SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babes-Bolyai University Cluj**
1.2 Faculty	**Mathematics and Computer Science**
1.3 Department	**Mathematics and Computer Science in Hungarian**
1.4 Field of study	**Computer Science**
1.5 Study cycle	**master**
1.6 Study programme / Qualification	**Databases**

2. Information regarding the discipline

2.1 Name of the discipline	**Database Systems Implementation**						
2.2 Course coordinator	**Viorica Varga PhD**						
2.3 Seminar coordinator	**Viorica Varga PhD**						
2.4. Year of study	**1**	2.5 Semester	**2**	2.6. Type of evaluation	**exam**	2.7 Type of discipline	**required**

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	3	Of which: 3.2 course	2	3.3 seminar/laboratory	1
3.4 Total hours in the curriculum	42	Of which: 3.5 course	28	3.6 seminar/laboratory	14
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					28
Additional documentation (in libraries, on electronic platforms, field documentation)					30
Preparation for seminars/labs, homework, papers, portfolios and essays					50
Tutorship					
Evaluations					2
Other activities:					

3.7 Total individual study hours	110
3.8 Total hours per semester	152
3.9 Number of ECTS credits	8

4. Prerequisites (if necessary)

4.1. curriculum	• non
4.2. competencies	• developing applications on relational DBMSs (SQL, relational algebra - completed an introductory course on Databases) • sorting/searching techniques (quick/merge sorts, binary trees, hash tables - course on Design and Analysis of Algorithms)

5. Conditions (if necessary)

5.1. for the course	• Video projector

5.2. for the seminar /lab activities	• Visual Studio and Java on the computers in laboratories

6. Specific competencies acquired

Professional competencies	• have a good insight into how DBMSs function internally • understand how to analyse the performance of data-intensive systems • be familiar with a variety of programming techniques for large-scale data manipulation • apply the insights achieved to build the major components of a mini-DBMS.
Transversal competencies	• this course give the basics for query optimization

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• The course objective is the presentation of data storage in databases, buffer management, index techniques, query processing and the overview of query optimization in relational databases. • The students will be able to understand query processing in relational databases • Implementation of a simple Database Management System (DBMS).
7.2 Specific objective of the discipline	• Secondary-storage devices; disk access time; Input/Output model of computation; optimized disk access; • File and System Structure: page layout and access; buffer management; file organizations (heap, sorted, clustered); row stores versus column stores; • Indexes: Tree-structured (ISAM, B+tree); hash-based (static, extendible, linear); multi-dimensional (UB-tree, k-d-b tree, R-tree) • External Sorting: external n-way merge sort; sorting based on B+trees; • Query Evaluation: Selection (index-based, hash-based, arbitrary selection predicates); projection (duplicate elimination; hash-based, sorting-based); joins (nested-loops, index nested, block nested, sort-merge, hash joins); set operations; aggregation; impact of buffering, pipelining, blocking; evaluation techniques in existing systems;

8. Content

8.1 Course	Teaching methods	Remarks
1. The structure of the physical database. The structure of the magnetic disc. Optimization of Disk-Block Access. RAID (redundant arrays of independent disks)	Presentation	
2. Buffer-replacement policies	Presentation	
3. File organization: fixed-length records, variable-length records, sequential file, heap file, sorted file, multitable clustering file organization. Data	Presentation	

dictionary storage		
4. Ordered indices, dense and sparse indices and multilevel indices. Index Sequential Access Mechanism. Index update. Primary (clustering) and secondary (unclustering) indices.	Presentation	
5. B+-tree index files. Structure of a B+-tree. Queries on B+-trees. Algorithm for update.	Presentation	
6. Algorithm for delete in B+-tree. B+-tree file organization.	Presentation	
7. B-tree index files. Static hashing, hash indices. Dynamic hashing: extendable hashing, algorithms for update and delete in hash files. Comparison of ordered indexing and hashing.	Presentation	
8. Multiple-key access: using multiple single-key indices, indices on multiple keys, bitmaps indices.	Presentation	
9. Overview of query processing. Measures of query cost. Basic algorithm for selection implementation. (linear search, binary search, using indices, selections involving comparison)	Presentation	
10. Algorithms for external sorting.	Presentation	
11. Algorithms for projection, set operations, outer join and aggregation implementation.	Presentation	
12. Algorithms for join implementation (nested-loop join, block nested-loop join, indexed nested-loop join, merge join, hash join, cost of algorithms).	Presentation	
13. hash join, cost of algorithms Implementation of pipelining.	Presentation	
14. Overview of query optimization. Transformation of relational expressions, equivalence rules. Join ordering. Enumeration of equivalent expressions. Estimating statistics of expression results: selection size estimation, join size estimation, size estimation for other operations. Materialized view, it's maintenance and using it in query optimization.	Presentation	

Bibliography

[MUW00] H. Garcia-Molina, J. D. Ullman, J. Widom: *Database Systems - The Complete Book*, Prentice Hall Upper Saddle River, New Jersey, 2008.

[R02] R. Ramakrishnan: *Database Management Systems*, WCB McGraw-Hill, Boston, 2002.

[SKS06]A. Silberschatz, H. Korth, S. Sudarshan: *Database System Concepts*, McGraw-Hill, New York, 2006.

[V06] V. Varga, *Interogarea bazelor de date distribuite*, Casa Cărții de Știință, Cluj-Napoca, 2006.

8.2 Seminar / laboratory	Teaching methods	Remarks
1. Implementation of a complete single-user relational database management system. It involves a significant amount of coding. The project is highly structured, but there is enough slack in the specification so that creativity is both allowed and required.It is recommended to implement a server component and a client one. The client can be implemented as Windows interface, Web client or a command line parser.		
2. *The Record Management* (*RM*) *Component:* implement a set of functions for managing		

unordered files of database records. (There is recommended to use binary files to implement unordered files). You can consider fix length records; the management of variable length records is optional. One idea to implement the delete operation of a record is the logical delete. It means to store for every record in one bit, which store: the record is deleted or not. In order to not read the whole file to find deleted records and overwrite them with new ones, you can link the deleted records in a stack or queue. The top of the deleted records stack can be stored in the first record of the file. You have to store the system catalog. It will contain table names, index file names. For every table the file name, where the table is stored, the structure of the table, the constraints, the associated index files. For every index file, the search key, the type of it. You can implement the catalog in XML file. In Catalog.xml you can find an example.		
3. *The Indexing* (*IX*) *Component:* implement a facility for building indexes on records stored in unordered files. The indexing facility will be based on B+ trees or dynamic hashing.		
4. *The Indexing* (*IX*) *Component:* implement a facility for building indexes on records stored in unordered files. The indexing facility will be based on B+ trees or dynamic hashing.		
5. *The System Management* (*SM*) *Component:* This part will implement various database and system utilities, including data definition commands (at least integer and character data type), including primary key and foreign key constraint (primary key have to be implemented for one or more columns, but foreign key is optional to implement for more than one column), index definition commands and catalog management. For primary key you will create index file automatic. The System Management component will rely on the Record Management and Indexing components from Parts 1 and 2. It also will use a command-line parser or a graphical user interface.		
6. *The Query Language* (*QL*) *Component:* In this part students will implement a query language, which consists of user-level data manipulation commands, both queries and updates (SQL Select, Insert, Update, Delete can be used). The Query Language component will use a command-line parser or a graphical user interface. The queries have to be processed, using algorithms presented at the course. Features you have to implement in Select statement: selection, projection, join of tables, aggregation, cumulative functions, (subquery, order by is optional for extra points).		
7. Create a database with 3 tables with the project.		

Update the data and run queries.		
Bibliography http://inst.eecs.berkeley.edu/~cs186/sp07/projects.html http://research.cs.wisc.edu/coral/mini_doc/minibase.html		

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- This course is in concordance with the program of similar courses in other universities:
  http://scpd.stanford.edu/search/publicCourseSearchDetails.do?method=load&courseId=11782
  http://www.cs.ox.ac.uk/teaching/courses/databasesystemsimplementation/

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	exam	written test	30
10.5 Seminar/lab activities	mini DBMS project	solve a problem with the project	70
10.6 Minimum performance standards			
➢ working mini DBMS project ➢ 50% in exam			

Date	Signature of course coordinator	Signature of seminar coordinator
..22 April 2013....	assoc. prof. Viorica Varga	assoc. prof. Viorica Varga
	.....	.....

Date of approval	Signature of the head of department
..... 30 April 2013.....	.....….....