Student(QandA\_ID, SID, CID, subject\_name, text, date)

### SQL Definition:

CREATE TABLE QnA\_Entry\_Student(

QnA\_ID INT AUTO\_INCREMENT,

SID INT NOT NULL,

CID INT NOT NULL,

subject\_name VARCHAR(30) NOT NULL,

text VARCHAR(300) NOT NULL,

date DATE,

PRIMARY KEY (QnA\_ID),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## QnA\_Entry\_Instructor

### Relational Model:

QnA\_Entry\_Instructor(QandA\_ID, IID, CID, subject\_name, text, date)

### SQL Definition:

CREATE TABLE QnA\_Entry\_Instructor(

QnA\_ID INT AUTO\_INCREMENT,

IID INT NOT NULL,

CID INT NOT NULL,

subject\_name VARCHAR(30) NOT NULL,

text VARCHAR(300) NOT NULL,

date DATE,

PRIMARY KEY(QnA\_ID),

FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Complaint\_Entry\_Student

### Relational Model:

Complaint\_Entry\_Student(Complaint\_ID, SID, CID, subject\_name, text, date, refund\_request)

### SQL Definition:

CREATE TABLE Complaint\_Entry\_Student(

Complaint\_ID INT AUTO\_INCREMENT,

SID INT NOT NULL,

CID INT NOT NULL,

subject\_title VARCHAR(30) NOT NULL,

text VARCHAR(300) NOT NULL,

date DATE,

refund\_request BOOLEAN,

PRIMARY KEY (Complaint\_ID),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Complaint\_Entry\_Instructor

### Relational Model:

Complaint\_Entry\_Instructor(Complaint\_ID, SID, CID, subject\_name, text, date)

### SQL Definition:

CREATE TABLE Complaint\_Entry\_Instructor(

Complaint\_ID INT AUTO\_INCREMENT,

IID INT NOT NULL,

CID INT NOT NULL,

subject\_name VARCHAR(30) NOT NULL,

text VARCHAR(300) NOT NULL,

date DATE,

PRIMARY KEY (Complaint\_ID),

FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Wishlist

### Relational Model:

Wishlist(SID, CID, date)

### SQL Definition:

/\*

Binary relationship between Student and Course

Attributes: date

\*/

CREATE TABLE Wishlist(

SID INT NOT NULL,

CID INT NOT NULL,

date DATE,

PRIMARY KEY (SID, CID),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Enrolls

### Relational Model:

Enrolls(SID, CID, progress, rating, date)

### SQL Definition:

/\*

Binary relationship between Student and Course

Attributes: progress, rating

\*/

CREATE TABLE Enrolls(

SID INT NOT NULL,

CID INT NOT NULL,

progress NUMERIC(5, 2) DEFAULT 0.00,

rating NUMERIC(5, 2),

date DATE,

PRIMARY KEY (SID, CID),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Certificate

### Relational Model:

Certificate(SID, CID, certificate\_text, date)

### SQL Definition:

/\*

Binary relationship between Student and Course

Attributes: date

\*/

CREATE TABLE Certificate(

SID INT NOT NULL,

CID INT NOT NULL,

certificate\_text VARCHAR(300),

date DATE,

PRIMARY KEY (SID, CID),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Teaches

### Relational Model:

Teaches(IID, CID)

### SQL Definition:

/\*

Binary relationship between Instructor and Course

\*/

CREATE TABLE Teaches(

IID INT NOT NULL,

CID INT NOT NULL,

PRIMARY KEY (IID, CID),

FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Evaluates\_Complaint\_Entry\_Student

### Relational Model:

Evaluates\_Complaint\_Entry\_Student(AID, Complaint\_ID, response, isRefund)

### SQL Definition:

/\*

Binary relationship between Admin and Complaint\_Entry

Attention: FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO ACTION ON UPDATE CASCADE

\*/

CREATE TABLE Evaluates\_Complaint\_Entry\_Student(

AID INT NOT NULL,

Complaint\_ID INT NOT NULL,

response VARCHAR(300),

isRefund BOOLEAN,

PRIMARY KEY (AID, Complaint\_ID),

FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (Complaint\_ID) REFERENCES Complaint\_Entry\_Student(Complaint\_ID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Evaluates\_Complaint\_Entry\_Instructor

### Relational Model:

Evaluates\_Complaint\_Entry\_Instructor(AID, Complaint\_ID, response)

### SQL Definition:

/\*

Binary relationship between Admin and Complaint\_Entry

Attention: FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO ACTION ON UPDATE CASCADE

\*/

CREATE TABLE Evaluates\_Complaint\_Entry\_Instructor(

AID INT NOT NULL,

Complaint\_ID INT NOT NULL,

response VARCHAR(300),

PRIMARY KEY (AID, Complaint\_ID),

FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (Complaint\_ID) REFERENCES Complaint\_Entry\_Instructor(Complaint\_ID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Discount

### Relational Model:

Discount(AID, CID, rate)

### SQL Definition:

/\*

Binary relationship between Admin and Course

Attributes: rate (e.g. %30)

Attention: FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO ACTION ON UPDATE CASCADE

\*/

CREATE TABLE Discount(

AID INT NOT NULL,

CID INT NOT NULL,

rate NUMERIC(5, 2),

PRIMARY KEY (AID, CID),

FOREIGN KEY (AID) REFERENCES Admin(AID) ON DELETE NO ACTION ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

## Zoom\_Session

### Relational Model:

Zoom\_Session(Session\_ID, IID, CID, invite\_link, start\_date);

/\*

Binary relationship between Instructor and Course

Attributes: invite\_link, start\_date

Attention: it uses its own key to allow instructors to create several links

\*/

CREATE TABLE Zoom\_Session(

Session\_ID INT AUTO\_INCREMENT,

IID INT NOT NULL,

CID INT NOT NULL,

invite\_link VARCHAR(1024),

start\_date TIMESTAMP,

PRIMARY KEY (Session\_ID),

FOREIGN KEY (IID) REFERENCES Instructor(IID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);

### SQL Definition:

## Take\_Quiz

### Relational Model:

Take\_Quiz(SID, CID, content\_num, grade)

### SQL Definition:

/\*

Binary relationship between Student and Quiz

Attributes: grade (updated according to Student\_Take\_Quiz\_Question)

Attention: its key is Student(SID) U Quiz(CID, QID)

\*/

CREATE TABLE Take\_Quiz(

SID INT NOT NULL,

CID INT NOT NULL,

content\_num INT NOT NULL,

grade NUMERIC(5, 2),

PRIMARY KEY (SID, CID, content\_num),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (content\_num) REFERENCES Quiz(content\_num) ON DELETE CASCADE ON UPDATE CASCADE

);

## Take\_Quiz\_Question

### Relational Model:

Take\_Quiz\_Question(SID, CID, content\_num, question\_num, answer, isTrue)

### SQL Definition:

/\*

Binary relationship between Student and Quiz\_Question

keeps students' answers to each quiz question

Attributes: isTrue

\*/

CREATE TABLE Take\_Quiz\_Question(

SID INT NOT NULL,

CID INT NOT NULL,

content\_num INT NOT NULL,

question\_num INT NOT NULL,

answer ENUM('choice1', 'choice2', 'choice3') NOT NULL,

isTrue BOOLEAN,

PRIMARY KEY (SID, CID, content\_num, question\_num),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID, content\_num, question\_num) REFERENCES Quiz\_Question(CID, content\_num, question\_num) ON DELETE CASCADE ON UPDATE CASCADE

);

## Take\_Lecture

### Relational Model:

Take\_Quiz\_Lecture(SID, CID, content\_num, isCompleted)

### SQL Definition:

/\*

Binary relationship between Student and Lecture

keeps students' completion to each lecture (lecture\_completed\_threshold in lecture (e.g. %70 for each lecture))

Attributes: isCompleted

\*/

CREATE TABLE Take\_Lecture(

SID INT NOT NULL,

CID INT NOT NULL,

content\_num INT NOT NULL,

isCompleted BOOLEAN,

PRIMARY KEY (SID, CID, content\_num),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID, content\_num) REFERENCES Lecture(CID, content\_num) ON DELETE CASCADE ON UPDATE CASCADE

);

## Evaluations

### Relational Model:

Evaluations(SID, CID, evaluation)

### SQL Definition:

CREATE TABLE Evaluations(

SID INT NOT NULL,

CID INT NOT NULL,

evaluation VARCHAR(300),

PRIMARY KEY (SID, CID),

FOREIGN KEY (SID) REFERENCES Student(SID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (CID) REFERENCES Course(CID) ON DELETE CASCADE ON UPDATE CASCADE

);