

# BLG361E – DATABASE MANAGEMENT SYSTEMS

2015-2016 FALL

	<b>CRN: 11768</b>	<b>CRN: 11767</b>
<b>Instructors:</b>	Assoc. Prof. Dr. Şule Öğüdücü	Lec. H. Turgut Uyar
<b>Office:</b>	EEB 3305	EEB 2422
<b>Office Hours:</b>	<i>send e-mail for appointment</i>	Mon 16:30-17:30 Wed 13:30-14:30
<b>Assistants:</b>	Nagehan İlhan, Tahir Sandıkkaya Gönül Uludağ, Emrullah Gazioğlu	

**Textbook:** Chris J. Date, "An Introduction to Database Systems", Addison-Wesley, ISBN 0-321-19784-4, 2004.

## Weekly Plan

Week	Date	Lecture	Exercise
1	16 Sep. 2015	Introduction	Python
	23 Sep. 2015	HOLIDAY	
2	30 Sep. 2015	Relational Model	Flask
3	7 Oct. 2015	Relational Model - SQL	Flask
4	14 Oct. 2015	Relational Algebra	Rel
5	21 Oct. 2015	Relational Algebra - SQL	SQL
6	28 Oct. 2015	Application Development	Flask – SQL, SQLAlchemy
7	4 Nov. 2015	Database Design	Normalization, E/R Diagrams
8	11 Nov. 2015	Concurrency	Concurrency
9	18 Nov. 2015	Midterm Exam	
10	25 Nov. 2015	NoSQL	MongoDB, Neo4J (Cypher)
11	2 Dec. 2015	NoSQL - XML	XQuery, BaseX
12	9 Dec. 2015	Query Optimization	Query Optimization
13	16 Dec. 2015	Project Presentations and Demos	
14	23 Dec. 2015	Project Presentations and Demos	

## Grading

- midterm exam: 25%, project: 30%, final exam: 45%
- conditions for participation at the final exam:
  - at least 70% attendance (30/42 hours)
  - project grade at least 30/100
- students with overall average under 40/100 will fail

## Important Notes

- **Ninova:** You have to follow the course announcements, your exam results and attendance status on the Ninova system (<http://ninova.itu.edu.tr/>). Check the Ninova site regularly for updates.
- **ITU Email Accounts:** Course related e-mail notifications will be sent to your ITU account; check it regularly.
- **Email Etiquette:** When sending e-mail to the instructors or assistants, use your ITU account and always include your full name at the end of the message.
- **Classroom Announcements:** You are expected to know everything that was announced in the classroom. There does not necessarily have to be an announcement on Ninova. If you have missed a class, ask your classmates if there were any announcements that you need to know.
- **Attendance:** Attendance will be checked on all three hours of the sessions and announced as "-", "1", "2", or "3" on Ninova.
- **Plagiarism:** Any form of cheating or plagiarism will not be tolerated. This includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources (including Internet pages, even if attributed). Serious offenses will be reported to the faculty administration for disciplinary measures.

## PROJECT

Students will form teams of 5 members each and develop a web application about a branch of sports. Every team member will be responsible for implementing the database and interface operations of at least three entities. An entity can be thought of as being represented as a table in the database which is referenced by at least one other table.

The technologies that have to be used are the Python (version 3.4 or later) programming language (<https://www.python.org>), the Flask web framework (<http://flask.pocoo.org/>), and an SQL database. The SQL database must be accessed via a dbapi2 compatible driver. Using object-relational mappers or any abstraction layer over the SQL language is not allowed. The documentation will be prepared using Sphinx (<http://sphinx-doc.org/>).

At the end of the term, all teams will submit the following items:

- source code (zip)
- report sources (zip)
- printable report (pdf)
- presentation (pptx or pdf)

The Git version control system will be used for coordinating team work and evaluating the progress of team members. The projects will be hosted on GitHub (<https://github.com/>), and deployed to IBM Bluemix DevOps Services (<https://hub.jazz.net/>). The documentation will also be hosted on GitHub and it will be publicly available on the Read The Docs site (<http://readthedocs.org/>).

## Schedule

*Note:* The deadline for each work item is the end of the working hours before the class day, i.e. Tuesday 17:30. So, for example "by the 2nd week" means "by 29 Sep. 2015, 17:30".

Each team proposes a project which will be checked and approved by the instructors. The proposal forms are submitted to the Ninova system by the 2nd week. The proposals can include 1 to 5 members; students from different sections can be in the same team. Every

student must submit the proposal document, even though the forms for all members of a team will be the same. Due to resource limitations, the instructors might decide to reorganize the teams. The number of teams working on the same branch of sports will be limited; if the number of teams proposing to work in one area of the arts is too large, the better proposal(s) will be chosen and the other teams will be assigned to another branch of sports.

Project teams and topics are finalized and announced on the 3rd week by the instructors. The repository setup and initial deployment must be finished by the 4th week.

Each team member completes the basic add-delete operations of at least one entity by the 7th week.

Each team member completes the add-delete-update-find operations of at least two entities by the 10th week.

All teams submit their code and presentation documents to the Ninova system at the beginning (Monday 10:00) of the 13th week.

All teams give a hands-on demo to the responsible TA according to the announced schedule on the 13th and 14th weeks. Some (if possible, all) of the teams present their projects to the class. The teams that are not selected to give a presentation in the classroom also have to attend these sessions.

All teams upload the source files and the printable version of their reports on the 14th week to the Ninova system. The report consists of an installation guide, a user manual, and a technical manual. Every student gives detailed information about the objects that she/he is responsible for. A printed team report will also be handed in at the designated report box at the department secretarial office.

## **Grading**

Each student will get grades for coding, report, and presentation. Grades will be weighted as 55% for coding, 30% for report, and 15% for presentation.

Each student's coding grade will be determined by her/his project work and by her/his performance at the demo session. The demo session has a strict "no show" policy and absent group members will get 0 for the coding grade. The intermediate progress checks on weeks 7 and 10 will be coefficients for the final coding grade. These coefficients will be between 0.7 and 1. For example, a student who shows no progress up until the check on the 10th week will get the coefficients 0.7 and 0.7 and therefore the final coding grade will be multiplied by 0.49. Note that it's not just the amount of committed code that contributes to the progress evaluation:

- Commits that do not provide enough information in the commit message are ignored. For example, commits with no commit message, or commits with unknown username. Make sure that your username shows up correctly on the commit logs.
- Commits that obviously break the code contribute no marks. For example, commits that contain conflicts. You can spot conflicts by the lines that start with ">>>".

The report grade has an upper limit with respect to the coding grade, i.e. it cannot be much higher than the coding grade.

The teams that are selected for the presentation will be graded according to their presentation performance in addition to the presentation document. The teams that do not present their project, or the students who do not attend the presentation session, will get 0 for their presentation grade.