EPS	**BSC IN SYSTEMS ENGINEERING**
	COURSE: ESTRUCTURES DE DADES I DE LA INFORMACIÓ (DATA AND INFORMATION STRUCTURES)
	TEACHING STAFF: MARTA OLIVA/JOSEP M. RIBÓ
	COURSE: 2nd | **CREDITS: 15** | **TYPE: COMPULSORY**

1. OBJECTIVES

The main objectives of this course are: firstly, introducing the student into the object-oriented paradigm as a development model of medium and large programs. This course also presents the most frequent memory data structures, specially those which allow a direct key access to the data. Finally, the student will learn to use databases as a tool to achieve permanent information storage. The student will learn to design and create a DB in order to work with it afterwards.

During the practical sessions, it is expected that the student will recognize and use SQL sentences with a view to implement, maintain and consult information of a relational database.

2. STRUCTURE

This is an annual course consisting of 8 theoretical credits, 3 practical credits and 4 laboratory credits. The schedule will be indicated in the theoretical classes.

3. PROGRAMA

- THEORETICAL PROGRAM

1. A program design model: the object-oriented programming (OOP).

1.1. Basis of the OOP.
    1.1.1.1. Abstraction. Exemples. How to manage the abstraction when programming.
1.2. References to objects and dynamic objects.
1.3. The programming language C++. Introduction.
1.4. Derived classes. Inheritance.
1.5. Polymorphism.
1.6. Generic classes.
1.7. Friend functions.
1.8. Operators overload.
1.9. Exception management.
1.10. A Data Structure library proposal.

2. Sequential access data structures.

2.1. Introduction.

2.2. The class *List*. Operations and iterators.
2.3. The class *List.* Specification.
2.4. The class *List*. Implementation.
2.5. The class *Queue.* Specification and implementation.
2.6. Generic algorithms.
2.7. Sequential files.

3. Direct access structures. Tables.

3.1. Introduction.
3.2. Specification of the class *Table.*
3.3. Some implementations of the class *Table.*
3.4. The implementation with dispersion tables (hash).
3.5. Hash functions.
3.6. Hash strategies.
3.6.1.1.1.Closed Hash. Open Hash. Efficiency. More considerations.
3.7. The relationships implemented as multilists.
3.8. Direct access files.
3.8.1.1.1.Concept. Specification and implementation.

4. Trees.

4.1. Definitions and basic properties.
4.2. Specification of the types of binaries tree and
4.2.1.General tree.
4.3. An implementation proposal.
4.4. Range of trees.
4.4.1.Presentation of the ranges. Transformation to iterative
4.4.2.Ranges. Transformation of recursive general actions to Iterative ones.
4.5. Adding iterators to binary trees*.*
4.5.1. Specification. An implementation proposal: Trees with threads.
4.5.2. 4.6.The type *CuaPrior* (Cua with priorities).
4.5.3.Specification. A proposal of implementation: The heaps.
4.5.4.Applications to the sorting of vectors and to the fusion of filing groups.
4.6. The Search Binary Trees. The A.V.L. trees
4.6.1.Concept. Costs. Implementation. Applications to the sorting of them and implementation of ties.
4.7. Examples of complex data structures.

5. Basic concepts.

5.1. Introduction.
5.2. Benefits of the DB approach and its uses.
5.3. When not to use a DBMS.

6. DB system architecture.

6.1. Three schema layers architecture.
6.2. Data independence.
6.3. DB administrator. Data dictionary.
6.4. Types of data.

7. Development of a DB system.

7.1. The information systems in the organizations.
7.2. The development of a DB system.
7.3. The DB design process.

8. The relational model.

8.1. Description of the model. Basic concepts.

8.2. Relationship properties.
8.3. Integrity constraints.
8.4. How to avoid incorrect states.
8.5. Relational algebra. Basic operations.
8.6. A small example.

9. The conceptual and logic design.
9.1. Introduction to UML.
9.2. Types and attributes.
9.3. Relationships.
9.3.1. Generalization and specialization.
9.3.2. Associations.
9.4. Miscelania.
9.5. Translation from UML to the relational model.

10. Standardization.
10.1. Anomalies in a DB schema.
10.2. First normal form (1NF).
10.3. Functional premises.
10.4. Second normal form (2NF).
10.5. Third normal form (3NF).
10.6. Boyce-Codd Normal Form (BCNF).

11. The physical design of the DB.
11.1. Introduction
11.1.1. Storage structures. File types.
11.1.2. Access. B-trees and B+-trees.
11.2. Factors affecting the physical design.
11.3. DB physical design decisions.
11.4. How to adjust an operating DB.

- **PRACTICE PROGRAM OF THE FIRST SEMESTER**

Three practicals must be done during the semester.
For more detailed information about these practices, turn to the virtual campus document *Presentació de les pràctiques d'EDALG i EDI-1rQ*.

- **LABORATORY PROGRAM OF THE SECOND SEMESTER**

1. Introduction to the usage of a relational DBMS.

2. Table maintenance.
2.1. Table creation.
2.2. Table updating.
2.3. Table deletion.
2.4. Table copy.
2.5. View creation.

3. Index creation and deletion.

4. Information maintenance.
4.1. Register insertion.
4.2. Register updating.
4.3. Register deletion.

5. Resource granting.

    5.1. Permission granting
    5.2. Permission revocation.

6. **Information retrieval.**
    6.1. Basic retrieval sentences.
    6.2. Information filtering.
    6.3. Retrieval with advanced predicates.
    6.4. Retrieval optimization.

4. COURSE MATERIALS AND SOFTWARE

In the theoretical and exercise-solving lessons, we will use the material created for the development of the course available in the virtual campus, in the Resources section. This material consists in transparencies of the lessons and a collection of exercises.

In the lab sessions, we will use the dossiers created by the lab supervisor, which correspond to each individual session of the course. PostgreSQL will be used in all the sessions.

5. BIBLIOGRAPHY

Basic reading

- Franch X. *Estructures de dades. Especificació, disseny i implementació*. Edicions UPC, 1999. Existeix també en versió electrònica.
- Horowitz E., Sahni S. *Fundamentals of Data Structures in C++*. Computer Science Press, 1990.
- Ribó, J.M. C++ orientat a objectes. Quaderns EUP núm 7. Paperkite editorial. 1999.
- Elmasri, R. & Navathe, S.B. *Fundamentals of Database Systems*. 3rd ed. Addison-Wesley, 2000.
- Connolly, T. & Begg, C. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 3rd ed. Addison-Wesley, 2001.
- http://www.postgresql.org/idocs/

Complementary reading

- Budd T. Data *Structures in C++ using the Standard Template Library* Addison Wesley, 1998.
- Stroustrup B. *El lenguaje de programación C++*. 3a edición. Addison Wesley, 1991.
- Meyer B. *Object-oriented Software Construction.* Prentice Hall, 1988.
- McFadden, F.R., Hoffer, J.A. & Prescott, M.B. Modern Database Management. 5th ed. Addison-Wesley, 1999.
- Date, C.J. Introducción a los Sistemas de Bases de Datos. 7a. ed. Prentice Hall. 2001.
- Silberschatz, A., Korth, H. & Sudarshan, S. *Fundamentos de Bases de Datos*. 4a. ed. McGraw Hill, 2002.
- Gulutzan, P. & Pelzer, T. *SQL-99 Complete, Really*. R&D Books. 1999.

6. ASSESSMENT

The course offers two exam periods: June and September.

In order to pass the course, some practicals on the first and the second semester must be done and the theoretical examinations must be passed. There will be four mid-term exams during the course (the November and February exams correspond to the first semester and the April and the June exams correspond to the second semester). Those students who passed the practicals of the first semester during the previous year have the possibility to validate them. The practicals of the second semester cannot be validated, unless the theoretical part is also passed (in both cases, the result must be equal to or higher than 5).

The final mark in any of the two examination sessions is calculated as follows:

Nfinal=NPrac1Q*0,25+NPrac2Q*0,12+NTeoria1Q*0,25+NTeoria2Q*0,38

where:

- NPrac1Q and NPrac2Q are, respectively, the global mark of the practiceals of the first and the second semester.
- NTeoria1Q and NTeoria2Q are, respectively, the global mark of the theoretical part of the first and the second semester.

  NTeoria1Q and NTeoria2Q is the average of the mid-term exams which take place each semester (in the June exam session) or the mark which corresponds to that semester on the September exam session.

In order to pass the course, a minimum of 4 points in each of the parts NPrac1Q, NPrac2Q, Nteoria1Q, Nteoria2Q is required.

In order to pass the assignments of the first semester, an assignment validation examination must be passed. The assignments of both semesters must be submitted on the deadline.

In February, students may improve the first mid-term mark through a new exam which will include the contents of the November mid-term exam. In June, students may improve the mark of any of the three previous mid-term exams.

The NPrac2Q can be improved by means of a practical exam on the same day of the global examination in both exam sessions (June and September).