

# CS 564 / Spring 2015: Course Project

## Project Description

Students in teams of 3 to 4 are required to select an application that needs a database and build a database application from start to finish.

## Goals

Finding an application for which database systems would be required  
Modeling the domain of the application, and defining the application functionalities  
Designing and implementing the schema  
Populating the database (this should **not** be the main focus of the project)  
Writing the code needed to embed the database system in the application.

## Schedule

There are a number of intermediate deadlines that you must meet in order to ensure a successful project.

- Fri Feb 13: [Group information due \(Stage 1\)](#)
- Wed Mar 4: [Initial ER design due \(Stage 2\)](#)
- Fri Mar 20: [ER translation due \(Stage 3\)](#)
- Fri Apr 10: [SQL queries \(Stage 4\)](#)
- Fri Apr 24: [Screenshots of the application \(Stage 5\)](#)
- Thur-Fri May 7-8: [Application demo \(Stage 6\)](#)

## Suggested Application Domains (subject to change)

You are free to propose domains that you want to work on, as long as you think the domain that you want to work with will have to manage a lot of data and that this data will best be managed in an RDBMS. You can also pick one of the suggested domains below:

### Movies domain:

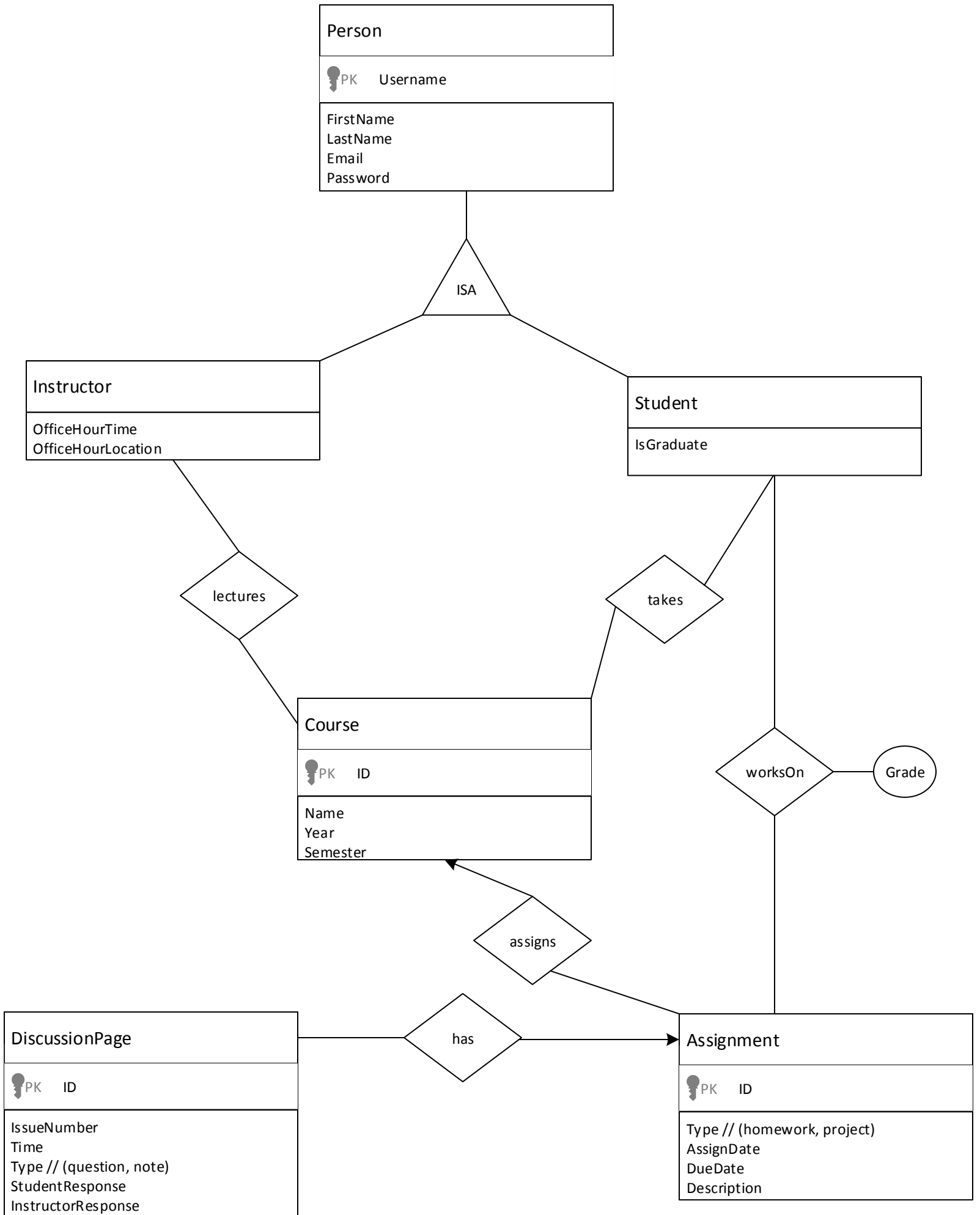
In this domain you would be modeling entities movies, their actors, directors, genres, playing times, reviews, and so on. There exist several sources on the Web from which you can get data to populate such a database. You can support various queries such as finding specific playing times, finding movies playing in Madison, directed by a given director. You can also support updates to the reviews section of the database (e.g., viewers giving their own opinions). Another functionality is to provide personal profiles of people (i.e., the movies they like) and then try to recommend movies to them based on profiles of viewers with similar tastes.

### Books domain:

In this domain you would be modeling entities such as books, their authors, topics (which may be a complex hierarchy). You may also model various attributes of the authors, the institutions they belong to, etc. You can support a buy/sell service of used books, books used in specific university courses. A personal profile, similar to the one for movies is also a possibility. Pointing an interested buyer to a web source to buy a book is also an interesting option.

### Apartments:

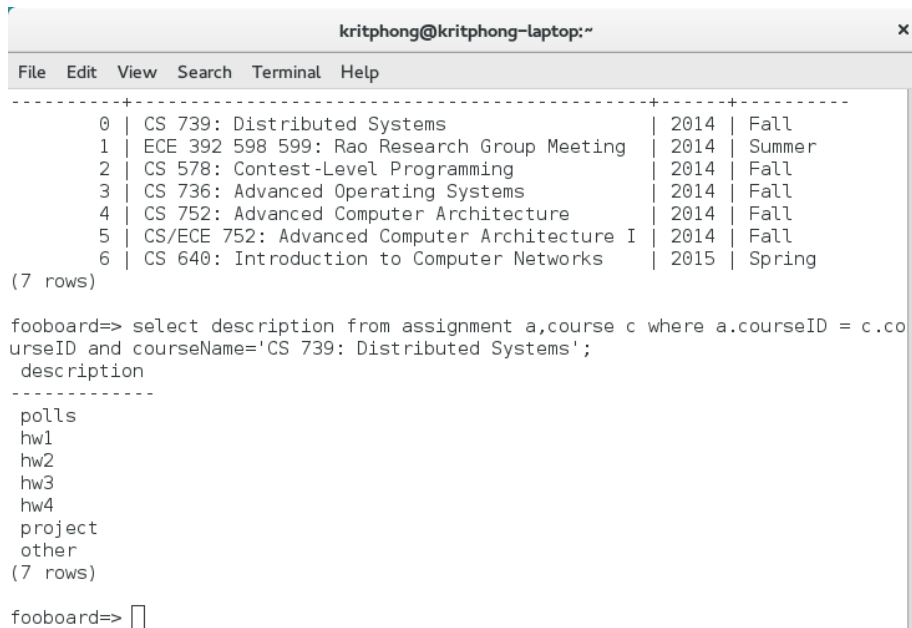
This domain would require modeling apartments and their properties, areas of town and their various properties (e.g., bus lines, crime rate distance from various landmarks). You would provide an interface for offering apartments for rent, finding apartments.



## FooBoard

Kritphong Mongkhonvanit    mongkhonvani@wisc.edu  
Jia Zhang                    jzhang378@wisc.edu  
Kai Da Zhao                 kzha032@cs.wisc.edu  
Tyler Demuth                tmdemuth@wisc.edu

## Stage 5



```
kritphong@kritphong-laptop:~  
File Edit View Search Terminal Help  
-----+-----+-----+-----+-----+-----+-----+-----  
0 | CS 739: Distributed Systems | 2014 | Fall  
1 | ECE 392 598 599: Rao Research Group Meeting | 2014 | Summer  
2 | CS 578: Contest-Level Programming | 2014 | Fall  
3 | CS 736: Advanced Operating Systems | 2014 | Fall  
4 | CS 752: Advanced Computer Architecture | 2014 | Fall  
5 | CS/ECE 752: Advanced Computer Architecture I | 2014 | Fall  
6 | CS 640: Introduction to Computer Networks | 2015 | Spring  
(7 rows)  
  
fooboard=> select description from assignment a,course c where a.courseID = c.courseID and courseName='CS 739: Distributed Systems';  
description  
-----  
polls  
hw1  
hw2  
hw3  
hw4  
project  
other  
(7 rows)  
  
fooboard=> 
```

Figure 1: Show all assignments in course CS 739.

```

kritphong@kritphong-laptop:~$ psql -h boardfoo.chcwfyg4pdzo.us-west-2.rds.amazo
naws.com fooboard -p 5432 -U user1 -W
Password for user user1:
psql (9.4.1)
SSL connection (protocol: TLSv1.2, cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256,
compression: off)
Type "help" for help.

fooboard=> select * from course;
 courseid |          coursename          | year | semester
-----+-----+-----+-----
      0 | CS 739: Distributed Systems  | 2014 | Fall
      1 | ECE 392 598 599: Rao Research Group Meeting | 2014 | Summer
      2 | CS 578: Contest-Level Programming | 2014 | Fall
      3 | CS 736: Advanced Operating Systems | 2014 | Fall
      4 | CS 752: Advanced Computer Architecture | 2014 | Fall
      5 | CS/ECE 752: Advanced Computer Architecture I | 2014 | Fall
      6 | CS 640: Introduction to Computer Networks | 2015 | Spring
(7 rows)

fooboard=> insert into course VALUES(7,'CS 564: Database Management Systems', 20
15, 'Spring');
INSERT 0 1
fooboard=>

```

Figure 2: Course table before inserting CS 564.

```

kritphong@kritphong-laptop:~$ psql -h boardfoo.chcwfyg4pdzo.us-west-2.rds.amazo
naws.com fooboard -p 5432 -U user1 -W
Password for user user1:
psql (9.4.1)
SSL connection (protocol: TLSv1.2, cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256,
compression: off)
Type "help" for help.

fooboard=> insert into course VALUES(7,'CS 564: Database Management Systems', 20
15, 'Spring');
INSERT 0 1
fooboard=> select * from course;
 courseid |          coursename          | year | semester
-----+-----+-----+-----
      0 | CS 739: Distributed Systems  | 2014 | Fall
      1 | ECE 392 598 599: Rao Research Group Meeting | 2014 | Summer
      2 | CS 578: Contest-Level Programming | 2014 | Fall
      3 | CS 736: Advanced Operating Systems | 2014 | Fall
      4 | CS 752: Advanced Computer Architecture | 2014 | Fall
      5 | CS/ECE 752: Advanced Computer Architecture I | 2014 | Fall
      6 | CS 640: Introduction to Computer Networks | 2015 | Spring
      7 | CS 564: Database Management Systems | 2015 | Spring
(8 rows)

fooboard=>

```

Figure 3: Course table after inserting CS 564.

```

kritphong@kritphong-laptop:~
File Edit View Search Terminal Help
  2 | CS 578: Contest-Level Programming | 2014 | Fall
  3 | CS 736: Advanced Operating Systems | 2014 | Fall
  4 | CS 752: Advanced Computer Architecture | 2014 | Fall
  5 | CS/ECE 752: Advanced Computer Architecture I | 2014 | Fall
  6 | CS 640: Introduction to Computer Networks | 2015 | Spring
  7 | CS 564: Database Management Systems | 2015 | Spring
(8 rows)

fooboard=> delete from course where courseName='CS 564: Database Management Systems';
DELETE 1
fooboard=> select * from course;
courseid | courseName | year | semester
-----+-----+-----+-----
  0 | CS 739: Distributed Systems | 2014 | Fall
  1 | ECE 392 598 599: Rao Research Group Meeting | 2014 | Summer
  2 | CS 578: Contest-Level Programming | 2014 | Fall
  3 | CS 736: Advanced Operating Systems | 2014 | Fall
  4 | CS 752: Advanced Computer Architecture | 2014 | Fall
  5 | CS/ECE 752: Advanced Computer Architecture I | 2014 | Fall
  6 | CS 640: Introduction to Computer Networks | 2015 | Spring
(7 rows)

fooboard=> 

```

Figure 4: Course table after deleting CS 564.

```

kritphong@kritphong-laptop:~
File Edit View Search Terminal Help
@wisc.edu | 0
ZachMiller@wisc.edu | 0
SoroushKhaleghi@wisc.edu | 0
NaveenNishanthNagarajan@wisc.edu | 1
(4 rows)

fooboard=> insert into worksOn values('ZachMiller@wisc.edu',1,90);
INSERT 0 1
fooboard=> select * from worksOn;
email | assignmentid | grade
-----+-----+-----
ZachMiller@wisc.edu | 1 | 90
(1 row)

fooboard=> insert into worksOn values('ZachMiller@wisc.edu',2,87);
INSERT 0 1
fooboard=> select * from worksOn;
email | assignmentid | grade
-----+-----+-----
ZachMiller@wisc.edu | 1 | 90
ZachMiller@wisc.edu | 2 | 87
(2 rows)

fooboard=> 

```

Figure 5: WorksOn table before and after inserting grades for a student.

```

kritiphong@kritiphong-laptop:~
File Edit View Search Terminal Help
HarneetSingh@wisc.edu | 4 | 68.21
HarneetSingh@wisc.edu | 5 | 84.6
HarneetSingh@wisc.edu | 6 | 28.88
HarneetSingh@wisc.edu | 7 | 78.05
HarneetSingh@wisc.edu | 8 | 54.26
KaiZhao@wisc.edu | 0 | 70.08
KaiZhao@wisc.edu | 1 | 5.36
KaiZhao@wisc.edu | 2 | 18.78
KaiZhao@wisc.edu | 3 | 61.26
KaiZhao@wisc.edu | 4 | 86.26
KaiZhao@wisc.edu | 5 | 55.77
KaiZhao@wisc.edu | 6 | 26.01
KaiZhao@wisc.edu | 7 | 70.18
KaiZhao@wisc.edu | 8 | 59.29
Jasonferiante@wisc.edu | 0 | 90.89
Jasonferiante@wisc.edu | 1 | 67.98
fooboard=> select avg(grade) from worksOn w, assignment a where w.email = 'KaiZhao@wisc.edu' and w.assignmentId = a.assignmentId and a.courseId = 7;
      avg
-----
37.72222222222222
(1 row)
fooboard=>

```

Figure 6: Querying a student's average grade of a course.

```

kritiphong@kritiphong-laptop:~
File Edit View Search Terminal Help
52 | 7 | project | Fri Apr 17 2015 17:58:06 GMT-0500 (CDT) | project_stage4 | 7 | CS
564: CS 564 Database Management Systems | 2015 | Spring
53 | 7 | project | Fri Apr 24 2015 12:33:40 GMT-0500 (CDT) | project_stage51 | 7 | CS
564: CS 564 Database Management Systems | 2015 | Spring
54 | 7 | logistics | Mon Feb 23 2015 13:47:27 GMT-0600 (CST) | logistics | 7 | CS
fooboard=> select duedate from assignment a, course c where c.courseName='CS 564: CS 564 Database Management Systems' and a.courseId = c.courseId;
      duedate
-----
Tue Jan 20 2015 18:25:29 GMT-0600 (CST)
Fri Feb 27 2015 17:23:53 GMT-0600 (CST)
Thu Mar 19 2015 13:53:20 GMT-0500 (CDT)
Fri Apr 10 2015 22:43:04 GMT-0500 (CDT)
Fri Apr 10 2015 16:45:06 GMT-0500 (CDT)
Tue Jan 20 2015 18:25:29 GMT-0600 (CST)
Mon Jan 26 2015 22:45:09 GMT-0600 (CST)
Fri Mar 13 2015 00:52:42 GMT-0500 (CDT)
Mon Feb 02 2015 11:30:18 GMT-0600 (CST)
(9 rows)
fooboard=>

```

Figure 7: Querying all the due dates of assignments of CS 564.