Syllabus
Advanced Databases
(5 ECTS)

Alexander Breyman (abreyman@hse.ru)
Department of Software Engineering

1. Course Description

a. Pre-requisites

The course is based on the knowledge of foundations of discrete mathematics (including logic and set theory), computer science, and computer programming. The students are assumed to be familiar with the topic of databases and have a proper knowledge of the database design and implementation, including entity-relationship data modeling, relational model, algebra and calculus, functional dependencies and normalization theory, relational query languages, including SQL.

b. Abstract

The objective of Advanced databases course delivery is to form professional competencies related to design and implementation of non-relational databases, including object-oriented, object-relational, deductive, multidimensional and semi-structured database models. Students will get a grasp on strengths and weaknesses of wide spectrum of approaches to data storage, search and retrieval, resulting in informed choice of database model.

It is a double module course, which is delivered in modules #1 and #2 of the first academic year. Number of credits is **5**. Total course length is **190** academic hours including **56** auditory hours (**22** Lecture **(L)** hours and ***34*** Seminar **(S)** hours) and **134** Self-study (**SS**) hours. Academic control forms are one home assignment, one test, and one written exam after module #2.

2. Learning Objectives

The course is based on "Advanced databases" course of Eindhoven University of Technology (Eindhoven, Netherlands), Faculty of Math and Computer Science, author and principal lecturer – Prof. Dr. Toon Calders.

While Databases course of bachelor curricula covered many of the core concepts behind database modeling, design and implementation, there are many other considerations that should be addressed for successful scientific or industry career in this field. The main objective of this course is to expand students' view and introduce advanced topics, including object-oriented extensions, deductive databases and appropriate query languages, data warehouses and online analytical data processing, and XML. The additional topics covered in this course will help students become more proficient in choosing appropriate technologies for projects and will expand their knowledge base so that they have a better understanding of the field. By the end of the course, students should have a solid grasp on business intelligence tools and XML, which will prove to be invaluable as they progress further in computer science studies.

This course studies different database models and their properties. The models that will be discussed are:

- Object-oriented databases;
- Deductive databases (Datalog);
- Data warehousing and online analytical processing (ROLAP, MOLAP and HOLAP);
- The XML data model (query languages XQuery and XPath, DTDs).

For these conceptual models the course will concentrate on the following points: Why was the database model introduced? Which of the shortcomings of other models does it address? What are the most important concepts and notions for the database model? How is the model implemented? Which are the main techniques? The importance of understanding the internals of a particular database model cannot be overemphasized as it is closely connected to its limitations.

3. Learning Outcomes

After taking this course the student should have achieved the following objectives:

Knows the shortcomings and restrictions of relational data model. Can reason about expressibility of relational query languages using notion of locality.

Knows data storage methods usable for object-oriented program systems, including pure object database systems and object-relational mappers, its advantages and disadvantages.

Knows models and methods of organization of deductive databases using Datalog language. Knows implementation and optimization techniques for Datalog translator.

Knows the reference architectures of data warehouses and is aware of the basic functionality offered by available commercial and free data warehousing systems. Master methods and tools for creating analytical database solutions. Can choose dimensions for multidimensional database, group them into hierarchies and define aggregates. Knows principles of view choice for materialization.

Knows XML document model, DTD and XML Schema. Can write and understand XPath and XQuery expressions. Can create and interpret XLST transformations.

Students should be able to understand the language of studies models, choose and use appropriate models and programming languages, implement systems using chosen models, methods and tools.

The course contributes to the development of the following competencies:

Research activities.

The ability to reflect (evaluate, process, analyze and synthesize) scientific methods and ways of activity for application in practice (СК-М1).

The ability to introduce concepts and models, create and verify new methods and tools for professional activity for application in practice (СК-М2).

The ability to learn new research methods in self-development mode, to correct own scientific and production activity profile (СК-М3).

Analytical activities.

The ability to analyze, verify and evaluate its completeness for data found and received from various sources in course of professional activity; to recover and to synthesize missing information if necessary (СК-М6).

The ability to select and develop methods for analysis of objects of professional activity taking into account current trends in software engineering (ПК10, ИК-М1.1.НИД (ПИ)).

The ability to perform analysis, synthesis and optimization of solutions to ensure the quality of the objects of professional activity (ПК11, ИК-М1.2.НИД (ПИ)).

Project activities.

The ability to organize individual and collective research work (ПК12, ИК-М1.3.НИД (ПИ)).

The ability to perform project work in the field of software engineering on the basis of system approach, to build and use models for description and forecasting of various phenomena, to carry out its qualitative and quantitative analysis (ПК15, ИК-М3.1.ПД (ПИ)).

Technological activities.

The ability to apply modern technologies of software complexes development using automated systems

of planning and management, to control the quality of the developed software products. (ПК18, ИК-М4.1.ПТД_ПИ2 (ПИ)).

Course Plan

№	Title of the topic	Total Hours	Classroom hours		Self-study
			Lectures	Seminars	
	Module #1				
1	Introduction and overview of the course.	***14***	2	2	10
2	Shortcomings of the relational model.	***16***	2	4	10
3	Object-oriented databases	***30***	2	6	22
4	Deductive databases.	***35***	4	4	27
	Module #1 totals	***95***	**10**	**16**	**69**
	Module #2				
5	Data warehousing and online analytical processing	***43***	4	8	31
6	Semistructured data.	***50***	6	10	34
7	Conclusion	***2***	2		
	Module #2 totals	***95***	**12**	**18**	**65**
	Total:	**190**	**22**	**34**	**134**

Topic 1. Introduction and overview of course.
Lecture outline:

- Structure of course.
- Restrictions of relational model.
- Object-oriented databases.
- Deductive databases.
- Data warehousing and OLAP.
- Semistructured data and XML.
- Project.
- Grading.

Practical studies:

1. Relational query languages refresher
    - Formulate queries for given typical relational database using relational algebra, SQL and Datalog.
    - Express graph-theoretic queries (e.g.: give all nodes that do not have any incoming edges) in either SQL or relational algebra.

Topic 2: Shortcomings of the relational model.
Lecture outline:

- Why relational model is insufficient.
- Expressivenes of relational algebra.
- Limitations of relational query languages concerning inexpressibility of certain queries classes, e.g. transitive closure and other recursive queries.
- Gaifman locality.
- Historical overview of non-relational models.
- Nested relational algebra.
- Nesting in SQL dialects.
- Extending SQL with recursion.

Practical studies:

1. Locality
    - Construct the Gaifman graph for given database and identify the spheres and neighborhoods for given tuples.
    - Construct recursive SQL queries.
    - Determine the locality rank for given query.
2. Limitations of the relational model
    - Write an SQL query using a recursive view definition.
    - Express queries using nested relational algebra.
    - Rewrite nested relational algebra expressions into flat ones.

Topic 3. Object-oriented databases.

Lectures outline:

- Applications that require storage and manipulation of complex data.
- Object-oriented programming languages for complex objects manipulation.
- Mapping objects to tuples in relations.
- Impedance mismatch.
- Extending SQL with complex types: collections, structures, inheritance, references.
- Methods for complex types.
- Notion of persistence.
- Persistent programming languages.
- Persistent objects.
- Object identity.
- Query languages for object-oriented databases.
- Object-relational databases.
- Object-relational mapping.
- Language-integrated query

Practical studies:

1. Complex types in SQL:1999 and SQL:2003
    - Compose queries with expressions to refer to RAW type components.
    - Write queries using observer methods.
    - Extend complex type for handling evolving data structure.
    - Define new type for given data structure .
2. OO-DBMS db4o
    - Create simple persistent objects, store, retrieve, update and delete it.
    - Compose native (NQ) queries returning objects that satisfy given conditions.
    - Write QBE queries returning objects that satisfy given conditions.
    - Create complex structured persistent objects, write and execute queries for different update depth.
    - Create collections and arrays, write and execute queries.
    - Create persistent class hierarchy, store and retrieve objects of subclasses.
3. Object-relational mapping with Hibernate
    - Create class-to table mappings using Hibernate annotations.
    - Make Hibernate configuration file.
    - Create data access objects using Hibernate persistence API.
4. Entity data model and language-integrated query (LINQ)
    - Create Entity data model for given domain.
    - Write LINQ queries for data retrieval.
    - Write LINQ queries for data update.

Topic 4. Deductive databases.

Lectures outline:

- Logic programming and Prolog.
- Combining rules and facts in one database.
- Intensional and extensional relations.
- Datalog syntax.
- Semantics of the Datalog program.
- Non-recursive Datalog programs.
- Negation.
- Safety of rule.
- Model-theoretic semantics.
- Recursive Datalog programs.
- Fixpoint evaluation.
- Stratified programs and strata.
- Aggregation.
- Equivalence of relational algebra and safe Datalog with negation, without recursion and aggregation.
- Evaluation and optimization of Datalog programs: avoiding repeated and unnecessary inferences, filtering with “magic sets”, indexing, materialization.

Practical studies:

1. Datalog without recursion
    - Write a Datalog program that defines intensional views for certain kinship types.
    - Give a relational algebra expressions for views defined by given Datalog programs.
    - Give a minimal model of a Datalog program.
    - Reason about safeness of Datalog program.
    - Give a stratified model contents of a Datalog program.
2. Recursive Datalog
    - Show steps of naïve and semi-naïve implementation of recursion for computing given intensional view.
    - Give a relational algebra expressions for views defined by given Datalog programs.

Topic 5. Data warehousing and OLAP.

Lectures outline:

- Requirements to data management from decision support systems.
- Historical, summarized, integrated data.
- Statistical and analytical queries.
- Business intelligence applications.
- Three-tier architecture.
- Extract-transform-load process.
- Data warehouse, data mart.
- Online analytical processing (OLAP).
- Conceptual models for decision support.
- Multidimensional view on the data.
- Cross-tabulation.
- Data cubes.
- Operations with data cubes: roll-up, drill-down, pivot, slice & dice, select.
- Query languages for supporting OLAP.
- SQL extensions: Group by cube, group by rollup.

- Multidimensional expressions (MDX).
- Data explosion problem.
- View materialization: optimal set of views.
- Partial order on views, cost model, greedy algorithm.
- Relational OLAP (ROLAP): Star schema, snowflake schema, snowflake constellation.
- Multi-dimensional OLAP (MOLAP): multicubes and hypercubes, sparse and dense dimensions.
- Indexing of dimensions: b-tree, bitmap, join indexes.
- Hybrid OLAP (HOLAP).

Practical studies:

1. Data warehousing and datacubes
    - Give a relational schema for certain data warehouse with sales data.
    - Express datacube aggregation in SQL:1999.
    - Count datacube cells for given relation in dense and sparse settings.
    - Determine sizes of views for given datacube.
    - Choose views to materialize by applying the greedy algorithm.
    -
2. ROLAP, MOLAP and HOLAP
    - Choose set of views for materialization for demographic database based on uniform access path distribution assumption.
    - Choose set of views for materialization for demographic database based on known workload.
    - Propose suitable physical organization for given datacube.
    - Construct bitmap-indexes and use it for query answering.

Topic 6. Semistructured data.

Lectures outline:

- Semistructured data for human and machine consumption.
- XML: tags, elements, attributes, values.
- Well-formed and valid documents.
- Namespaces.
- XPath: axes, node-tests, predicates.
- XQuery: expressions, functions.
- FLWOR expressions.
- XQuery data model.
- DTD and XML Schema.
- Simple and complex types of XML Schema.
- Light XQuery.
- Extensible Stylesheet Language Transformations (XSLT): templates, parameters, variables.
- XML data management.

Practical studies:

1. XML and XPath
    - Find and correct errors in given XML document.
    - Construct tree-representation of given XML document.
    - Write XPath-expression that selects given elements and attributes.
    - Predict result of XPath expression execution.
    - Reason about equivalency of given XPath queries.
    - Give unabbreviated versions for abbreviated XPath expressions.

2. XQuery
    - Predict result of given XQuery quieris and explain it.
    - Write XQuery expressions for given queries.
    - Transform one XML document into another using XQuery.
    - Write XQuery function for transforming elements and attributes.
    - Write a function for transitive closure computing.
3. DTDs, XML Schema, XPath, XQuery
    - Check validity of document given DTD.
    - Construct DTD for given constraints.
    - Check validity of document given XML Schema.
    - Construct XML Schema for given constraints.
    - Reason about equivalency of given XPath queries.
    - Write XPath and XQuery expressions for data extraction from marked-up text document.
4. XSLT
    - Predict result of given XLST template execution and explain it.
    - Write XSLT template for given queries.
    - Transform one XML document into another using XSLT.
5. Large XML repositories
    - For given document storage layout using SQL express XML navigational axes in SQL expressions.
    - Use partitioning scheme for answering XPath-expressions efficiently.
    - Construct edge-based relational representation of XML document and give SQL-query for computing the result of XPath-expression.

Topic 7. Conclusion.
Lecture outline:

The last theory lecture will conclude the course with a summary of the material covered during the course. Students will get the opportunity to ask questions. We will review and discuss some past

5 Reading List

a) Required

- Foster, E. C., Godbole S. (2016) **Database Systems: A Pragmatic Approach**, Second Edition [Электронный ресурс] / Elvis C. Foster, Shripad Godbole. – Электрон. текстовые данные. – Apress, 2016. – 644 p. – 978-1-4842-119-22. — Режим доступа: https://proxylibrary.hse.ru:2184/book /10.1007%2F978-1-4842-1191-5
- Vaisman A., Zimányi E. **Data Warehouse Systems. Design and Implementation**. [Электронный ресурс] /Alejandro Vaisman, Esteban Zimányi. – Электрон. текстовые данные. – Springer, 2014. – 625 p. – 978-3-642-54655-6. — Режим доступа: https://proxylibrary.hse.ru:2184/book/10.1007%2F978-3-642-54655-6

b) Optional

- Ul Haq Q.S. Data Mapping for Data Warehouse Design [Электронный ресурс] / Qamar Shahbaz Ul Haq. – Электрон. текстовые данные. – Morgan Kaufmann Publishers, 2016. – 181 p. – 978-012-80518-56. — Режим доступа: https://proxylibrary.hse.ru:2258/toc.aspx?bookid=106793
- Linstedt D., Olschimke M. Building a Scalable Data Warehouse with Data Vault 2.0 Design [Электронный ресурс] / Daniel Linstedt and Michael Olschimke. – Электрон. текстовые данные. –Morgan Kaufmann Publishers, 2016. – 684 p. – 978-012-80251-09. — Режим доступа: https://proxylibrary.hse.ru:2258/toc.aspx?bookid=100369

- Celko J. Joe Celko's Complete Guide to NoSQL: What Every SQL Professional Needs to Know about Nonrelational Databases [Электронный ресурс] /Joe Celko. – Электрон. текстовые данные. –Morgan Kaufmann Publishers, 2014. – 244 p. – 978-012-40719-26. — Режим доступа: https://proxylibrary.hse.ru:2258/toc.aspx?bookid=67013
- Majkić Z. Big Data Integration Theory. Theory and Methods of Database Mappings, Programming Languages, and Semantics [Электронный ресурс] / Zoran Majkić. – Электрон. текстовые данные. – Springer, 2014. – 516 p. – 978-3-319-04156-8. — Режим доступа: https://proxylibrary.hse.ru:2066/10.1007/978-3-319-04156-8

6. Grading System

Attendance record provides data for calculating the ratio of the number of lectures, attended by student, to overall number of lectures and practices. Value of attendance component of final grade formula (A) is that ratio multiplied by 10 and rounded to the nearest integer value.

Practice activity during practice hours is assessed by evaluating of student involvement into discussions as well as quality of exercise performance during seminars. Value of practice activity component of final grade formula (PA) is an integer value from interval [0,10].

The course Advanced databases includes a group project, compulsory to all students. Students will work in groups of from 3 up to 5 students on either one of the suggested topics, or on a subject of their own choice. Projects can be of two types: build an application using the techniques studied in the course or study some of the topics more in-depth. Results of group project of first type should be presented in form of report that consists of design document, implementation description, results of testing. Mandatory appendixes are source code for application and database creation script. Results of second type project should be presented in a form of survey or research paper. Report should be submitted to LMS not later that for 7 calendar days before assigned date of its presentation (announced on first lecture). Project should be presented and demonstrated by all group members. Each group member should demonstrate complete understanding of all project details and give correct answers to at least two questions of instructor. Value of group project component of final grade formula (P) is an integer value from interval [0,10] consists of the common score for the report and presentation (from 0 to 5; same score to all group members) and individual student score for the answers to the questions (from 0 to 5). If a student misses the project presentation because of some valid reason, s/he receives «absence» grade. If a student misses the project presentation because of any other reason, s/he receives grade based on individual score set to 0.

Exam at the end of the second module (module 2 of 1st year of study) implies arrangement of the oral examination for all students enrolled to the course. Topics covered by the test embraces all course material. If a student misses the exam because of some valid reason, s/he receives «absence» grade. The exam (E) is assessed on usual ten-point scale.

The overall and accumulated course grades Go and Ga (10-point scale each) are calculated as follows:

$Ga = 0.2A + 0.5PA + 0.3P;$

$Go = 0.6Ga + 0.4E.$

The overall and accumulated course grades Go and Ga (10-point scale each) include results achieved by students in their attendance record A, practice activities PA, group project P and exam E; it is rounded up to an integer number of points. The rounding procedure accounts for students' practice activities during seminars. Intermediate assessment retakes are not allowed.

7. Guidelines for Knowledge Assessment

Home assignment for group project:

The course Advanced databases includes a group project, compulsory to all students. Students will work in groups of up to 5 students on either one of the suggested topics, or on a subject of their own choice.

Projects can be of two types:
1. Application: build an application using the techniques studied in the course. Motivate the choices you made during the project; e.g., why did you choose to store user data in XML, why do you use an object oriented database instead of a relational one, etc. The grade is not only determined by the complexity of your application, but also (and even more) by the appropriate use of the right technologies.
2. Research: study some of the topics more in-depth; e.g.: one of the topics is OLAP and it is argued that such systems can more efficiently handle ad-hoc analytical queries. Test this by constructing a database and benchmark the performance of certain types of queries in both systems. What is the influence of a particular indexing technique? Other example: take one of the research papers and validate the results of the paper by repeating the experiments; make a critical analysis, show weaknesses and opportunities for the approach proposed in the paper.
The only restriction on the subject of the project is that it needs to involve the techniques that have been studied the course; i.e., the subject needs to relate to at least one of the following topics:
1. Object-oriented databases (this include persistent programming languages) or object-relational extensions to relational databases,
2. Datalog or other rule-based deduction engines,
3. Semistructured data (XML),
4. Data warehousing and Online Analytical Processing (OLAP)

In the project students can use available implementations and tools; i.e., you do not have to write your own XQuery processor or your own Datalog engine. You can use existing database management systems, such as MySQL, Postgress, MonetDB for relational support, Galax, MonetDB/XQuery for XPath and XQuery functionality, available Java and PHP libraries, Object-oriented databases; Objectivity, ObjectStore, etc.

A good guideline to take into account when formulating a project proposal is that the workload is expected to be between 30 to 40 hours per group member. Please notice that the main workload should be on course-related topics; e.g., spending two weeks on making a dazzling GUI might turn out to be less effective for getting a good grade than working a couple of days on improving the database structure.

To successfully complete the project, several deliverables need to be provided:
Project proposal. Latest date proposals will be accepted: 30.10.2019. This document must contain the following information:
1. Group members;
2. Project description:
(application) Which data are you planning to use? What will be the functionalities of the application? Make a detailed list of the functionalities.
(research) which papers will you discuss? What are the claims you will test? What experiments will you consider? How will you evaluate the results?
3. A planning; how long do you expect to work on each of the components?
4. Overview of the technologies you are planning to use.
The complete project proposal report is expected to be around 2 pages (excluding figures).
Final report. Due date: 20.12.2019.
(application) This document describes the inside of the application; how is the data stored? What are the DTDs/schema's that have been used? Which queries are used by the system in order to retrieve the data? In general, all technical details of which you think they show your ability to use the material that was lectured.
(research) Which papers have been read? Give a short summary in your own words. List the claims that have been tested. Explain the experimental setup. Discuss the outcome. There are no

strict lower or upper bounds on the length of this document, but a good guideline is around 15 pages, including illustrative queries, XML fragments, screen shots, etc. Please note that not only completeness but certainly also succinctness is an important quality for a good report.

7. Examination type

Exam at the end of the second module (module 2 of 1st year of study) implies arrangement of the oral examination for all students enrolled to the course. Topics covered by the test embraces all course material:

- Expressivenes of relational algebra.
- Limitations of relational query languages.
- Gaifman locality.
- Nested relational algebra.
- Extending SQL with recursion.
- Mapping objects to tuples in relations.
- Extending SQL with complex types: collections, structures, inheritance, references.
- Persistent programming languages.
- Query languages for object-oriented databases.
- Object-relational mapper.
- Intensional and extensional relations.
- Semantics of the Datalog program.
- Notion of safety of datalog rule.
- Model-theoretic semantics.
- Recursive Datalog programs.
- Fixpoint evaluation.
- Stratified programs and strata.
- Evaluation and optimization of Datalog programs: avoiding repeated and unnecessary inferences.
- Evaluation and optimization of Datalog programs: filtering with “magic sets”.
- Evaluation and optimization of Datalog programs: indexing and materialization.
- Requirements to data management from decision support systems.
- Extract-transform-load process.
- Conceptual models for decision support.
- Multidimensional view on the data.
- Operations with data cubes: roll-up, drill-down, pivot, slice & dice, select.
- Query languages for supporting OLAP.
- SQL extensions: Group by cube, group by rollup.
- Multidimensional expressions (MDX).
- View materialization: optimal set of views, partial order on views, cost model, greedy algorithm.
- Relational OLAP (ROLAP): Star schema, snowflake schema, snowflake constellation.
- Multi-dimensional OLAP (MOLAP): multicubes and hypercubes, sparse and dense dimensions.
- Indexing of dimensions: bitmap indexes.
- Indexing of dimensions: join indexes.
- Hybrid OLAP (HOLAP).
- XML: tags, elements, attributes, values.
- Well-formed and valid documents.

- XPath: axes, node-tests, predicates.
- XQuery expressions.
- XQuery functions.
- FLWOR expressions.
- XQuery data model.
- DTD.
- XML Schema: simple and complex types.
- Extensible Stylesheet Language Transformations (XSLT): templates, parameters, variables.

8. Methods of Instruction

Course studies are organized in the form of lectures and practical studies. Besides traditional forms, some active and interactive forms are provided: discussion of real industry case studies; proposing and discussing group projects topics and its planned outcomes, using interactive simulators for database languages.

9. Special Equipment and Software Support

9.1 Software

Software access: internal network, in accordance with license and contract.

- Microsoft Windows 7 Professional RUS
- Microsoft Windows 8.1 Professional RUS
- Microsoft Windows 10
- Apple Mac OS
- Microsoft Visual Studio 2015 Community (or later versions)
- SQL Server Management Studio

9.2 Remote support

LMS is used for remote course support.

9.3 Material and technical resources

Projector for lectures and practical studies.