

User

Username (primary key)	Not null	Unique	varchar(15)
Password	Not null		varchar(8 - 22)
Name	Not null		varchar(15)
Age			Int > 5
Country			varchar(20)
E-mail	Not null	Unique	Email domain
Published posts			integer(0 - 50000)
Profile Pic.			url domain

Other

Type	Not null		enum
Membership started			Date domain
Membership expiry			Date domain

Post

ID (primary key)	Not null	Unique	varchar(6)
Context	Not null		varchar(512)
Publish date			Date domain

Day			integer(1 - 31)
Month			integer(1-12)
Year			integer(1975 - 2035)
Publish time	Not null		varchar(5)
Last edit	Not null		varchar(16)
Likes (calc)	Not null		integer(0 - 10000)

Admin-Request

Result			integer(0 - 1)
Check time			varchar(10)

Request

ID (primary key)	Not null	unique	varchar(6)
Submission time			varchar(10)

Question

Title	Not null		varchar(0 - 1000)
--------------	----------	--	-------------------

Sample Test

ID (primary key)	Not null	Unique	varchar(8)
Date			varchar(10)
Uni held			varchar(0 - 40)
# of questions			integer(1 - 30)

Lessons

ID (primary key)	Not null		varchar(6)
Name			varchar(10)

Topic

Lesson ID (primary key)	Not null	unique	varchar(6)
Name			varchar(0 - 22)
Subtopic			varchar(0 - 22)

Course

ID (primary key)	Not null	Unique	varchar(6)
Title	Not null		varchar(0 - 20)
Schedule			varchar(20)
Capacity	Not null		integer(1 - 120)
Attendee			integer(1-120)
Number of sessions			integer(0 - 40)
Price			varchar(0 - 10)

Content

ID (primary key)	Not null	Unique	varchar(10)
Title	Not null		varchar(0 - 20)
Author			varchar(0 - 40)
Stuck	Not null		integer(0 - 100)
Price			varchar(0 - 10)

Book

ISBN	Not null	Unique	varchar(16)
Publisher	Not null		varchar(0 - 22)

Edition			integer(1 - 10)
----------------	--	--	-----------------

Handout

# of pages			integer(1 - 1000)
-------------------	--	--	-------------------

Relations

- User(Username, Password, Name, Age, Country, E-mail, Published Posts, Profile Pic.)
- Admin(**User.Username**)
- Other(**User.Username**, Type, Membership Started, Membership Expiry)
- Post(ID, Context, Publish Date Day, Publish Date Month, Publish Date Year, Publish Time, Last Edit, Likes, **Publish.Username, Publish.LessonID, Publish.TopicName**)
- Question(**Post.ID**, Title, **SampleTest.ID**)
- Answer(**Post.ID, Question.Post.ID**)
- Like(**User.Username, Post.ID**, Time)
- Bookmark(**User.Username, Post.ID**, Time)
- Lesson(LessonID, Name)
- Topic(**Lesson, Name**, Subtopic1, Subtopic2)
- SampleTest(TestID, Date, Uni. Held, Number Of Questions, **Lesson**)
- Content(ID, Title, Author, Stock, Price, **Share.Username**)
- Book(**Content.ID**, ISBN, Publisher, Edition)
- Handout(**Content.ID**, # of pages)
- Course(ID, Title, Schedule, Capacity, Attendee, # of sessions, Price, **About.Lesson, Instructor.Other.Username**)
- Request(ID, Submission Time)
- StartCourse(**Request.ID, Submit.Other, Submit.Lesson**)
- UpgradeToInstructor(**Request.ID, Submit.Other**)
- CheckRequest(**Admin.Username, Request.ID**, CheckTime, Result)
- Enroll(**Course.ID, Other.Username**, Enrollment Date)
- Reference(**Content.ID, Course.ID**)

Constraints

1. محدودیت پسورد (در domain لحاظ شود).
2. تعداد موجود از محتوا نباید منفی شود (در domain لحاظ شود).
3. ~~(طبق ER) ادمین نمی تواند خود درخواستی ثبت کند.~~
4. ~~کاربر عادی نباید بتواند درخواستی را بررسی کند (ما هنوز ممیزی برای تفاوت بین ادمین و عادی نداریم).~~
5. ~~جواب نباید برای یک جواب پست شود (فقط برای یک پرسش. اینجا هم تفاوت سوال و پرسش لحاظ نشده است).~~
6. برگزار کننده یک کلاس باید از نوع کاربر عادی استاد باشد. (done)
7. ثبت کننده درخواست استادی باید از نوع کاربر عادی باشد. (done)
8. ~~ثبت درخواست خرید محتوا به شرط موجود بودن آن محتوا قابل ثبت است.~~
9. اگر درخواست استادی تایید شد باید نوع کاربر عوض شود. (done)
10. تعداد شرکت کننده های هر کلاس از ظرفیت آن کلاس بیشتر نیست. (done)
11. ~~هر دوره آموزشی تنها بایک درس در ارتباط است.~~
12. کاربر استاد نباید بتواند برای شرکت در دوره ای که خودش استاد آن است ثبت نام کند. (done)
13. تریگر برای ادد کردن هر نوع یوزر به جدول یوزر (done)
14. تریگر برای صفات مجازی

شک:

- در پست ما صفت یوزر داریم نیازی به داشتن رفرنس در سوال و جواب هم هست؟
- اونایی که توی زیرنوع ننوشتیم رو از not null در آوردم.
- چجوری بعد از true شدن یک درخواست دوره آیدی براش بسازیم؟

مشکلات:

ساختن primary key

صفات مجازی. (یه چیزی ننوشتیم تست شود.)

ویو تعریف نکردیم.

با تایید تشکیل دوره توسط یک آموزنده باید یک کورس ساخته شود (نوشته شده ولی باگ دارد.)

محدودیت صفت ها تمام نشده مثلا پسورد حداقل ۸ کاراکتر.

محدودیت های جزئی. مثلا last edit باید بعد از publish time باشد و مسائلی از این دست.

lesson(id, name)

course(id, title, schedule, capacity, attendee,
#of_sessions, price, lesson_id,
instructor_non_admin_username)

start_course(request_id, submission_time,
submit_non_admin_user, submit_lesson_id)

upgrade(request_id, submission_time,
submit_non_admin_user)

check(admin_username, Request_id, Date, time, result)