Course title: PHYSICAL DESIGN OF DATABASES

Lecturers	Full Prof. Alen Lovrenčić, Ph.D., Miljenko Novaković, mag. educ. phys. et inf.
Language of instruction	Croatian and English
Study level	Master
Study programme	Databases and Knowledge Bases
Semester	2nd (summer)
ECTS	5
Goal	This course complements the knowledge required for profound understanding of Systems Database Management Systems (DBMS), their functioning and practical application. This course uses knowledge from many other courses, for example Databases 1, Algorithms, Data structures etc., and represents their logical continuation in the similar context. The students are acquainted with the way a software system works. Software systems are the second most dominant system regarding their presence in a computer, right after the operating systems. DBMS is quite a complex system which itself contains various aspects that enable its functioning. In this course the students are introduced to many of these aspects appearing in most known DBMSs. This course is therefore useful not only to students planning to explore DBMS development, (it is assumed that the number of such students is limited) but to all the students interested to use a DBMS. The course will help them understand possibilities and limits of the system they are working with, which are inevitable owing to objective theoretical difficulties. It will also help them understand unfavorable criticism related to that particular system, accounted for by a lack of inadequate solutions that the system offers. Finally, the course helps the students choose the appropriate DBMS. In addition, students who intend to work in the field of information science, especially in the database field, are given a different perspective of the database world. They are introduced to the lower, physical level and algorithms which enable declarative work,
General and specific learning outcomes	
Content	**1. Introduction (2 hours)** The concept of DBMS. Historical development of DBMS. Physical, conceptual and outside level of DBMS. Language for database work. Relation model. Charts and attributes. **2. Discs and databases (2 hours)** Managing disc space. Managing storage. Documents. Documents of fixed and variable length. Pages. Pages of variable and fixed length. Data banks. Indexes. Systematic DBMS catalogues. Sorting in secondary memory. Sorting by connecting and evaluation of complexity. **3. Organization of data banks and data bank objects** Heap data banks. Sorted data banks. Hash data banks. Choice of data bank system. Representation of relation objects. Documents. Fixed length documents. Document titles. Notes. Variable length documents. Changes in documents. **4. Linear indexes (2 hours)** Sequential data banks. Dense and rare indexes. Primary and secondary indexes. Composite indexes. Indexes with doubled keys. B+-tree indexes. Organization of

	B-tree index. Searching B+-tree index. Adding value to B+-tree index. Deleting values from B+-tree index. Evaluation of operation complexities in B+-tree index. **5. Hash chart-based indexes (2 hours)** Hash chart-based indexes. Static hashing. Expansive hashing. Additions to expansive Hash chart. Deleting from expansive Hash chart. Evaluation of operation complexities in expansive Hash index. Linear hashing. Adding elements into linear Hash chart. Deleting elements from linear hash chart. Evaluation of operation complexities in linear Hash chart. **6. Indexes for multidimensional data (2 hours)** Data cubes. Grid data banks. Searching grid data banks. Inserting values into Grid data banks. Evaluation of operation complexities in Grid data banks. R-trees. Operation on R-trees. Evaluation of operation complexities on R-trees. Index bitmaps. GiST and GiN indeces **7. Queries (2 hours)** Relation algebra. Selection, projection, natural connection, -connection operators. Cartesian product. Group operators– union, difference, cut. Elimination of multiple lines. Grouping and aggregation. Sorting. Chart review. Sorting while reviewing. Reviewing complexity. **8. Queries (cont.)** Single-phase algorithms for relation operators. Selection case – without index, unsorted data. Selection – without index, sorted data. Selection – B+-tree index. Selection – hash index. Sorting-based projection. Hashing-based projection. Projection and indexes. **9. Queries (cont.) (2 hours)** Natural connection – nested loops. Connection done by connection-based sorting. Connection with help of hashing. Union and difference with help of sorting. Union and difference with help of hashing. Sorting-based double-phase algorithms. Hashing-based double-phase algorithms. Index-based algorithms. Multi-phase algorithms. **10. Optimizing and query translating** Parsing of a query. Syntax analysis. Translation trees. Query plans. Rules for query improvements. Rules for commutability and associability. Selection rules. Pushing away selections. Rules on connections and products. Rules for erasing double lines. Rules linked to grouping and aggregation. **11. Optimizing queries (cont.) (2 hours)** Advancing query plan. Grouping associative and cumulative operators. SDBM-led statistics. Estimation of plan cost. Estimation of exit size. Estimation of inter-result size. Estimation of projection size. Estimation of selection size. Estimation of connection size. Connections with several conditions. Connections of a bigger relation number. Estimation of other operators. **12. Optimizing queries (cont.)** Heuristics for lowering the plan price. Left and right argument of the connection and their difference. Connection trees. Algorithm for optimizing connection sequence based on dynamic programming. Algorithm for optimizing connection sequence based on greed method. Creation of final question plan. Choice of selection method. Choice of connection method. Forwarding for unary operators. Forwarding for binary operators.

	13. Competitiveness in DBMS (2 hours) Concept of transactions. Serial and serialized plans. Locking. Implementation of locking and unlocking. Deadlocks. Solving deadlocks by finding cycles in a directed mathematical graph. Prevention of deadlocks. Problem of livelocks. Planning locking. Double-phase locking. Divided and exclusive keys. Incremental keys. Problem of phantom documents. **14. Competitiveness in DBMS (cont.) (2 hours)** Competition control in tree-indexes. Ensuring competitiveness using time traps and shots of database. Problem of dirty data. Connecting locking and time traps. Transactions which read uncertified data. View seriability. Solving deadlocks using time traps. Very long transactions. Problems of long transactions. Sagas. Compensation of transactions. **15. Recovery from system crash** Logs. Other data structures for recovery of database. Logging protocol by writing in advance. Work with control points. Analysis phase. Transaction repetition phase. Phase of deleting unfinished transactions. Media recovery. Media mirroring. RAID5 discs.
Exercises	Seminars: Seminar projects deal with new methods which appear in DBMS, and with related areas not otherwise covered. They can also be connected to criticism of the way of implementation of certain theoretical concepts in DBMSs used by students. This individual student's project is significant for acquiring knowledge about the area, the real DBMS and differences between theory and practice. For that reason seminar projects have a major influence on student's final course grade. Exercises: Laboratory exercises are based on developing algorithms connected to databases in C programming language.
Realization and examination	Classes: lectures, seminars and exercises Examination: Seminars (50%), written examination (25%), oral examination (25%)
Related courses	1. Berkeley: http://inst.eecs.berkeley. edu/~cs186/sp18/ 2. Cornell: http://www.cs.cornell.edu/Courses/cs432/2003fa 3. Princeton: http://www.cs.princeton.edu/courses/archive/spring03/cs425 4. Washington: http://www.cs.washington.edu/education/courses/444/ 5. Toronto: http://www.cs.toronto.edu/~csc443h 6. Brown: http://www.cs.brown.edu/courses/cs127 7. Yale: http://zoo.cs.yale.edu/classes/cs437
Literature	Basic: Oracle Database Concepts, Oracle Corp., 2015. Ramakrishnan, R.: Database Management Systems, McGraw- Hill, 1998 Garcia-Molina, H.; Ullman, J.D.; Widom, J.: Database System Implementation, Prentice-Hall, 2000 Additional: Ullman, J.D.; Widom, J: First Course in Database Systems, Prentice-Hall, 2001. P. Atzeni, V. De Antonellis: Relational Database Theory, The Benjamin/Cummings, 1993. Aho, A.; Hopcroft, J.; Ullman, J.D.: Algorithms and data structures, Addison-Wesley, 1983 J. van Leeuwen: Handbook of Theoretical Computer science, Elsevier & MIT Press, 1990.

	D.E. Knuth: The art of Computer programming: Searching and Sorting, Addison-Wesley, 1973.