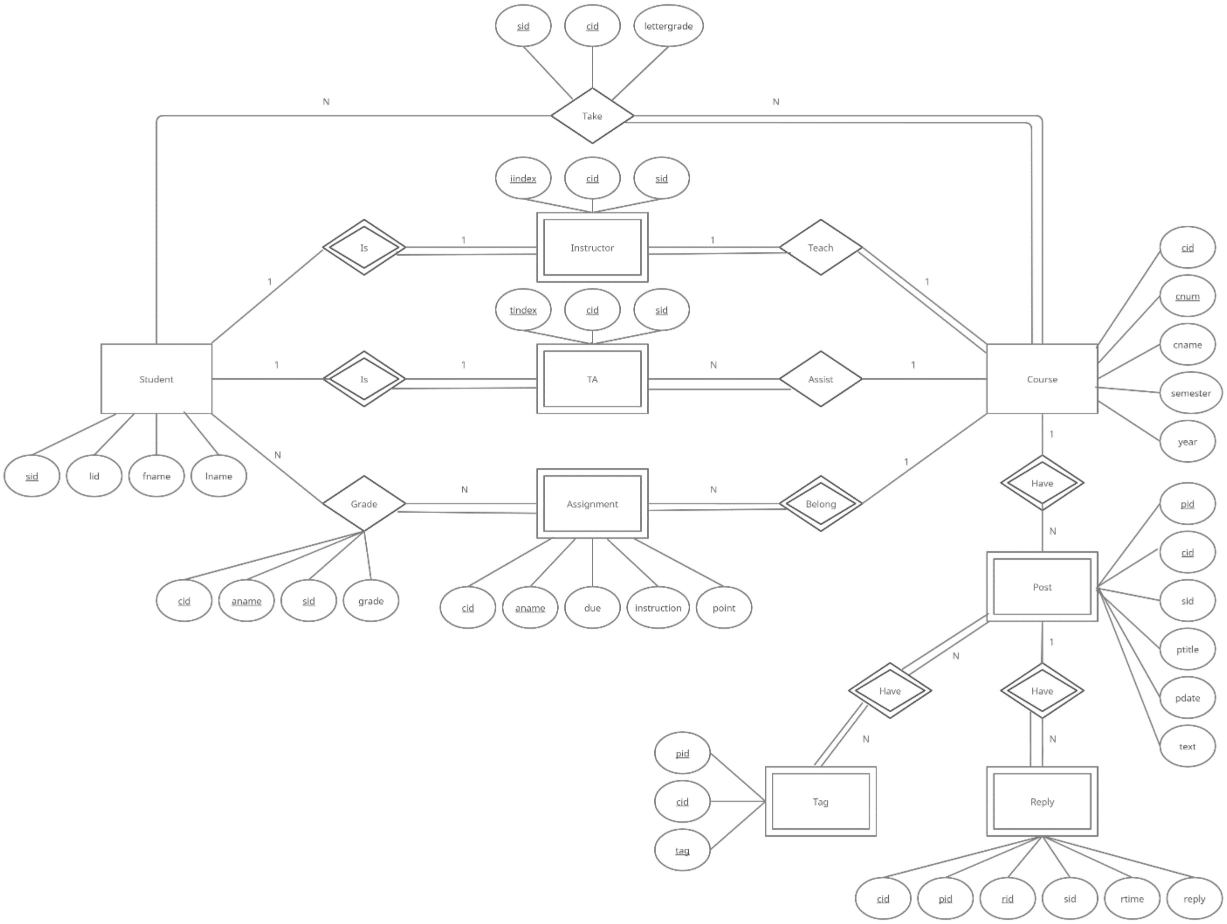
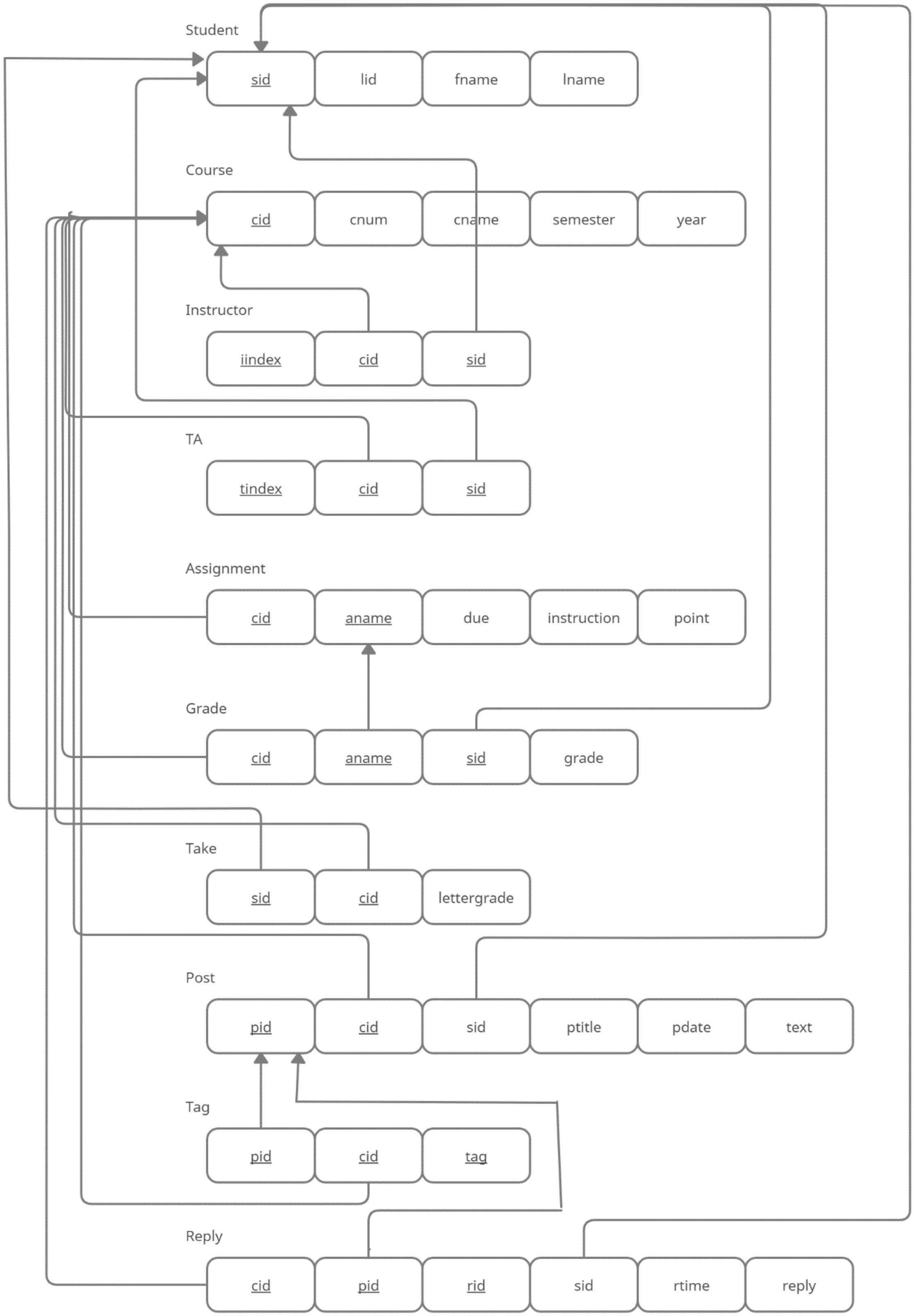
CS377 Final Project

Yulin Hu

1.



2.


Student
sid
lid
fname
lname
Course
cid
cnum
cname
semester
year
Instructor
iindex
cid
sid
TA
tindex
cid
sid
Assignment
cid
aname
due
instruction
point
Grade
cid
aname
sid
grade
Take
sid
cid
lettergrade
Post
pid
cid
sid
ptitle
pdate
text
Tag
pid
cid
tag
Reply
cid
pid
rid
sid
rtime
reply

Student (sid, lid, fname, lname)

Course (cid, cnum, cname, semester, year)

Instructor (iindex, cid, sid)

TA (tindex, cid, sid)

Assignment (cid, aname, due, instruction, point)

Grade (cid, aname, sid, grade)

Take (sid, cid, lettergrade)

Post (pid, cid, sid, ptitle, pdate, text)

Tag (pid, cid, tag)

Reply (cid, pid, rid, sid, rtime, reply)

3.

The functional dependencies include:

Student (sid, lid, fname, lname)

{sid→lid, lid, fname, lname}

Course (cid, cnum, cname, semester, year)

{cid→cnum, cname, semester, year}

Instructor (iindex, cid, sid)

{iindex→cid, sid}

TA (tindex, cid, sid)

{tindex→cid, sid}

Assignment (cid, aname, due, instruction, point)

{cid, aname→due, instruction,point}

Grade (cid, aname, sid, grade)

{cid, aname, sid→grade}

Take (sid, cid, lettergrade)

{sid, cid→lettergrade}

Post (pid, cid, sid, ptitle, pdate, text)

{pid→cid, sid, ptitle, pdate,text}

Tag (pid, cid, tag)

Reply (cid, pid, rid, sid, rtime, reply)

{cid, pid, rid→sid, rtime, reply}

There is no partial dependency or transitive dependency, so the relational schema is in 3nf.

4. Successfully created and loaded: http://34.68.205.71/login.php

5. See the document in the folder.

6. Database Description:

The **student** table defines all the users including students, instructors, and TAs.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
sid	varchar(255)	Unique identifier for each user
lid	varchar(255)	Login ID
fname	varchar(255)	User first name
sname	varchar(255)	User last name

The **course** table defines all the existing courses.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
cid	varchar(255)	Unique identifier for each course
cnum	varchar(255)	Course number
cname	varchar(255)	Course name
semester	varchar(255)	The semester of the course
year	varchar(255)	The year of the course

The **instructor** table defines all the users who are instructors.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
iindex	varchar(255)	A unique index for all the instructors
cid	varchar(255)	Course ID of the course the instructor teaches
sid	varchar(255)	The instructor's sid

The **ta** table defines all the users who are TAs.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
tindex	varchar(255)	A unique index for all the TAs
cid	varchar(255)	Course ID of the course the TA assists
sid	varchar(255)	The TA's sid

The **assignment** table defines all the assignments corresponding to a course.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
cid	varchar(255)	The course the assignment belongs to
aname	varchar(255)	Assignment name
due	datetime	Due date of the assignment
instruction	varchar(255)	Text instructions of the assignment
point	int	Total point of the assignment

The **grade** table defines the assignment grades of a student in a course.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
cid	varchar(255)	The course the assignment belongs to
aname	varchar(255)	Assignment name
sid	varchar(255)	Student ID
grade	int	Student’s grade of the assignment

The **take** table defines the final letter grade of a student taking a course.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
sid	varchar(255)	Student who takes the course
cid	varchar(255)	Unique Course ID
lettergrade	varchar(255)	Student’s final letter grade of the course

The **post** table defines all the posts with details.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
pid	varchar(255)	Unique identifier for each post
cid	varchar(255)	Course ID of the post
sid	varchar(255)	Student who posted the post
ptitle	varchar(255)	Post title
pdate	datetime	Post date
text	varchar(255)	Post text

The **tag** table defines all the existing tags a certain post has.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
pid	varchar(255)	The post which has the tag
cid	varchar(255)	Course ID of the post
tag	varchar(255)	The tag (Tag here is single-valued)

The **reply** table defines all the replies with details a post has.

ATTRIBUTE NAME	ATTRIBUTE TYPE	DESCRIPTION
cid	varchar(255)	Course ID of the post
pid	varchar(255)	Post ID
rid	varchar(255)	Reply ID
sid	varchar(255)	Student who replied
rtime	datetime	Reply time
reply	varchar(255)	Reply message

7.

login.php

Enter student ID and login ID and click login. Wrong credentials will cause failure to login, with red notification “Error loginID or studentId”.

home.php

After logging in, student or instructor will arrive at the home page. The home page shows all the classes with final grades the user participated as a student (in time ascending order). This function will act as the transcript corresponding to question 11.

On the right most column, if the user is an instructor/TA, which means he/she has the permission to set grades or add assignments, there will be a “manage course” link that takes the instructor to the course. If the user is not an instructor/TA, then there will be a “show course” link, which takes the student to the course to see the assignments and grades. Students who have no access to the course will still be able to click in “show course”, but no information will be displayed. This page is corresponding to Question 10 and 11.

course.php

A student can check the course assignments and grades of the courses they have access to.

This is corresponding to Question 7.

An instructor/TA can:

1. Add an assignment (with assignment name, due date, instructions, point)
2. Set final grades for every student in this course (by entering the letter grade in front of “set grade” and clicking “set grade”). If you enter nothing and set grade, then the original grade will be wiped.
3. View all assignments with details.
4. Go to an assignment and set the assignment grade for each student (assignment.php).

assignment.php

An instructor/TA can access the page through course.php. In the assignment.php, an instructor/TA can go to an assignment and set the assignment grade for each student.

This is corresponding to Question 8.

post.php

After clicking into a course, anyone logged in could go to the post page by clicking post on the top left corner.

A user is able to check every post related to this course along with the replies under the post.

A user is able to post anything with a title and tag(s).

A user is able to reply to any existing post by entering in the reply box and click reply.

This is corresponding to Question 9.

***Model.php**

All the php files with name "Model" are where the SQL queries are. We call the functions to acquire result data from the database.

For example, CourseModel extends course.

index.php

Database connection configurations.

7-11.

See the web page and php files.

http://34.68.205.71/login.php