
Open Source Online University

Team Databaes

1 Idea

The rise of ‘Zoom University’ raises the question: Why do we need universities, especially ones not of a very high quality? Is there any limit to when a person can start picking up a new skill? Perhaps the biggest experience missing in taking courses online compared to going to college is the social element. But what if we could bring that in too? Especially now that “online” is a well accepted mode of “socializing”. Being online opens doors to easily incorporate desirable attributes of a university experience like international exposure, making friends from diverse backgrounds, ages and cultures etc.

We propose an open-source platform that allows users to transcend traditional boundaries like geography, experience and existing social circles to together undertake existing online courses. We aim to create a collaborative environment that simulates universities, without any arbitrary restrictions on ‘acceptance’, ‘nationality’ and age.

We build a database that contains details about various online courses from various sources such as Coursera, OCW etc. Users can then form study groups on an external platform of their choice to work on courses together. We will attempt to facilitate this by connecting users with similar goals, common languages and other preferences. Metrics to evaluate the efficacy of the groups and courses can then be tracked, providing valuable data on ideal settings for learning. The growth of our platform can be measured using the correlation between friendship formation through courses and average user assigned ratings of study groups found using the platform.

2 Database Description

Users initially join the platform and provide information on their subjects of interest and languages known. They can then start adding friendships with other users they know to get started.

For each course, multiple study groups can exist parallelly with one or more users starting the course at a similar time. Each study group can have a fixed set of allowed languages. An example of why this may be necessary is that north-Indians might prefer code-switching between Hindi and English. In English only groups, this won’t be allowed.

Users can find study groups for a course and language of their choice getting detailed information on different types of ratings, members and when it started. They can view the events list of the study group, which contains like meet times, deadlines, to-do list etc. They can then choose which group to join using a link that leads to the group which is provided (they can even join multiple if they want). This isolates the actual group from our platform, and allows it to be on any popular messaging/forum platforms as decided by the administrators, giving them maximum flexibility.

The study group allows different users to have different roles to allow proper organization and avoid spam. This allows application developers to build rich features for permissions management. The administrators need to maintain information regarding the study group such as the contribution of each user, current list of members, member removals, events calendar maintenance etc. They need to ensure course collaboration policies are followed and decorum is maintained. While our platform serves as an aggregator of study groups, we are not responsible for the actions of administrators of different study groups. With enough users using our platform, users administering will be incentivized to maintain a fair, well managed

and trustable study group.

Users can also make submissions to the platform in the form of posts. A post can either be a blog or a review. Blogs allow users to express their thoughts about the platform, a subject, their interests etc. Blogs have tags, which allow them to be categorized. Reviews allow users to rate courses on a scale from 1-10, and allow other users to evaluate courses at a glance. Blogs allow users to share their own insights, the outcomes of their learnings, motivation, study tips etc. which may be of interest to other users.

A lot of scope is available for software engineering such as implementing a UI/UX for user and study group profiles, automating study group information updating for administrators using external APIs for the respective study group platforms, implementing a social media-like feed, allowing instant-messaging between users in-platform etc. By virtue of this being a database project and not a real organization, we will not be covering these.

3 Entity Types

3.1 User [USER]

Information about each USER entity is stored in the database. Each user entity has userID [UserID], which is a composite key composed of username [Username] and discriminator number [DNum] as key attribute. This allows multiple users to have the same username, with the discriminator number distinguishing users with the same name. Other attributes are Email ID [Email], name [Name] which is a composite attribute [FName,MName,SName] and date of birth [DOB].The country [Country] in which the user lives is also stored. Their contribution score on the platform [UContribution] is a derived attribute using the [UserSgContribution] attribute of the PARTICIPATES_IN relationship type mentioned later.

3.2 Post [POST]

This is a weak entity identified by its relation to a USER. It has a post number [PostNumber], which is the serial number of the post by a particular

user as the partial key.

Its other attributes are: post title [PostTitle], and the content of the post [PostContent]. Two disjoint subclasses of posts are identified - REVIEW and BLOG. A post must be one of the two types(Total).

Review has the attribute of rating [ReviewRating], which takes the value between 1 to 10. A blog has the attribute tag [BlogTag] (multivalued).

3.3 Subject [SUBJECT]

Information about various subjects are stored in the database.Each subject is uniquely identified by its name[SubName].

The popularity score of the subject[SPopularity] is a derived attribute using the HAS_INTEREST_IN relationship type. Number of Courses[NumCourses] is another derived attribute also stored for each SUBJECT entity.

3.4 Course [COURSE]

The courses are uniquely identified by the key attribute Course ID[CourseID]. It maintains the attributes: the organization that is offering the course [CourseOrg], the name of the course [CourseName], the course duration in weeks [CourseDuration], the instructor(s) [CourseInstructor](multi-valued).

In addition to these, the platforms it is offered on [CoursePlatform], the medium of instruction [CourseMedium] (“Text”, “Audio”, “Video”, “Slides”, “Multimedia” or “Augmented Reality”), course difficulty [CourseDifficulty] (“Beginner”, “Intermediate”, “Advanced”) and the expected number of weekly hours of work [CourseHours] are also maintained. The attribute rating [CourseRating] is derived using Review - Course relationships as described below.

3.5 Language [LANGUAGE]

To keep a track of languages spoken that are spoken by a user, a course is offered in or a study group uses, we have a LANGUAGE entity in the database with it's ISO 639-3 Code as its key attribute [LangCode] (3 lower-case latin characters)

and name of the language[LangName] as simple attribute.

3.6 Study Group

[STUDY_GROUP]

The key attribute of a STUDY_GROUP is a link [SgURL] that leads to the study group. It's creation [SgCreated] and when it was last updated [SgUpdated] are stored as timestamp attributes. It contains an attribute to maintain a group to-do list for announcements and "pinned" information [Pins] (multivalued). It also maintains the derived attributes: number of users currently [NumUsers] using distinct users in the PARTICIPATES_IN relationship and average quality rating [Rating] using the UserSgRating attribute of the PARTICIPATES_IN relationship.

3.7 Study Group Events

[SG_EVENT]

SG_EVENT is a weak entity type of STUDY_GROUP. It has a partial key that stores the event number [EventNum] in a serial order for a particular STUDY_GROUP. It has two attributes, the title [EventTitle] and more information [EventInfo]. It contains the subclasses Meet and Target which are disjoint in nature. An event must be of one of the two types (Total Participation).

Meet has attributes: the meeting time [MeetTime] which is a timestamp and duration [MeetDuration] specified in number of minutes. Target has a single attribute: the date it was created [TargetSetDate] and when it has to be achieved by [TargetDeadline].

4 Relationship Types

4.1 USER - STUDY_GROUP - COURSE - LANGUAGE : PARTICIPATES_IN

An entity of this relationship type describes which USER is taking which COURSEs as a part of which STUDY_GROUPS and what LANGUAGEs are used in the STUDY_GROUP. Note that a

LANGUAGE of the STUDY_GROUP may not be one in which the course is available in. A detailed breakdown from the perspective of the different entity types involved is as follows:

Perspective of USER: USERS of the platform may be a part of multiple STUDY_GROUPS. The different STUDY_GROUPS a USER is part of might each have different LANGUAGE(s). They can obviously take multiple COURSEs. Since the aim of the platform is to facilitate learning together, a USER would mostly join at least one STUDY_GROUP for each course. It is still possible for the USER to link a dummy STUDY_GROUP if they wish to do it independently. Initially, a USER might be taking no COURSE, so the minimum bound for it in this relationship is 0.

Perspective of STUDY_GROUP: Each STUDY_GROUP can have multiple allowed LANGUAGES. A STUDY_GROUP might take up multiple COURSEs, example an introductory variant and in-depth variant of the same topic. Any STUDY_GROUP cannot exist without an associated LANGUAGE and COURSE and at least one USER. Hence the minimum bound for it in this relationship is 1.

Perspective of COURSE: Each COURSE has multiple USERS taking it and multiple STUDY_GROUPS catered to it. Initially a COURSE might have 0 USERS and thus 0 STUDY_GROUPS, and hence the minimum bound for it in this relationship is 0.

Perspective of LANGUAGE: Each LANGUAGE can be used in multiple STUDY_GROUPS and can be known by multiple USERS. Multiple COURSEs can be offered in a LANGUAGE too. As there may be no COURSE, USER or STUDY_GROUP in a language, it's minimum bound in this relationship is 0.

Therefore,
USER - [0,n] - PARTICIPATES_IN
STUDYGROUP - [1,n] - PARTICIPATES_IN
COURSE - [0,n] - PARTICIPATES_IN
LANGUAGE - [0,n] - PARTICIPATES_IN

Attributes:
User-Study Group based

- UserSgRole: User can either be a member or

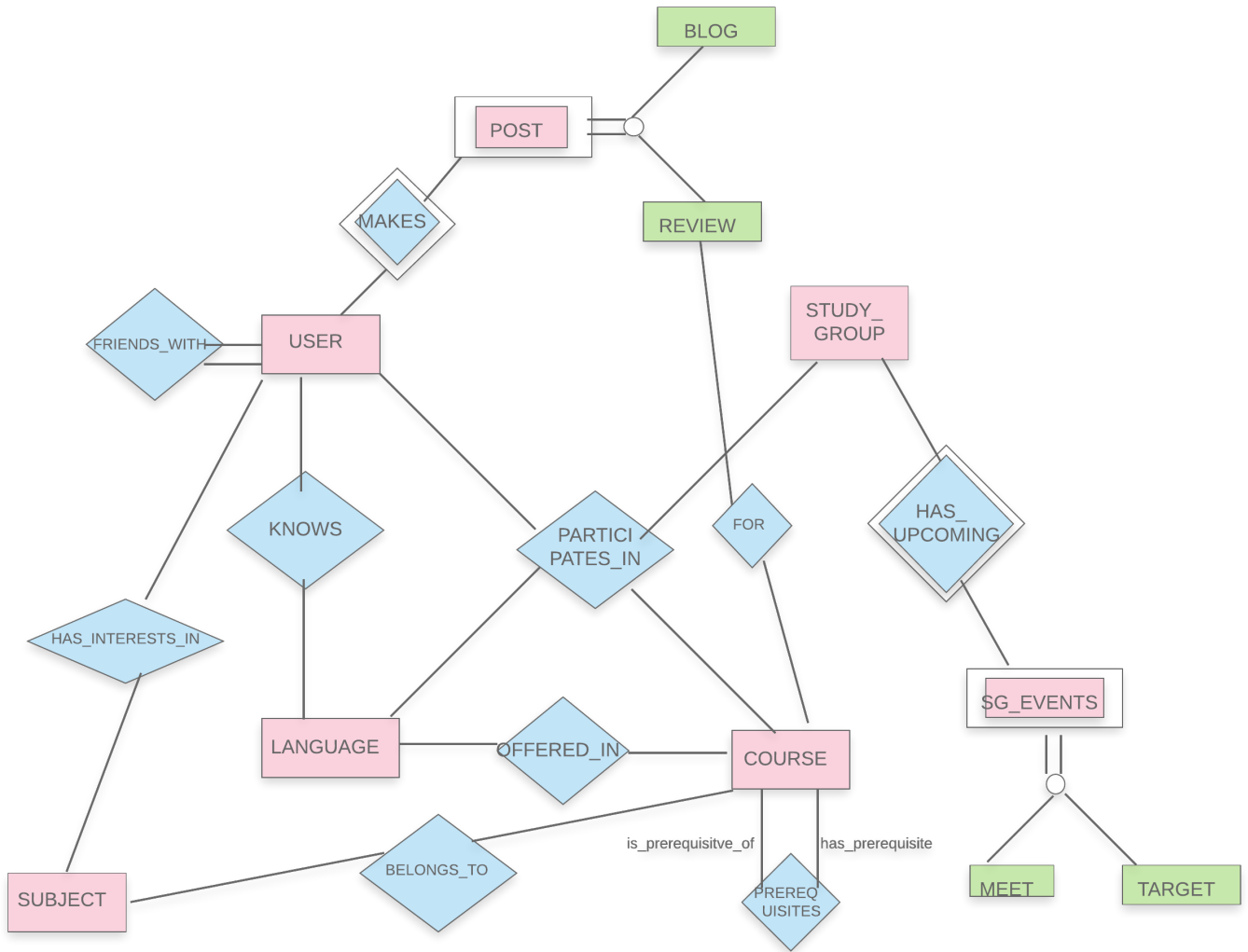


Figure 1: A diagram of the database in ER model style, without describing the attributes, for illustrative purposes

administrator of the group

- UserSgContrib: A numeric score from 1 to 10 based on user's participation levels in the group. This is provided by administrators of the group
- UserSgRating: Rating of the study group by user, on a scale from 1 to 10

User-Course based

- UserPerformance: The performance of the user in assignments and tests as a percentage.
- UserNumHours: The number of hours that user has invested in the course. Provided by the user.

- UserProgress: The user's progress towards completing the course content as a percentage.

Study Group-Course based:

- SgStatus: Describes whether the study group has already, plans to, or currently is doing a particular course. It takes one of the 3 values: "Completed", "Active" or "Planned".

4.2 USER KNOWS LANGUAGE

Since a user might not have a PARTICIPATING_IN relationship in one of their known languages, a separated relationship-type is required. USERS are asked to specify LAN-

GUAGES known by them to facilitate finding appropriate STUDY_GROUPS and COURSES. A user must know at least one language and a language can be spoken by many users. Therefore, USER - [1,n] - KNOWS - [0,n] - LANGUAGE.

Attributes:

Fluency: To check the fluency of the user in a particular language which can take the following values: “Elementary”, “Limited Working”, “Professional Working”, “Native/Bilingual”.

4.3 COURSE OFFERED_IN LANGUAGE

Since a course may be offered in languages not currently used by any entity of the PARTICIPATING_IN relationship type, a separate relationship type is required. A course can be offered in many languages (at least one) and there can be many courses in a particular language.

Therefore, COURSE - [1,n] - OFFERED_IN - [0,n] - LANGUAGE.

Attributes:

SubDub: To check whether the course is “subbed”, “dubbed” or “original” in that particular language.

4.4 STUDY_GROUP HAS_UPCOMING SG_EVENT

STUDY_GROUPS can schedule SG_EVENTS for meetings and deadlines. SG_EVENT is a weak entity type identified by HAS_UPCOMING. A study group can have many upcoming events but an upcoming event can belong to only one study group. Therefore,

STUDY_GROUP - [0,n] - HAS_UPCOMING - [1,1] - SG_EVENT.

4.5 USER MAKES POST

To promote activity on the platform, USERS can make POSTs on the platform. MAKES is the identifying relationship for the weak entity POST. The posts can be reviews for courses or blog posts. The time of creation of a post is stored. A post

must be related to exactly one user but a user can make multiple posts.

Therefore

USER - [0,n] - MAKES - [1,1] - POST

Attributes: Timestamp: The time of creation of the POST

4.6 REVIEW FOR COURSE

REVIEWS can be made for COURSES. As mentioned before, REVIEW is a subclass of POST and is thus already associated with a single USER. Every review must be related to exactly one course, while a course can have multiple reviews. Therefore,

REVIEW - [1,1] - FOR - [0,n] - COURSE

4.7 COURSE PREREQUISITE COURSE

This is a relationship type with the same participating entity type in distinct roles - has_prerequisite and is_prerequisite_of.

A course can be a prerequisite for many courses and a course can have many prerequisites.

Therefore,

COURSE - [0,n] - PREREQUISITE - [0,n] - COURSE.

Attributes:

PreReqImportance: How necessary knowing the prerequisite is to do the course. Takes values “Essential” and “Helpful”.

4.8 USER FRIENDS_WITH USER

A User can be friends with other users.

Therefore,

USER - [0,n] - FRIENDS_WITH - [0,n] - USER.

Attributes:

- Timestamp: To keep track of when they became friends.
- NumCommonCourses: This is a derived attribute to keep track of the number of common courses taken till date, including active ones.

- NumStudyGroups: This is a derived attribute to keep track of the number of common study groups the 2 USERS were a part of till date, including active ones.

4.9 USER HAS INTEREST IN SUBJECT

A user can be interested in multiple subjects, and a subject can have multiple users interested in it. Therefore,
 USER - [1,n] - HAS_INTEREST_IN - [0,n] - SUBJECT

Attributes:

InterestType: In what capacity the user is interested in the subject. It's values can be: "Research", "Professional", "Major", "Minor" and "Casual".

4.10 COURSE BELONGS_TO SUBJECT

A course can belong to multiple subjects (for example a course on data science can belong to Computer Science, Mathematics etc) and a subject can have many courses.

Therefore,
 COURSE - [1,n] - BELONGS_TO - [0,n] - SUBJECT.

5 Functional Requirements

5.1 Retrieval

1. Selection
 - (a) Details (as a tuple) of all subjects available
 - (b) Details (as a tuple) of all courses offered which have rating above a threshold.
2. Projection
 - (a) List of active study groups for a particular course
 - (b) List of courses offered in a particular language
3. Aggregation

- (a) Study group with maximum number of users for a particular course
- (b) Average rating of a particular course by all users who have taken the course
- (c) Average number of events organized by a study group

4. Search

- (a) Search for courses with specific keywords in their name. For example "probability".
- (b) Search friends details with a subset of their full name.

5.2 Analysis

1. **Are we connecting users meaningfully?:** Correlation between user friendships, courses taken and subjects of interest. This is done using the FRIENDS_WITH, HAS_INTEREST_IN and PARTICIPATES_IN relationship data. Example: How many new friends per study group are our users gaining?
2. **How to optimize collaborative learning?:** Study Group metrics like number of users, number of friendships between these users, frequency of meets etc. are either available or derivable for each STUDY_GROUP. We also have access to UserProgress, UserSgRating, UserPerformance for each course. These can be correlated to answer general questions about what is the ideal study group? Do users learn better when taking courses with pre-existing friends? These insights can be valuable not only for our platform, but also educators across the world.

5.3 Modification

5.3.1 Insertion

1. Registering a new user, with a minimum of one known language and subject interest.
2. Creating a new study group, which is related to a minimum of one user, language and course.

-
3. Adding details of newly released courses.

5.3.2 Update

1. Updating subject interest for user
2. Updating content of post by user, for both reviews and blogs
3. Updating status of study groups

5.3.3 Delete

1. Deleting post by user
2. Removing courses no longer offered