# University of Engineering and Technology
# School of Computer Science
# Syllabus of Course
# Academic Period 2018-II

1. **Code and Name:** CS2701. Databases I
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HL;
4. **Professor(s)**

**Lecturer**

- Heider Sanchez
  - PhD in Computer Science, UChile, Chile, 2017.
- Teófilo Chambilla Aquino
  - MSc in Computer Science, Universidad de Chile, Chile, 2016.

Meetings after coordination with the professor

5. **Bibliography**


[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

[Dat05] C.J. Date. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier, 2005.

[Die01] Suzanne W Dietrich. *Understanding Relational Database Query Languages, First Edition*. Prentice Hall, 2001.

[EN04] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.

[Har02] Jan L. Harrington. *Relational Database Design Clearly Explained, Second Edition*. Morgan Kaufmann, 2002.

[KS02] Henry F. Korth and Abraham Silberschatz. *Fundamentos de Base de Datos*. McGraw-Hill, 2002.

[RC04] Peter Rob and Carlos Coronel. *Database Systems: Design, Implementation and Management, Sixth Edition*. Morgan Kaufmann, 2004.

[SW04] Graeme Simsion and Graham Witt. *Data Modeling Essentials, Third Edition*. Morgan Kaufmann, 2004.

[WM01] Mark Whitehorn and Bill Marklyn. *Inside Relational Databases, Second Edition*. Springer, 2001.


6. **Information about the course**

(a) **Brief description about the course** Information management (IM) plays a major role in almost all areas where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of accessing and updating stored information, data modeling and abstraction, and physical file storage techniques.It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which (IM) methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable restrictions, including Scalability and usability.

(b) **Prerrequisites:**

- CS1102. Objects oriented programming I. ($2^{nd}$ Sem)
- CS1D02. Discrete Structures II. ($2^{nd}$ Sem)

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

**7. Specific goals of the Course**

- That the student learn to represent information in a database prioritizing the efficiency in the recovery of the same.
- That the student learn the fundamental concepts of the management of databases. This includes the design of databases, database languages and the realization of databases.
- Discuss the database model with the base in relational algebra, relational calculus and the study of SQL statements.

**8. Contribution to Outcomes**

**b)** An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

**d)** An ability to function on multidisciplinary teams. (**Usage**)

**i)** An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

**j)** Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

**b)** An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

**d)** An ability to function on multidisciplinary teams. (**Usage**)

**i)** An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

**j)** Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

**9. Competences (IEEE)**

**C1.** An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

**C2.** Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome d**

**C7.** Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome i**

**CS4.** Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome i**

**CS5.** Specify, design, and implement computer-based systems.⇒ **Outcome j**

**C1.** An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome b**

**C2.** Ability to have a critical and creative perspective in identifying and solving problems using computational thinking. ⇒ **Outcome d**

**C7.** Being able to apply the software engineering principles and technologies to ensure that software implementations are robust, reliable, and appropriate for their intended audience.⇒ **Outcome i**

**CS4.** Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.⇒ **Outcome i**

**CS5.** Specify, design, and implement computer-based systems.⇒ **Outcome j**

**10. List of topics**

1. Database Systems
2. Data Modeling
3. Indexing

4. Relational Databases
5. Query Languages
6. Relational Databases

**11. Methodology and Evaluation**
**Methodology**:

**Theory Sessions:**
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

**Lab Sessions:**
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

**Oral Presentations:**
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

**Reading:**
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.
**Evaluation System:**

**12. Content**

| **Unit 1: Database Systems (14)** | |
|---|---|
| **Competences Expected: C1,C7,CS4,CS5** | |
| **Learning Outcomes** | **Topics** |
| • Explain the characteristics that distinguish the database approach from the approach of programming with data files [Usage]<br>• Describe the most common designs for core database system components including the query optimizer, query executor, storage manager, access methods, and transaction processor [Usage]<br>• Cite the basic goals, functions, and models of database systems [Usage]<br>• Describe the components of a database system and give examples of their use [Usage]<br>• Identify major DBMS functions and describe their role in a database system [Usage]<br>• Explain the concept of data independence and its importance in a database system [Usage]<br>• Use a declarative query language to elicit information from a database [Usage]<br>• Describe facilities that datatbases provide supporting structures and/or stream (sequence) data, eg, text [Usage]<br>• Describe major approaches to storing and processing large volumes of data [Usage] | • Approaches to and evolution of database systems<br>• Components of database systems<br>• Design of core DBMS functions (e.g., query mechanisms, transaction management, buffer management, access methods)<br>• Database architecture and data independence<br>• Use of a declarative query language<br>• Systems supporting structured and/or stream content<br>• Approaches for managing large volumes of data (e.g., noSQL database systems, use of MapReduce). |
| **Readings :** [RC04], [EN04], [Dat05], [KS02] | |

| **Unit 2: Data Modeling (14)** | |
|---|---|
| **Competences Expected: C1,C2,C7,CS4,CS5** | |
| **Learning Outcomes** | **Topics** |
| • Compare and contrast appropriate data models, including internal structures, for different types of data [Usage]<br>• Describe concepts in modeling notation (eg, Entity-Relation Diagrams or UML) and how they would be used [Usage]<br>• Define the fundamental terminology used in the relational data model [Usage]<br>• Describe the basic principles of the relational data model [Usage]<br>• Apply the modeling concepts and notation of the relational data model [Usage]<br>• Describe the main concepts of the OO model such as object identity, type constructors, encapsulation, inheritance, polymorphism, and versioning [Usage]<br>• Describe the differences between relational and semi-structured data models [Usage]<br>• Give a semi-structured equivalent (eg, in DTD or XML Schema) for a given relational schema [Usage] | • Data modeling<br>• Conceptual models (e.g., entity-relationship, UML diagrams)<br>• Spreadsheet models<br>• Relational data models<br>• Object-oriented models<br>• Semi-structured data model (expressed using DTD or XML Schema, for example) |
| **Readings :** [SW04], [EN04], [KS02] | |

| **Unit 3: Indexing (4)** | |
|---|---|
| **Competences Expected: CS4,CS5** | |
| **Learning Outcomes** | **Topics** |
| • Generate an index file for a collection of resources [Usage]<br>• Explain the role of an inverted index in locating a document in a collection [Usage]<br>• Explain how stemming and stop words affect indexing [Usage]<br>• Identify appropriate indices for given relational schema and query set [Usage]<br>• Estimate time to retrieve information, when indices are used compared to when they are not used [Usage]<br>• Describe key challenges in web crawling, eg, detecting duplicate documents, determining the crawling frontier [Usage] | • The impact of indices on query performance<br>• The basic structure of an index<br>• Keeping a buffer of data in memory<br>• Creating indexes with SQL<br>• Indexing text<br>• Indexing the web (e.g., web crawling) |
| **Readings :** [WM01], [Dat05], [KS02] | |

**Unit 4: Relational Databases (14)**

**Competences Expected: 5**

| Learning Outcomes | Topics |
|---|---|
| • Prepare a relational schema from a conceptual model developed using the entity- relationship model [Usage]<br>• Explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint (including definition of the concept of a foreign key) [Usage]<br>• Demonstrate use of the relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) and the relational algebra operations developed specifically for relational databases (select (restrict), project, join, and division) [Usage]<br>• Write queries in the relational algebra [Usage]<br>• Write queries in the tuple relational calculus [Usage]<br>• Determine the functional dependency between two or more attributes that are a subset of a relation [Usage]<br>• Connect constraints expressed as primary key and foreign key, with functional dependencies [Usage]<br>• Compute the closure of a set of attributes under given functional dependencies [Usage]<br>• Determine whether a set of attributes form a superkey and/or candidate key for a relation with given functional dependencies [Usage]<br>• Evaluate a proposed decomposition, to say whether it has lossless-join and dependency-preservation [Usage]<br>• Describe the properties of BCNF, PJNF, 5NF [Usage]<br>• Explain the impact of normalization on the efficiency of database operations especially query optimization [Usage]<br>• Describe what is a multi-valued dependency and what type of constraints it specifies [Usage] | • Mapping conceptual schema to a relational schema<br>• Entity and referential integrity<br>• Relational algebra and relational calculus<br>• Relational Database design<br>• Functional dependency<br>• Decomposition of a schema; lossless-join and dependency-preservation properties of a decomposition<br>• Candidate keys, superkeys, and closure of a set of attributes<br>• Normal forms (BCNF)<br>• Multi-valued dependency (4NF)<br>• Join dependency (PJNF, 5NF)<br>• Representation theory |

**Readings :** [WM01], [Dat05], [KS02]

| **Unit 5: Query Languages (12)** | |
|---|---|
| **Competences Expected: C1,CS4,CS5** | |
| **Learning Outcomes** | **Topics** |
| • Create a relational database schema in SQL that incorporates key, entity integrity, and referential integrity constraints [Usage]<br>• Use SQL to create tables and retrieve (SELECT) information from a database [Usage]<br>• Evaluate a set of query processing strategies and select the optimal strategy [Usage]<br>• Create a non-procedural query by filling in templates of relations to construct an example of the desired query result [Usage]<br>• Embed object-oriented queries into a stand-alone language such as C++ or Java (eg, SELECT ColMethod() FROM Object) [Usage]<br>• Write a stored procedure that deals with parameters and has some control flow, to provide a given functionality [Usage] | • Overview of database languages<br>• SQL (data definition, query formulation, update sublanguage, constraints, integrity)<br>• Selections<br>• Projections<br>• Select-project-join<br>• Aggregates and group-by<br>• Subqueries<br>• QBE and 4th-generation environments<br>• Different ways to invoke non-procedural queries in conventional languages<br>• Introduction to other major query languages (e.g., XPATH, SPARQL)<br>• Stored procedures |
| **Readings :** [Die01], [EN04], [Cel05], [KS02] | |

**Unit 6: Relational Databases (12)**

**Competences Expected: C1,CS4,CS5**

| Learning Outcomes | Topics |
|---|---|
| • Prepare a relational schema from a conceptual model developed using the entity- relationship model [Usage]<br>• Explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint (including definition of the concept of a foreign key) [Usage]<br>• Demonstrate use of the relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) and the relational algebra operations developed specifically for relational databases (select (restrict), project, join, and division) [Usage]<br>• Write queries in the relational algebra [Usage]<br>• Write queries in the tuple relational calculus [Usage]<br>• Determine the functional dependency between two or more attributes that are a subset of a relation [Usage]<br>• Connect constraints expressed as primary key and foreign key, with functional dependencies [Usage]<br>• Compute the closure of a set of attributes under given functional dependencies [Usage]<br>• Determine whether a set of attributes form a superkey and/or candidate key for a relation with given functional dependencies [Usage]<br>• Evaluate a proposed decomposition, to say whether it has lossless-join and dependency-preservation [Usage]<br>• Describe the properties of BCNF, PJNF, 5NF [Usage]<br>• Explain the impact of normalization on the efficiency of database operations especially query optimization [Usage]<br>• Describe what is a multi-valued dependency and what type of constraints it specifies [Usage] | • Mapping conceptual schema to a relational schema<br>• Entity and referential integrity<br>• Relational algebra and relational calculus<br>• Relational Database design<br>• Functional dependency<br>• Decomposition of a schema; lossless-join and dependency-preservation properties of a decomposition<br>• Candidate keys, superkeys, and closure of a set of attributes<br>• Normal forms (BCNF)<br>• Multi-valued dependency (4NF)<br>• Join dependency (PJNF, 5NF)<br>• Representation theory |

**Readings :** [Har02], [EN04], [Dat05], [KS02]