# Syllabus
# Databases
# (6 ECTS)

## 1. Course Description

a. Pre-requisites: Programming, Discrete Mathematics, Introduction into Software Engineering, Algorithms and Data Structures

b. Abstract

The course attendants should develop skills and understanding in:

- the design methodology for databases and verifying their structural correctness;
- implementing databases and applications software in the relational model as well as in map/reduce paradigm;
- using querying languages, primarily SQL, and other database supporting software;
- applying the theory behind various database models and query languages;
- implementing security and integrity policies relating to databases;
- working in group settings to design and implement database projects.

## 2. Learning Objectives

The objective of the course is to expose to students topics of (mostly) online transactional processing and relational database theory, database design and implementation, including entity-relationship data modeling, relational model, algebra and calculus, functional dependencies and normalization theory, relational query languages, including SQL. 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.

This course studies different conceptual database models and their properties. 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 data modeling concepts and has an understanding of relational data model.
- Can compose queries to relational databases using relational algebra, tuple relational calculus and SQL.
- Knows methods of database design, including entity-relationship approach and normalization-based approach.
- Knows and is able to apply database application design and development methods usable for object-oriented program systems, including object-relational mappers, its advantages and disadvantages.

- Knows models and methods of internal organization of relational databases including file storage, indexing, query processing and transaction management issues.

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.

This course contributes to the development of the following competencies (according to the educational standards of National Research University "Higher Schools of Economics" – 09.03.04 "Software Engineering", Bachelor level, protocol approved on 30.01.2015, No1) described in the table below. More detailed description of the theory and practice that form the listed competencies as well as evaluation criteria are given in the corresponding sections of this syllabus (Course Plan and Course Assessment).

| Competency | Code | Descriptors | Education forms and methods for competency formation |
|---|---|---|---|
| Universal | УК-1, СК-Б1 | Able to learn, acquire new knowledge, skills, including in the area other than the professional | To successfully complete homework assignment, students have to find information about the data sources and other domain knowledge for wide variety of possible areas – economics, transportation, manufacturing, medicine, etc. |
| Universal | УК-3, СК-Б4 | Able to solve problems in professional activities on the basis of the analysis and synthesis | This course is divided onto sections covering wide variety of methods and tools for data management including storage, retrieval, exchange, and others, that can be used for synthesizing a software solution to data processing tasks. This could be achieved by analyzing and dividing the given complex task (e.g. homework assignment) onto a sequence of simpler tasks. |

| | | | |
|---|---|---|---|
| Universal | УК-5, СК-Б6 | Able to work with the information: locate, evaluate and use information from different sources necessary for solving scientific and professional problems (including on the basis of the system approach) | To successfully complete homework assignment, students have to find information about the data they will use for their project, understand and properly model the data. They also need to find the appropriate ways (usually seek for on-line documentation) to manage the data (store, read, exchange, etc.) |
| Universal | УК-7, СК-Б8 | Able to work in a team | Teamwork is mandatory to complete homework assignment. |
| Professional, Scientific research activities | ПК-1, ИК-1 | Able to apply main concepts, principles, theories and facts, connected to computer science, in the process of scientific research and problem solving. | Essay composing, Home assignments, Self-study |
| Professional, Scientific research activities | ПК-2, ИК-2 | Able to formalize in its subject area within the constraints of methods of research | Practical studies in data modeling, Home assignments (data modeling part) |
| Professional, Scientific research activities | ПК-3, ИК-3 | Prepared to use research techniques and tools to study objects of professional activity | Lectures, Practical studies in data model assessment and evaluation, Home assignments (solutions evalution part) |
| Professional, Analytical activities | ПК-6, ИК-6 | The ability to formalize the subject area of a software project and to develop specifications the software product components | Lectures, Practical studies, Home assignments |
| Professional, Project activities | ПК-10, ИК-10 | Able to design, develop and test software products. | Lectures, Practical studies, Home assignments |
| Professional, Project activities | ПК-11, ИК-11 | Able to read, understand, and extract the main idea from source code and documentation | Practical studies, Industrial cases reviews |
| Professional, Technology-oriented activities | ПК-15, ИК-15 | Able to use operating systems, network technologies, software interface development tools, languages and methods of formal specifications, database management systems | Lectures, Practical studies, Home assignments |
| Professional, Technology-oriented activities | ПК-16, ИК-16 | Able to use diverse software development technologies | Lectures, Practical studies, Home assignments |
| Professional, Technology-oriented activities | ПК-17, ИК-17 | Able to use core software development methods and tools | Practical studies, Home assignments: Microsoft Visual Studio, Microsoft SQL Server Management Studio |

| | | | |
|---|---|---|---|
| Professional, Technology-oriented activities | ПК-19, ИК-19 | The ability to understand life cycle standards and models. | Lectures, Practical studies, Home assignments |

4. Course Plan

| № | Topic title | Total hours | Classroom hours | | | Self-study |
|---|---|---|---|---|---|---|
| | | | Lectures | Seminars | Practice | |
| | **Module #1, 3rd year** | | | | | |
| **1** | Introduction. | ***8*** | 2 | | 2 | 4 |
| **2** | Data modeling. | ***12*** | 2 | | 2 | 8 |
| **3** | Database Design: The E/R and UML Approaches. | ***24*** | 4 | | 4 | 16 |
| **4** | Relational Model. | ***12*** | 2 | | 2 | 8 |
| **5** | Relational Database Design. | ***12*** | 2 | | 2 | 8 |
| **6** | Relational Query Languages. | ***12*** | 2 | | 2 | 8 |
| **7** | SQL. | ***34*** | 6 | | 6 | 22 |
| | **Module #1 totals** | **114** | **20** | | **20** | **74** |
| | **Module #2, 3rd year** | | | | | |
| **8** | Application Design and Development. | ***24*** | 4 | | 4 | 16 |
| **9** | Storage and File Structure. | ***12*** | 2 | | 2 | 8 |
| **10** | Indexing and Hashing. | ***22*** | 4 | | 4 | 14 |
| **11** | Query Processing. | ***22*** | 4 | | 4 | 14 |
| **12** | Transaction Management. | ***22*** | 4 | | 4 | 14 |
| **13** | Distributed and Parallel Databases. | ***12*** | 2 | | 2 | 8 |
| | **Module #4 totals** | **114** | **20** | | **20** | **74** |
| | **TOTAL** | **228** | **40** | | **40** | **148** |

| Assessment | Form | 3rd year | | | | Dept | Parameters |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | | |
| Current (week) | Test | 7 | | | | SE | Written test (SQL), 60 min. |
| | Quiz | * | * | | | SE | Each next week, first 5 minutes of lecture |
| | Essay | * | * | | | SE | Written essay, up to 2 pages, homework |
| Interim | Homework | | * | | | SE | Group presentation of home assignment followed by demonstration |
| Final | Exam | | * | | | SE | Written exam, 90 min. |

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

b. Optional


- Vihya V., Jeyaram G., Ishwarya K.R. Database Management Systems Edition [Электронный ресурс] /V. Vidhya, G. Jeyaram, and K.R. – Электрон. текстовые данные. – Alpha Science International, 2016. – 417 p. – 978-1-78332-318-0. — Режим доступа: https://ebookcentral.proquest.com/lib/hselibrary-ebooks/detail.action?docID=5248352
- Celko J. Joe Celko's SQL for Smarties: Advanced SQL Programming, Fifth Edition [Электронный ресурс] /Joe Celko. – Электрон. текстовые данные. –Morgan Kaufmann Publishers, 2015. – 853 p. – 978-012-80076-17. — Режим доступа: https://proxylibrary.hse.ru:2258/toc.aspx?bookid=78670
- 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
- Mohanty H., Bhuyan P., Chenthati D. Big Data. A Primer [Электронный ресурс] / Hrushikesha Mohanty, Prachet Bhuyan, Deepak Chenthati. – Электрон. текстовые данные. – Springer, 2015. – 184 p. – 978-81-322-2494-5. — Режим доступа: https://proxylibrary.hse.ru:2066/10.1007/978-81-322-2494-5


## 6. Grading System

Rounding procedure for grades (where applicable): up to an integer number of points.

Practice activity during practice hours is assessed by evaluating of student involvement into discussions as well as quality of exercise performance during practice. Practice activity grade ($O_{classroom}$) uses a ten-point scale.

Students have to write an essay (on topic proposed by instructor) once in each module (due 4th week of each module) with length of up to 2 pages. Grade $O_{essay}$ is an arithmetic averages of two essays grades (ten-point scale, rounding up to an integer number of points).

Students have to answer quiz questions in first 10 minutes of some lectures. Grade $O_{quiz}$ is an arithmetic averages of quiz grades respectively (ten-point scale, rounding up to an integer number of points). Average is calculated by division of sum of all student's quiz answers grades on a total quizzes count in a year.

Students have to answer test questions on 7th week of the 1st module. Grade $O_{test}$ for that test is on ten-point scale.

Value of $O_{homework}$ (homework assignment) component of final grade formula 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.

Written test at the end of the first module (last week of the 1st module) is assessed on usual ten-point scale.

Written exam at the end of the second module (2nd module) $O_{exam}$ is assessed on usual ten-point scale.

Cumulative grade for student's current results is calculated using the following formula:

$$O_{cumulative} = 0,7 * O_{current} + 0,3 * O_{classroom}$$

where

$$O_{current} = 0,2 \cdot O_{essay} + 0,2 \cdot O_{test} + 0,2 \cdot O_{quiz} + 0,4 \cdot O_{homework};$$

Final grade for student is calculated using the following formula:

$$O_{final} = 0,5\ O_{cumulative} + 0,5\ O_{exam}.$$

## 7. Guidelines for Knowledge Assessment

Home assignment has to be prepared in modules 1 and 2 by students in groups of up to 5 and includes design, implementation and testing of a database and database application for given subject area (chosen by group and approved by instructor or assigned by instructor). Results of the home assignment 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. Report should be submitted to LMS not later that for 7 calendar days before assigned date of its presentation (on the last week of module 2). 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.

Written test at the end of the first module (last week of the 1st module) implies arrangement of the written test (in lecture room) for all students enrolled to the course. Topics covered by the test embraces all course material of first module.

Written exam at the end of the second module (2nd module) implies arrangement of the written test (in lecture room) for all students enrolled to the course. Topics covered by the test embraces all course material.

## Topics for the home assignment

The course includes home assignment, compulsory to all students. Students will work in groups of up to 5 students towards designing and development of a relational database. Topics might be suggested by lecturer as well as proposed by students.

Example topic: Online auction system. The Online Auction System is a web-based application which allows Sellers to sell their items via Auctions. Buyers make bids. Last highest bid wins. After an Auction is closed, the winner pays for the item via a credit card. The seller is in charge of delivering the Item to the buyer.

The contents of the design document / explanatory note:

- planned end users, users' needs
- functional requirements and data restrictions in textual form
- non-functional requirements (optional)
- a preliminary database schema (based on requirements): a list of entities, relationships and attributes - either as an ER diagram or as a class diagram UML)
- functional and multi-valued (optional) dependencies that are derived from textual restrictions on data (e.g.: single trip has exactly one initial stop, exactly one final stop, and 0 or more intermediate stops; each point can be an initial stop of 0 or more trips)
- normalized database scheme (up to BCNF (mandatory) or 4NF (optional) with respect to the set of dependencies)
- SQL DDL script for database creation based on normalized schema
- SQL DML queries that implement functional requirements
- requests grouped into transactions

Evaluation:

- completeness and consistency of (textual) functional requirements - 0..1
- compliance with ER / UML functional requirements - 0..1
- completeness and consistency of the set of (formal) functional dependencies - 0..2
- correctness of normalization - 0..2
- the correctness of SQL DML requests is 0 .. [min (number of requests, 3)]

Bonuses:

- correctness of an example of problems due to undernormalization - 0..1
- correctness of accounting for non-functional requirements - 0..1

**Example Midterm Test**

Task 1. Entity-Relationship Modelling. Company has a few branches (each branch is in distinct city) and time to time send employees to business trips from one branch to another. Each trip has a planned and fact start and end dates, and a task assigned to it. Trip may be cancelled before its start, and date of cancellation is of interest. A pair of flights should be arranged and a hotel should be booked at destination city for each trip. Each flight has an origin and destination cities, air company, flight number, date and time of departure, as well as ticket price. Hotel has name, address (including city), and room charge for overnight stay. Draw an ER diagram for this domain.

Task 2. DDL. Build a relational database scheme for model from Task 1. (CREATE TABLEs).

Task 3. SQL. Find all hotels in Paris booked by employees from Berlin in current calendar year.

Task 4. SQL. Find company spending on hotel bookings in previous month grouped by city of hotel.

Task 5. SQL. Find company spending on hotel bookings in previous month grouped by city of employee who stayed in there.

Task 6. SQL. Find top 3 branches by number of trip cancellation in three days before trip start date.

Task 7. SQL. Find all pairs of employees who travelled in the same day between the same pair of cities in the same direction but on different flights.

**Example Quiz Questions**

1. In the column Phone Number a phone number is stored as a string (varchar(20)). Select all the correct versions of the function that will return the first three characters of this string.

- LEFT(PhoneNumber,3)
- SUBSTRING(PhoneNumber,1,3)
- SUBSTRING(PhoneNumber,3,1)
- CHARINDEX('[0-9][0-9][0-9]', PhoneNumber, 3)

2. An empty table Document has the following definition:
   CREATE TABLE Document (
   DocumentID UNIQUEIDENTIFIER NOT NULL PRIMARY KEY DEFAULT (NEWID())
   );
   What result will be output after executing the following sequence of commands?
   INSERT INTO Document DEFAULT VALUES;
   INSERT INTO Document DEFAULT VALUES;
   SELECT * FROM Document;

- A table consisting of two rows and one column containing two different GUID.
- A table consisting of two rows and one column containing integer values 1 and 2.
- The attempt to execute the second INSERT will output an error of primary key value duplication.
- The attempt to execute the first INSERT will output an error that a primary key can not be set by a default value.

3. In the database there is a table OrderDetails which stores detailed information on sales:
   CREATE TABLE OrderDetails (
   …
   OrderNo INT NOT NULL, // order number
   Name VARCHAR(100), // product name
   UnitPrice NUMERIC(10,2), // unit price
   Quantity INT // quantity
   );
   You create a query that should return a list of products, the unit price of each of which does not exceed 100, and the total revenue from each exceeds 10 000. In which part of the query should the following condition be put?
   (UnitPrice<=100)

- WHERE
- HAVING
- GROUP BY
- ON

4. In the database there is a table OrderDetails which stores detailed information on sales:

CREATE TABLE OrderDetails (
…
OrderNo INT NOT NULL, // order number
Name VARCHAR(100), // product name
UnitPrice NUMERIC(10,2), // unit price
Quantity INT // quantity
);

You create a query that should return a list of products, the unit price of each of which does not exceed 100, and the total revenue from each exceeds 10 000. In which parts of the query should the column name Name be placed?

- GROUP BY
- SELECT
- WHERE
- HAVING
- ON

5. Which statements about operators used in subqueries are correct?

- NOT operator can be used with operators IN, ANY, and ALL
- IN operator can be used with subqueries which can return only a single record (single-row)
- Comparison operator "=" can be used with subqueries which can return more than one record
- NOT IN operator is equivalent to IS NULL

6. Which of the following SQL operators requires the use of subquery?

- EXISTS
- BETWEEN
- IN
- LIKE
- NOT IN

7. Is it possible to write two subqueries in WHERE and compare their results (using '='operator)? WHERE (SELECT …) = (SELECT …)

- Yes, if each of the subqueries returns strictly a single value (a table of one row and one column)
- Yes, if each of the subqueries returns a table of one row and two or more columns
- Yes, if at least one of the subqueries returns strictly a single value (a table of one row and one column)
- Yes, always
- No, it is not

8. Is it possible to compare in WHERE the results of two subqueries which return tables of the same structure of two or more columns and an unknown number of rows? WHERE (SELECT A, B FROM …) <op> (SELECT C, D FROM…)

- No, it is not
- Yes, by using IN
- Yes, by using EXISTS
- Yes, by using ANY
- Yes, always

9. Which of the following may be included in the estimated query execution plan?

- Data read from tables
- Search in the indexes
- Sort operations
- Data on the number of rows actually read in the tables
- Partition operations

10. What is a query execution plan? Choose all of the correct options.

- A representation of a query which presents the order of execution of table data fetch operations and relational operations.
- A procedural representation of declarative query.
- A journal of operations that have been performed for DBMS to obtain the query result.
- Estimation of query execution time.
- Estimation of query execution cost

11. Can a query execution plan contain a sort operation if ORDER BY is not used in the query itself?

- It can, to prepare the data for sort-merge
- !It can, to eliminate duplicates from the query result
- It can, if the table is partitioned
- It can, if an index is created by the field, by which the sorting is performed
- It can not in any way

12. List all options of indexes that an optimizer can use for faster execution of query SELECT A FROM T WHERE B=1. Value B=1 occurs in two rows of the table.

- CREATE INDEX IX_T_B(B)
- CREATE INDEX IX_T_BA(B,A)
- CREATE INDEX IX_T_AB(A,B)
- CREATE INDEX IX_T_A(A)

**Example Essay Topic**

Chris Anderson's paper «The End of Theory: The Data Deluge Makes the Scientific Method Obsolete» (http://www.wired.com/science/discoveries/magazine/16-07/pb_theory) describe a new science paradigm in face of availability of huge amounts of data.

In spite of those ideas, propose a list of possible data sources for discovering how local weather conditions affect sales and consumer traffic of a large international fast food restaurant chain. Which datasets may be useful for that purpose? (Some types of possible questions: Did restaurants along the highway get more traffic in drive-thru during regular vs. abnormal weather? Do restaurants need to staff more if different weather conditions? Did people prefer drive-thru in extreme weather conditions irrespective of the geography?)

**Topics for course final assessment**

- Basic database system concepts.
- Database environment.
- Database planning.
- Three level database architecture.
- Basic relational model concepts.
- Object-oriented model.

- Object-relational model.
- Semi-structured model.
- Mathematical and database relations. Relation schema. Relational database.
- Integrity constraints.
- Relation keys. Key constraint. Foreign key constraint.
- Relational algebra operations. Selection. Projection.
- Relational algebra operations. Set operations.
- Relational algebra operations. Join, equijoin, antijoin, theta-join. Natural join.
- Relational algebra operations. Division.
- Tuple relational calculus: atoms, formulas, queries.
- SQL data types.
- SQL. Table declaration. Primary keys, unique constraints, default values, nullable attributes.
- SQL. Check constraints.
- SQL. Foreign key constraints. Handling foreign key violations.
- SQL. Database schema modifications.
- SQL. Single-table queries. Filtering conditions. Logical operations IN, ALL, EXISTS.
- SQL. Join queries. Join types: cross, natural, inner, outer, self.
- SQL. Duplicates elimination.
- SQL. Set operations.
- SQL. Nested queries. Correlated nested queries.
- SQL. Aggregate functions.
- SQL. Grouping and group filtering.
- SQL. Query result sorting.
- SQL. INSERT.
- SQL. UPDATE .
- SQL. DELETE.
- SQL. Views: creation, use and updating.
- SQL. Triggers: creation, activation, execution. Multiple triggers.
- SQL. View materialization.
- Stored procedures.
- Entity-relationship model. Entities and attributes. Entity types. Keys.
- Entity-relationship model. Relationships. Attributes and roles. Relationship type, degree and cardinality. Relationship participation constraints.
- Entity type hierarchies. Specialization and generalization. Total and partial unions.
- Unified modeling language class diagram. Association and aggregation in UML. Generalization hierarchies in UML. Multiplicity indicators in UML.
- Objectives of normalization. Limitations of E/R design. Data redundancy.
- Anomalies: insertion, deletion, update.
- Functional dependencies. Axioms of functional dependencies.

- Closure. Minimal cover of a set of dependencies.
- Desirable properties of decompositions: attributes preservation, dependency preservation, lossless join.
- First normal form.
- Full functional dependencies. Second normal form.
- Transitive dependency. Third normal form. Boyce/Codd normal form (BCNF).
- Multivalued dependencies. Fourth normal form.
- Fifth normal form.
- Domain/Key normal form (DKNF).
- BCNF decomposition algorithm and its properties.
- Normalization drawbacks.
- JDBC architecture, connecting to DBMS, preparing and executing queries, using result sets and cursors.
- Handling exceptions in JDBC.
- Transactions in JDBC.
- Object-relational mapping.
- Design patterns for data persistence. Active Record pattern.
- Design patterns for data persistence. Data Mapper pattern.
- Hibernate.
- Datafiles: blocks and extents. Block structure. Fixed and variable record formats. Large objects (LOBs).
- B-tree index. B-tree insertions and deletions.
- Hash index. Hash functions.Extendible hashing.
- Bitmap index.
- Join index.
- GiST.
- Relational algebra translation. Query tree.
- Relational algebra equivalences.
- Cost-based optimization. Cost factors and estimation.
- External merge sort.
- Duplicate elimination.
- Implementing set operations.
- Sort-based and hash-based projection.
- Computing selection without indexes.
- Computing selection with clustered index.
- Computing selection with b-tree index.
- Computing selection with hash index.
- Computing joins: nested loops.
- Computing joins: block nested loops.
- Computing joins: sort-merge join.
- Computing joins: hash-join.
- Basic concepts of transactions. ACID properties.

- Schedules: serial, serializable. Methods to ensure serializability.
- Optimistic and pessimistic concurrency.
- Two-phase locking. Locking and deadlocks. Implementing isolation levels with locks.
- Snapshot isolation.
- Write-ahead log. Redo and undo records. Recovery from crash.
- Distributed transactions. Two-phase commit protocol.
- Distributed database architectures.
- Software components and functions of distributed DBMS.
- Transaction management for distributed DBMS.
- Distributed recovery from failures.
- Distributed query processing.
- Data integration.
- Parallel database systems.

## 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

Projector for lectures and practical studies.

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