IS6420 - 002 Database Theory and Design

Reverse-engineering a transactional database for the learning platform **Udemy**

GROUP 06 | Rylan Tribush u1497366
Ketki Kulkarni u1476451
Vedika Garg u1478855
Anish Khairnar u1420954

October 24, 2023

TABLE OF CONTENT

EXECUTIVE SUMMARY

Udemy is a global marketplace for learning and teaching online where millions of students master new skills and achieve their goals by learning from an extensive library of over 150,000 courses taught by expert instructors.

The objective of the project is to analyze and understand important components of the transactional database that supports Udemy's widely used online learning platform. Our group project will specifically concentrate on comprehending and dissecting two significant database interactions that involve students and Udemy courses.

Subscription and Completion Relationship:

The focus of the initial database interaction is to understand the intricate correlation between students and their course subscriptions and completions. By examining this aspect, our objective is to acquire knowledge about the mechanisms that regulate course access, progress tracking, and successful course completion. This comprehension will aid in enhancing user experience and refining course recommendations, among other advantages.

Course Categorization Database:

The second primary objective of our project is to analyze the underlying structure of the database responsible for course categorization and its integration with student subscriptions. A comprehensive investigation into this correlation will provide insights into Udemy's methodology for organizing its extensive course library into different categories and the impact of this categorization on student engagement. This understanding can guide the creation of user-friendly navigation features and personalized learning paths for students.

UDMEY APPLICATION

INTRODUCTION

Udemy, a global marke-tplace for online learning and te-aching, has emerged as a frontrunne-r in democratizing education. Since its launch in 2010, Ude-my has revolutionized the traditional e-ducation model by offering a diverse, scalable, and cost-efficient le-arning solution. Through its unique platform, expert instructors are connected with learne-rs seeking personal de-velopment sessions and course-s across various disciplines and languages.

VISION

Udemy's mission is to enhance lives through learning by providing a broad range of educational resources that are accessible and affordable to people from all over the world. Through its comprehensive platform, Udemy enables people to progress in their professional lives, acquire essential skills, and pursue hobbies and interests regardless of their financial situation or location.

OBJECTIVES

1. Accessibility and Affordable: In order to break down barriers in education, it is imperative to provide affordable, accessible, and diverse courses to individuals across the world.
2. Diversity & Quality: Ensure a diverse and high-quality collection of courses that cater to various proficiency levels and disciplines by maintaining and expanding a well-curated catalog.
3. Innovate: Consistently utilize and incorporate technological advancements to enhance the learning experience, rendering it more interactive, personalized, and user-friendly.
4. Empowerment: Enable individuals to follow their professional and personal interests and advance their careers by engaging in ongoing education.

PRODUCT & SERVICES

Udemy is a leading provider of e-learning solutions that offers customized learning experiences through progress tracking and personalized content recommendations. The platform provides a dynamic course catalog that is regularly updated by instructors to ensure that the content remains current and relevant. Upon completion of a course, learners are awarded certificates that enhance their professional credibility. This comprehensive approach underscores Udemy's commitment to providing an engaging, up-to-date, and rewarding educational journey.

HOW DOES UDEMY USE TRANSACTIONAL DATABASES?

Udemy's operations heavily depend on its transactional database, which is crucial for efficiently managing user subscriptions, course enrollments, and progress tracking. This database operates in real-time, ensuring precise and accurate information for many users. The system plays a vital role in organizing courses, maintaining comprehensive records to provide a personalized user journey. Additionally, it oversees the important task of issuing certificates upon course completion, which is essential for learners' professional growth. This well-structured digital infrastructure guarantees a seamless, interactive, and fulfilling learning experience, catering to both educational and professional development requirements.

USER REQUIREMENTS

We would like to reverse engineer aspects of Udemy's transactional databases as it is a widely used online learning platform that enables individuals to develop and enroll in courses on various topics. The requirements that we have identified are for an e-learning platform, and they relate to user roles, account administration, course administration, payment system, and other features. The business requirement is mentioned below in greater detail -

1. **User Roles**:

- *Students* - The main users of the platform are Students, who enroll in courses offered by Instructors in order to gain knowledge. Their primary activities include viewing courses, signing up for courses, monitoring progress, and obtaining certificates.
- *Instructor* - Instructors are the creators of courses on the platform, who require tools to upload and maintain course content.

2. **Account Management:**

- **Common Attributes for Both Users:**
- *First Name and Last Name* **-** These attributes are necessary for personal identification and interaction on the platform.
- *Email ID* **-** Unique email addresses are used for registration and communication.
- *Username and Password* **-** The username is a combination of FirstName and LastName and the password is essential for secure login and account access.

3. **Course Management:** Attributes for Each Course are listed below:

- *Unique Course ID* **-** A unique identifier is crucial for tracking courses and distinguishing them from one another.
- *Course Name* **-** The name of the course is how it's presented to students.
- *Description* **-** A detailed course description is necessary to inform potential students about the course content.
- *Duration* **-** This provides an estimate of the time commitment required to complete the course.
- *Updating Date* **-** Keeping track of when the course was last modified helps users identify the most recent content.
- *Number of Lectures* **-** This is valuable information for students to gauge the course's depth.
- *Category* - Categorizing courses by subject matter (e.g., Science, Arts, Technology) aids in search and discovery.

4. **Student-Specific Requirements:** Course Enrollment for students are as below**:**

- *Ability to browse and enroll in various courses*: Students should easily find courses and enroll in them.
- *Capture Course Fee and Enrollment Date*: Recording fees and dates is essential for tracking student transactions.

5. **Certification for a particular course is as follows:**

- *On completion of a course, students should receive a certification*: Certificates of completion add value to the courses.
- *Each certificate should have a unique Certification Id, Date of Issue, and Title for verification*: These details provide the description of the course and enhance the credibility and traceability of certificates.

5. **Instructor-Specific Requirements:**

- *Instructors should be able to upload new courses with all required course attributes*: The platform provide user-friendly tools for creating and updating courses.

7. **Payment Model for Students:**

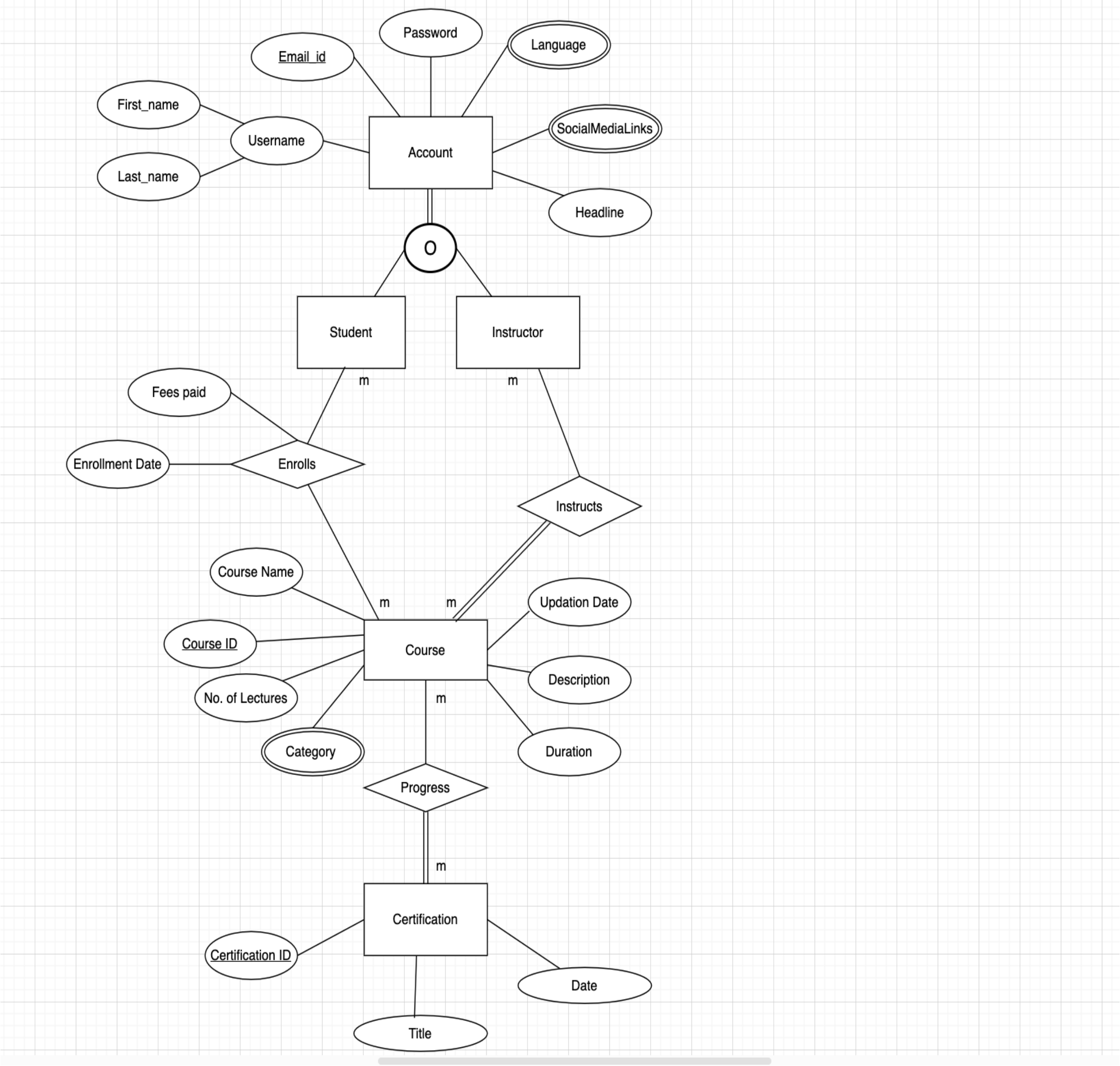
- Students are charged a one-time fee for each course they enroll in, facilitating straightforward payment transactions.

8. **Additional Functionalities such as Search and Filter:**

- *Users should be able to search for courses and filter them based on categories, instructors* **-** Enhanced search and filtering capabilities improve the user experience.

DATABASE DESIGN

1. CONCEPTUAL MODEL



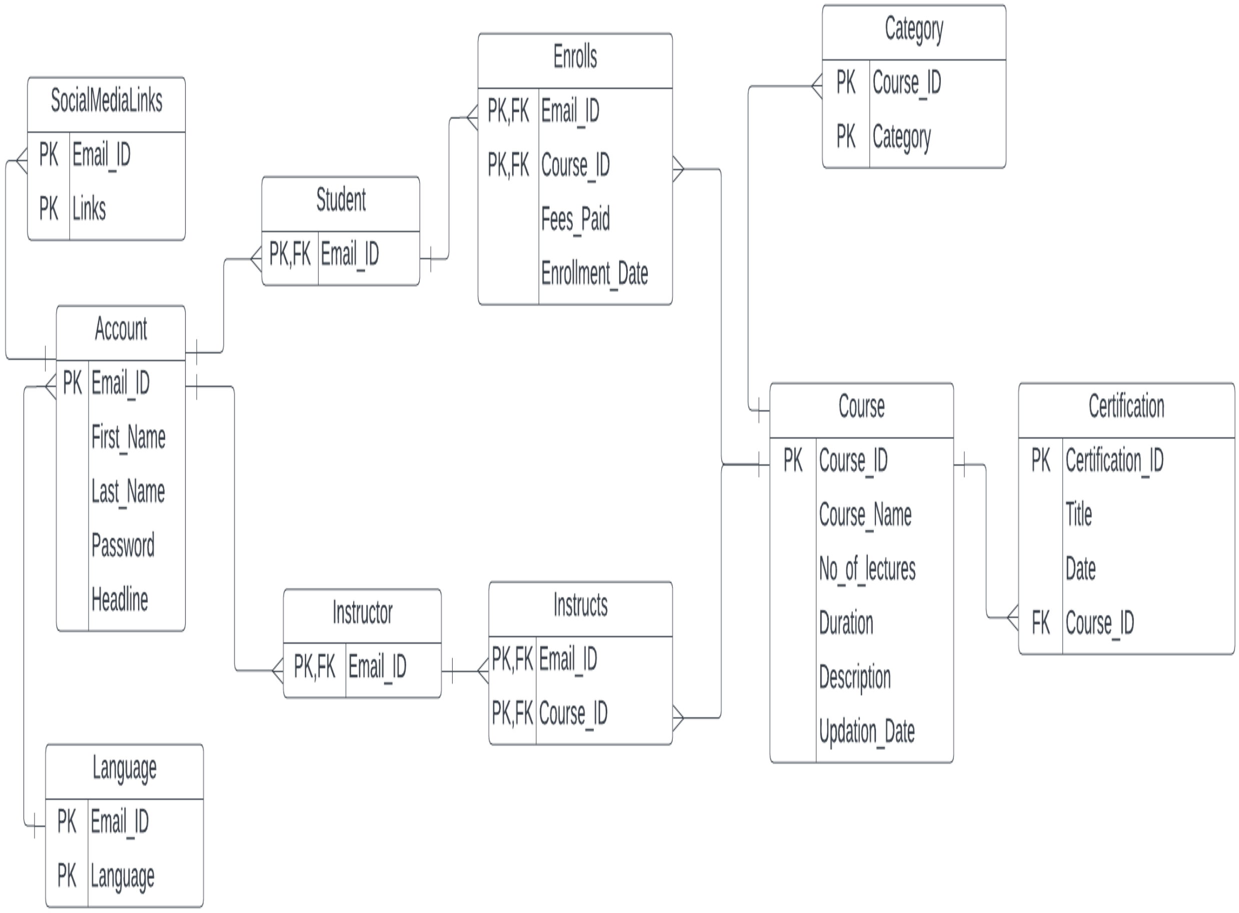
The ER (Entity Relationship) diagram shows how the different parts of Udemy platform are connected. The Account is the main part, and it lets users use the platform. Each Account has a unique username and other information like email address as an identifier and language(multi-valued) preference. This makes the platform more personal for each user. Here, at Udemy once an account is created, the user can be instructor as well as student so there is an overlapping for these two.

Students start learning after they enroll in courses. Students can enroll in multiple courses. Their progress in each class is tracked, which helps both students and instructors see how engaged they are and how well they are learning. Courses are not just listed by name or ID (unique identifier for each class) but also by content, length, and when they were last updated. This gives potential students a promising idea of what they are signing up for.

Instructors are the main people who make or teach the content. Instructor can upload/teach many courses. They are linked to courses through the Instructs relationship, which makes sure they can share their knowledge and update course materials as needed.

Finally, the model shows how important certifications are. When students finish their courses, the platform gives them certifications to show how hard they worked and how much they learned. These certifications have unique IDs and titles, and they show that students have gained skills and knowledge on the platform.

2.LOGICAL MODEL



The above logical model indicates the relationship between various Udemy entities such as Account, Course, and Certification. The model shows that the account can be a student as well as an Instructor, i.e., the account acts as a supertype for the subtype's student and instructor.

An account can be recognized with the help of a unique Email ID. The platform lets you enter your first name, last name, password, and headline. It also allows you to enter multiple social media links and languages.

The students are allowed to enroll in multiple courses as indicated by the Enrolls entity. At the time of enrollment, the platform notes the enrollment date and the fees paid for a particular course. On the other hand, an instructor instructs the course and is allowed to upload multiple courses. The Email_ID acts as both the primary key as well as the foreign key for the student and instructor entities.

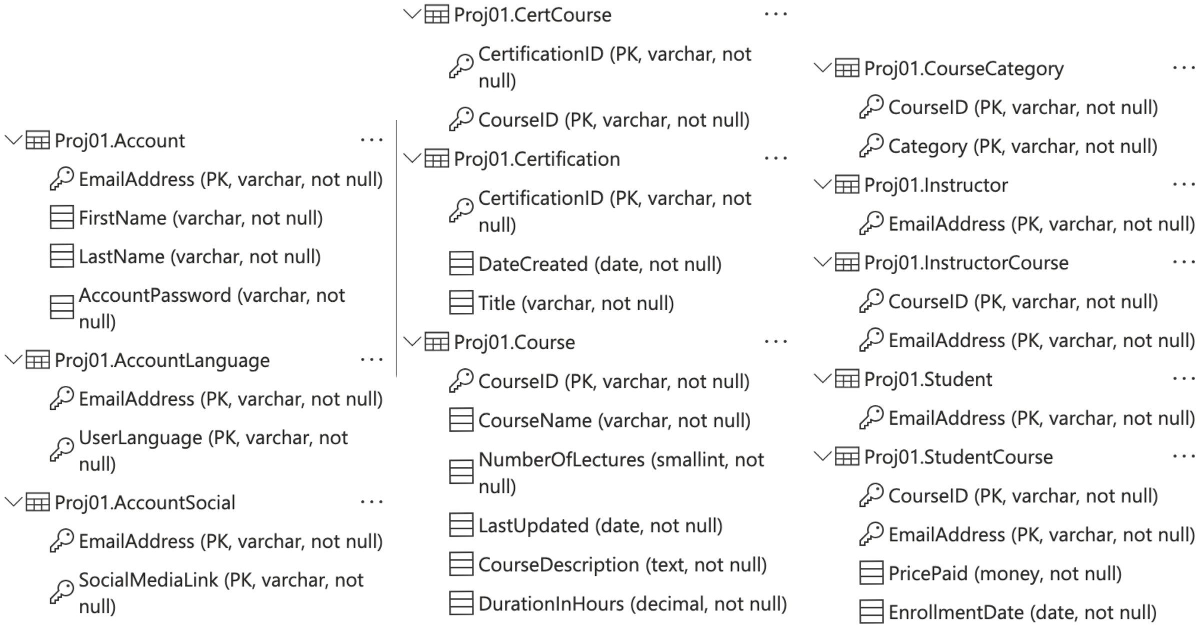
A Course can be uniquely identified via the Course_ID. Udemy also stores the information about the course name, duration, description, updation_date, and no_of_lectures. The courses can be assigned to multiple categories. This information helps a student find a suitable course for enrollment.

Once the Course is completed, a certificate is generated. The students can use this certificate as proof of course completion. The Certification can be tracked using the unique Certification_ID. It also has information regarding the title and date. The Course_ID acts as a foreign key for the Certification entity.

DATABASE IMPLEMENTATION

PHYSICAL MODEL

The physical implementation of Udemy’s database follows naturally from the logical model above. Data types were chosen conservatively to accommodate a wide range of names and user inputs. The database structure and choice of data types is summarized in the following screenshots:



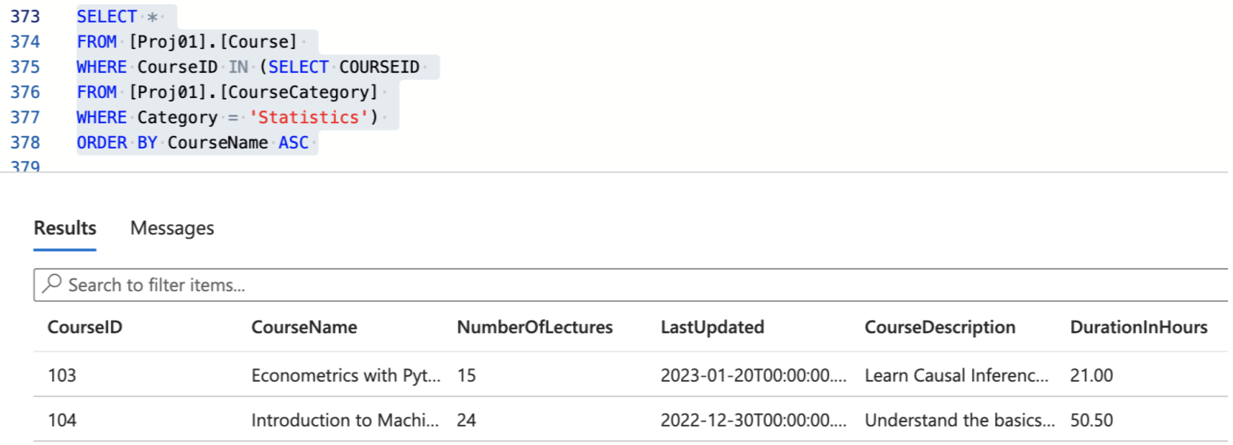
- Proj01.Account
  - EmailAddress (PK, varchar, not null)
  - FirstName (varchar, not null)
  - LastName (varchar, not null)
  - AccountPassword (varchar, not null)
- Proj01.AccountLanguage
  - EmailAddress (PK, varchar, not null)
  - UserLanguage (PK, varchar, not null)
- Proj01.AccountSocial
  - EmailAddress (PK, varchar, not null)
  - SocialMediaLink (PK, varchar, not null)
- Proj01.CertCourse
  - CertificationID (PK, varchar, not null)
  - CourseID (PK, varchar, not null)
- Proj01.Certification
  - CertificationID (PK, varchar, not null)
  - DateCreated (date, not null)
  - Title (varchar, not null)
- Proj01.Course
  - CourseID (PK, varchar, not null)
  - CourseName (varchar, not null)
  - NumberOfLectures (smallint, not null)
  - LastUpdated (date, not null)
  - CourseDescription (text, not null)
  - DurationInHours (decimal, not null)
- Proj01.CourseCategory
  - CourseID (PK, varchar, not null)
  - Category (PK, varchar, not null)
- Proj01.Instructor
  - EmailAddress (PK, varchar, not null)
- Proj01.InstructorCourse
  - CourseID (PK, varchar, not null)
  - EmailAddress (PK, varchar, not null)
- Proj01.Student
  - EmailAddress (PK, varchar, not null)
- Proj01.StudentCourse
  - CourseID (PK, varchar, not null)
  - EmailAddress (PK, varchar, not null)
  - PricePaid (money, not null)
  - EnrollmentDate (date, not null)

FINAL REQUIREMENTS

Category	Description	Status
Student	Interacts with the Course by enrolling	Completed
Student	Student provides rating and reviews to the course and the instructor	Future Consideration
Student	The student pays a one-time fee at the time of enrollment	Completed
Instructor	Uploads the course and updates it	Completed
Instructor	Receives a monthly payment based on the number of courses uploaded	Future Consideration
Course	Can be searched based on the category and instructor	Completed
Certification	A unique certificate is generated at the time of course completion	Completed

DEMONSTRATION OF FEATURES

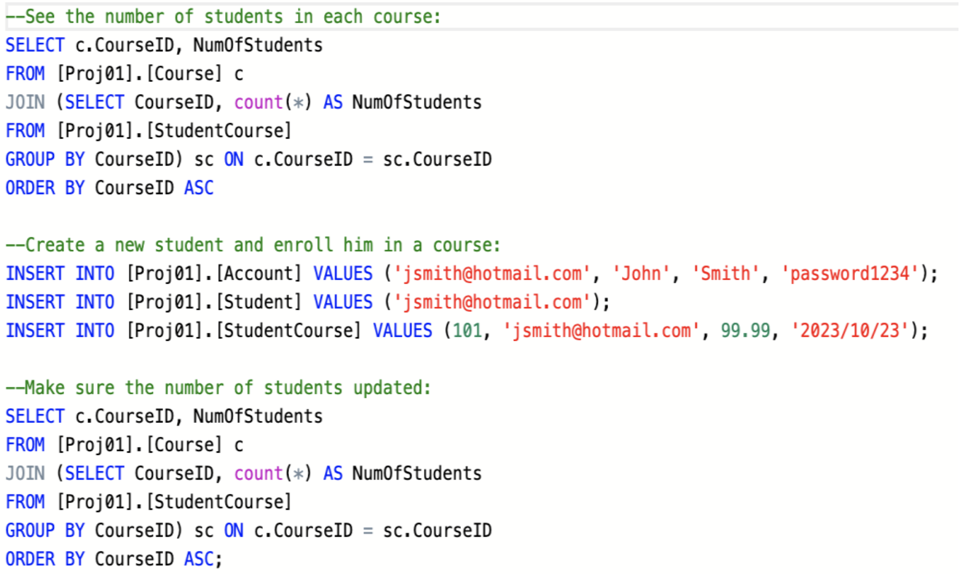
#1. Filter course list by subject—Students need to have the ability to filter classes by subject matter. Udemy has many thousands of course offerings, so prospective students need the ability to navigate the long list of possibilities. In the example below, only courses in the “statistics” category are displayed.



```
SELECT *
FROM [Proj01].[Course]
WHERE CourseID IN (SELECT COURSEID
FROM [Proj01].[CourseCategory]
WHERE Category = 'Statistics')
ORDER BY CourseName ASC
```

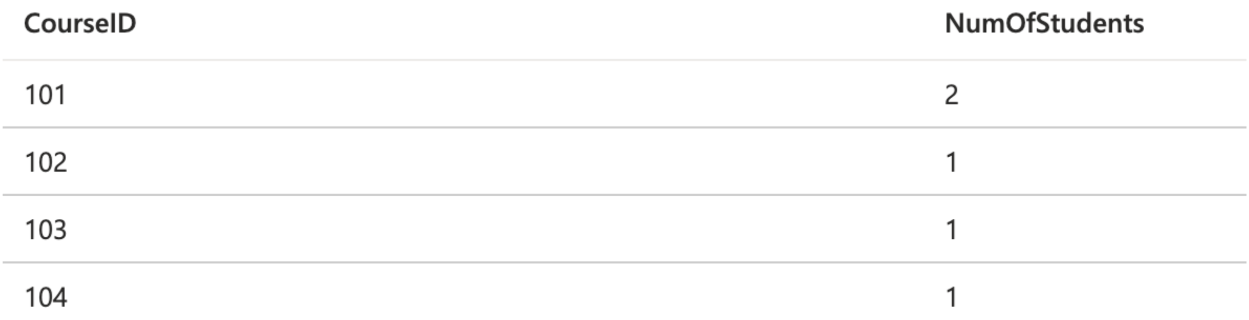
CourseID	CourseName	NumberOfLectures	LastUpdated	CourseDescription	DurationInHours
103	Econometrics with Pyt...	15	2023-01-20T00:00:00....	Learn Causal Inferenc...	21.00
104	Introduction to Machi...	24	2022-12-30T00:00:00....	Understand the basics...	50.50

#2. Create a new student and add him to a class—The database needs to be able to accommodate the addition of new students and their enrollment into courses. In the example below, a new student account is created for John Smith. He is then enrolled in the Web Deveolpment course. Before and after the addition, the number of students per course is calculated.



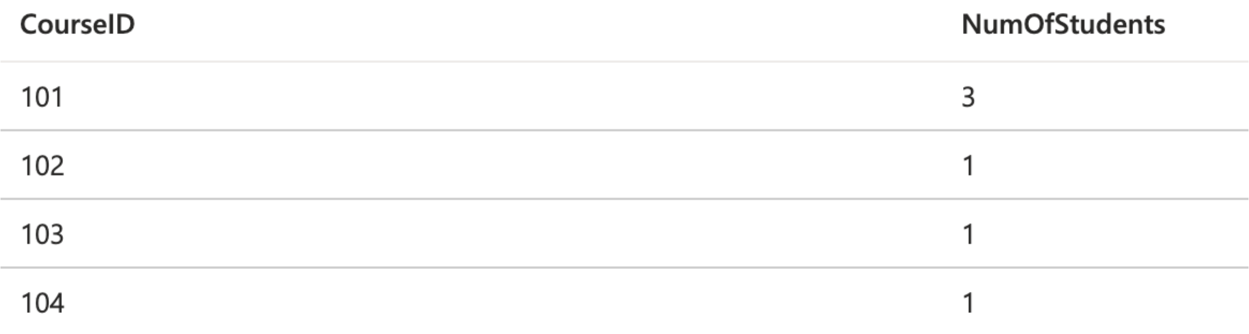
```
--See the number of students in each course:
SELECT c.CourseID, NumOfStudents
FROM [Proj01].[Course] c
JOIN (SELECT CourseID, count(*) AS NumOfStudents
FROM [Proj01].[StudentCourse]
GROUP BY CourseID) sc ON c.CourseID = sc.CourseID
ORDER BY CourseID ASC

--Create a new student and enroll him in a course:
INSERT INTO [Proj01].[Account] VALUES ('jsmith@hotmail.com', 'John', 'Smith', 'password1234');
INSERT INTO [Proj01].[Student] VALUES ('jsmith@hotmail.com');
INSERT INTO [Proj01].[StudentCourse] VALUES (101, 'jsmith@hotmail.com', 99.99, '2023/10/23');

--Make sure the number of students updated:
SELECT c.CourseID, NumOfStudents
FROM [Proj01].[Course] c
JOIN (SELECT CourseID, count(*) AS NumOfStudents
FROM [Proj01].[StudentCourse]
GROUP BY CourseID) sc ON c.CourseID = sc.CourseID
ORDER BY CourseID ASC;
```

Before:



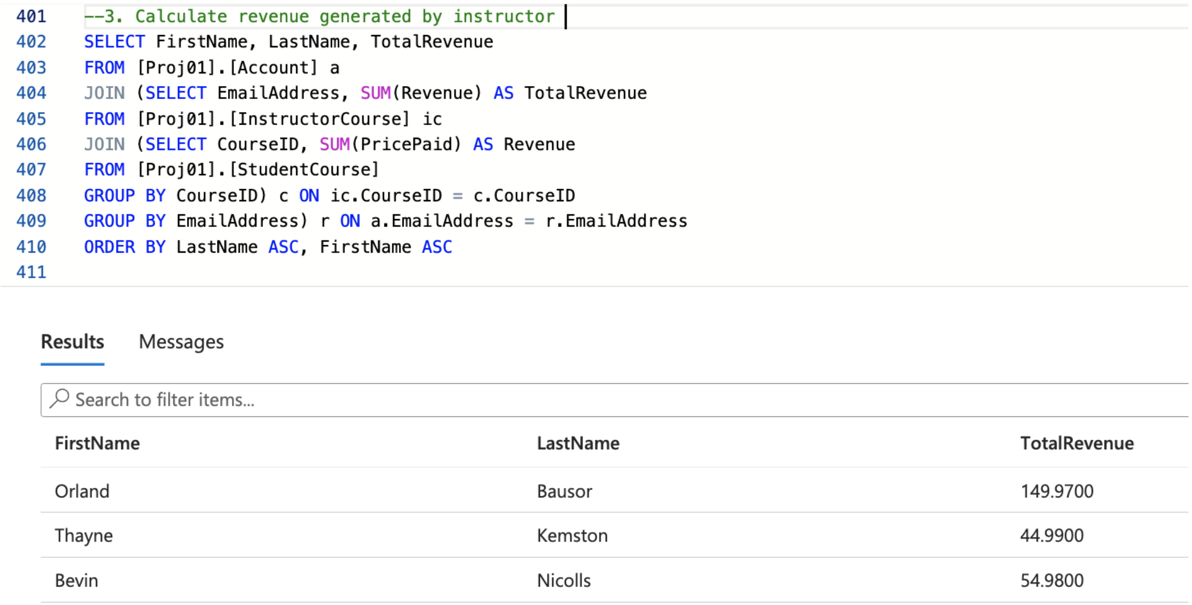
CourseID	NumOfStudents
101	2
102	1
103	1
104	1

After:



CourseID	NumOfStudents
101	3
102	1
103	1
104	1

#3. Calculate the revenue generated by each instructor—Udemy will want to identify its most important instructors. The query below calculates the amount of revenue generated by each instructor, based on the price paid by the enrolled students.



```sql
--3. Calculate revenue generated by instructor
SELECT FirstName, LastName, TotalRevenue
FROM [Proj01].[Account] a
JOIN (SELECT EmailAddress, SUM(Revenue) AS TotalRevenue
FROM [Proj01].[InstructorCourse] ic
JOIN (SELECT CourseID, SUM(PricePaid) AS Revenue
FROM [Proj01].[StudentCourse]
GROUP BY CourseID) c ON ic.CourseID = c.CourseID
GROUP BY EmailAddress) r ON a.EmailAddress = r.EmailAddress
ORDER BY LastName ASC, FirstName ASC
```

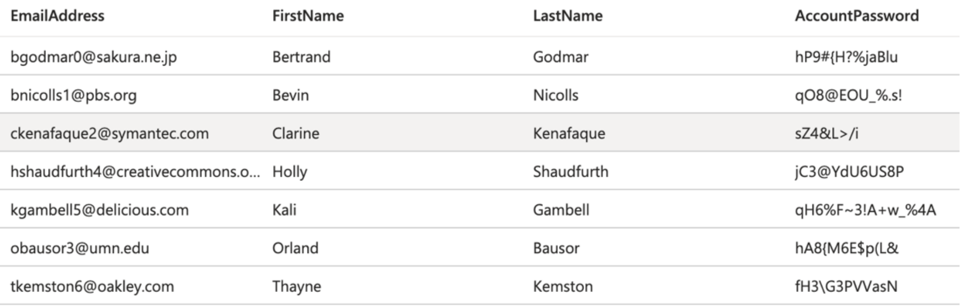
FirstName	LastName	TotalRevenue
Orland	Bausor	149.9700
Thayne	Kemston	44.9900
Bevin	Nicolls	54.9800

APPENDIX

Screenshot of Sample Data

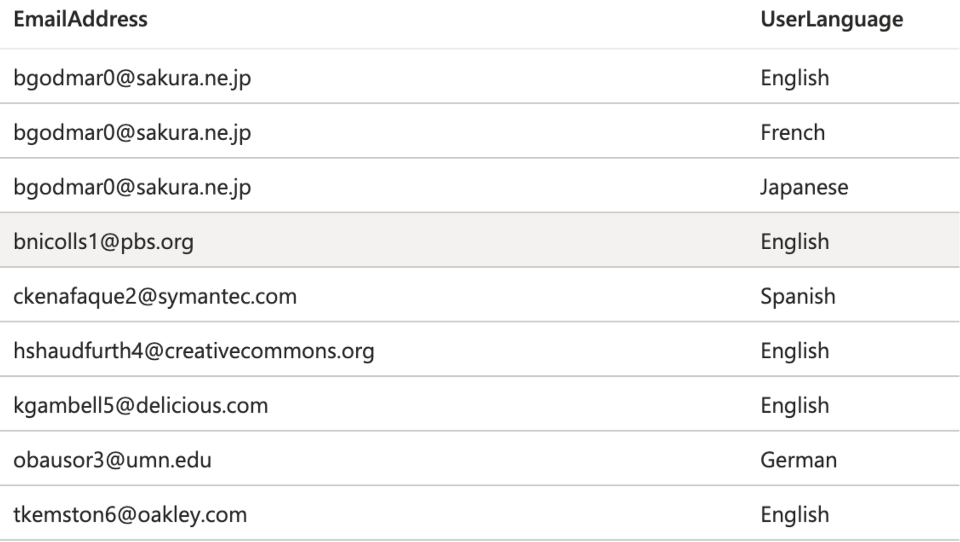
Sample data were generated in order to test and demonstrate the features of the database. One of the key attributes of the database is that an individual account can be both a student and a teacher simultaneously. Additionally, the database implementation needs to accommodate the fact that accounts can have multiple languages and courses can have multiple categories. The sample data chosen reflects these essential business requirements.

Account:



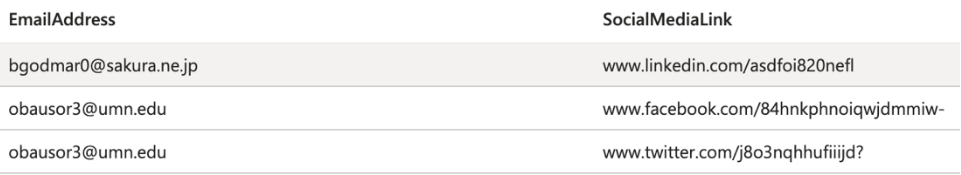
EmailAddress	FirstName	LastName	AccountPassword
bgodmar0@sakura.ne.jp	Bertrand	Godmar	hP9#{H?%jaBlu
bnicolls1@pbs.org	Bevin	Nicolls	qO8@EOU_%.s!
ckenafaque2@symantec.com	Clarine	Kenafaque	sZ4&L>/i
hshaudfurth4@creativecommons.o...	Holly	Shaudfurth	jC3@YdU6US8P
kgambell5@delicious.com	Kali	Gambell	qH6%F~3!A+w_%4A
obausor3@umn.edu	Orland	Bausor	hA8{M6E$p(L&
tkemston6@oakley.com	Thayne	Kemston	fH3\G3PVVasN

AccountLanguage:



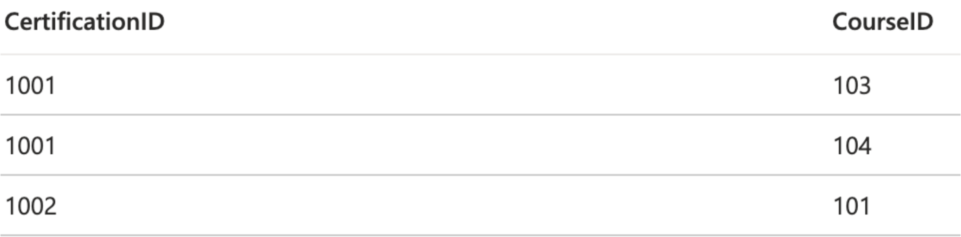
EmailAddress	UserLanguage
bgodmar0@sakura.ne.jp	English
bgodmar0@sakura.ne.jp	French
bgodmar0@sakura.ne.jp	Japanese
bnicolls1@pbs.org	English
ckenafaque2@symantec.com	Spanish
hshaudfurth4@creativecommons.org	English
kgambell5@delicious.com	English
obausor3@umn.edu	German
tkemston6@oakley.com	English

AccountSocial:



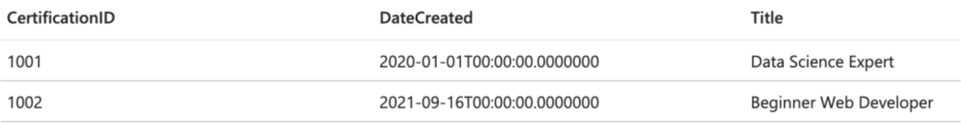
EmailAddress	SocialMediaLink
bgodmar0@sakura.ne.jp	www.linkedin.com/asdfoi820nefl
obausor3@umn.edu	www.facebook.com/84hnkphnoiqwjdmmiw-
obausor3@umn.edu	www.twitter.com/j8o3nqhhufiiijd?

CertCourse:



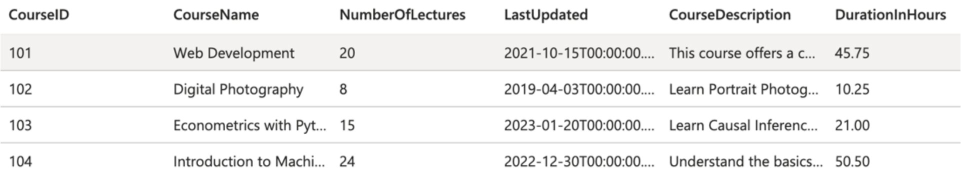
CertificationID	CourseID
1001	103
1001	104
1002	101

Certification:



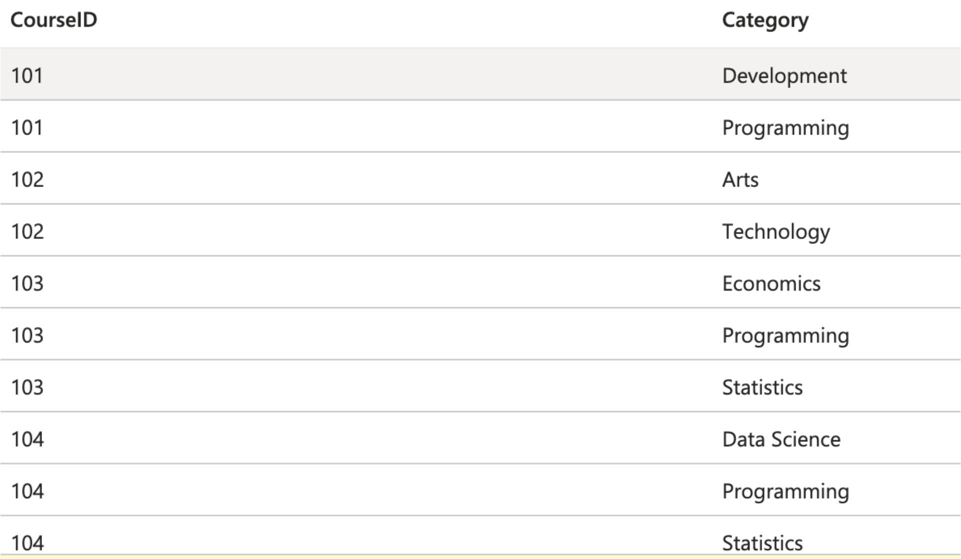
CertificationID	DateCreated	Title
1001	2020-01-01T00:00:00.0000000	Data Science Expert
1002	2021-09-16T00:00:00.0000000	Beginner Web Developer

Course:



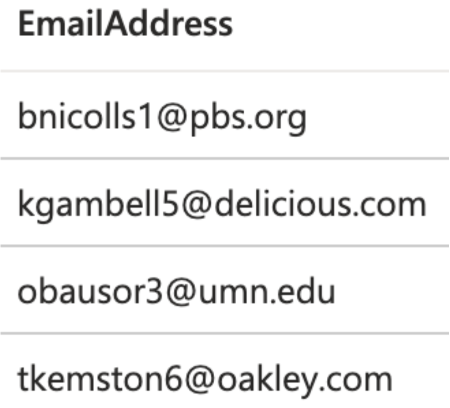
CourseID	CourseName	NumberOfLectures	LastUpdated	CourseDescription	DurationInHours
101	Web Development	20	2021-10-15T00:00:00....	This course offers a c...	45.75
102	Digital Photography	8	2019-04-03T00:00:00....	Learn Portrait Photog...	10.25
103	Econometrics with Pyt...	15	2023-01-20T00:00:00....	Learn Causal Inferenc...	21.00
104	Introduction to Machi...	24	2022-12-30T00:00:00....	Understand the basics...	50.50

CourseCategory:



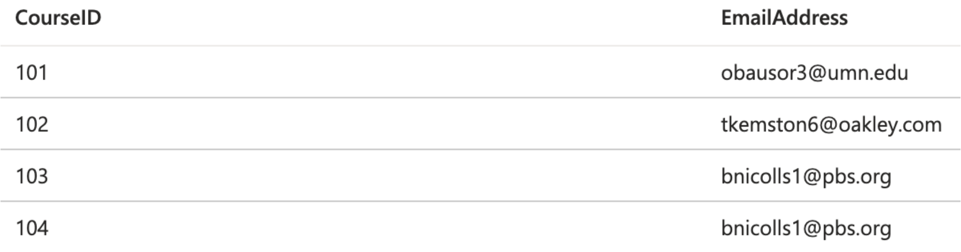
CourseID	Category
101	Development
101	Programming
102	Arts
102	Technology
103	Economics
103	Programming
103	Statistics
104	Data Science
104	Programming
104	Statistics

Instructor:



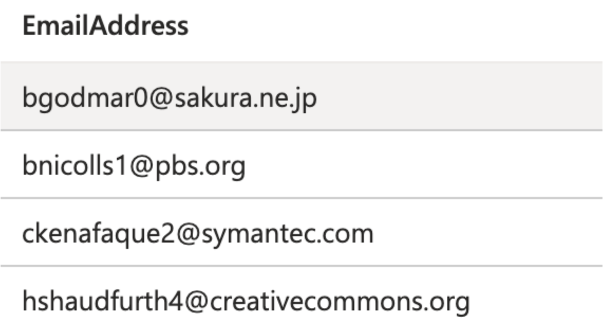
EmailAddress
bnicolls1@pbs.org
kgambell5@delicious.com
obausor3@umn.edu
tkemston6@oakley.com

InstructorCourse:



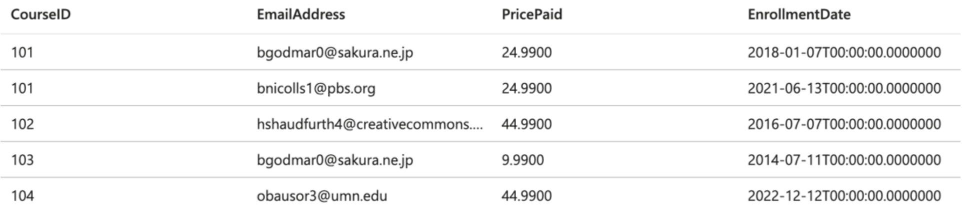
CourseID	EmailAddress
101	obausor3@umn.edu
102	tkemston6@oakley.com
103	bnicolls1@pbs.org
104	bnicolls1@pbs.org

Student:



EmailAddress
bgodmar0@sakura.ne.jp
bnicolls1@pbs.org
ckenafaque2@symantec.com
hshaudfurth4@creativecommons.org

StudentCourse:



CourseID	EmailAddress	PricePaid	EnrollmentDate
101	bgodmar0@sakura.ne.jp	24.9900	2018-01-07T00:00:00.0000000
101	bnicolls1@pbs.org	24.9900	2021-06-13T00:00:00.0000000
102	hshaudfurth4@creativecommons....	44.9900	2016-07-07T00:00:00.0000000
103	bgodmar0@sakura.ne.jp	9.9900	2014-07-11T00:00:00.0000000
104	obausor3@umn.edu	44.9900	2022-12-12T00:00:00.0000000

Source Code

```sql
CREATE SCHEMA Proj01;
GO

DROP TABLE IF EXISTS Proj01.CertCourse;
DROP TABLE IF EXISTS Proj01.Certification;
DROP TABLE IF EXISTS Proj01.InstructorCourse;
DROP TABLE IF EXISTS Proj01.StudentCourse;
DROP TABLE IF EXISTS Proj01.CourseCategory;
DROP TABLE IF EXISTS Proj01.Course;
DROP TABLE IF EXISTS Proj01.BankAccount;
DROP TABLE IF EXISTS Proj01.CreditDebit;
DROP TABLE IF EXISTS Proj01.Instructor;
DROP TABLE IF EXISTS Proj01.Student;
DROP TABLE IF EXISTS Proj01.AccountSocial;
DROP TABLE IF EXISTS Proj01.AccountLanguage;
DROP TABLE IF EXISTS Proj01.Account;

CREATE TABLE Proj01.Account (
EmailAddress VARCHAR(100) NOT NULL,
FirstName VARCHAR(100) NOT NULL,
LastName VARCHAR(100) NOT NULL,
AccountPassword VARCHAR(50) NOT NULL,
CONSTRAINT Account_PK PRIMARY KEY (EmailAddress)
);

CREATE TABLE Proj01.AccountLanguage (
EmailAddress VARCHAR(100) NOT NULL,
UserLanguage VARCHAR(100) NOT NULL,
CONSTRAINT AccountLanguage_PK PRIMARY KEY (EmailAddress, UserLanguage),
CONSTRAINT AccountLanguage_FK FOREIGN KEY (EmailAddress)
REFERENCES Proj01.Account(EmailAddress)
);

CREATE TABLE Proj01.AccountSocial (
EmailAddress VARCHAR(100) NOT NULL,
SocialMediaLink VARCHAR(100) NOT NULL,
CONSTRAINT AccountSocial_PK PRIMARY KEY (EmailAddress, SocialMediaLink),
CONSTRAINT AccountSocial_FK FOREIGN KEY (EmailAddress)
REFERENCES Proj01.Account(EmailAddress)
);

CREATE TABLE Proj01.Student (
EmailAddress VARCHAR(100) NOT NULL,
CONSTRAINT Student_PK PRIMARY KEY (EmailAddress),
CONSTRAINT Student_FK FOREIGN KEY (EmailAddress)
REFERENCES Proj01.Account(EmailAddress)
);

CREATE TABLE Proj01.Instructor (
EmailAddress VARCHAR(100) NOT NULL,
CONSTRAINT Instructor_PK PRIMARY KEY (EmailAddress),
CONSTRAINT Instroctor_FK FOREIGN KEY (EmailAddress)
REFERENCES Proj01.Account(EmailAddress)
);
```

```
CREATE TABLE Proj01.Course (
CourseID VARCHAR(20) NOT NULL,
CourseName VARCHAR(100) NOT NULL,
NumberOfLectures SMALLINT NOT NULL,
LastUpdated DATE NOT NULL,
CourseDescription TEXT NOT NULL,
DurationInHours DECIMAL(5, 2) NOT NULL,
CONSTRAINT Course_PK PRIMARY KEY (CourseID),
);

CREATE TABLE Proj01.CourseCategory (
CourseID VARCHAR(20) NOT NULL,
Category VARCHAR(50) NOT NULL,
CONSTRAINT CourseCategory_PK PRIMARY KEY (CourseID, Category),
CONSTRAINT CourseCategory_FK FOREIGN KEY (CourseID) REFERENCES Proj01.Course(CourseID)
);

CREATE TABLE Proj01.StudentCourse (
CourseID VARCHAR(20) NOT NULL,
EmailAddress VARCHAR(100) NOT NULL,
PricePaid MONEY NOT NULL,
EnrollmentDate DATE NOT NULL,
CONSTRAINT StudentCourse_PK PRIMARY KEY (CourseID, EmailAddress),
CONSTRAINT StudentCourse_FK1 FOREIGN KEY (CourseID) REFERENCES
Proj01.Course(CourseID),
CONSTRAINT StudentCourse_FK2 FOREIGN KEY (EmailAddress) REFERENCES
Proj01.Student(EmailAddress)
);

CREATE TABLE Proj01.InstructorCourse (
CourseID VARCHAR(20) NOT NULL,
EmailAddress VARCHAR(100) NOT NULL,
CONSTRAINT InstructorCourse_PK PRIMARY KEY (CourseID, EmailAddress),
CONSTRAINT InstructorCourse_FK1 FOREIGN KEY (CourseID) REFERENCES
Proj01.Course(CourseID),
CONSTRAINT InstructorCourse_FK2 FOREIGN KEY (EmailAddress) REFERENCES
Proj01.Instructor(EmailAddress)
);

CREATE TABLE Proj01.Certification (
CertificationID VARCHAR(20) NOT NULL,
DateCreated DATE NOT NULL,
Title VARCHAR(500) NOT NULL,
CONSTRAINT Certification_PK PRIMARY KEY (CertificationID)
);

CREATE TABLE Proj01.CertCourse (
CertificationID VARCHAR(20) NOT NULL,
CourseID VARCHAR(20) NOT NULL,
CONSTRAINT CertCourse_PK PRIMARY KEY (CertificationID, CourseID),
CONSTRAINT CertCourse_FK1 FOREIGN KEY (CertificationID)
REFERENCES Proj01.Certification(CertificationID),
CONSTRAINT CertCourse_FK2 FOREIGN KEY (CourseID)
REFERENCES Proj01.Course(CourseID)
```

```
);

INSERT INTO [Proj01].[Account] VALUES ('bgodmar0@sakura.ne.jp', 'Bertrand', 'Godmar',
'hP9#{H?%jaBlu');
INSERT INTO [Proj01].[Account] VALUES ('bnicolls1@pbs.org', 'Bevin', 'Nicolls',
'qO8@EOU_%.s!');
INSERT INTO [Proj01].[Account] VALUES ('ckenafaque2@symantec.com', 'Clarine',
'Kenafaque', 'sZ4&L>/i');
INSERT INTO [Proj01].[Account] VALUES ('obausor3@umn.edu', 'Orland', 'Bausor',
'hA8{M6E$p(L&');
INSERT INTO [Proj01].[Account] VALUES ('hshaudfurth4@creativecommons.org', 'Holly',
'Shaudfurth', 'jC3@YdU6US8P');
INSERT INTO [Proj01].[Account] VALUES ('kgambell5@delicious.com', 'Kali', 'Gambell',
'qH6%F~3!A+w_%4A');
INSERT INTO [Proj01].[Account] VALUES ('tkemston6@oakley.com', 'Thayne', 'Kemston',
'fH3\G3PVVasN');

INSERT INTO [Proj01].[AccountLanguage] VALUES ('bgodmar0@sakura.ne.jp', 'English');
INSERT INTO [Proj01].[AccountLanguage] VALUES ('bgodmar0@sakura.ne.jp', 'French');
INSERT INTO [Proj01].[AccountLanguage] VALUES ('bgodmar0@sakura.ne.jp', 'Japanese');
INSERT INTO [Proj01].[AccountLanguage] VALUES ('bnicolls1@pbs.org', 'English');
INSERT INTO [Proj01].[AccountLanguage] VALUES ('ckenafaque2@symantec.com', 'Spanish');
INSERT INTO [Proj01].[AccountLanguage] VALUES ('obausor3@umn.edu', 'German');
INSERT INTO [Proj01].[AccountLanguage] VALUES ('hshaudfurth4@creativecommons.org',
'English');
INSERT INTO [Proj01].[AccountLanguage] VALUES ('kgambell5@delicious.com', 'English');
INSERT INTO [Proj01].[AccountLanguage] VALUES ('tkemston6@oakley.com', 'English');

INSERT INTO [Proj01].[AccountSocial] VALUES ('bgodmar0@sakura.ne.jp',
'www.linkedin.com/asdfoi820nefl');
INSERT INTO [Proj01].[AccountSocial] VALUES ('obausor3@umn.edu',
'www.facebook.com/84hnkphnoiqwjdmmiw-');
INSERT INTO [Proj01].[AccountSocial] VALUES ('obausor3@umn.edu',
'www.twitter.com/j8o3nqhhufiiijd?');

INSERT INTO [Proj01].[Student] VALUES ('bgodmar0@sakura.ne.jp');
INSERT INTO [Proj01].[Student] VALUES ('bnicolls1@pbs.org');
INSERT INTO [Proj01].[Student] VALUES ('ckenafaque2@symantec.com');
INSERT INTO [Proj01].[Student] VALUES ('obausor3@umn.edu');
INSERT INTO [Proj01].[Student] VALUES ('hshaudfurth4@creativecommons.org');

INSERT INTO [Proj01].[Instructor] VALUES ('bnicolls1@pbs.org');
INSERT INTO [Proj01].[Instructor] VALUES ('obausor3@umn.edu');
INSERT INTO [Proj01].[Instructor] VALUES ('kgambell5@delicious.com');
INSERT INTO [Proj01].[Instructor] VALUES ('tkemston6@oakley.com');

INSERT INTO [Proj01].[Course] VALUES (101, 'Web Development', 20, '2021/10/15',
'This course offers a comprehensive look into the entire web development process',
45.75);
INSERT INTO [Proj01].[Course] VALUES (102, 'Digital Photography', 8, '2019/04/03',
'Learn Portrait Photography, Landscape Photography, and more',
10.25);
INSERT INTO [Proj01].[Course] VALUES (103, 'Econometrics with Python', 15,
'2023/01/20',
'Learn Causal Inference & Statistical Modeling with Python',
21.0);
```

```
INSERT INTO [Proj01].[Course] VALUES (104, 'Introduction to Machine Learning', 24,
'2022/12/30',
'Understand the basics of Machine Learning in R',
50.5);

INSERT INTO [Proj01].[CourseCategory] VALUES (101, 'Development');
INSERT INTO [Proj01].[CourseCategory] VALUES (101, 'Programming');
INSERT INTO [Proj01].[CourseCategory] VALUES (102, 'Arts');
INSERT INTO [Proj01].[CourseCategory] VALUES (102, 'Technology');
INSERT INTO [Proj01].[CourseCategory] VALUES (103, 'Economics');
INSERT INTO [Proj01].[CourseCategory] VALUES (103, 'Statistics');
INSERT INTO [Proj01].[CourseCategory] VALUES (103, 'Programming');
INSERT INTO [Proj01].[CourseCategory] VALUES (104, 'Data Science');
INSERT INTO [Proj01].[CourseCategory] VALUES (104, 'Statistics');
INSERT INTO [Proj01].[CourseCategory] VALUES (104, 'Programming');

INSERT INTO [Proj01].[InstructorCourse] VALUES (101, 'obausor3@umn.edu');
INSERT INTO [Proj01].[InstructorCourse] VALUES (102, 'tkemston6@oakley.com');
INSERT INTO [Proj01].[InstructorCourse] VALUES (103, 'bnicolls1@pbs.org');
INSERT INTO [Proj01].[InstructorCourse] VALUES (104, 'bnicolls1@pbs.org');

INSERT INTO [Proj01].[StudentCourse] VALUES (101, 'bgodmar0@sakura.ne.jp', 24.99,
'2018/01/07');
INSERT INTO [Proj01].[StudentCourse] VALUES (103, 'bgodmar0@sakura.ne.jp', 9.99,
'2014/07/11');
INSERT INTO [Proj01].[StudentCourse] VALUES (101, 'bnicolls1@pbs.org', 24.99,
'2021/06/13');
INSERT INTO [Proj01].[StudentCourse] VALUES (104, 'obausor3@umn.edu', 44.99,
'2022/12/12');
INSERT INTO [Proj01].[StudentCourse] VALUES (102, 'hshaudfurth4@creativecommons.org',
44.99, '2016/07/07');

INSERT INTO [Proj01].[Certification] VALUES (1001, '2020/01/01', 'Data Science
Expert');
INSERT INTO [Proj01].[Certification] VALUES (1002, '2021/09/16', 'Beginner Web
Developer');

INSERT INTO [Proj01].[CertCourse] VALUES (1001, 103);
INSERT INTO [Proj01].[CertCourse] VALUES (1001, 104);
INSERT INTO [Proj01].[CertCourse] VALUES (1002, 101);

--Demonstration of Features

--1. Search for course by category
SELECT *
FROM [Proj01].[Course]
WHERE CourseID IN (SELECT COURSEID
FROM [Proj01].[CourseCategory]
WHERE Category = 'Statistics')
ORDER BY CourseName ASC


--2. Add a new student to the database

--See the number of students in each course:
```

```
SELECT c.CourseID, NumOfStudents
FROM [Proj01].[Course] c
JOIN (SELECT CourseID, count(*) AS NumOfStudents
FROM [Proj01].[StudentCourse]
GROUP BY CourseID) sc ON c.CourseID = sc.CourseID
ORDER BY CourseID ASC



--Create a new student and enroll him in a course:
INSERT INTO [Proj01].[Account] VALUES ('jsmith@hotmail.com', 'John', 'Smith',
'password1234');
INSERT INTO [Proj01].[Student] VALUES ('jsmith@hotmail.com');
INSERT INTO [Proj01].[StudentCourse] VALUES (101, 'jsmith@hotmail.com', 99.99,
'2023/10/23');

--Make sure the number of students updated:
SELECT c.CourseID, NumOfStudents
FROM [Proj01].[Course] c
JOIN (SELECT CourseID, count(*) AS NumOfStudents
FROM [Proj01].[StudentCourse]
GROUP BY CourseID) sc ON c.CourseID = sc.CourseID
ORDER BY CourseID ASC;

--3. Calculate revenue generated by instructor
SELECT FirstName, LastName, TotalRevenue
FROM [Proj01].[Account] a
JOIN (SELECT EmailAddress, SUM(Revenue) AS TotalRevenue
FROM [Proj01].[InstructorCourse] ic
JOIN (SELECT CourseID, SUM(PricePaid) AS Revenue
FROM [Proj01].[StudentCourse]
GROUP BY CourseID) c ON ic.CourseID = c.CourseID
GROUP BY EmailAddress) r ON a.EmailAddress = r.EmailAddress
ORDER BY LastName ASC, FirstName ASC
```

Time Tracking

Date	Team Member	Hours Spent	Description	Comments
9/16/2023	All	1	first meeting	
9/17/2023	All	0.5	met and emailed title and details	
9/18/2023	Anish	1	Created User requirement	
10/5/2023	Rylan	1	rough draft of ER	
10/6/2023	All	1	progress meeting	
10/15/2023	Vedika, Ketki	2	Renewed ER draft	Based on discussions and suggestions by Prof. Boyle in the class
10/18/2023	Anish, Ketki, Vedika	0.5	Meeting with the Professor	Meeting with Professor to discuss the Conceptual Model
10/18/2023	All	0.5	Team Meeting	Discussed changes in ER Diagram
10/18/2023	Anish	0.5	Changes in user requirement	
10/18/2023	Ketki	1.5	ER diagram- UML	
10/18/2023	Vedika	0.75	Logical Model	Created the Logical Model and updated it based on discussions
10/18/2023	Rylan	3	SQL Code	Drafted code to create database
10/18/2023	Rylan	2	SQL Code	
10/19/2023	Ketki	0.5	Conceptual Modeling description	
10/19/2023	Vedika	0.5	Logical Modeling description	
10/20/2023	Anish	2.25	Report generation	
10/20/2023	Vedika	0.75	Report generation	
10/22/2023	Rylan	2	SQL Code	Further refinement of code and demonstration of features
10/22/2023	Anish, Ketki, Vedika	2	Worked on the Report and Presentation	