

# 15-415: Database Applications

School of Computer Science  
Carnegie Mellon University, Qatar  
Spring 2018

## 1 Overview

**Title:** Database Applications

**Units:** 12 units

**Pre-requisites:** Grades of “C” or better in 15-121 (i.e., Introduction to Data Structures) and 15-213 (i.e., Introduction to Computer Systems)

**Lectures:** Sunday and Tuesday, 1:30 – 2:50 PM, Room 2147

**Recitations:** Wednesday, 4:30 – 5:20 PM, Room 2147

**Webpage:** <http://www.qatar.cmu.edu/~mhhammou/15415-s18/>

### Course Description:

This course presents an in-depth treatment of database management systems (DBMSs), with an emphasis on how to *design, create, refine* and efficiently *use a relational* database as well as *build* and *optimize* internals of DBMSs. As such, the course offers a combination of application-centric and systems-centric discussions on classical and modern DBMSs, with a focus on relational databases. Specifically, we will first discuss the Entity-Relationship and the Relational models. Second, we will cover relational algebra and calculus as a foundation for relational query languages. Third, we will study the commercial Structured Query Language (SQL) which allows creating, manipulating and querying relational databases. Fourth, we will show how users can connect to a DBMS and execute SQL queries from within high-level programming languages using Django. Consequently, students will have the chance to develop full-fledged applications using a three-tier architecture, which encompasses a front-end web-based or standalone GUI tier, a middle logic-processing tier and a DBMS back-end tier.

Afterwards, we will initiate the discussion on the *internals* of DBMSs. In particular, we will first study hash-based (e.g., extendible and linear) and tree-based (e.g., ISAM and B+ trees) indexing schemes, which are crucial for expediting query processing in relational databases. Other DBMS internals like disk space and buffer managers will be examined as well. Second, we will explain how relational operators and query plans can be implemented, evaluated and optimized. Third, we will elaborate on *transaction management* which forms the underpinning for concurrent execution and failure recovery in DBMSs. Specifically, we will define transactions, cover its Atomic, Consistency, Isolation and Durability (or in short, *ACID*) properties, and demonstrate how DBMSs can ensure such properties. Finally, we will conclude our discussion with other types of databases like NoSQL databases (e.g., the Google’s BigTable) and Cassandra.

To this end, we note that students will be given an intensive hands-on experience through three large programming projects. Particularly, students will develop multiple applications that interface with a DBMS (treating it as a “black box”), and write code that implements some internal modules of relational DBMSs (e.g., B+ tree).

## Instructors:

Mohammad Hammoud  
[mhhammou@gatar.cmu.edu](mailto:mhhammou@gatar.cmu.edu),  
Room 1006, 4454-8506,  
Office Hours: Wednesday, 10:30AM – 12:00PM.

Yousuf Ahmad  
[myahmad@gatar.cmu.edu](mailto:myahmad@gatar.cmu.edu),  
Room 1014, 4454-2826,  
Office Hours: Sunday: 3:00PM – 4:30PM.

## Teaching Assistant:

Tamim Jabban  
[tamim@cmu.edu](mailto:tamim@cmu.edu),  
Room 1004, 4454-8496,  
Office Hours: Monday: 10:30AM – 12:00PM.

## 2 Objectives

Database management continues to gain importance as data is growing exponentially and made ever more accessible. As pointed out previously, this course aims at presenting an in-depth treatment of database management systems (DBMSs) with an emphasis on relational databases. For that sake, our objective is two-fold, application-oriented and systems-oriented. In particular, we intend to focus on the usage of databases via explaining: (1) how to design and implement a database from 'cradle-to-grave' for real-world enterprises using the entity-relationship and the relational models, (2) how to query and manipulate a database using SQL, and (3) how to refine and speed up data manipulation and queries using auxiliary data structures called indexes. From a systems-oriented perspective, our goal is to concentrate on some internal modules of DBMSs and discuss the details of constructing: (1) buffer and disk space managers, (2) query optimizers, (3) and concurrency and crash recovery managers for a DBMS.

Although relational DBMSs are still the dominant type of databases nowadays, new types of storage layers like NoSQL are emerging and attracting a great deal of attention, especially for large-scale datasets (i.e., Big Data). Hence, we further seek to expose students to the most recent visions that drive the databases field as well as some of the resultant and modern DBMSs. Accordingly, we will discuss Big Data and its ramifications on designing database systems, and present one popular analytics engine for processing Big Data (i.e., Hadoop MapReduce). Furthermore, we will examine the Google's BigTable as a practical and real-world NoSQL database. To this end, the course can prove to be richly rewarding in more than one way!

## 3 Learning Outcomes

This course incorporates *thirteen* major Learning Outcomes. In particular, after finishing this course, students will be able to:

1. Describe a wide range of data involved in real-world organizations using the entity-relationship (ER) data model.

2. Apply the relational model, specify integrity constraints, and explain how to create a relational database using an ER diagram.
3. Analyze and apply two formal query languages, relational calculus and algebra, associated with the relational model.
4. Indicate how Structured Query Language (SQL) builds upon relational calculus and algebra and effectively apply SQL to create, query and manipulate relational databases.
5. Design and develop multi-tiered, full-fledged standalone and web-based applications using back-end databases and SQL integrated within general-purpose programming languages.
6. Employ Object-Relational Mapping (ORM) frameworks (e.g., Django) to design and develop an application's business logic using high-level object-oriented programming constructs, which obviate the need for manual SQL programming.
7. Appreciate how database management systems create, manipulate and manage files of fixed-length and variable-length records on disks.
8. Compare, contrast, create and operate various static and dynamic tree-based (e.g., ISAM and B+ trees) and hash-based (e.g., extendable and linear hashing) indexing schemes using an I/O cost model for analyzing the suitability of such schemes for different kinds of workloads.
9. Explain and evaluate various algorithms for relational operations (e.g., join) using techniques such as iteration, indexing and partitioning.
10. Analyze and apply different query evaluation plans and describe the various tasks of a typical relational query optimizer, including translating SQL queries into relational algebra and estimating costs of alternative query plans, among others.
11. Discuss the concepts, motivations and properties of transactions in databases, describe how they can be interleaved and managed *correctly*, and indicate how a DBMS can ensure atomicity and durability when systems fail or entirely crash.
12. Identify alternative architectures for parallel and distributed databases, describe how data can be partitioned and distributed across networked nodes of a DBMS, and suggest how queries and segmented data can be optimized and managed in a distributed environment.
13. Appreciate the scale of Big Data, discuss some popular analytics engines for Big Data processing and denote the applicability of NoSQL databases for Big Data storage.

### 3 Course Textbook

Raghu Ramakrishnan and Johannes Gehrke, "**Database Management Systems**", Third Edition, McGraw-Hill, 2002.

### 4 Course Organization

The participation of students in the course will involve five forms of activities:

- Attending lectures and recitations.
- Solving assignments (involving writing and/or coding).
- Solving large programming projects.
- Taking exams and quizzes.
- Participating in class discussions.

## 5 Assessment

Each student will receive a numeric score with a corresponding letter grade, based on a weighted average of the following:

1. **Projects:** The projects will count for a total of 40% of your final score. There will be 3 projects throughout the course. All projects are *individual* projects (i.e., no teamwork is allowed). The projects are worth 10%, 15% and 15% respectively.

You are encouraged to submit the projects on time. For all projects except the final one, the following rules apply. If you submit one day late, we will deduct 25% of the project score as a penalty. If you submit two days late, 50% will be deducted. The project will not be graded (and you will receive a zero score on it) if you are more than two days late. However, there is a **grace-days quota** for projects. In particular, you will be given **3 grace days** for all projects, except for the final one. You can use the grace days as needed. For instance, you can submit your first project three days late and still not receive any penalty. In this case, you will be penalized starting from the 4<sup>th</sup> day after the deadline. Be aware, however, that when you entirely consume your grace-days quota, you will be left with no grace days for the remaining projects.

Note that the final project is unique in two aspects. First, you cannot use grace days for it. Hence, if you are left with some grace days before the final project, you will lose them all. As such, plan wisely for how to utilize your grace-days quota. Second, there will not be any penalty system for this project either. That is, if you are one day late in submitting the project, it will not be graded and you will receive a zero score on it.

2. **Exams:** There will be two in-class exams – midterm and final – which combined will account for 30% of your final score. The midterm is worth 10% and the final is worth 20%.
3. **Problem Sets:** There will be 5 assignments that will test you on some problem analysis and solving skills. At the end of the semester, the assignment bearing the lowest score will be dropped. The 4 assignments will altogether contribute 15% towards your final score. It follows that each assignment is worth 3.75%.
4. **Quizzes:** There will be 2 quizzes, which together account for 10% of your final score. These quizzes are meant to test your understanding and preparation for the concepts covered throughout the course.
5. **Class/Recitation Participation and Attendance:** Your attendance of both, classes and recitations, as well as your participation in discussions during presentations will count for 5% of your final score.

Table 1 below shows the breakdown of the five forms of activities that the course involves, alongside the quantity and the overall weight of each activity. Take into account that small differences in scores can make the difference between two letter grades. Letter grades will be determined by absolute standards. The total score will be plotted as a histogram. Cutoff points are determined by examining the quality of students' work on the borderlines. Individual cases, especially those near the cutoff points may be adjusted upward or downward based on factors such as attendance, class participation, improvement observed throughout the course, exam performance, and special circumstances.

Type	#	Weight
Projects	3	40%
Exams	2	30%
Problem Sets	5	15%
Quizzes	2	10%
Class/Recitation Participation and Attendance	41	5%

**Table 1:** Breakdown of the Five Activities Involved in the Course.

## 6

## 7 Getting Help

For urgent communication with the instructor and the teaching assistant, it is best to send an email (preferred) or give a phone call. If you want to talk to any of them in person, remember that their posted office hours are merely nominal times when they guarantee that they will be in their offices. You are always welcome to visit them outside of their office hours if you need help or want to talk about the course.

We ask that you follow a few simple guidelines. The instructor normally works with his office door being open. Whenever the office door is open, he welcomes visits from students. However, if his office door is closed, this means that he is busy with meetings or phone calls, thus prefers not to be disturbed.

We will use the course webpage as the central repository for all information about the class. Through the webpage, you can:

1. Obtain copies of any handouts or assignments. This is especially useful if you miss a class or lose a document.
2. View announcements that relate to the course.
3. Find links to any electronic data you need for your assignments.
4. Read clarifications and changes made to any assignments, schedules, or policies, if necessary.
5. Provide healthy feedback about the course.

Lastly, you can use **Piazza** for asking questions and receiving answers without using emails! Posting your questions on Piazza will help the whole class benefit, and will certainly avoid redundancy. Find our class Piazza page at:

<https://piazza.com/qatar.cmu/spring2018/15415/home>

## 8 Policies

### **Working Alone on Assignments/Projects**

Assignments/projects that are assigned to students should be performed individually. This course does not include any team project or assignment.

### **Handing in Assignments/Projects**

All assignments/projects are due at 11:59PM (one minute before midnight) on the specified due date. All hand-ins are electronic and should be submitted using the AFS file system: `/afs/qatar.cmu.edu/usr10/mhhammou/www/15415-s18/handin/userid/`, where *userid* is your Andrew user id.

### **Making up Exams, Assignments and Projects**

Missed exams, assignments and projects can be made up on a case by case basis, but only if you make prior arrangements with the instructors. However, you should have a good reason for doing so. You need a written consent from the instructors for making up exams, assignments or projects. It is your responsibility to get your projects and assignments done on time. Be sure to work far enough in advance to avoid unexpected problems, such as illness, unreliable or overloaded computer systems, etc.

### **Appealing Grades**

After each exam, assignment, and/or project is graded, you have **7** calendar days to appeal your grade. All your appeals should be provided in writing. If after appealing you are still not satisfied, please visit the instructors. If you have questions about an exam, an assignment or a project grade, please visit the instructors directly.

## 9 Cheating

Each project or assignment must be the sole work of the student turning it in. Projects and assignments will be closely monitored, and students may be asked to explain suspicious similarities with any write-up or piece of code available. The following are guidelines on what cheating is and is not:

### **What is cheating?**

1. Sharing code or other electronic files: either by copying, retyping, looking at, or supplying a copy of a file.
2. Sharing written assignments: either by re-writing, looking at, or supplying a copy of an assignment.

### **What is NOT cheating?**

1. Clarifying ambiguities or vague points in class handouts.
2. Helping others use the computer systems, networks, compilers, debuggers, profilers, or other system facilities.
3. Helping others with high-level design issues.
4. Helping others debug their codes.

Consequently, be aware of what constitutes cheating (and what does not) when interacting with your colleague students. Same rules of cheating as above apply when collaborating with other students. In short, you cannot share written assignments, code, and/or other electronic files with other students. If you are unsure, ask the teaching staff.

Finally, be sure to store your work in protected directories. The penalty for cheating is severe, and might jeopardize your whole career as a student – cheating is not worth the trouble. By cheating in the course, you are cheating yourself; the worst outcome of cheating is missing an opportunity to learn. Besides, you will be removed from the course and assigned a failing grade. We also place a record of the incident in your permanent university profile.

## 10 Health & Wellness

### 10.1 Learning Disabilities

Carnegie Mellon University is committed to providing reasonable accommodations for all persons with disabilities. To access accommodation services you are expected to initiate the request and submit a Voluntary Disclosure of Disability Form to the office of Health & Wellness or CaPS-Q. In order to receive services/accommodations, verification of a disability is required as recommended in writing by a doctor, licensed psychologist or psycho-educational specialist. The office of Health & Wellness, CaPS-Q and Office of Disability Resources in Pittsburgh will review the information you provide. All information will be considered confidential and only released to appropriate persons on a need to know basis.

Once the accommodations have been approved, you will be issued a Summary of Accommodations Memorandum documenting the disability and describing the accommodation. You are responsible for providing the Memorandum to your professors at the beginning of each semester.

For more information on policies and procedures, please visit <https://scotty.qatar.cmu.edu/qword/student-affairs/office-of-health-and-wellness/assistance-for-individuals-with-disabilities/>

### 10.2 Taking Care of Yourself

Do your best to maintain a healthy lifestyle this semester by eating well, exercising, getting enough sleep, and taking some time to relax. This will help you achieve your goals and cope with stress.

All of us benefit from support during times of struggle. You are not alone. There are many helpful resources available on campus and an important part of the college experience is learning how to ask for help. Asking for support sooner rather than later is often helpful.

If you or anyone you know experiences any academic stress, difficult life events, or feelings like anxiety or depression, we strongly encourage you to seek support. Counseling and Psychological Services (CaPS-Q) is here to help: call 4454 8525 or make an appointment to see the counselor by emailing [student-counselling@qatar.cmu.edu](mailto:student-counselling@qatar.cmu.edu). Consider reaching out to a friend, faculty, or family member you trust for help.

If you or someone you know is feeling suicidal or in danger of self-harm, call someone immediately, day or night, at 5554-7913. **If the situation is life threatening, call 999.**

# 11 Class Schedule

Table 2 demonstrates the tentative schedule of the class. The schedule indicates the project and the assignment activities as well. Any changes will be announced and reflected on the course webpage. An updated schedule will be always maintained on the course webpage.

Week	Ses sion	Date	Topic	Teaching Method	Reading	Projects	Prob. Solving Assignments
1	1	7 Jan	Administrivia and Introduction	Lecture	Syllabus, R&G C1		
	2	9 Jan	The Entity-Relationship Model	Lecture	R&G C2		Start PS1
	3	11 Jan	Case study on Entity Relationship Diagram	Recitation	Notes from TA		
2	4	14 Jan	The Relational Model	Lecture	R&G C3		
	5	16 Jan	The Relational Model (Continued)	Lecture	R&G C3		
	6	18 Jan	Case study on Relational Algebra	Recitation	Notes from TA		
3	7	21 Jan	Relational Algebra	Lecture	R&G C4.2		End PS1
	8	23 Jan	Relational Calculus	Lecture	R&G C4.3		Start PS2
	9	25 Jan	Hands-on Relational Algebra & TRC Expressions	Recitation	Notes from TA		
4	10	28 Jan	The SQL Query Language (Part I)	Lecture	R&G C5	Start P1	
	11	30 Jan	The SQL Query Language (Part II)				
	12	1 Feb	Hands-on PostgreSQL	Recitation	Notes from TA		
5	13	4 Feb	The SQL Query Language (Part III)	Lecture	R&G C5		End PS2
	14	6 Feb	ORM	Lecture	Notes from the Instructor		
	15	8 Feb	Quiz I	N/A	N/A		
6	16	11 Feb	Storing Data - Disks, Buffers, and Files	Lecture	R&G C9		
	-	13 Feb	<b>Sports Day</b>	<b>No Class</b>			
	17	15 Feb	Hands-on Django	Recitation	Notes from TA	End P1 Start P2	
7	18	18 Feb	File Organizations and Indexing & Midterm Overview	Lecture	Notes from the Instructor		
	19	20 Feb	<b>Midterm</b>	<b>Exam 1</b>			
	20	22 Feb	Overview of P2	Recitation	Notes from TA		
8	21	25 Feb	Tree-Based Indexing Schemes	Lecture	R&G C10		Start PS3
	22	27 Feb	Tree-Based Indexing Schemes (Part II)	Lecture	R&G C10		
	23	01 Mar	More on P2 and Django	Recitation	Notes from TA		
<b>Spring Break March 4 - 9</b>							
9	24	11 Mar	Hash-Based Indexing Schemes	Lecture	R&G C11		
	25	13 Mar	A Brief Primer on Query Evaluation and External Sorting	Lecture	R&G C12 & C13		
	26	15 Mar	Case study on Query Plans and Cost Estimation + Overview of P3	Recitation	Notes from TA		
10	27	18 Mar	External Sorting (Continued)	Lecture	R&G C12 & C13		End PS3
	28	20 Mar	P2 Student Presentations	N/A	N/A		

	29	22 Mar	More on P3 - Understanding B+ Trees	Recitation	Notes from TA	End P2 Start P3	
11	30	25 Mar	Relational Operators (Part I)	Lecture	R&G C12 & C14		Start PS4
	31	27 Mar	Relational Operators (Part II)	Lecture	R&G C12 & C14		
	32	29 Mar	Query Optimization	Lecture	R&G C15		
12	33	1 Apr	Query Optimization (Continued)	Lecture	R&G C15		
	34	3 Apr	Concurrency Control (Part I)	Lecture	R&G C16		
	35	5 Apr	Quiz II	N/A	N/A		
13	36	8 Apr	Concurrency Control (Part II)	Lecture	R&G C16		End PS4
	37	10 Apr	Logging and Recovery (Part I)	Lecture	R&G C16		Start PS5
	38	12 Apr	Logging and Recovery (Part II)	Lecture	R&G C18		
14	39	15 Apr	NoSQL Databases & Hadoop	Lecture	R&G C18	End P3	
	40	17 Apr	NoSQL Databases & Hadoop	Lecture	Notes from Instructor		
	41	19 Apr	Final Exam Review	Lecture	Notes from the Instructor		End PS5
	42	TBA	Final	Exam 2			

**Table 2:** Tentative Time-Line of the Course.

**Notations:**

- **PS:** Problem Set
- **R&G Cx.y.z:** Chapter x (section y, sub-section z) from the course textbook
- **TBA:** To Be Announced