QS STARS RATED FOR EXCELLENCE 2012

FPT Education

ĐẠI HỌC FPT

DBI Assignment

Student: Cao Duc Duy

Student’s Roll Number: HE176716

Class: AI1706

Subject: Introduction to Databases (DBI202)

Instructor: Le Phuong Chi

Table of Contents

A. **Background:**

I. General:

1. A brief description of the database including any assumptions made during the design.
2. An ERD (Entity Relationship Diagram) that fully describes the database.
3. The relational schema derived from the ERD that is at least in 3NF.
4. The set of database statements used to create the tables used in database.
5. 10 queries that demonstrate the usefulness of the database.
6. The trigger, store procedure, and the index should be added.

II. Specification:

1. This is an individual assignment which show steps of designing and developing a database for online sharing learning resources.
2. The database must consist of at least six tables that have been populated with data. The database is to support queries that would typically be submitted to the system for the topical area have chosen.
3. Populate the database with a small amount of data. The data should be meaningful but does not need to be extensive.
4. Using appropriate SQL commands create a set of database tables in MS SQL Server 2008+.
5. Database contains one view, one trigger, on store procedure and an index.
6. Create 10 sample queries that demonstrate the expressiveness of database system. Queries demonstrates different aspects of the system.

III. Destination:

Design and develop a database system for **Udemy online education.**

IV. Tools

1. Microsoft SQL Server Management Studio 19
2. Microsoft Word 2019 Professional
3. https://www.draw.io
4. https://www.generatedata.com
5. https://www.canva.com

B. Preview:

I. Introduce Udemy:

Udemy is a platform that allows instructors to build online courses on their preferred topics. Using Udemy's course development tools, instructors can upload videos, source code for developers, PowerPoint presentations, PDFs, audio, ZIP files and any other content that learners might find helpful.

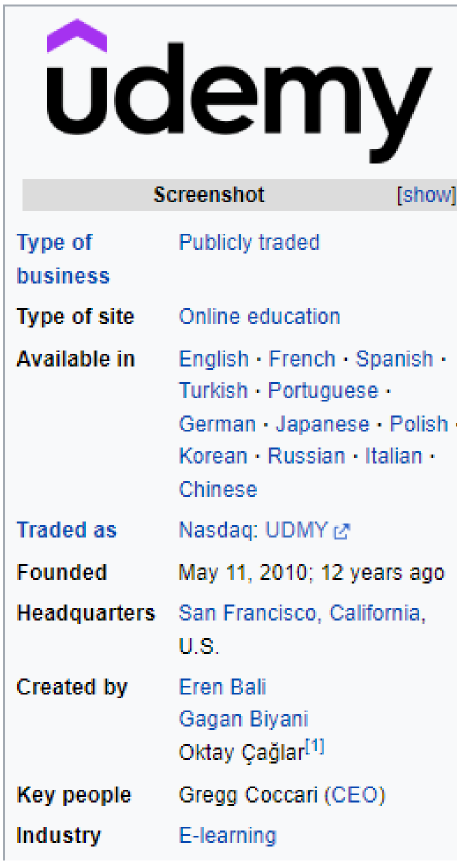
Instructors can also engage and interact with users via online discussion boards.

Courses are offered across a wide breadth of categories, including business and entrepreneurship, academics, the arts, health and fitness, language, music, and technology. Most classes are in practical subjects such as AWS and Azure training, Excel software or using an iPhone camera. Udemy also offers Udemy Business (formerly Udemy for Business), enabling businesses access to a targeted suite of over 7,000 training courses on topics from digital marketing tactics to office productivity, design, management, programming, and more. With Udemy Business, organizations can also create custom learning portals for corporate training.

Courses on Udemy can be paid or free, depending on the instructor.

In April 2013, Udemy offered an app for Apple iOS, allowing students to take classes directly from iPhones; The Android version was launched in January 2014. As of January 2014, the iOS app had been downloaded over 1 million times, and 20 percent of Udemy users access their courses via mobile.

Udemy, Inc.



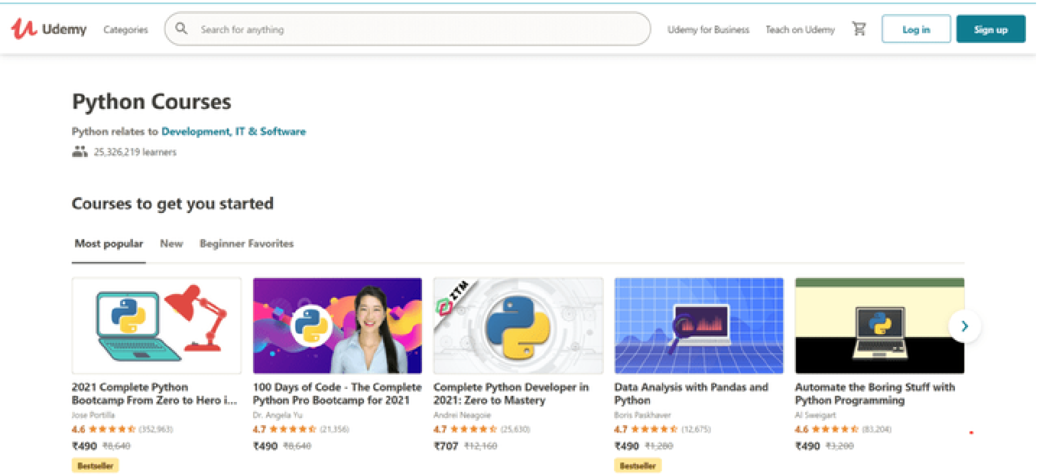
Type of business	Publicly traded
Type of site	Online education
Available in	English · French · Spanish · Turkish · Portuguese · German · Japanese · Polish · Korean · Russian · Italian · Chinese
Traded as	Nasdaq: UDMY
Founded	May 11, 2010; 12 years ago
Headquarters	San Francisco, California, U.S.
Created by	Eren Bali Gagan Biyani Oktay Çağlar[1]
Key people	Gregg Coccari (CEO)
Industry	E-learning

(Source: https://en.wikipedia.org/wiki/Udemy)

II. More about Udemy:

Udemy is a massive open online course provider, and its learning experience arranges coursework into a series of modules and lessons that can include videos, text notes and assessment tests. Udemy's video player has functional features like closed captioning and note-taking functions.

Massive open online course (MOOC):
Udemy is part of the MOOC movement available outside the traditional university system, and has been noted for the variety of courses offered. Udemy as well as others in the MOOC offers a fantastic option for students who want to delve into complicated university level courses while studying on their own time. It often provides the same skills but at a fraction of the cost.



Python Courses

Python relates to Development, IT & Software

25,326,219 learners

Courses to get you started

2021 Complete Python Bootcamp From Zero to Hero i...
Jose Portilla
4.6 ★★★★☆ (352,963)
₹490 ~~₹8,640~~
Bestseller

100 Days of Code - The Complete Python Pro Bootcamp for 2021
Dr. Angela Yu
4.7 ★★★★☆ (21,356)
₹490 ~~₹8,640~~

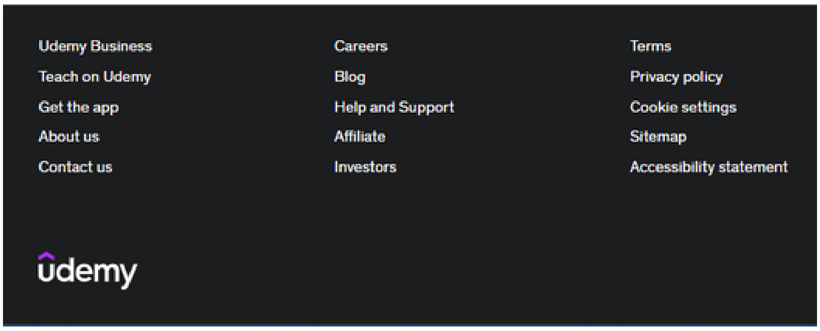
Complete Python Developer in 2021: Zero to Mastery
Andrei Neagoie
4.7 ★★★★☆ (25,630)
₹707 ~~₹12,160~~

Data Analysis with Pandas and Python
Boris Paskhaver
4.7 ★★★★☆ (12,675)
₹490 ~~₹1,280~~
Bestseller

Automate the Boring Stuff with Python Programming
Al Sweigart
4.6 ★★★★☆ (83,204)
₹490 ~~₹3,200~~

Udemy Main Features and Information:

- Categories
- Specializations (Sub-topics)
- Courses
- Lectures
- Instructors
- Students (Learners)
- Certificates
- Forum
- Terms - Help & Support



Udemy Business
Teach on Udemy
Get the app
About us
Contact us

Careers
Blog
Help and Support
Affiliate
Investors

Terms
Privacy policy
Cookie settings
Sitemap
Accessibility statement

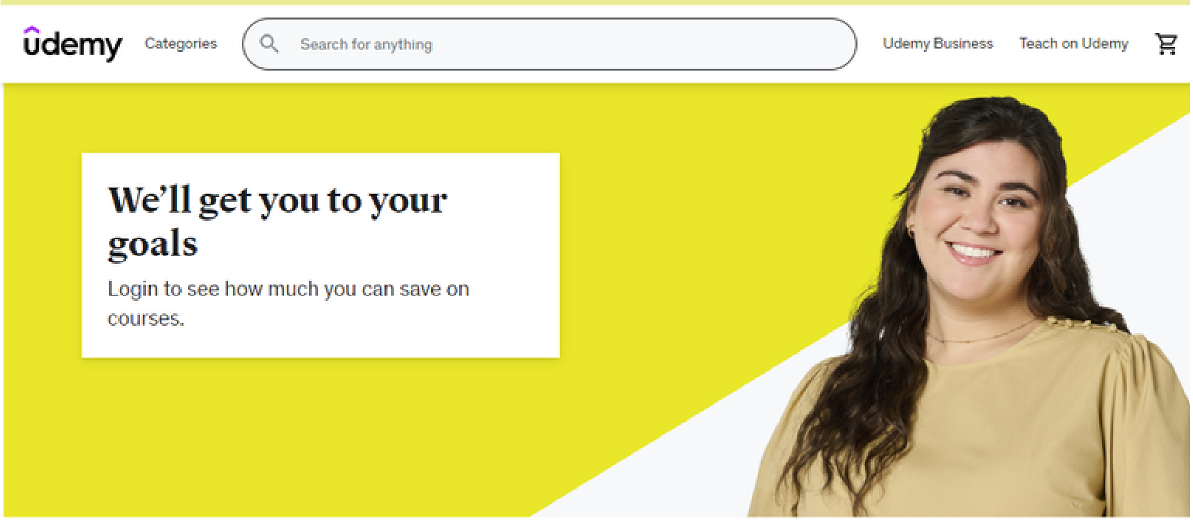
(Source: https://www.udemy.com/)

There are a lot of sections and side information, but the main purpose is to design a concise database to operate the website and the teaching and learning process of Udemy. That's why we will only focus on the main items and ignore the unnecessary ones.

1, Categories

Udemy has user-friendly interface by dividing courses into detailed categories based on the skills users want to learn and master. Instead of immediately in the eye are the courses with confusing names.

When users feel interested and click on each category, they will go deeper into each content one by one.



We'll get you to your goals

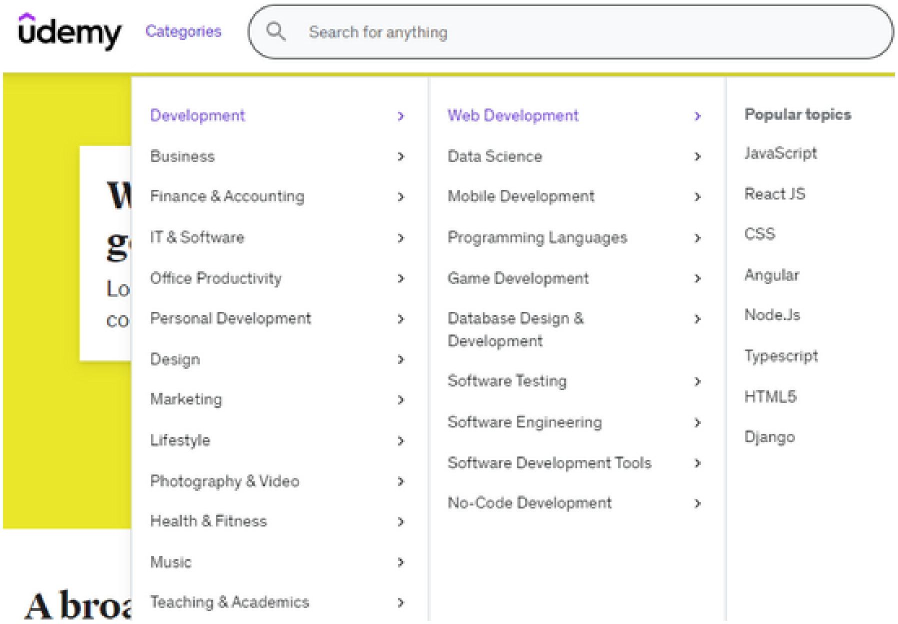
Login to see how much you can save on courses.

We did our research to check how many online topics are posted on Udemy overall. From our research, Udemy offers courses on almost any topic we can think of.

Udemy divides the teaching knowledge into 13 main categories. Ranging from specialized knowledge to life skills. Only from the category sections can we see the skills, knowledge and courses develop that skill and also including information about topics that people are interested in.

Udemy Main Categories:

- Development
- Business
- Finance & Accounting
- IT & Software
- Office Productivity
- Personal Development
- Design
- Marketing
- Lifestyle
- Photography & Video
- Health & Fitness
- Music
- Teaching & Academics

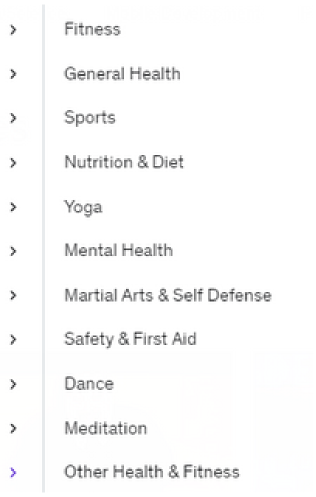


(Source: https://www.udemy.com/)

2, Specializations

Specializations are identified through the names of each category and skill (Ex: Data Science in Development).

Or we can call specializations a sub-branch of the category. For example, when you want to improve your health (Health and Fitness), you can choose to do that by learning to dance (Dance) or meditate (Mental Health).

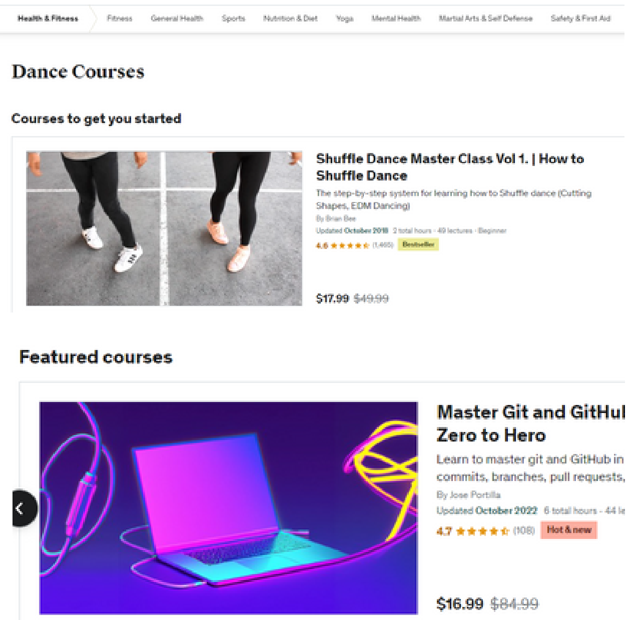


› Fitness
› General Health
› Sports
› Nutrition & Diet
› Yoga
› Mental Health
› Martial Arts & Self Defense
› Safety & First Aid
› Dance
› Meditation
› Other Health & Fitness

When clicking on the Dance specialization. The courses will be displayed, moreover the web will give us the useful information and advices.

- Featured Course:

Shows basic information of recommended courses such as name, lecturers, updated date, duration, number of lectures, difficulty, rating such as number of stars, price or even "Best seller" information.



Health & Fitness › Fitness General Health Sports Nutrition & Diet Yoga Mental Health Martial Arts & Self Defense Safety & First Aid

Dance Courses

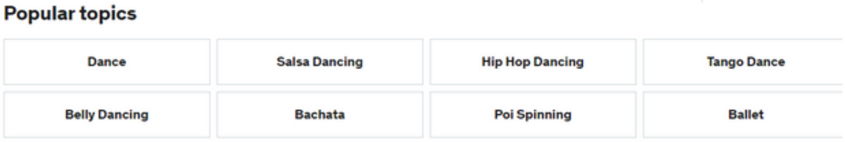
Courses to get you started

Shuffle Dance Master Class Vol 1. | How to Shuffle Dance
The step-by-step system for learning how to Shuffle dance (Cutting Shapes, EDM Dancing)
By Brian Bee
Updated October 2018 2 total hours · 49 lectures · Beginner
4.6 ★★★★★ (1,465) Bestseller
$17.99 ~~$49.99~~

Featured courses

Master Git and GitHul
Zero to Hero
Learn to master git and GitHub in
commits, branches, pull requests,
By Jose Portilla
Updated October 2022 6 total hours · 44 le
4.7 ★★★★★ (108) Hot & new
$16.99 ~~$84.99~~

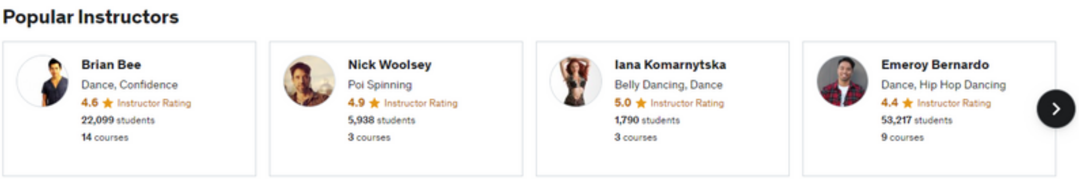
- Popular topics in specialization:



Popular topics

Dance	Salsa Dancing	Hip Hop Dancing	Tango Dance
Belly Dancing	Bachata	Poi Spinning	Ballet

- Popular Instructors in specialization:

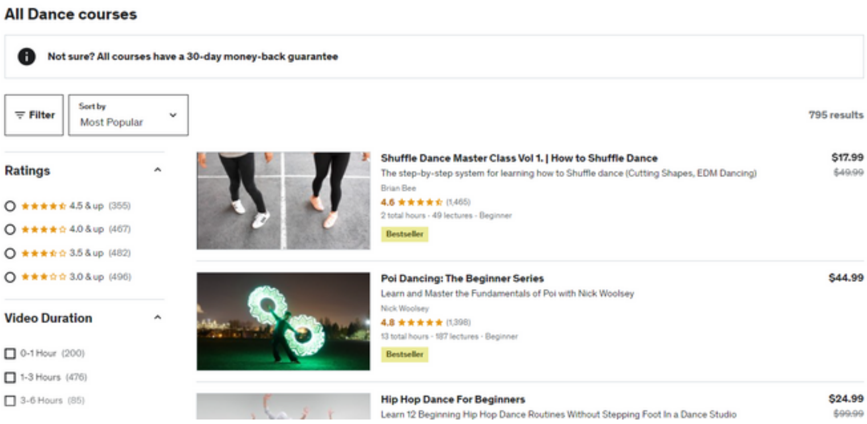


Popular Instructors

Brian Bee Dance, Confidence 4.6 ★ Instructor Rating 22,099 students 14 courses	Nick Woolsey Poi Spinning 4.9 ★ Instructor Rating 5,938 students 3 courses	Iana Komarnytska Belly Dancing, Dance 5.0 ★ Instructor Rating 1,790 students 3 courses	Emeroy Bernardo Dance, Hip Hop Dancing 4.4 ★ Instructor Rating 53,217 students 9 courses

- All courses in specialization:

Shows the basic information (course name, lecturers, updated date, duration, number of lectures, difficulty, price) of all the courses of that specialization and has a convenient filtering mechanism.



All Dance courses

Not sure? All courses have a 30-day money-back guarantee

Filter | Sort by Most Popular | 795 results

Ratings
○ ★★★★☆ 4.5 & up (355)
○ ★★★★☆ 4.0 & up (467)
○ ★★★☆☆ 3.5 & up (482)
○ ★★★☆☆ 3.0 & up (496)

Video Duration
□ 0-1 Hour (200)
□ 1-3 Hours (476)
□ 3-6 Hours (85)

Shuffle Dance Master Class Vol 1. | How to Shuffle Dance — $17.99 ~~$49.99~~
The step-by-step system for learning how to Shuffle dance (Cutting Shapes, EDM Dancing)
Brian Bee
4.6 ★★★★★ (1,465)
2 total hours · 49 lectures · Beginner
Bestseller

Poi Dancing: The Beginner Series — $44.99
Learn and Master the Fundamentals of Poi with Nick Woolsey
Nick Woolsey
4.8 ★★★★★ (1,398)
13 total hours · 187 lectures · Beginner
Bestseller

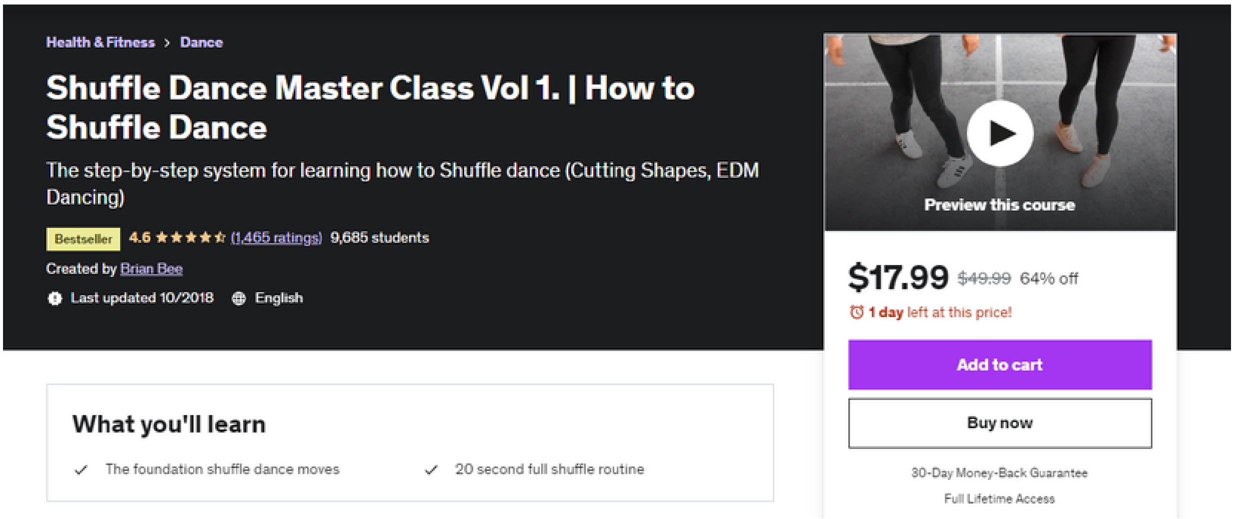
Hip Hop Dance For Beginners — $24.99 ~~$99.99~~
Learn 12 Beginning Hip Hop Dance Routines Without Stepping Foot In a Dance Studio

(Source: https://www.udemy.com/)

3, Courses

Courses are simply lectures and communications aimed at helping to specialize learners, helping them master a certain skill previously displayed in Categories and Specializations.

The information of a course includes: Course name, basic content, updated date, language, number of stars, number of subscribers, price, discount, requirements and description.



Health & Fitness > Dance

Shuffle Dance Master Class Vol 1. | How to Shuffle Dance

The step-by-step system for learning how to Shuffle dance (Cutting Shapes, EDM Dancing)

Bestseller 4.6 ★★★★☆ (1,465 ratings) 9,685 students

Created by Brian Bee

Last updated 10/2018 English

What you'll learn

✓ The foundation shuffle dance moves

✓ 20 second full shuffle routine

$17.99 $49.99 64% off

1 day left at this price!

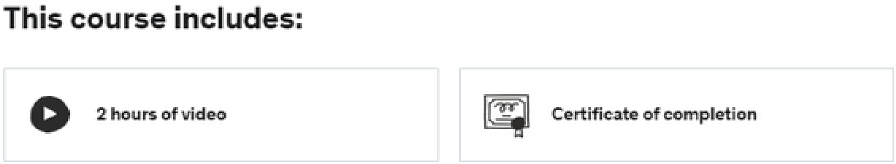
Add to cart

Buy now

30-Day Money-Back Guarantee

Full Lifetime Access

- Basic contents:

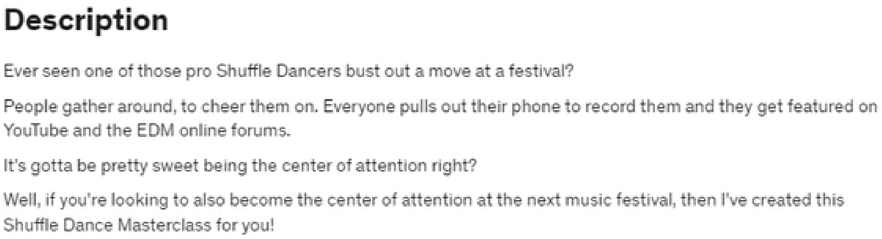


This course includes:

2 hours of video

Certificate of completion

- Description:



Description

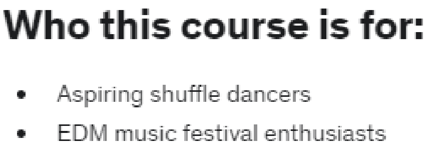
Ever seen one of those pro Shuffle Dancers bust out a move at a festival?

People gather around, to cheer them on. Everyone pulls out their phone to record them and they get featured on YouTube and the EDM online forums.

It's gotta be pretty sweet being the center of attention right?

Well, if you're looking to also become the center of attention at the next music festival, then I've created this Shuffle Dance Masterclass for you!

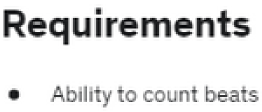
- Target learners:



Who this course is for:

- Aspiring shuffle dancers
- EDM music festival enthusiasts

- Requirements:

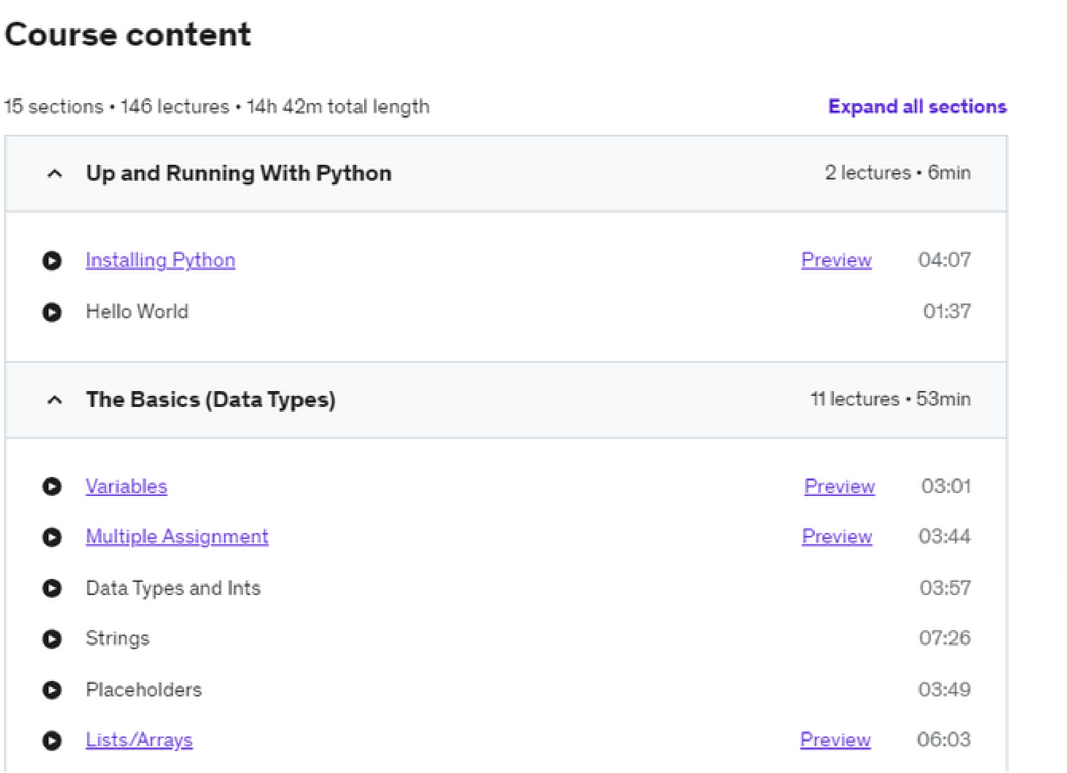


Requirements

- Ability to count beats

(Source: https://www.udemy.com/)

After clicking learn, the specific content of the course will be displayed, including modules, sections, videos, time each video. When clicking on the video or reading, we have started the lesson of that section.



Course content

15 sections • 146 lectures • 14h 42m total length

Expand all sections

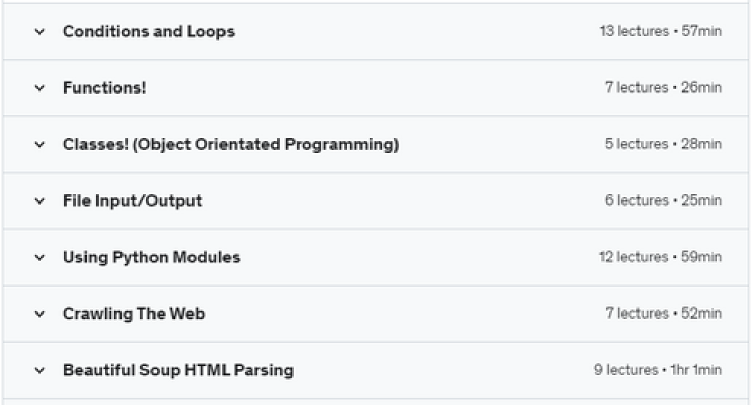
Up and Running With Python 2 lectures • 6min

Installing Python	Preview	04:07
Hello World		01:37

The Basics (Data Types) 11 lectures • 53min

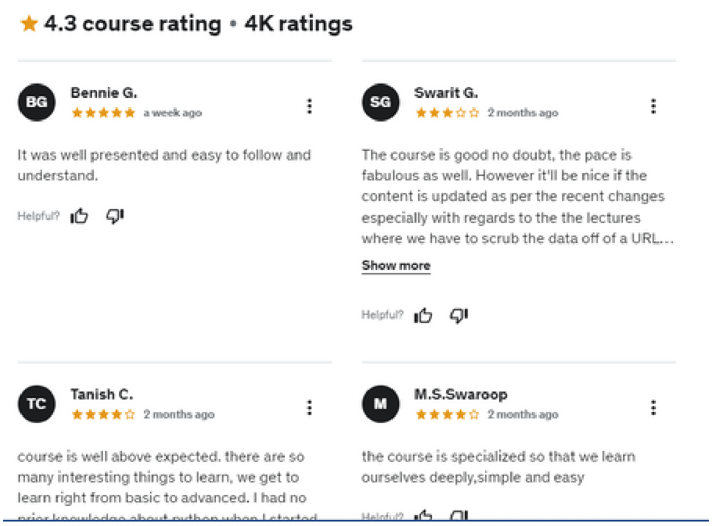
Variables	Preview	03:01
Multiple Assignment	Preview	03:44
Data Types and Ints		03:57
Strings		07:26
Placeholders		03:49
Lists/Arrays	Preview	06:03

- Sections:



Conditions and Loops	13 lectures • 57min
Functions!	7 lectures • 26min
Classes! (Object Orientated Programming)	5 lectures • 28min
File Input/Output	6 lectures • 25min
Using Python Modules	12 lectures • 59min
Crawling The Web	7 lectures • 52min
Beautiful Soup HTML Parsing	9 lectures • 1hr 1min

- Courses Ratings:



★ 4.3 course rating • 4K ratings

BG Bennie G. ★★★★★ a week ago

It was well presented and easy to follow and understand.

Helpful?

SG Swarit G. ★★★☆☆ 2 months ago

The course is good no doubt, the pace is fabulous as well. However it'll be nice if the content is updated as per the recent changes especially with regards to the the lectures where we have to scrub the data off of a URL...

Show more

Helpful?

TC Tanish C. ★★★★☆ 2 months ago

course is well above expected. there are so many interesting things to learn, we get to learn right from basic to advanced. I had no prior knowledge about python when I started

M M.S.Swaroop ★★★★☆ 2 months ago

the course is specialized so that we learn ourselves deeply,simple and easy

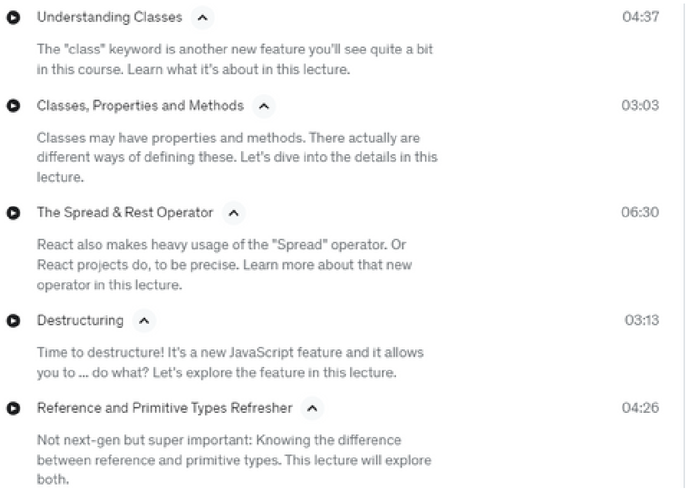
Helpful?

(Source: https://www.udemy.com/)

4, Lectures:

The lectures will be in each course, it can be divided into chapters, or the course has only 1 lecture. It shows the number, lecture title, type, subtitles, duration (video) or estimated reading time and also a small description summarizing the content of the lecture. Lectures will often be presented through videos or articles, readings, quizzes. Students can also download materials available on the website.

- Lectures in Course:



Understanding Classes 04:37

The "class" keyword is another new feature you'll see quite a bit in this course. Learn what it's about in this lecture.

Classes, Properties and Methods 03:03

Classes may have properties and methods. There actually are different ways of defining these. Let's dive into the details in this lecture.

The Spread & Rest Operator 06:30

React also makes heavy usage of the "Spread" operator. Or React projects do, to be precise. Learn more about that new operator in this lecture.

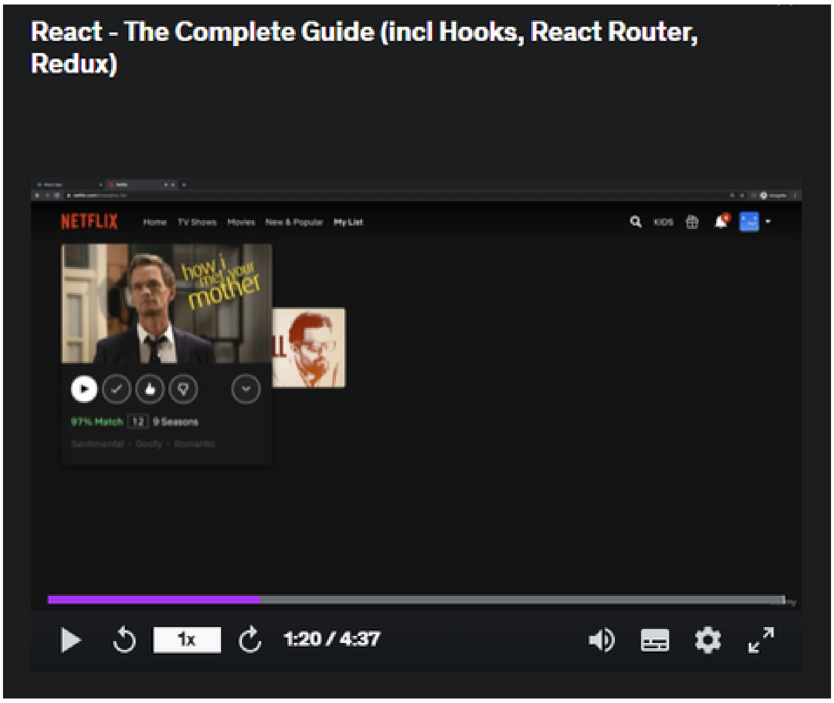
Destructuring 03:13

Time to destructure! It's a new JavaScript feature and it allows you to ... do what? Let's explore the feature in this lecture.

Reference and Primitive Types Refresher 04:26

Not next-gen but super important: Knowing the difference between reference and primitive types. This lecture will explore both.

- Inside each lecture:

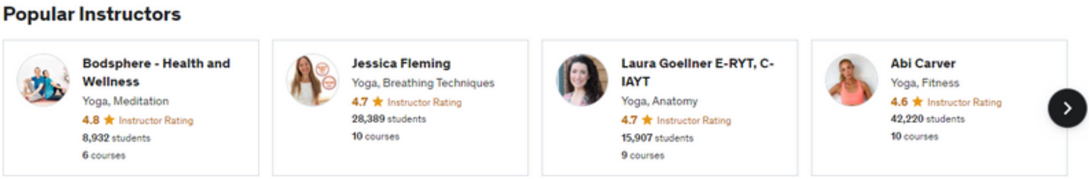


React - The Complete Guide (incl Hooks, React Router, Redux)

(Source: https://www.udemy.com/)

5, Instructors

Each course will have one or more instructors to guide and teach through lessons. There are many instructors on Udemy, they are ranked and rated according to the number of stars and the number of people enrolled.



Popular Instructors

Bodsphere - Health and Wellness
Yoga, Meditation
4.8 ★ Instructor Rating
8,932 students
6 courses

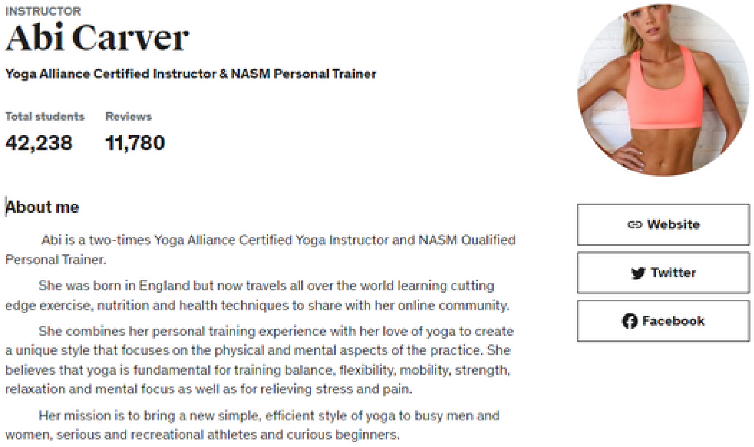
Jessica Fleming
Yoga, Breathing Techniques
4.7 ★ Instructor Rating
28,389 students
10 courses

Laura Goellner E-RYT, C-IAYT
Yoga, Anatomy
4.7 ★ Instructor Rating
15,907 students
9 courses

Abi Carver
Yoga, Fitness
4.6 ★ Instructor Rating
42,220 students
10 courses

- Instructor's Bio:

Each instructor will have a separate profile that includes their basic information such as their major, name, date of birth, gender, country, number of students, description and links to some social networking sites.



INSTRUCTOR

Abi Carver

Yoga Alliance Certified Instructor & NASM Personal Trainer

Total students	Reviews
42,238	11,780

About me

Abi is a two-times Yoga Alliance Certified Yoga Instructor and NASM Qualified Personal Trainer.

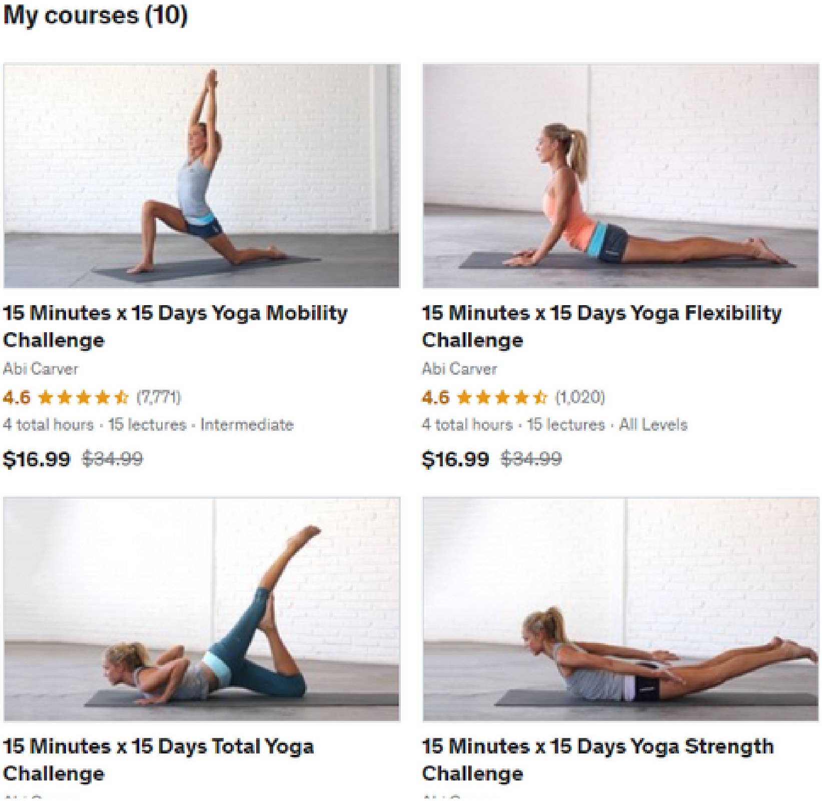
She was born in England but now travels all over the world learning cutting edge exercise, nutrition and health techniques to share with her online community.

She combines her personal training experience with her love of yoga to create a unique style that focuses on the physical and mental aspects of the practice. She believes that yoga is fundamental for training balance, flexibility, mobility, strength, relaxation and mental focus as well as for relieving stress and pain.

Her mission is to bring a new simple, efficient style of yoga to busy men and women, serious and recreational athletes and curious beginners.

- Courses taught by them:

Including information that introduces learners to their course.



My courses (10)

15 Minutes x 15 Days Yoga Mobility Challenge
Abi Carver
4.6 ★★★★☆ (7,771)
4 total hours · 15 lectures · Intermediate
$16.99 ~~$34.99~~

15 Minutes x 15 Days Yoga Flexibility Challenge
Abi Carver
4.6 ★★★★☆ (1,020)
4 total hours · 15 lectures · All Levels
$16.99 ~~$34.99~~

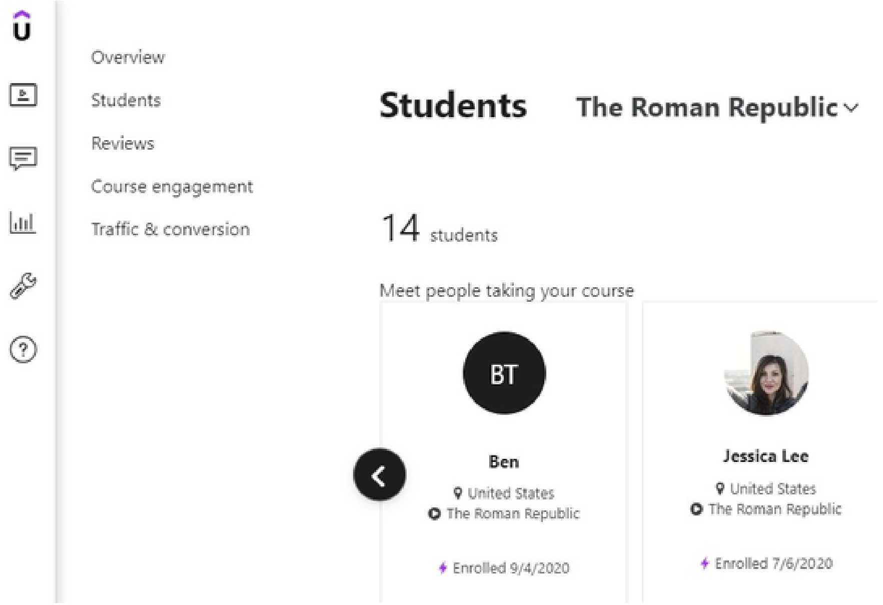
15 Minutes x 15 Days Total Yoga Challenge

15 Minutes x 15 Days Yoga Strength Challenge

(Source: https://www.udemy.com/)

6, Students

Students are the primary audience of the site, who participate in courses, services, or help from the site and instructors. As customers, they sign up for or purchase courses from instructors with the site acting as an intermediary.



Students The Roman Republic

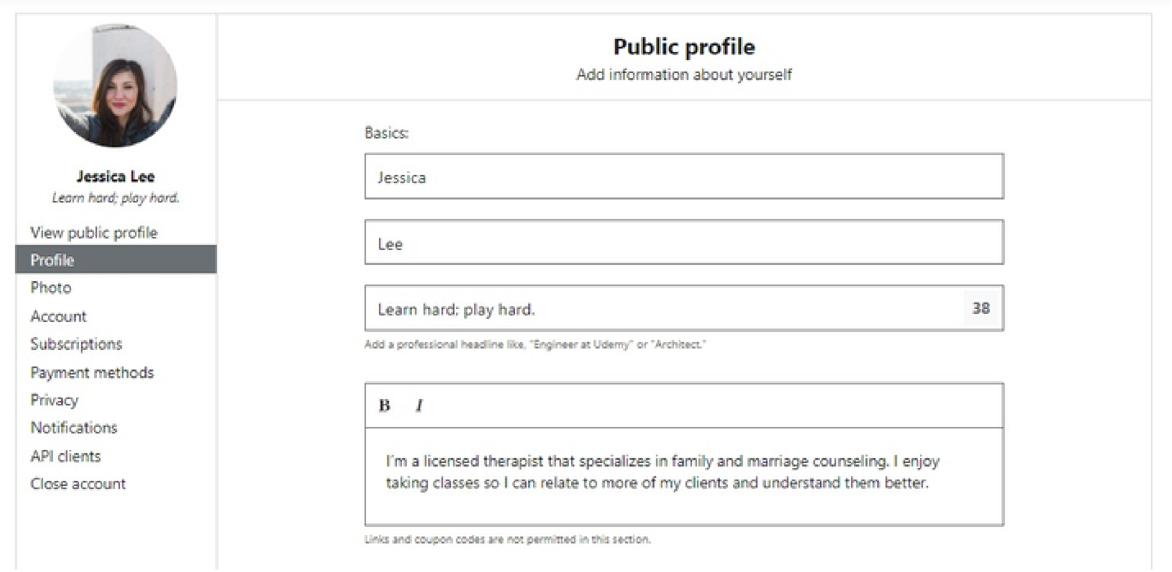
14 students

Meet people taking your course

BT
Ben
United States
The Roman Republic
Enrolled 9/4/2020

Jessica Lee
United States
The Roman Republic
Enrolled 7/6/2020

Similar to instructors, students also have their own profile, including name, date of birth, gender, country descriptions,.... Because they attend the course as a customer, they also have information such as website account, bank account, payments information (billing date, ...) and access to a specific course or their own discount.


Jessica Lee
Learn hard; play hard.

Public profile
Add information about yourself

Basics:

Jessica

Lee

Learn hard; play hard. 38

Add a professional headline like, "Engineer at Udemy" or "Architect."

B *I*

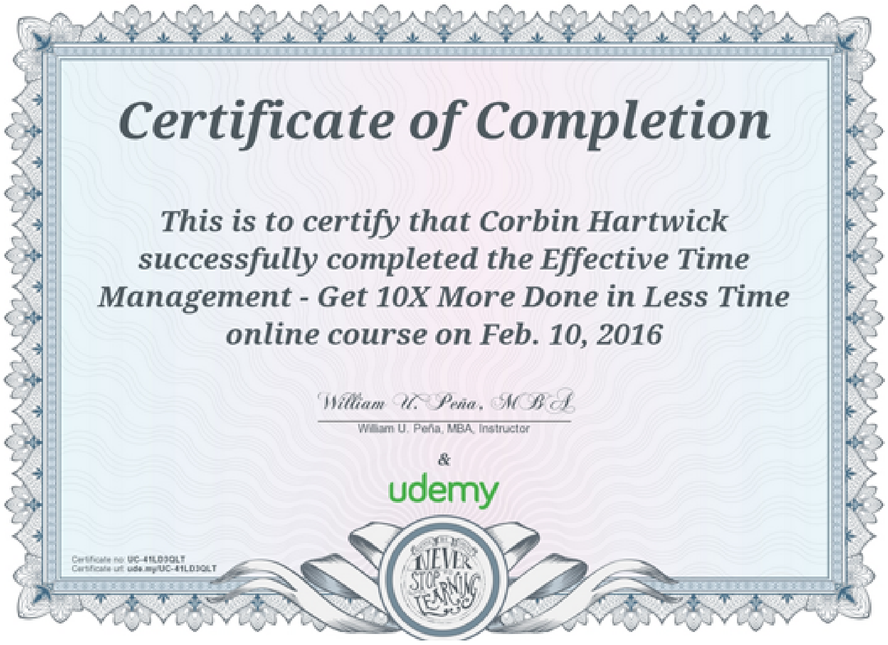
I'm a licensed therapist that specializes in family and marriage counseling. I enjoy taking classes so I can relate to more of my clients and understand them better.

Links and coupon codes are not permitted in this section.

(Source: https://www.udemy.com/)

7, Certificates:

In the certificate, the necessary information is shown such as the student's name, the course name, the name and signature of the instructor, the date of completion.



Certificate of Completion

This is to certify that Corbin Hartwick successfully completed the Effective Time Management - Get 10X More Done in Less Time online course on Feb. 10, 2016

William U. Peña, MBA

William U. Peña, MBA, Instructor

&

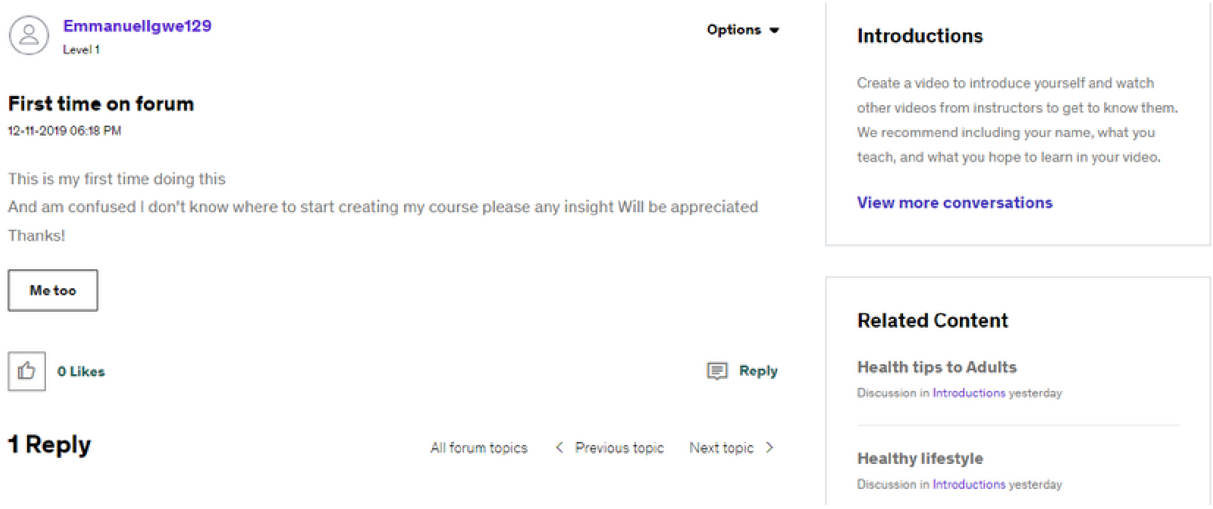
udemy

Certificate no: UC-41LD3QLT
Certificate url: ude.my/UC-41LD3QLT

NEVER STOP LEARNING

8, Forums:

Forum is a place for students to exchange with each other, teachers answer questions, interact directly with their own students. Some information in the forum such as posts, topics, questions, answers, likes, related content



Emmanuellgwe129
Level 1

First time on forum
12-11-2019 06:18 PM

This is my first time doing this
And am confused I don't know where to start creating my course please any insight Will be appreciated
Thanks!

1 Reply

Introductions

Create a video to introduce yourself and watch other videos from instructors to get to know them. We recommend including your name, what you teach, and what you hope to learn in your video.

View more conversations

Related Content

Health tips to Adults
Discussion in Introductions yesterday

Healthy lifestyle
Discussion in Introductions yesterday

(Source: https://www.udemy.com/)

9, Terms - Help & Support:

- Terms:

Although not directly related to the content, but it is one of the important things to operate a website, manage stably between students and instructors, make policies to keep benefits for all 3 parties. As well as commitments between website owners and customers such as privacy or billing issues, content that violates the policy will also be strictly prohibited.

I wouldn't consider Terms as a relation, but it is an important element of the Udemy site so that I can set up constrants and rules for relations in the database and attributes of the relation.

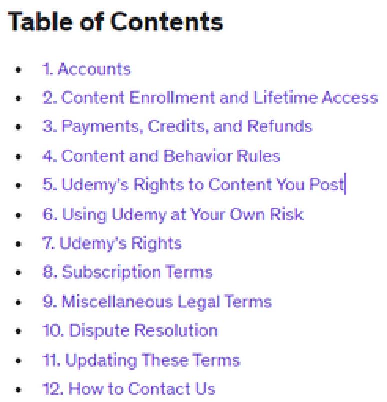
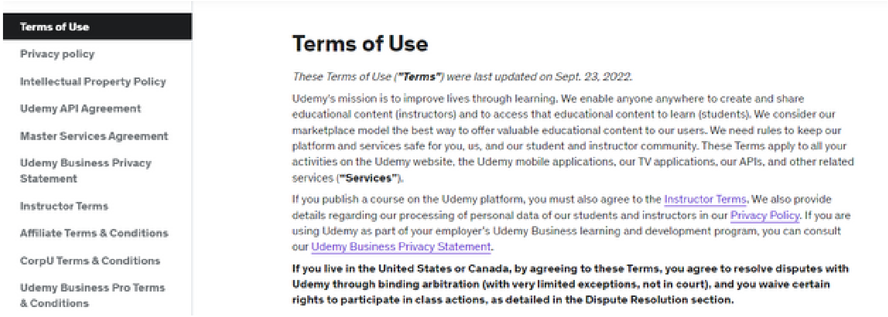


Table of Contents

- 1. Accounts
- 2. Content Enrollment and Lifetime Access
- 3. Payments, Credits, and Refunds
- 4. Content and Behavior Rules
- 5. Udemy's Rights to Content You Post
- 6. Using Udemy at Your Own Risk
- 7. Udemy's Rights
- 8. Subscription Terms
- 9. Miscellaneous Legal Terms
- 10. Dispute Resolution
- 11. Updating These Terms
- 12. How to Contact Us


Terms of Use
Privacy policy
Intellectual Property Policy
Udemy API Agreement
Master Services Agreement
Udemy Business Privacy Statement
Instructor Terms
Affiliate Terms & Conditions
CorpU Terms & Conditions
Udemy Business Pro Terms & Conditions

Terms of Use

*These Terms of Use (**"Terms"**) were last updated on Sept. 23, 2022.*

Udemy's mission is to improve lives through learning. We enable anyone anywhere to create and share educational content (instructors) and to access that educational content to learn (students). We consider our marketplace model the best way to offer valuable educational content to our users. We need rules to keep our platform and services safe for you, us, and our student and instructor community. These Terms apply to all your activities on the Udemy website, the Udemy mobile applications, our TV applications, our APIs, and other related services (**"Services"**).

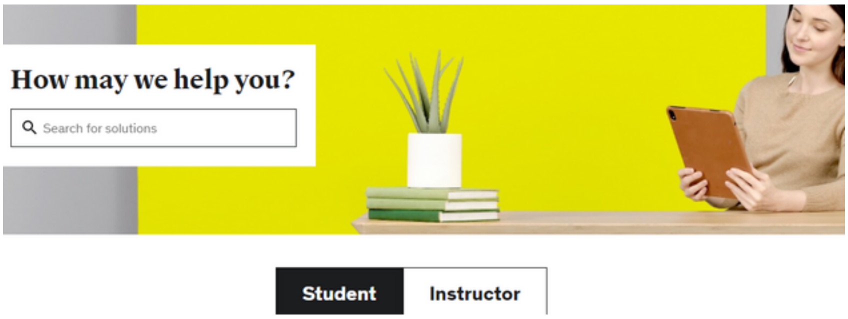
If you publish a course on the Udemy platform, you must also agree to the Instructor Terms. We also provide details regarding our processing of personal data of our students and instructors in our Privacy Policy. If you are using Udemy as part of your employer's Udemy Business learning and development program, you can consult our Udemy Business Privacy Statement.

If you live in the United States or Canada, by agreeing to these Terms, you agree to resolve disputes with Udemy through binding arbitration (with very limited exceptions, not in court), and you waive certain rights to participate in class actions, as detailed in the Dispute Resolution section.

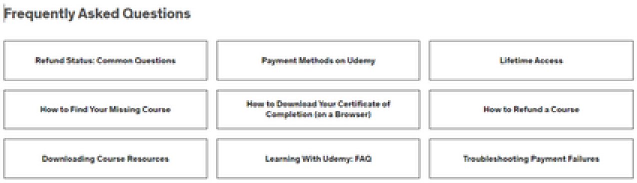
- Help & Support:

This is a place where students or instructors can ask questions to the website administrators about the terms, how to register, how to join the course,.. all the basic but necessary information is displayed on this page.

The information is presented in the form of a solution which we can search (query). In addition, there are Frequent Asked Questions, the answers are also divided into topics.

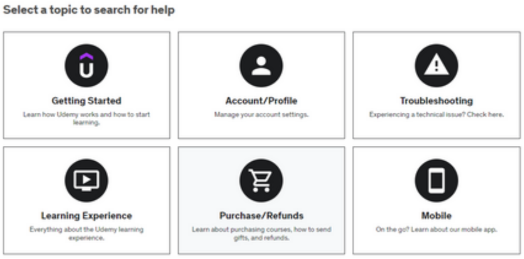


(Source: https://www.udemy.com/)



Frequently Asked Questions

Refund Status: Common Questions	Payment Methods on Udemy	Lifetime Access
How to Find Your Missing Course	How to Download Your Certificate of Completion (on a Browser)	How to Refund a Course
Downloading Course Resources	Learning With Udemy: FAQ	Troubleshooting Payment Failures



Select a topic to search for help

Getting Started Learn how Udemy works and how to start learning.	Account/Profile Manage your account settings.	Troubleshooting Experiencing a technical issue? Check here.
Learning Experience Everything about the Udemy learning experience.	Purchase/Refunds Learn about purchasing courses, how to send gifts, and refunds.	Mobile On the go? Learn about our mobile app.

C. Entity Relationship Diagram (Chen's Notation):

I. Business Rules and Assumption Rules

- Business Rules:

1. One category must have at least one course.
2. One course must be in at least one category.
3. One course must contain at least one lecture.
4. One lecture belongs to only one course. (Udemy's Content rules)
5. One course must have at least one lecture type "Test". (To get certificate)
6. One course must be taught by at least one instructor.
7. The price of a course must be greater than or equal to 0 (paid or free).
8. The duration of a lecture must be more than 0 and there is no time limit.
9. Students and instructors must be at least 18 years of age to create an account on Udemy.
10. Instructors must have a bachelor's degree or higher or a valid certificate in their specialization.
11. Phone number/email used to register account must be unique.
12. The discount must be non-negative and less than or equal the course price.
13. Student cannot transfer, assign, or share subscription with anyone else.
14. Instructors may not grant licenses to their content to students directly, and any such direct license shall be null and void and a violation of these Terms.

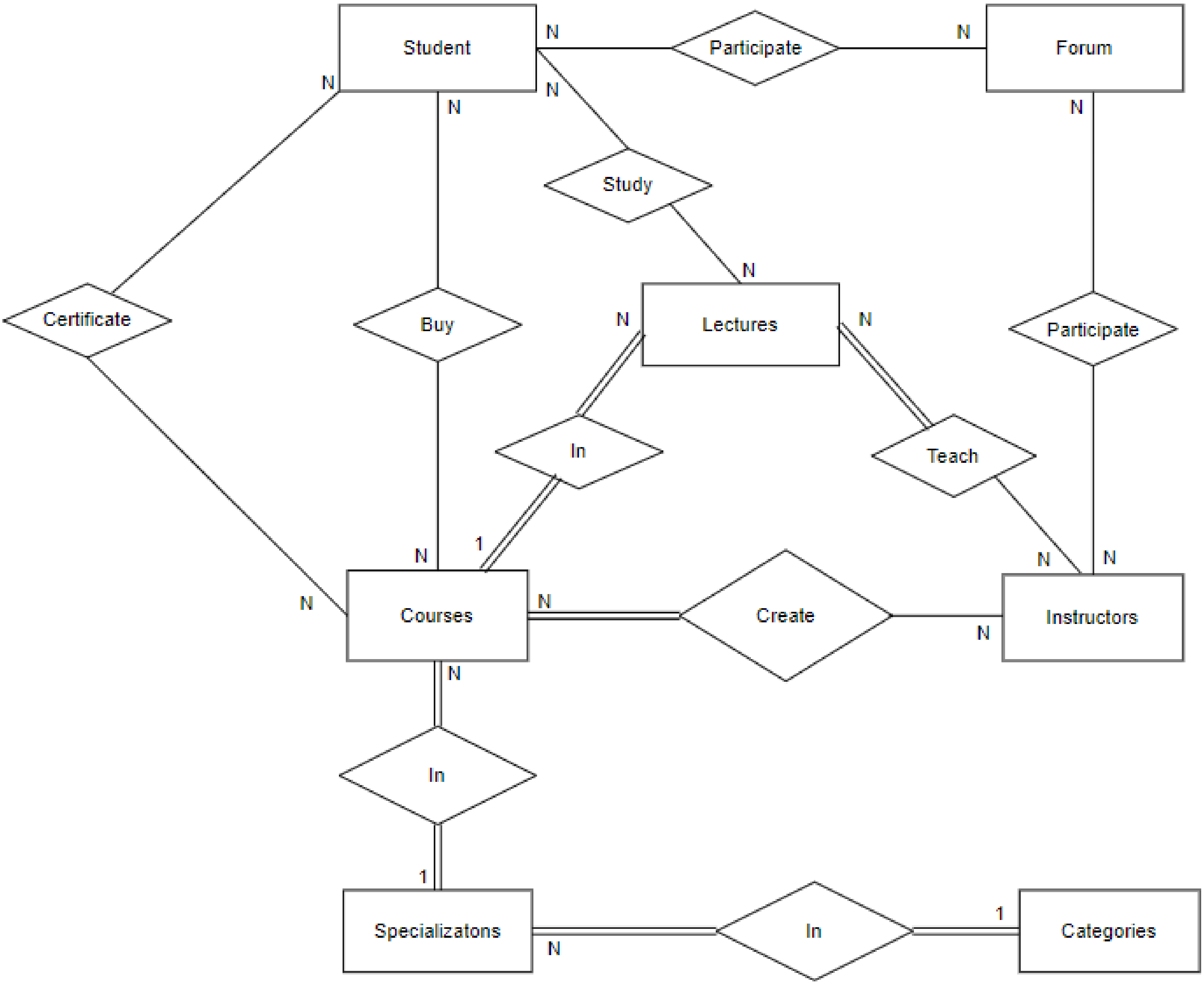
- Assumption Rules:

1. A student taking the course may have limited access time or lifetime access.
2. A student can request a 30-day refund from a course (Leave the course).
3. Billing date must be before enrolling date.
4. After the course expires, students will not be able to participate unless re-enrolling.

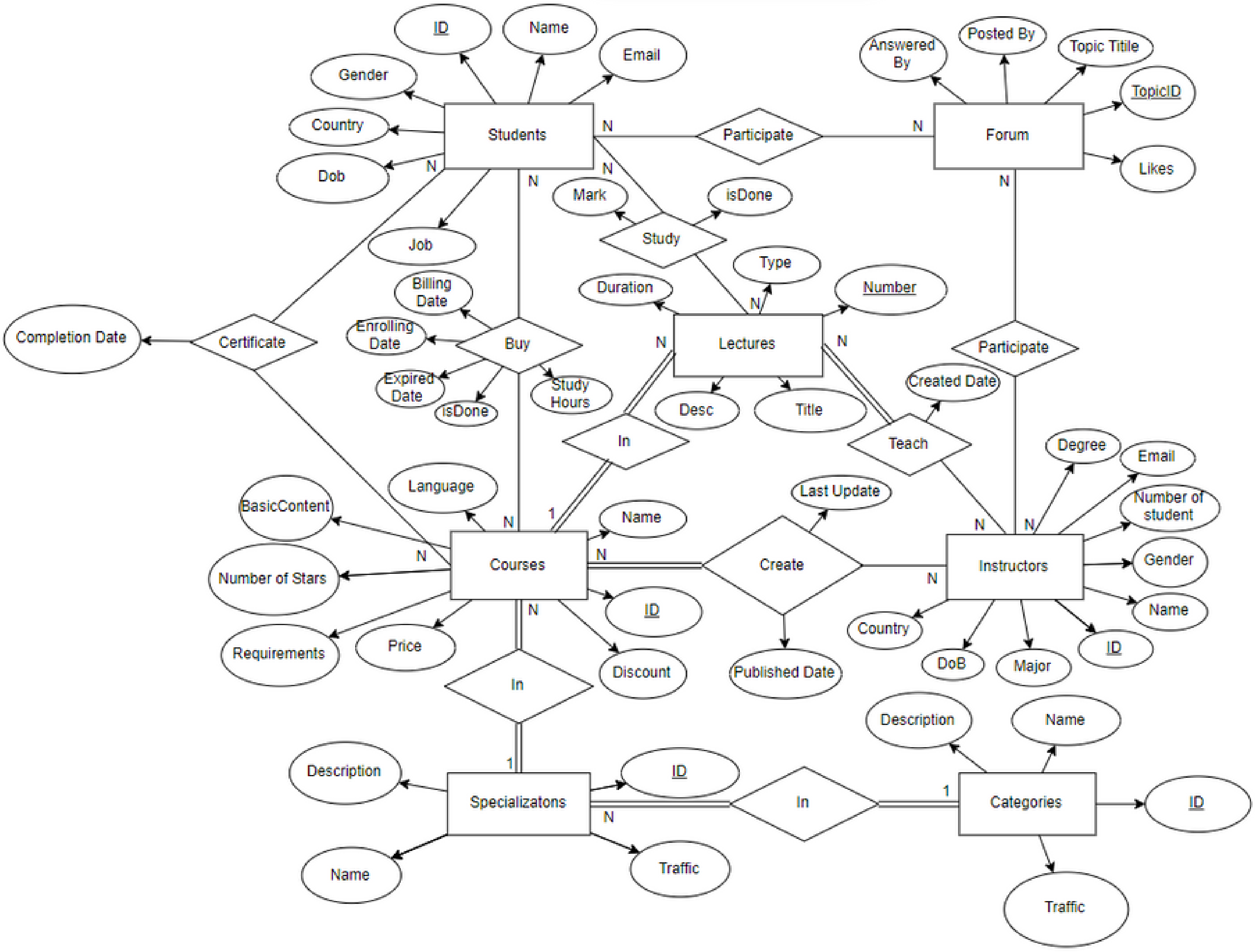
Some business rules are taken directly from Term - Help & Support by Udemy.
https://www.udemy.com/terms/
https://support.udemy.com/hc/en-us

II. Entity Relationship Diagram (ERD)s:

1. Simplified Entity Relationship Diagram

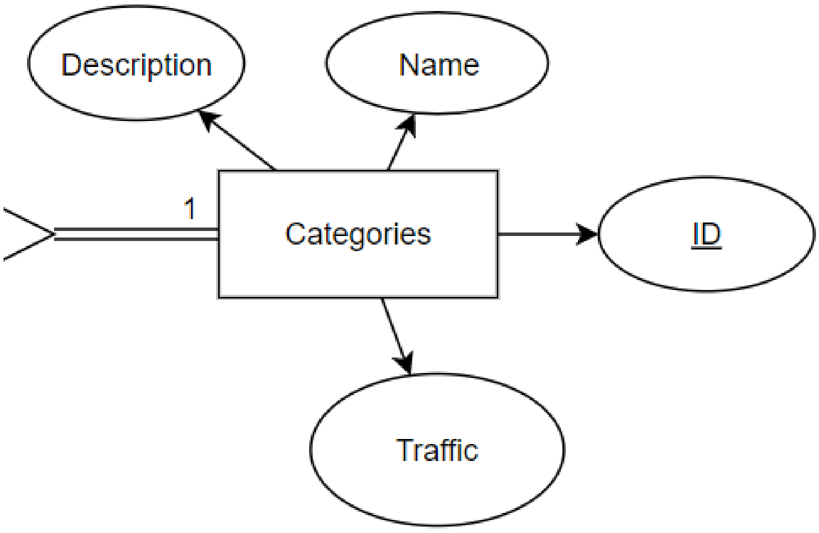


2. Full Entity Relationship Diagram for Database



III. Explanations for entities:

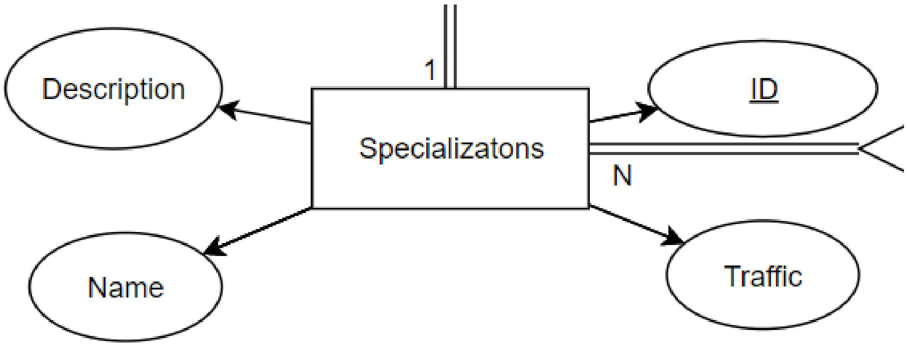
1. Categories



Categories Entity has 4 main properties: ID (CategoryID), Name, Traffic, Description.

- ID: A unique ID to represent the distinction and serve to look up each category.
- Name: Name of the category.
- Traffic: Number of people clicking, following, interested in the category.
- Description: Some background information about category.

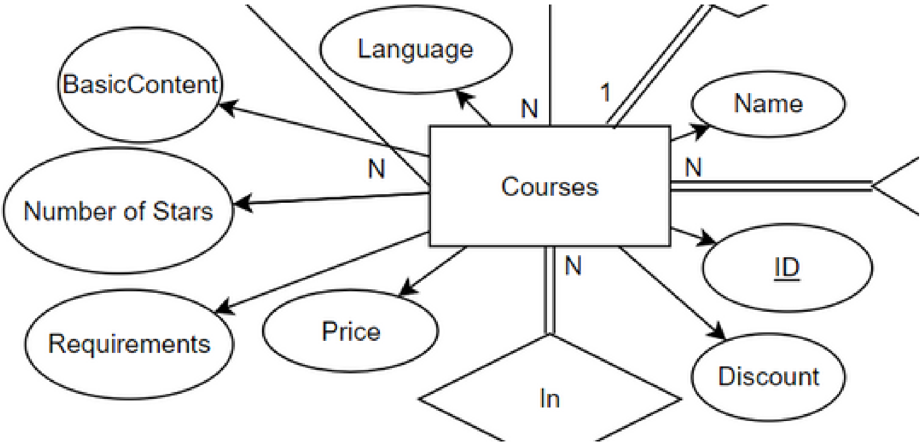
2. Specializations



Specializations Entity has 4 main properties: ID (SpecializationID), Name, Traffic, Description.

- ID: A unique ID to represent the distinction and serve to look up each specialization.
- Name: Name of the category.
- Traffic: Number of people clicking, following, interested in the specialization.
- Description: Some background information about specialization.

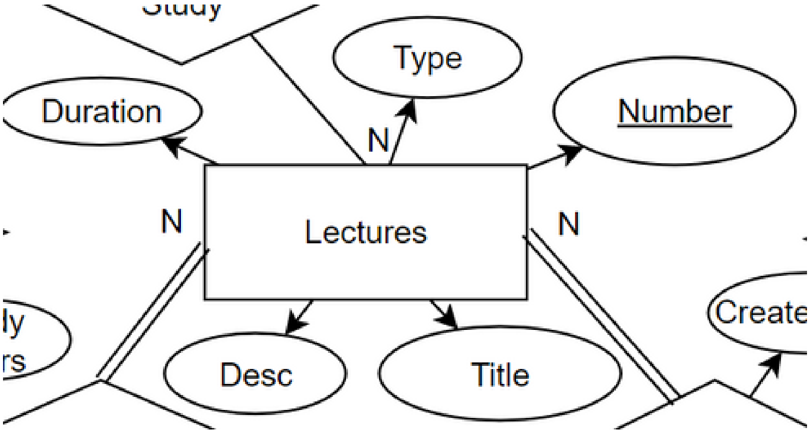
3. Courses



Courses Entity has 8 main properties: ID (CourseID), Name, Language, Basic Content, Requirements, Price, Discount, Number of Stars.

- ID: A unique ID to represent the distinction and serve to look up each course.
- Language: Supported language in the course.
- Basic Content: An overview of some of the basic content of the course.
- Requirements: Requirements for course participants.
- Price: Price excluding tax of the course.
- Discount: Course discount percentage/
- Number of Stars: Number of star votes from users for the course.

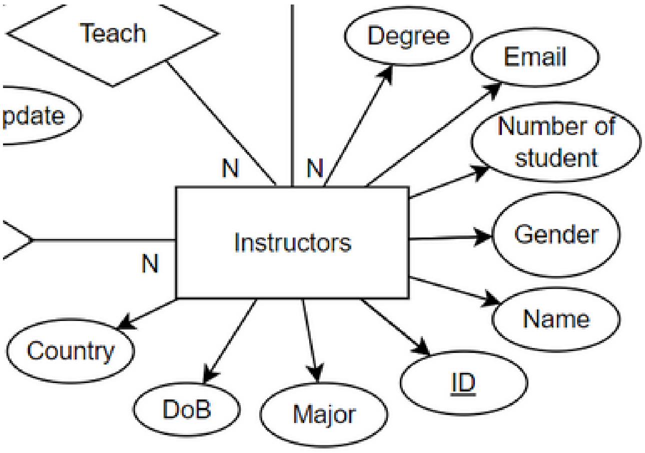
4. Lectures



Lectures Entity has 8 main properties: Number, Title, Type, Duration, Description.

- Number: Lesson number of the lecture in the course.
- Title: Name of lecture.
- Type: Lecture can be of 5 types 'Quiz', 'Test', 'Reading', 'Video', 'Article'.
- Duration: Estimated duration of the lecture.
- Description: Some background information about lecture.

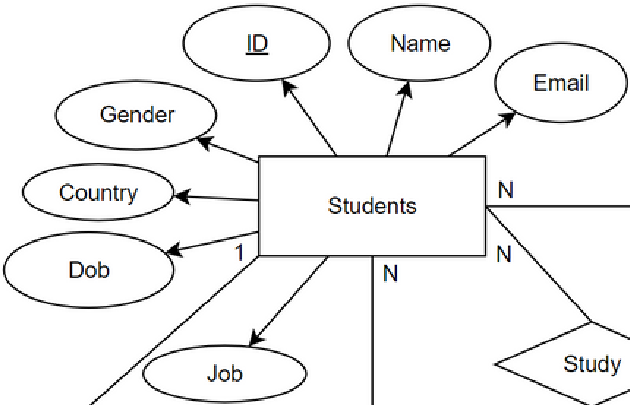
5. Instructors



Instructors Entity has 8 main properties: ID (InstructorID), Name, Gender, Major, Date of Birth, Country, Degree, Number of students, Email, Description.

- ID: A unique ID to represent the distinction and serve to look up each instructor.
- Name: Instructor's name.
- Gender: Instructor's gender.
- Major: Instructor's teaching specialty.
- DoB: Instructor's date of birth.
- Degree: Instructor's degree.
- Number of Student: Number of students on all instructor's courses.
- Email: Unique instructor's email
- Description: Some background information about instructor.

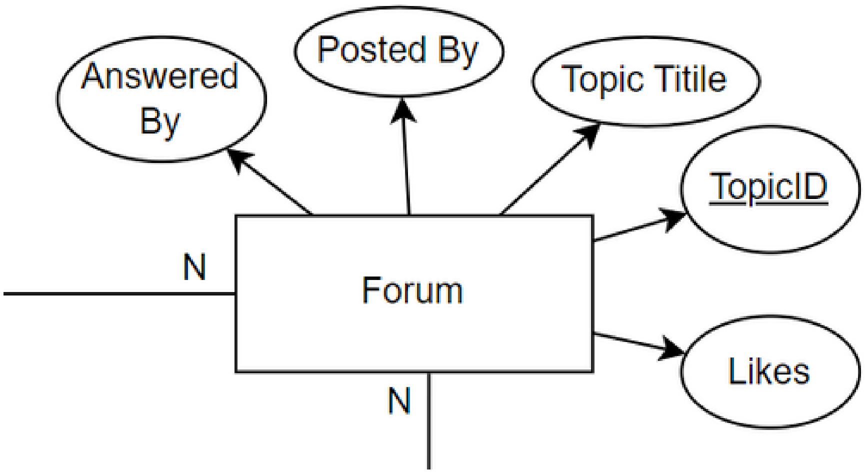
6. Students



Student Entity has 7 main properties: ID (Student ID), Name, Gender, Date of Birth, Country, Job, Email.

- ID: A unique ID to represent the distinction and serve to look up each student.
- Name: Student's name.
- Gender: Student's gender.
- DoB: Student's date of birth.
- Country: Student's living country.
- Job: Student's current occupation.
- Email: Unique student's email

7. Forum



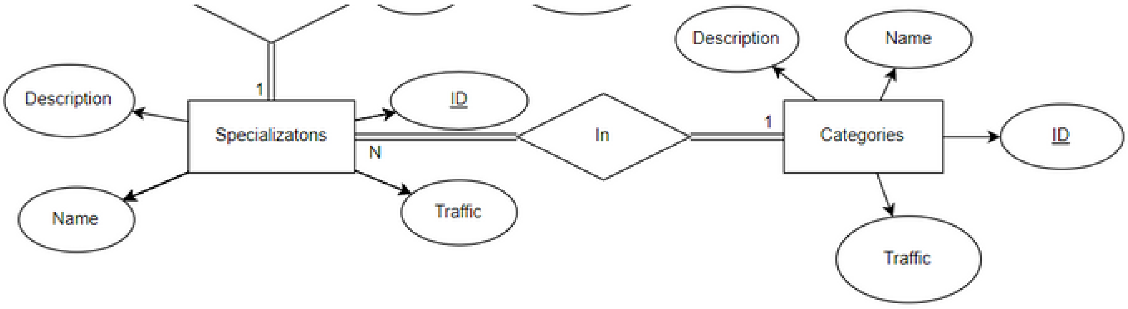
Forum Entity has 5 main properties: TopicID, Topic Title, Posted By, Answered By, Likes.

- ID: A unique ID makes topics have their own numbers and makes it easier to find topics in database.
- Title: Title of the topic.
- Posted By: The person who posted or asked the question.
- Answered By: The person who answered the question or commented on the post.
- Likes: The number of likes or votes from both students and instructors.

We consider the person who asks the question is always a student and the person who answers is an instructor for the sake of simplification.

IV. Explanation for Entity Relationships:

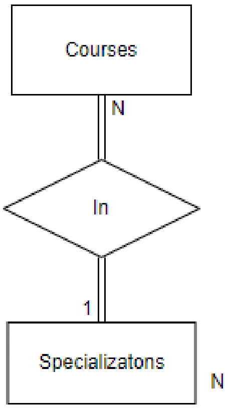
1. Categories - Specialization Relationship



These 2 entities are connected to each other based on the principle:

- A category can contain many specializations.
- But a specialization can only belong to 1 category.
- A category must contain at least 1 specialization.
- A specialization must be in a certain category.

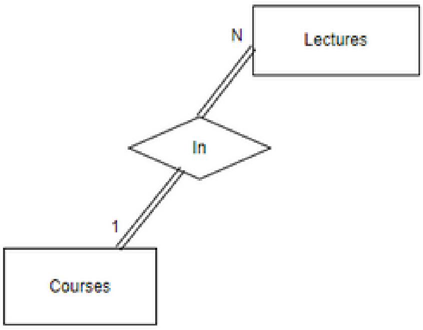
2. Specialization - Courses Relationship



These 2 entities are connected to each other based on the principle:

- A specialization can contain many courses.
- But a course can only belong to 1 specialization.
- A specialization must contain at least 1 course.
- A course must be in a certain specialization.

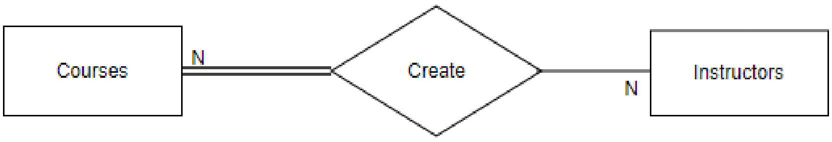
3. Courses - Lectures Relationship



These 2 entities are connected to each other based on the principle:

- A course can contain many lectures.
- But a lecture can only belong to 1 course.
- A course must contain at least 1 lecture.
- A lecture must be in a certain course.

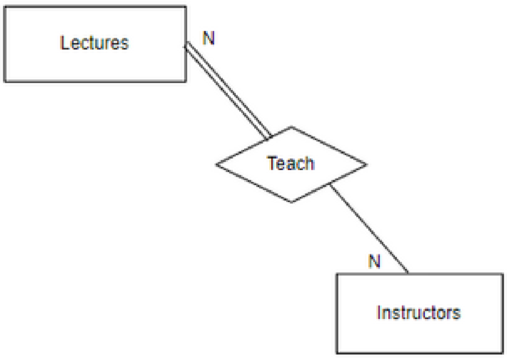
4. Instructors - Courses Relationship



These 2 entities are connected to each other based on the principle:

- An instructor can create many courses.
- A course can be created by multiple instructors.
- An instructor may not need to create a course yet.
- A course must be created by at least one instructor.

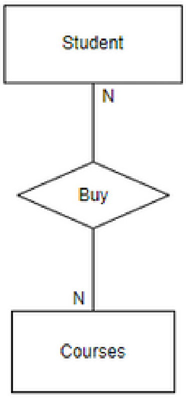
5. Instructors - Lectures Relationship



These 2 entities are connected to each other based on the principle:

- An instructor can teach many lectures.
- A lecture can be taught by multiple instructors.
- An instructor may not need to teach a lecture yet.
- A lecture must be taught by at least one instructor.

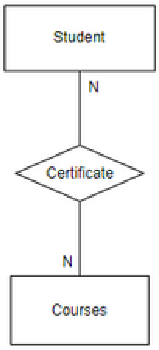
6. Students - Courses Relationship (Buy)



These 2 entities are connected to each other based on the principle:

- A student can buy (subscribe) many courses.
- A course can be bought by multiple students.
- A student does not need to buy any course (just sign up for an account).
- A course created may not yet be sold to any student.

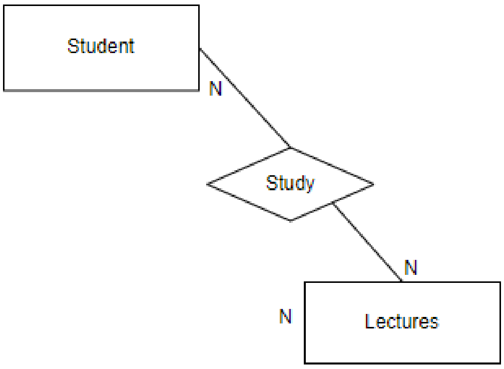
7. Students - Courses Relationship (Certifcate)



A certificate is given to students when they complete a particular course. These 2 entities are connected to each other based on the principle:

- A student can earn a certificate from many courses.
- One course can be in the certificate of many students.
- A student may not have any certificates.
- A course that no one may have completed (get a certificate).

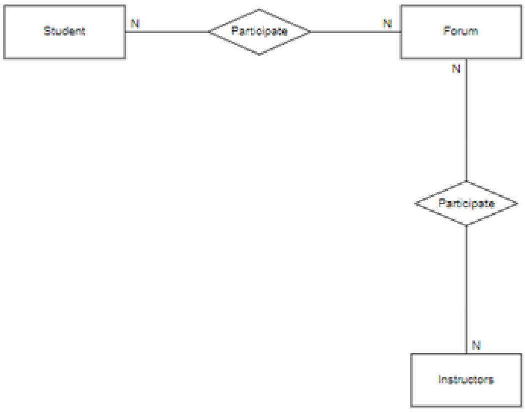
8. Students - Lectures Relationship



These 2 entities are connected to each other based on the principle:

- A student can study many lectures.
- A lecture can be studied by multiple students.
- A student does not need to study any lecture.
- A lecture may not yet be studied by any student.

9. Students - Forum - Instructors Relationship



As mentioned above, Forum is a place to connect Students and Instructors. Both students and instructors participate in the forum and create content in it. So, the relationship will combine all these 3 entities.

- Many students and instructors can join the forum.
- Topics in the Forum can be discussed by everyone.
- Students and instructors of Udemy are not required to join the forum.
- Maybe there are no participants on the forum yet.

D. Relational Mapping

I. Step 1: Mapping Regular Entity

- For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.

- Choose one of the key attributes of E as the primary key for R.

- If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R

Results after implementation:

- **Categories** (ID, Name, Traffic, Description)
- **Specializations** (ID, Name, Traffic, Description)
- **Course** (ID, Name, BasicContent, Language, NumberOfStars, NumberOfSubscribers, Price, Discount, Requirements)
- **Instructors** (ID, Name, Gender, Major, DoB, Country, Degree, NumberOfStudents, Email, Description)
- **Students** (ID, Name, Gender, Country, DoB, Job, Email, Description)
- **Forum** (TopicID, TopicTitile, Likes)

II. Step 2: Weak entities handling

- For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.
- Also, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).
- The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

Results after implementation:

- **Lectures** (Number, *CourseID*, Title, Type, Duration, Description) with *CourseID* referenced ID from table Courses. Combination of Number and CourseID is the primary key of table Lectures.

III. Step 3: Mapping of Binary 1:1 Relation Types

For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.

Since there is no such relationship in our ERD, I will skip this step.

IV. Step 4: Mapping of Binary 1:N Relationship Types

- For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.
- Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.
- Include any simple attributes of the 1:N relation type as attributes of S.

Results after implementation:

- Add the CategoryID attribute as foreign key to the Specializations table.
  => **Specializations** (ID, *CategoryID*, Name, Traffic, Description)
- Add the SpecializationID attribute as foreign key to the Courses table.
  => **Course** (ID, *SpecializationID*, Name, BasicContent, Language, NumberOfStars, NumberOfSubscribers, Price, Discount, Requirements)
- Add the CourseID attribute as foreign key to the Lectures table. Since Lectures is a weak entity, we have added CourseID in step 2.

V. Step 5: Mapping of Binary M:N Relationship Types

- For each regular binary M:N relationship type R, create a new relation S to represent R.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.
- Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

Results after implementation:

- We see between Instructors and Courses there is an M-N relationship. From there I will create an Instructors_Courses table to represent this relationship and its simple attributes.

=> **Instructors_Courses** (InstructorID, CourseID, PublishedDate, LastUpdated) with InstructorID referenced ID from table Instructors, CourseID referenced ID from table Courses. Combination of InstructorID and CourseID is the primary key of table Instructors_Courses.

- Similar to other relationships, we get the result:

=> **Instructors_Lectures** (InstructorID, LectureNum, CourseID, CreatedDate) with InstructorID referenced ID from table Instructors, LectureNum and CourseID referenced from the primary key of table Lectures. Combination of InstructorID and LectureNum, CourseID is the primary key of table Instructors_Lectures.

=> **Students_Courses** (StudentID, CourseID, BillingDate, EnrollingDate, ExpiredDate, isDone, StudyHours) with StudentID referenced ID from table Students, CourseID referenced ID from table Courses. Combination of StudentID and CourseID is the primary key of table Students_Courses.

=> **Students_Lectures** (StudentID, LectureNum, CourseID, Mark, isDone) with StudentID referenced ID from table Students, LectureNum and CourseID referenced from the primary key of table Lectures. Combination of StudentID and LectureNum, CourseID is the primary key of table Students_Lectures.

In my database design, there are 2 special points which are Certificate table and Forum table. In the Certificate table is still the relationship between Students and Courses but shows the second relationship "Achieve Certificate". In Forum table, although it looks like the relationship between Students and Instructors, it is actually the interaction between the two entities mentioned above and the server of the forum. So, I decided to make Forum a separate entity.

Result table:

- **Certificate** (StudentID, CourseID, CompletionDate) with StudentID referenced ID from table Students, CourseID referenced ID from table Courses. Combination of StudentID and CourseID is the primary key of table Certificate.

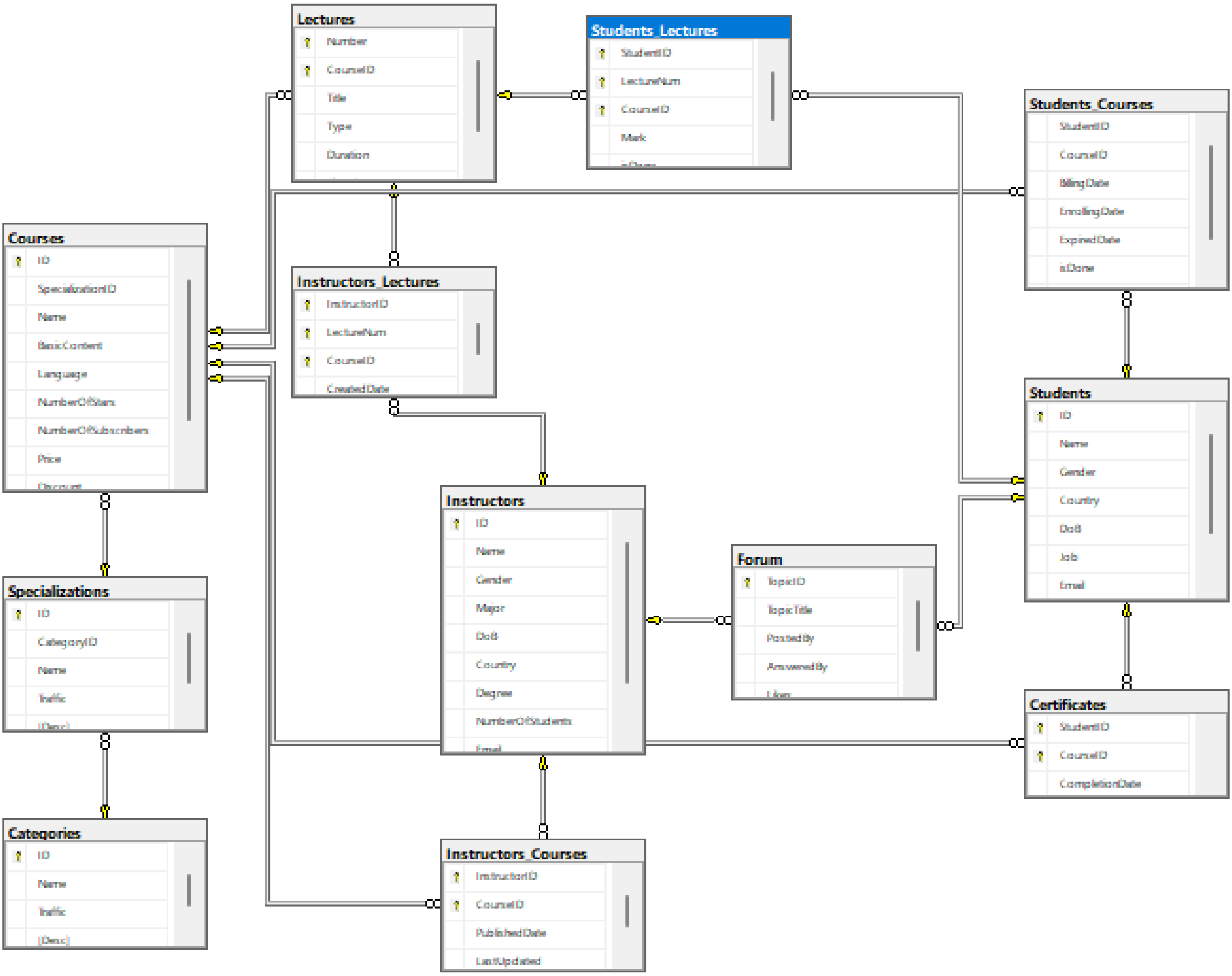
- **Forum** (TopicID, TopicTitle, *PostedBy*, *AnsweredBy*, Likes) with *PostedBy* referenced ID from table Students and *AnsweredBy* referenced ID from table Instructors. (For simplicity, we assume that the questioner is always a student, and the respondent is always an instructor). The primary key of the table will still be TopicID.

E. Logical Design

I. Relational Schema

- **Categories** (ID, Name, Traffic, Description)
- **Specializations** (ID, *CategoryID*, Name, Traffic, Description)
- **Course** (ID, *SpecializationsID*, Name, BasicContent, Language, NumberOfStars, NumberOfSubscribers, Price, Discount, Requirements)
- **Lectures** (Number, CourseID, Title, Type, Duration, Description)
- **Instructors** (ID, Name, Gender, Major, DoB, Country, Degree, NumberOfStudents, Email, Description)
- **Students** (ID, Name, Gender, Country, DoB, Job, Email, Description)
- **Forum** (TopicID, TopicTitle, *PostedBy*, *AnsweredBy*, Likes)
- **Instructors_Courses** (InstructorID, CourseID, PublishedDate, LastUpdated)
- **Instructors_Lectures** (InstructorID, LectureNum, CourseID, CreatedDate)
- **Students_Courses** (StudentID, CourseID, BillingDate, EnrollingDate, ExpiredDate, isDone, StudyHours)
- **Students_Lectures** (StudentID, LectureNum, CourseID, Mark, isDone)
- **Certificate** (StudentID, CourseID, CompletionDate)

II. Database Diagram



III. Table Analysis

1. Categories:

Attributes	Data type	Allow null
ID (Primary key)	varchar(4)	No
Name	varchar(100)	No
Traffic	int	Yes
Description	varchar(250)	Yes

Description of constraints:

- Because ID is the primary key, the attribute to distinguish each category should not be null.
- The category name is the first information the user needs to know so it cannot be null.

SQL command of table Categories:

```
create table Categories
(
[ID] varchar(4) primary key,
[Name] varchar(100) not null,
Traffic int,
[Desc] varchar(250)
)
```

2. Specializations:

Attributes	Data type	Allow null
ID (Primary key)	varchar(5)	No
CategoryID (Foreign key)	varchar(4)	No
Name	varchar(100)	No
Traffic	int	Yes
Description	varchar(250)	Yes

Description of constraints:

- Because ID is the primary key, the attribute to distinguish each specialization should not be null.
- Since CategoryID is a foreign key and do not interfere with the primary key of the table. But according to Udemy, every specialization has to be in a specific category so it can't be null.
- The specialization name is the first information the user needs to know so it cannot be null.

SQL command of table Specializations:

```
create table Specializations
(
[ID] varchar(5) primary key,
[CategoryID] varchar(4) not null foreign key references Categories([ID]),
[Name] varchar(100) not null,
Traffic int,
[Desc] nvarchar(250)
)
```

3. Courses:

Attributes	Data type	Allow null
ID (Primary key)	varchar(6)	No
SpecializationID (Foreign key)	varchar(5)	No
Name	varchar(100)	No
BasicContent	varchar(250)	Yes
Language	varchar(10)	Yes
NumberOfStars	smallint	Yes
NumberOfSubscribers	int	Yes
Price	money	No
Discount	float	No
Requirements	varchar(250)	Yes

Description of constraints:

- Because ID is the primary key, the attribute to distinguish each course should not be null.
- Since SpecializationD is a foreign key and do not interfere with the primary key of the table. But according to Udemy, every course has to be in a specific specialization so it can't be null.
- The course name is the first information the user needs to know so it cannot be null.
- Price and discount are two essential pieces of information for students as buyers to be interested in, so it shouldn't be null.
- The price of a course must be a non-negative number (charged or free) and the discount must also be a non-negative number less than or equal to the price.
- NumberOfStars must be greater than or equal to 0 and less than or equal to 5.

SQL command of table Courses:

```
create table Courses
(
[ID] varchar(6) primary key,
[SpecializationID] varchar(5) not null foreign key references Specializations([ID]),
[Name] varchar(100) not null,
[BasicContent] varchar(250),
[Language] varchar(10),
[NumberOfStars] smallint,
[NumberOfSubscribers] int,
Price money not null,
Discount float not null,
Requirements varchar(250),
check (Price >= 0 and Discount >=0 and Discount <= Price),
check (0 <= [NumberOfStars] and [NumberOfStars] <= 5)
)
```

4. Lectures:

Attributes	Data type	Allow null
Number	smallint	No
CourseID (Foreign key)	varchar(6)	No
Title	varchar(100)	No
Type	varchar(10)	No
Duration	float	Yes
Desc	varchar(250)	Yes
Primary Key	Number + CourseID	

Description of constraints:

- Number and CourseID with CourseID being a foreign key referenced from the Course table's ID. Combine the two to form a primary key, so both should not have null values to distinguish each record in the relation.
- The lecture's title is the first information the user needs to know so it cannot be null.
- Lecture must be in 1 of 5 types: Quiz, Test, Reading, Video, Article and cannot be null.
- Duration of a lecture must be greater than 0.

SQL command of table Lectures:

```
create table Lectures
(
[Number] smallint not null,
CourseID varchar(6) not null foreign key references Courses([ID]),
[Title] varchar(100) not null,
[Type] varchar(10) not null,
[Duration] float,
[Desc] varchar(250),
primary key (Number, CourseID),
check(Duration > 0 and Type in ('Quiz', 'Test', 'Reading', 'Video', 'Article'))
)
```

5. Instructors:

Attributes	Data type	Allow null
ID (Primary key)	varchar(10)	No
Name	varchar(100)	No
Gender	varchar(5)	No
Major	varchar(100)	No
DoB	date	Yes
Country	varchar(25)	Yes
Degree	varchar(250)	No
NumberOfStudents	int	Yes
Email	varchar(25)	No
Desc	varchar(250)	Yes

Description of constraints:

- Because ID is the primary key, the attribute to distinguish each instructor should not be null.
- Name, gender, major (teaching majors) are basic information of a instructor so it should not be null.

- Since Udemy has some rules about the instructor's qualification as mentioned above, so Degree is not null.
- Email is information to register for an account, so it must be unique and cannot be null.
- The gender of a instructor as a rule can only be male or female.
- According to Udemy's rules, to register an account, the person must be over 18 years old.

SQL command of table Instructors:

```
create table Instructors
(
[ID] varchar(10) primary key,
[Name] varchar(100) not null,
[Gender] varchar(5) not null,
[Major] varchar(100) not null,
[DoB] date,
Country varchar(25),
[Degree] varchar(250) not null,
[NumberOfStudents] int,
[Email] varchar(25) unique not null,
[Desc] varchar(250),
check ((Gender = 'Male' or Gender = 'Female') and datediff(year, DoB, getDate()) >= 18)
)
```

6. Students:

Attributes	Data type	Allow null
ID (Primary key)	varchar(10)	No
Name	varchar(100)	No
Gender	varchar(5)	No
DoB	date	Yes
Country	varchar(25)	Yes
Job	varchar(25)	Yes
Email	varchar(25)	No
Desc	varchar(250)	Yes

Description of constraints:

- Because ID is the primary key, the attribute to distinguish each student should not be null.
- Name, gender are basic information of a student so it should not be null.
- Email is information to register for an account, so it must be unique and cannot be null.
- The gender of a student as a rule can only be male or female.
- According to Udemy's rules, to register an account, the person must be over 18 years old.

SQL command of table Students:

```
create table Students
(
[ID] varchar(10) primary key,
[Name] varchar(100) not null,
[Gender] varchar(5) not null,
[DoB] date,
Country varchar(25),
[Job] varchar(25),
[Email] varchar(25) unique not null,
[Desc] varchar(250),
check ((Gender = 'Male' or Gender = 'Female') and datediff(year, DoB, getDate()) >= 18)
)
```

7. Forum:

Attributes	Data type	Allow null
TopicID (Primary key)	varchar(10)	No
TopicTitle	varchar(100)	No
PostedBy	varchar(10)	No
AnsweredBy	varchar(10)	No
Likes	int	Yes

Description of constraints:

- Because TopicID is the primary key, the attribute to distinguish each topic should not be null.
- The topic's title is the first information the user needs to know so TopicTitle cannot be null.
- PostedBy and AnsweredBy are information that is always present in any post, so it should not be null. Furthermore, both are foreign keys referenced from ID of the Students and Instructors tables.

SQL command of table Forum:

```
create table Forum
(
[TopicID] varchar(10) primary key,
[TopicTitle] varchar(100) not null,
[PostedBy] varchar(10) not null foreign key references Students(ID),
[AnsweredBy] varchar(10) not null foreign key references Instructors(ID),
[Likes] int
)
```

8. Certificates:

Attributes	Data type	Allow null
StudentID (Foreign key)	varchar(10)	No
CourseID (Foreign key)	varchar(6)	No
CompletionDate	date	No
Primary key	StudentID + CourseID	

Description of constraints:

- All 3 information StudentID (Student's basic information), CourseID (Course's basic information), CompletionDate (course completion date) are very important to what a certificate represents. So all 3 can't have null value.
- StudentID and CourseID are both foreign keys referencing Students and Courses tables. The combination of 2 attributes creates a primary key.
- Course completion date must be on or before today.

SQL command of table Certificates:

```
create table [Certificates]
(
[StudentID] varchar(10) not null foreign key references Students([ID]),
[CourseID] varchar(6) not null foreign key references Courses([ID]),
[CompletionDate] date not null,
primary key(StudentID, CourseID),
check([CompletionDate] <= getDate())
)
```

9. Instructors_Courses:

Attributes	Data type	Allow null
InstructorID (Foreign key)	varchar(10)	No
CourseID (Foreign key)	varchar(6)	No
PublishedDate	date	No
LastUpdated	date	Yes
Primary key	InstructorID + CourseID	

Description of constraints:

- InstructorID is the ID of the instructor that created the course(s) referenced from the Instructors table, CourseID is the ID of the course generated from the instructor(s) that referenced from the Courses table. Both generate primary key so cannot be null.
- Publish date is an important information so it should not be null.
- The publish date must be or before the update date and be or before the current time.

SQL command of table Instructors_Courses:

```
create table Instructors_Courses
(
[InstructorID] varchar(10) not null foreign key references Instructors(ID),
[CourseID] varchar(6) not null foreign key references Courses([ID]),
[PublishedDate] date not null,
[LastUpdated] date,
primary key(CourseID, InstructorID),
check([PublishedDate] <= getDate() and [LastUpdated] <= getDate() and [PublishedDate] <=
[LastUpdated])
)
```

10. Instructors_Lectures:

Attributes	Data type	Allow null
InstructorID (Foreign key)	varchar(10)	No
LectureNum (Foreign key)	smallint	No
CourseID (Foreign key)	varchar(6)	No
CreatedDate	date	Yes
Primary key	InstructorID + LectureNum + CourseID	

Description of constraints:

- InstructorID is the ID of the instructor that taught the lecture(s) referenced from the Instructors table, LectureNum and CourseID of the lecture taught by instructor(s) that referenced from primary key of the Lectures table. Both generate primary key so cannot be null.
- Creation date must be or before current time.

SQL command of table Instructors_Lectures:

```
create table Instructors_Lectures
(
[InstructorID] varchar(10) not null foreign key references Instructors(ID),
LectureNum smallint not null,
CourseID varchar(6) not null,
[CreatedDate] date,
foreign key (LectureNum, CourseID) references Lectures(Number, CourseID),
primary key(InstructorID, CourseID, LectureNum),
check([CreatedDate] <= getDate())
)
```

11. Students_Courses:

Attributes	Data type	Allow null
StudentID (Foreign key)	varchar(10)	No
CourseID (Foreign key)	varchar(6)	No
BillingDate	datetime	No
EnrollingDate	datetime	No
ExipiredDate	datetime	No
AvgMark	numeric(4,2)	No
isDone	bit	No
StudyHours	float	Yes
Primary key	StudentID + CourseID	

Description of constraints:

- StudentID is the ID of the student that study (subscribe/buy) the course(s) referenced from the Students table, CourseID is the ID of the course studied by the student(s) that referenced from the Courses table. Both generate primary key so cannot be null.
- BillingDate, EnrollingDate, ExpiriedDate are important information for a student when they register for any course so it cannot be null.
- AvgMark (GPA) are important information for student assessment so it should not be null.
- isDone is information whether the student has completed the course or not, it is important information for the certification so it should not be null.
- BillingDate must be or before EnrollingDate and EnrollingDate must be or before ExpiredDate.
- StudyHours must be a non-negative number.

SQL command of table Students_Courses:

```
create table Students_Courses
(
[StudentID] varchar(10) not null foreign key references Students([ID]),
[CourseID] varchar(6) not null foreign key references Courses([ID]),
[BillingDate] datetime not null,
[EnrollingDate] datetime not null,
[ExpiredDate] datetime not null,
[AvgMark] numeric(4,2) not null,
[isDone] bit not null,
[StudyHours] float,
check ([BillingDate] <= [EnrollingDate] and [EnrollingDate] < [ExpiredDate] and StudyHours
>= 0)
)
```

12. Students_Lectures:

Attributes	Data type	Allow null
StudentID (Foreign key)	varchar(10)	No
LectureNum (Foreign key)	smallint	No
CourseID (Foreign key)	varchar(6)	No
Mark	numeric(4,2)	No
isDone	bit	No
Primary key	StudentID + LectureNum + CourseID	

Description of constraints:

- StudentD is the ID of the student that study the lecture(s) referenced from the Students table, LectureNum and CourseID of the lecture studied by student(s) that referenced from primary key of the Lectures table. Both generate primary key so cannot be null.
- Mark (Grade) are important information for student assessment and to calculate the average score, so it should not be null.
- isDone is information whether the student has completed the lecture or not, therefore complete the whole course, it is important information for the certification so it should not be null.

SQL command of table Students_Lectures:

```
create table Students_Lectures
(
[StudentID] varchar(10) not null foreign key references Students([ID]),
LectureNum smallint not null,
CourseID varchar(6) not null,
Mark numeric(4,2) not null,
[isDone] bit not null,
foreign key (LectureNum, CourseID) references Lectures(Number, CourseID),
primary key(StudentID, CourseID, LectureNum)
)
```

F. Queries, Store Procedures and Trigger

I. Sample Queries:

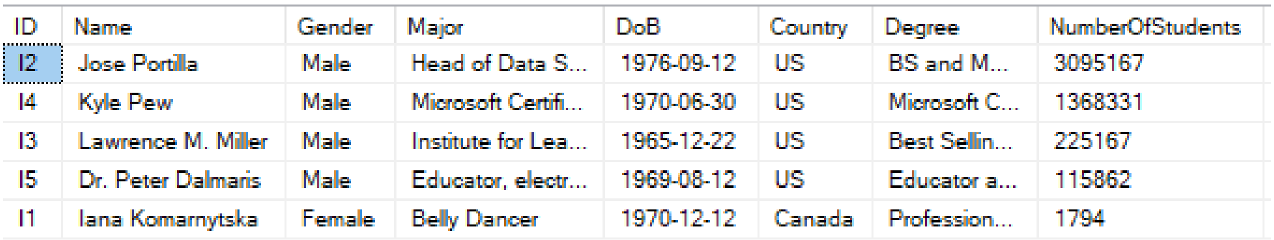
1. Queries using ORDER BY

Requests: Rank the teachers with the most students.

Query:

```
select * from Instructors
order by NumberOfStudents desc
```

Result:



ID	Name	Gender	Major	DoB	Country	Degree	NumberOfStudents
I2	Jose Portilla	Male	Head of Data S...	1976-09-12	US	BS and M...	3095167
I4	Kyle Pew	Male	Microsoft Certifi...	1970-06-30	US	Microsoft C...	1368331
I3	Lawrence M. Miller	Male	Institute for Lea...	1965-12-22	US	Best Sellin...	225167
I5	Dr. Peter Dalmaris	Male	Educator, electr...	1969-08-12	US	Educator a...	115862
I1	Iana Komarnytska	Female	Belly Dancer	1970-12-12	Canada	Profession...	1794

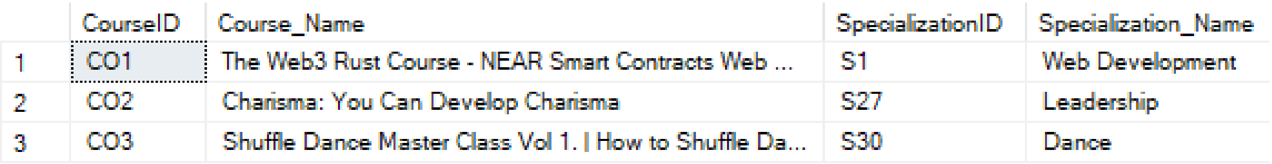
2. Queries using Inner Join

Requests: Find out specialization of courses.

Query:

```
select C.ID [CourseID], C.[Name] [Course_Name], C.SpecializationID, S.[Name]
[Specialization_Name]
from Courses C inner join Specializations S
on C.SpecializationID = S.ID
```

Result:



	CourseID	Course_Name	SpecializationID	Specialization_Name
1	CO1	The Web3 Rust Course - NEAR Smart Contracts Web ...	S1	Web Development
2	CO2	Charisma: You Can Develop Charisma	S27	Leadership
3	CO3	Shuffle Dance Master Class Vol 1. | How to Shuffle Da...	S30	Dance

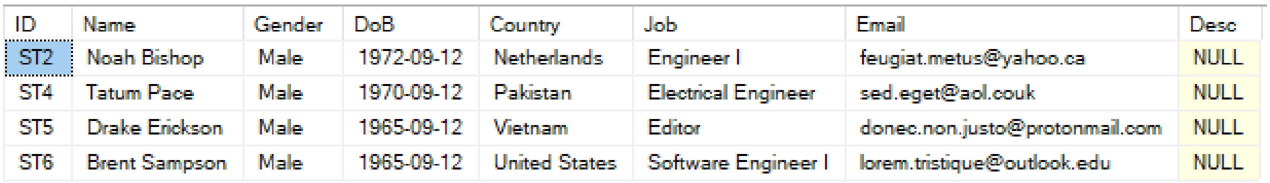
3. Queries using aggregate functions

Requests: Retrieve all students who are 40 years or older.

Query:

```
sselect * from Students
where DATEDIFF(year, DoB, GETDATE()) >= 40
```

Result:



ID	Name	Gender	DoB	Country	Job	Email	Desc
ST2	Noah Bishop	Male	1972-09-12	Netherlands	Engineer I	feugiat.metus@yahoo.ca	NULL
ST4	Tatum Pace	Male	1970-09-12	Pakistan	Electrical Engineer	sed.eget@aol.couk	NULL
ST5	Drake Erickson	Male	1965-09-12	Vietnam	Editor	donec.non.justo@protonmail.com	NULL
ST6	Brent Sampson	Male	1965-09-12	United States	Software Engineer I	lorem.tristique@outlook.edu	NULL

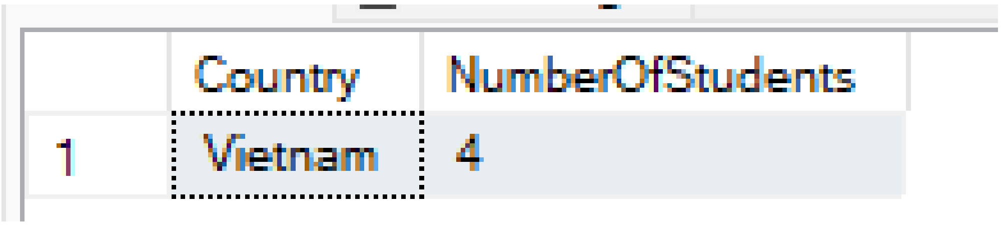
4. Queries using GROUP BY and HAVING clauses

Requests: Display the number of students from Vietnam.

Query:

```
select Country, count(*) [NumberOfStudents]
from Students
group by Country
having Country = 'Vietnam'
```

Result:



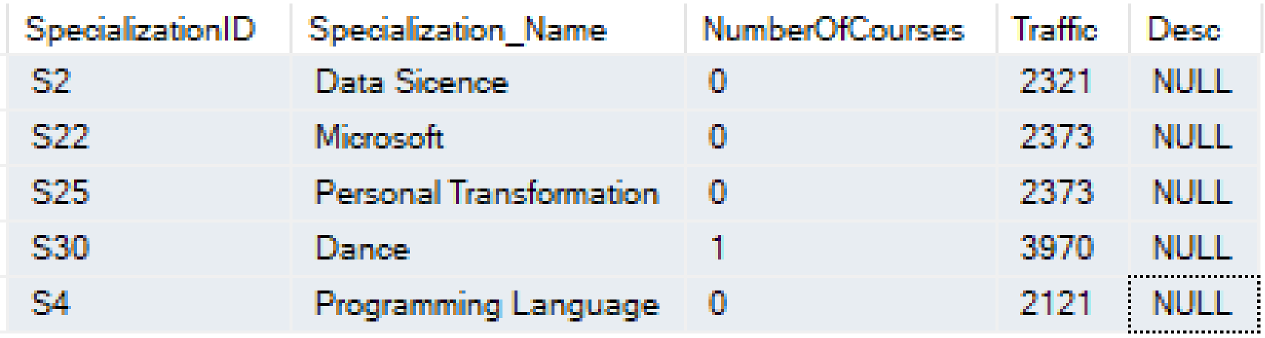
	Country	NumberOfStudents
1	Vietnam	4

Requests: Display the number of Courses in specialization where specialization has traffic greater than 2000 **(**According to Udemy, Specialization always has at least 1 course. But the number of specializations and course is very large, so we will consider the case = 0, in which case it is a new specialization and no course has been added yet**).**

Query:

```
select S.ID [SpecializationID], S.[Name] [Specialization_Name], [NumberOfCourses] =
        case when count(C.ID) is null then 0 else count(C.ID) end,
        Traffic, S.[Desc]
from Specializations S left outer join Courses C
on S.ID = C.SpecializationID
group by S.ID, S.[Name], Traffic, S.[Desc]
having Traffic >= 2000
```

Result:



SpecializationID	Specialization_Name	NumberOfCourses	Traffic	Desc
S2	Data Sicence	0	2321	NULL
S22	Microsoft	0	2373	NULL
S25	Personal Transformation	0	2373	NULL
S30	Dance	1	3970	NULL
S4	Programming Language	0	2121	NULL

5. Queries that use sub – queries as relation

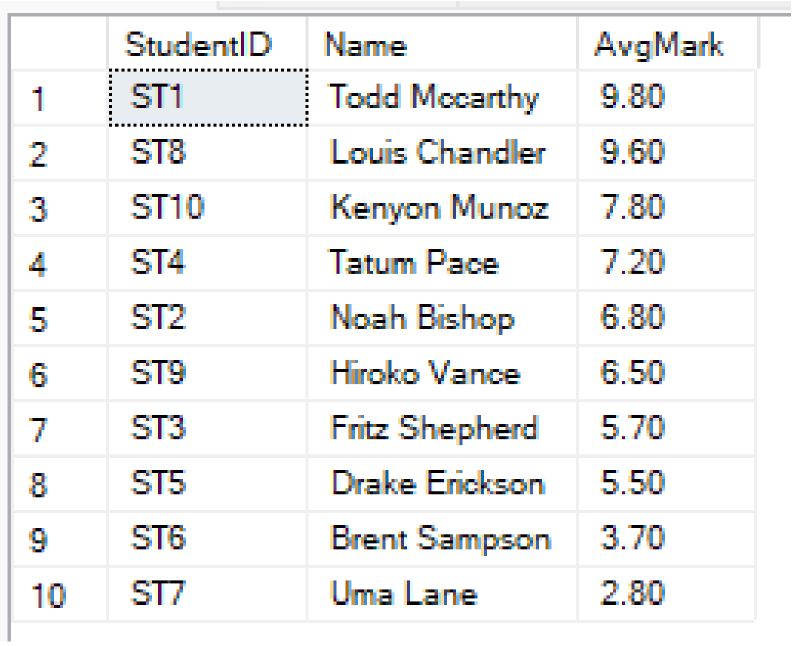
Requests: Find out 5 people with the highest score and 5 people with the lowest score.

Query:

```
with S as(
select top 5 StudentID, [Name], AvgMark
from Students S, Students_Courses C
where S.ID = C.StudentID
order by AvgMark desc),
R as(
select top 5 StudentID, [Name], AvgMark
from Students S, Students_Courses C
where S.ID = C.StudentID
order by AvgMark)

select * from (select * from S union select * from R) M
order by AvgMark desc
```

Result:



	StudentID	Name	AvgMark
1	ST1	Todd Mccarthy	9.80
2	ST8	Louis Chandler	9.60
3	ST10	Kenyon Munoz	7.80
4	ST4	Tatum Pace	7.20
5	ST2	Noah Bishop	6.80
6	ST9	Hiroko Vance	6.50
7	ST3	Fritz Shepherd	5.70
8	ST5	Drake Erickson	5.50
9	ST6	Brent Sampson	3.70
10	ST7	Uma Lane	2.80

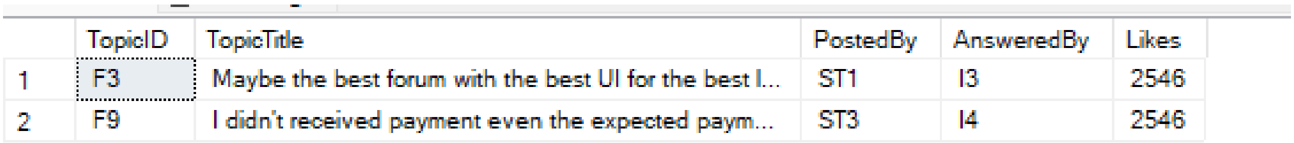
6. A query that uses a sub-query in the WHERE clause

Requests: Display topics with the most likes.

Query:

```
select * from Forum
where Likes = (select top 1 Likes from Forum order by Likes desc)
```

Result:



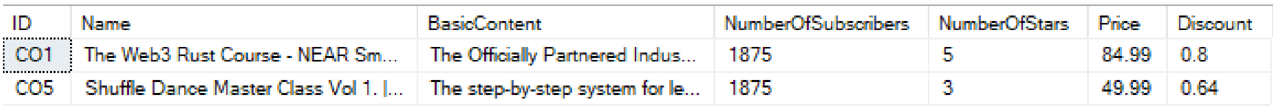
	TopicID	TopicTitle	PostedBy	AnsweredBy	Likes
1	F3	Maybe the best forum with the best UI for the best l...	ST1	I3	2546
2	F9	I didn't received payment even the expected paym...	ST3	I4	2546

Requests: Display courses with the most students, if it's equal, then rank it by number of stars.

Query:

```
select ID, [Name], BasicContent, NumberOfSubscribers, NumberOfStars, Price, Discount
from Courses
where NumberOfSubscribers = (select top 1 NumberOfSubscribers from Courses)
order by NumberOfStars desc
```

Result:



ID	Name	BasicContent	NumberOfSubscribers	NumberOfStars	Price	Discount
CO1	The Web3 Rust Course - NEAR Sm...	The Officially Partnered Indus...	1875	5	84.99	0.8
CO5	Shuffle Dance Master Class Vol 1. I...	The step-by-step system for le...	1875	3	49.99	0.64

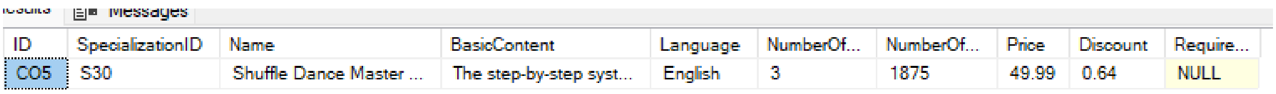
7. Queries that use partial matching in the WHERE clause:

Requests: Display courses that contain the word 'Dance' or 'dance'.

Query:

```
select * from Courses
where [Name] like '%Dance%' or [Name] like '%dance%'
```

Result:



ID	SpecializationID	Name	BasicContent	Language	NumberOf...	NumberOf...	Price	Discount	Require...
CO5	S30	Shuffle Dance Master ...	The step-by-step syst...	English	3	1875	49.99	0.64	NULL

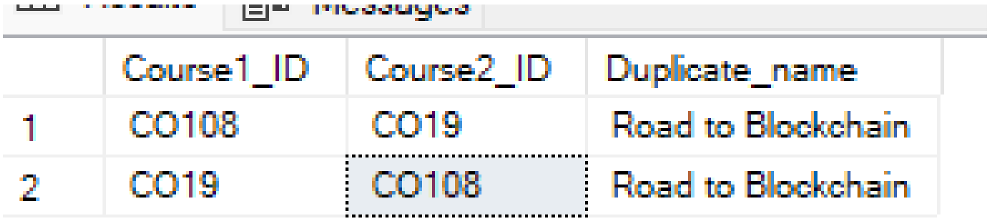
8. Queries that use self-join:

Requests: Display different courses with the same name (duplicated).

Query:

```
select C1.ID [Course1_ID], C2.ID [Course2_ID], C1.[Name] [Duplicate_name]
from Courses C1, Courses C2
where C1.[Name] = C2.[Name] and C1.ID != C2.ID
```

Result:



	Course1_ID	Course2_ID	Duplicate_name
1	CO108	CO19	Road to Blockchain
2	CO19	CO108	Road to Blockchain

II. Stored Procedures:

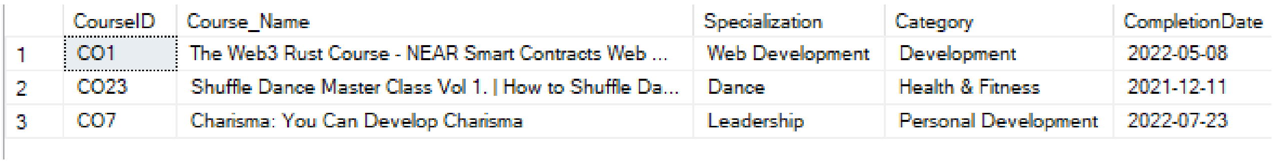
Create stored procedure sp_Certificate with a goal: Enter the student's name to show all the certificates they have.

Stored procedure:

```
create procedure sp_Certificate (@studentID varchar(10))
as
begin
select CourseID, Courses.[Name] [Course_Name], Specializations.[Name] [Specialization],
Categories.[Name] [Category], CompletionDate
from Certificates, Courses, Specializations, Categories
where StudentID = @studentID and CourseID = Courses.ID and SpecializationID =
Specializations.ID and CategoryID = Categories.ID
end
```

Testing: I will test with student with ID 'ST1' => @student = ST1

Output:



	CourseID	Course_Name	Specialization	Category	CompletionDate
1	CO1	The Web3 Rust Course - NEAR Smart Contracts Web ...	Web Development	Development	2022-05-08
2	CO23	Shuffle Dance Master Class Vol 1. | How to Shuffle Da...	Dance	Health & Fitness	2021-12-11
3	CO7	Charisma: You Can Develop Charisma	Leadership	Personal Development	2022-07-23

III. Trigger:

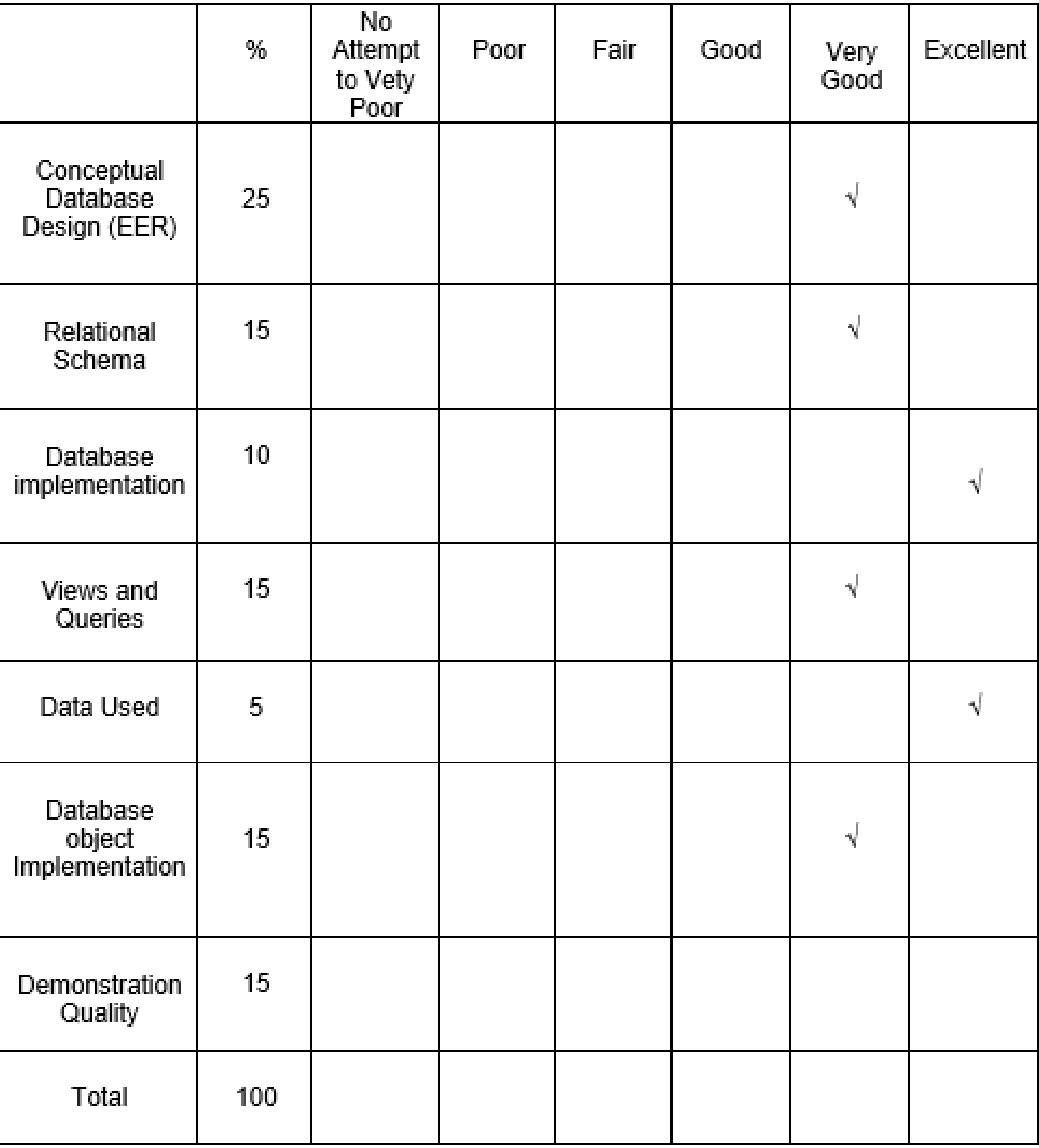
Create a trigger tg_AvgMark on the Students_Lectures table so that after inserting, deleting, updating records in this table, the database will automatically update the average course score in the Students_Courses table of each student that has just updated.

Trigger:

```
create trigger tg_AvgMark on Students_Lectures
after insert, delete, update
as
begin
update Students_Courses
set AvgMark = (select avg(Mark) from Students_Lectures
    where Students_Courses.StudentID = StudentID
    and Students_Courses.CourseID = CourseID
    group by StudentID, CourseID)
where StudentID in (select StudentID from inserted union select StudentID from deleted)
end
```

--- End ---

Self-Assessment Sheet



	%	No Attempt to Vety Poor	Poor	Fair	Good	Very Good	Excellent
Conceptual Database Design (EER)	25					√	
Relational Schema	15					√	
Database implementation	10						√
Views and Queries	15					√	
Data Used	5						√
Database object Implementation	15					√	
Demonstration Quality	15						
Total	100						