Project Milestone 1:
UVA StudyBuddies Project

Christian Boxley - clb7cy@virginia.edu
Sammy Lahrime - sl8rn@virginia.edu
Geoff Hicks - gdh8jh@virginia.edu
Sam Book - slb8pm@virginia.edu

04 October, 2020

1. Develop E-R diagram

1.1. Assumptions

Each user account can submit many feedback, each piece of feedback can be submitted by at most one user account

Each submitted feedback can be reviewed by at most one administrators, each administrator can review multiple feedbacks

Each user account can be assisted by at most one member of the support team, each member of the support team can assist many user accounts

Each user account can design at most one profile, each profile can be designed by at most one user account

Each user account can upload many documents, each document can be uploaded by at most one user account

Each document contains at most one piece of technical information, each piece of technical info is contained by at most one document

There are two types of documents: quizzes and flashcards

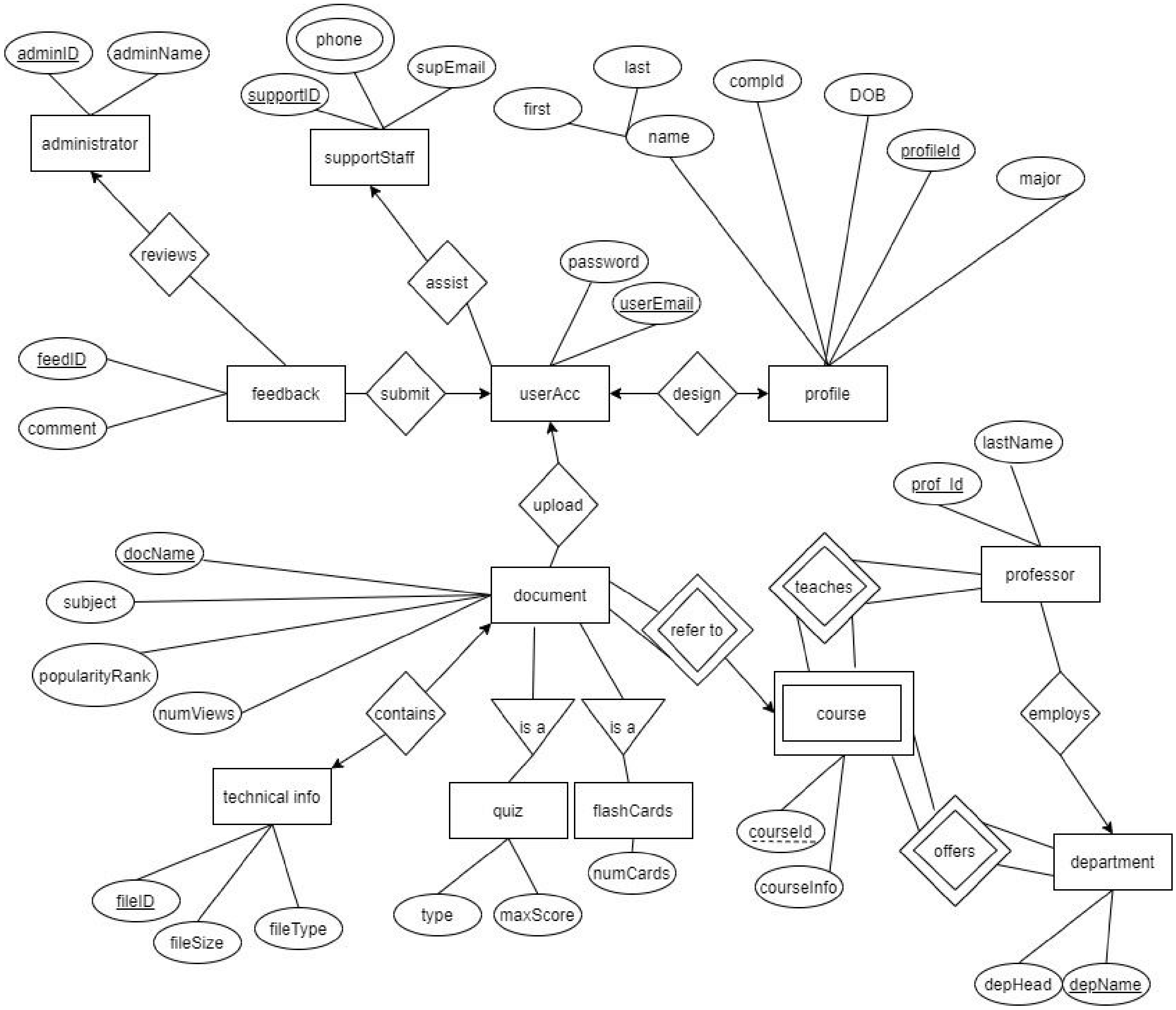
Each document must refer to at most one course, each course can refer to many documents

Each course must be taught by at least one professor, and each professor must teach at least one course

Each course must be offered by at least one department, and each department must offer at least one course

Each profesor can be employed by at most one department, and each department can employ many professors

1.2. E-R Diagram



2. Convert E-R diagram into tables

2.1. Schema Creation

administrator(adminID, adminName)

reviews(feedID,adminID)

feedback(feedID, comment)

submit(feedID, userEmail)

userAccount(userEmail, password)

assist(userEmail, supportID)

supportStaff(supportID, supEmail)

supportStaff_phone(supportID, phone)

profile(profileID, userEmail, major, DOB,compid, first_name, last_name)

upload(docName, userEmail)

document(docName, courseID, subject, popularityRank, numViews)

flashCards(docName, numCards)

quiz(docName, type, maxScore)

technical_info(fileID, docName, fileType, fileSize)

course(docName, profID, depName, courseID, courseInfo)

department(depName, courseID, depHead)

professor(profID, lastName, courseID)

employs(profID, depName)

3. Decompose tables using 3NF or BCNF

3.1. Functional Dependencies

administrator
adminID -> adminName

reviews
feedID -> adminID

feedback
feedID -> comment

submit
feedID -> userEmail

userAccount
userEmail -> password

assist
userEmail -> supportID

supportStaff
supportID -> supEmail

profile
profileID -> userEmail, major, DOB,compid, first_name, last_name

upload
docName -> userEmail

document
docName -> courseID, subject, popularityRank, numViews

flashcards
docName -> numCards

quiz

docName -> type, maxScore

technical_info
fileID -> docName, fileType, fileSize

course
docName, profID, depName, courseID -> courseInfo

department
depName -> courseID, depHead

professor
profID -> lastName, courseID

employs
profID -> depName

3.2. Decomposition Process

For this section, we chose to use 3NF decomposition.
Note: As the “supportStaff_phone” table was already fully decomposed, we chose to leave it out of our decomposition process.

(1) Copy FDs as they originally were

adminID -> adminName
feedID -> adminID
feedID -> comment
feedID -> userEmail
userEmail -> password
userEmail -> supportID
supportID -> supEmail
profileID -> userEmail, major, DOB,compid, first_name, last_name
docName -> userEmail
docName -> courseID, subject, popularityRank, numViews
docName -> numCards
docName -> type, maxScore
fileID -> docName, fileType, fileSize
docName, profileID, depName, courseID -> courseInfo
depName -> courseID, depHead
profID -> lastName, courseID
profID -> depName

(2) Combine any functional dependencies with similar left hand sides

adminID -> adminName
feedID -> adminID, comment, userEmail
userEmail -> password, supportID
supportID -> supEmail
profileID -> userEmail, major, DOB,compid, first_name, last_name
docName -> userEmail, courseID, subject, popularityRank, numViews, numCards, type, maxScore
fileID -> docName, fileType, fileSize
docName, profileID, depName, courseID -> courseInfo
depName -> courseID, depHead
profID -> depName, lastName, courseID

(3) Remove Extraneous Attributes

For this step, we will look for any functional dependencies (FD) that have two or more attributes on the left-hand side and see if we can eliminate any. The only FD that fits this criteria is “docName, profileID, depName, courseID -> courseInfo”. For this, we will first look at docName closure (docName+), profileID closure (profileID+), depName closure (depName+), and courseID closure (courseID+).

docName+ includes docName, userEmail, courseID, subject, popularityRank, numViews, numCards, type, and maxScore

depName+ includes depName, courseID, depHead

profileID+ includes userEmail, major, DOB, compid, first_name, last_name

courseID+ includes courseID

Since it appears that we cannot eliminate docName, depName, profileID, or courseID there are no changes we can make to “docName, profileID, depName, courseID -> courseInfo”. Thus, there are no extraneous attributes that can be eliminated, so our list of functional dependencies remains the same as it was in step 2. If we include the “supportStaff_phone” table that we initially excluded from our decomposition process, this translates to:

adminID, adminName // feedID, adminID, comment, userEmail // userEmail, password, supportID // supportID, supEmail // supportID, phone // profileID, userEmail, major, DOB, compid, first_name, last_name // docName, userEmail, courseID, subject, popularityRank, numViews, numCards, type, maxScore // fileID, docName, fileType, fileSize // docName, profileID, depName, courseID, courseInfo // depName, courseID, depHead // profID, depName, lastName, courseID

3.3. Final Normalized Table Schema Statements

administrator(adminID, adminName)
feedback(feedID, adminID, comment, userEmail)
user(userEmail, password, supportID)
supportStaff(supportID, supEmail)
supportStaff_phone(supportID, phone)
profile(profileID, userEmail, major, DOB,compid, first_name, last_name)
document(docName, userEmail, courseID, subject, popularityRank, numViews, numCards, type, maxScore)
technical_info(fileID, docName, fileType, fileSize)
course(docName, profileID, depName, courseID, courseInfo)
department(depName, courseID, depHead)
professor(profID, depName, lastName, courseID)