Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Emblem of the department (in the presence)

Department of computer technology

Organization of databases

curriculum (Syllabus)

Details of the Course	
Level of higher education	*First (undergraduate)*
Field of Study	*12 Information technologies*
Specialty	*123 Computer engineering*
Educational program	*Computer Engineering*
Type of Course	*Normative*
Mode of Studies	*Full-time*
Year of studies, semester	*2nd year, spring semester*
ETCS workload	*36 lectures, 18 practical, 66 self-study*
Semester control/ control measures	*Final Test*
Timetable	
Language of teaching	*English*
Information about the head of the course / teachers	*PhD, Assoc. Prof. Andriy Boldak*
Access to the course	Presentations https://edu-db.github.io/p1 https://edu-db.github.io/p2 https://edu-db.github.io/p3 https://edu-db.github.io/p4 https://edu-db.github.io/p5 http://jace-dev.herokuapp.com/design/use-case-tutorial http://jace-dev.herokuapp.com/design/conceptual-data-modeling http://jace-dev.herokuapp.com/design/ER-modeling Tests http://jace-dev.herokuapp.com/design/uml-editor http://jace-dev.herokuapp.com/design/MySQLTraner Repository template for performing laboratory work https://github.com/boldak/database_basics_template Methodical instructions for performing laboratory work

	https://github.com/boldak/database_basics_template/blob/master/guidelines/guidelines.md Video recordings of lectures https://t.me/COMSYS_DB_2022/112 https://t.me/COMSYS_DB_2022/99 https://t.me/COMSYS_DB_2022/201 https://t.me/COMSYS_DB_2022/202 https://t.me/COMSYS_DB_2022/275 https://t.me/COMSYS_DB_2022/276 https://t.me/COMSYS_DB_2022/201?single https://t.me/COMSYS_DB_2022/202?single https://t.me/COMSYS_DB_2022/363 https://t.me/COMSYS_DB_2022/348 https://t.me/COMSYS_DB_2022/335 https://t.me/COMSYS_DB_2022/379 https://t.me/COMSYS_DB_2022/367

Course Program

1. Description of the course, its purpose, subject of study and learning outcomes

The purpose of the educational discipline is to form students' abilities related to the practical application of existing database management systems; the use of effective models of data provision based on the study of the subject area, methods of analysis, search and use of existing database management systems; using existing relational database management systems.
The subject of the academic discipline is the theory and methodology of designing databases, as well as the development of application software (software) using database management systems (DBMS).
After mastering the academic discipline, students must demonstrate the following results: the basis of the system approach to software design, software lifecycle models, stages of information support design and implementation, methods of subject area analysis and formulation of requirements for the software system, methods of describing system functionality, basics of using the UML language for conceptual modeling of the subject area, methods informational and datalogical modeling of data, relational algebra, the language of structured SQL queries, the basics of DBMS administration, software architecture using databases, object-relational display design templates.
skills: collect and analyze requirements for system functionality, perform conceptual modeling of the subject area, develop infological and datalogical models, implement a database using the SQL structured query language, perform data manipulation, design a software layer responsible for object-relational mapping .
gain experience: analysis of requirements for a software system, design of databases as part of software systems that use various methods of data processing, implementation of application software using databases.

2. . Pre-requisites and post-requisites of the course (place in the structural and logical scheme of training according to the relevant educational program)

The discipline is a component of professional and technical training and is directly related to other regulatory disciplines of natural science, basic and professional training
"Discrete mathematics", "Object-oriented programming", "Organization of computer networks", "Software development technologies".
The basic knowledge for studying the discipline is the theory of sets and operations on them, the basics of object-oriented programming, and design patterns.

3. Course content

Chapter 1. Methodological principles of database design
Topic 1.1. Basic provisions of the system approach to software design

Topic 1.2. Software life cycle models. Characteristics of the information support development process.

Chapter 2. Fundamentals of database design

Topic 2.1. The stage of development of requirements for the functionality of the software system.

Topic 2.2. The stage of designing scenarios of interaction between the system and the user. Use case model.

Topic 2.3. The stage of conceptual design of software system information support.

Topic 2.4. Infographic model "Entity-relationship"

Topic 2.5. Datalogic models. Relational scheme. Object-oriented domain model.

Topic 2.6. Relational algebra

Topic 2.7. Database management systems

Chapter 3. SQL structured query language

Topic 3.1. SQL structured query language. Data definition language.

Topic 3.2. SQL structured query language. A data manipulation language.

Topic 3.3. Complex queries in the SQL language. Queries from multiple tables.

Topic 3.4. Complex queries in the SQL language. Grouping and sorting results.

Chapter 4. Use of databases in software applications.

Topic 4.1. Software architecture using databases

Topic 4.2. Design patterns for organizing database access.

Topic 4.3. Basics of ODBC and JDBC.

Topic 4.4. Generalized implementation of the Data Access Object1 template

Topic 4.5. The main trends in the development of information technologies related to the use of data.

4. Educational materials and resources

1. An introduction to the Unified Modeling Language , https://developer.ibm.com/articles/an-introduction-to-uml/
2. Tutorials: Discover UML , https://www.developer.com/design/tutorials-discover-uml/
3. Database Design in DBMS Tutorial: Learn Data Modeling , https://www.guru99.com/database-design.html
4. Database Structure and Design Tutorial , https://www.lucidchart.com/pages/database-diagram/database-design
5. Database Modeling in UML , https://sparxsystems.com/resources/tutorials/uml/datamodel.html
6. V. I. Haydarzhi, O. A. Datsyuk. Fundamentals of designing and using databases: training. manual — K.: Polytechnic; Periodicals, 2004.
7. G. A. Gaina Fundamentals of database design: teaching. manual Grif MON.— K.: Condor, 2008.
8. V.V. Pasichnik, V.A. Reznichenko Organization of databases and knowledge: a textbook. Grif MON.— K.: BHV Publishing Group, 2006.

Additional materials

9. Bauer C. Hibernate In Action / C. Bauer, G. King. – Greenwich: Manning Publications, 2004.
10. Bernard E. Hibernate Search in Action / E. Bernard, J. Griffin. – Manning Publications, 2008.

Educational content

5. Methods of mastering the Course (educational component)

Names of sections, topics	Distribution of study time				
	In total	including			
		Lectures	Practical (seminar) classes	Laboratory work (computer workshop)	Self-study
1	2	3	4	5	6
Chapter 1. Methodological principles of database design					
Topic 1.1. Basic provisions of the system approach to software design	2	2			
Topic 1.2. Software life cycle models. Characteristics of the information support development process.	5	2			4
Together by chapter 1	**7**	**4**			**4**
Chapter 2. Fundamentals of database design					
Topic 2.1. The stage of development of requirements for the functionality of the software system.	10	2		1	4
Topic 2.2. The stage of designing scenarios of interaction between the system and the user. Use case model.	10	2		1	4
Topic 2.3. The stage of conceptual design of software system information support.	10	2		1	4
Topic 2.4. Infographic model "Entity-relationship"	5	2		1	4
Topic 2.5. Datalogic models. Relational scheme. Object-oriented domain model.	5	2		2	4
Topic 2.6. Relational algebra	5	2		2	4
Topic 2.7. Database management systems	5	2		2	4
Together by chapter 2	**50**	**14**		**10**	**28**
Chapter 3. SQL structured query language.					
Topic 3.1. SQL structured query language. Data definition language.	6	2		1	4
Topic 3.2. SQL structured query language. A data manipulation language.	8	2		1	4
Topic 3.3. Complex queries in the SQL language. Queries from multiple tables.	6	2		1	4

Topic 3.4. Complex queries in the SQL language. Grouping and sorting results.	6	2		1	4
Together by section 3	**26**	**8**		**4**	**16**
Chapter 4. Use of databases in software applications.					
Topic 4.1. Software architecture using databases	4	2			3
Topic 4.2. Design patterns for organizing database access.	8	2		1	3
Topic 4.3. Basics of ODBC and JDBC.	8	2		1	3
Topic 4.4. A generalized implementation of the Data Access Object template	8	2		2	3
Topic 4.5. The main trends in the development of information technologies related to the use of data.		2			3
Together under section 4	**34**	**10**		**4**	**15**
Preparation for the test	**2**				**2**
Test	**1**				**1**
Total per semester	**120**	**36**		**18**	**48**

Lecture classes

lecture no	The name of the topic of the lecture and a list of the main questions (a list of didactic tools, references to the literature and self-study tasks)
1	**Basic provisions of the system approach to software design**. Definition of system and external environment, "black box" model, system classification, decomposition process, system structure, classification and generalization process, methods of describing system dynamics, program as an object of system analysis.
2	**Software life cycle models. Characteristics of the information support development process**. Definition of the software life cycle, cascade, iterative and spiral models of the software life cycle, Rational Unified Process methodology, agile methods, stages of information support development, responsibility, prerequisites and results of activities at the stages of database development.
3	**The stage of development of requirements for the functionality of the software system.** Analysis of the subject area in order to identify requirements for the program, requests of interested parties, business actors, business scenarios, scenario specification, scenario verification, system boundary drawing, formulation of functionality requirements.
4	**The stage of designing scenarios of interaction between the system and the user. Use case model.** Identifying use cases. Definition of actors of the system. Definition of use cases and relationships between use cases. Use case diagrams. Action diagrams. Study of the interaction of objects within the framework of object-oriented analysis of systems. Diagrams of implementations of use cases of the system. Use case flow diagrams. Specification of case scenarios.
5	**The stage of conceptual design of software system information support.** Standard stereotypes for business entities. Identification of business entities. Diagrams of business

	entities. Study of the interaction of objects within the framework of object-oriented analysis of systems. Definition of relationships between business entities. Associations between business entities. Aggregate relations of business entities. Define roles between business entities. Associative business entities. Defining inheritance relationships between business entities. Object-oriented design and use of design patterns.
6	**Infographic model "Entity-relationship".** Model "entity - connection". Concept of domain, entity, properties, connection. Entity subtypes and supertypes. ER modeling of the subject area. Basic concepts. Types of ER diagrams. Types of communication on ER-diagrams. Rules for converting the model of business entities into the "entity-relationship" model.
7	**Datalogic models. Relational schema. Object-oriented domain model.** Relational table, table header and body, type, attributes and instances. Constraints, primary and alternate keys, foreign key, additional semantics of foreign keys. Rules for transforming the entity-relationship model into a relational scheme. Rules for transforming the entity-relationship model into an object-oriented model. Object-relational mapping.
8	**Relational algebra.** Basic concepts and definitions. Relational algebra. Operations of relational algebra. Examples of application of relational algebra. Properties of relational algebra operations. Codd (with variable tuples) and Pirot (with variable domains) relational calculus.
9	**Database management systems.** DBMS MySQL General characteristics of DBMS MySQL. DB objects: tables, forms, queries, reports. The structure of the MySQL DBMS. MySQL server installation and configuration. Using the MySQL Workbench application for database management, forward and reverse engineering in MySQL Workbench.
10	**SQL structured query language. Data definition language.** The SQL language. SQL statements. DDL (Data Definition Language) operators are database object definition operators.
11	**SQL structured query language.** A data manipulation language. DML (Data Manipulation Language) operators - data manipulation operators. Examples of using data manipulation operators. INSERT - insert rows into the table. UPDATE - restoring rows in the table. DELETE - delete rows in the table. Examples of using the SELECT statement. Selecting data from one table. Syntax of conditional WHERE expressions. The order of execution of the SELECT statement. Implementation of relational algebra by means of the SELECT operator. SQL relational completeness. Direct product operator. Projection operator. Selection operator. Union operator. Connection operator. The intersection operator.
12	**Complex queries in the SQL language. Queries from multiple tables.** Selecting data from several tables. Use of correlation names (aliases, pseudonyms). Using subqueries. Using union, intersection and difference. Types of connections and their use.
13	**Complex queries in the SQL language. Grouping and sorting results.** Sorting query results. Using aggregate functions in queries. Using aggregate functions with groupings.
14	**Software architecture using databases.** DBMS in "client-server" architecture. Client-server architecture. Open systems. Clients and servers of local networks. System architecture "client-server". Database servers. Interaction of different types of DBMS. Access and data exchange technologies between different types of DBMS. Development of network databases. Development problems and methods of designing database "client-server" applications. Distributed databases. Distributed databases. WEB-access to the database, an example of php MyAdmin, systems with a thin client, WEB-services, RESTfull-services.
15	**Design patterns for organizing database access.** Templates Active Record, Access Data Object. Comparison of templates. Solving the object caching problem.
16	**Basics of ODBC and JDBC.** Using the Brige template to organize unified access to databases. Characteristics and usage of ODBC. Characteristics and use of JDBC, classes of session, driver, request, result. Retrieving metadata via JDBC.

17	**A generalized implementation of the Data Access Object template.** Characteristics of the Hibernate library, xml-description of the display, use of annotations for the description of the display, Hibernate-session factory, DAO-classes, factory usage, Hibernate settings.
18	**The main trends in the development of information technologies related to the use of data.** New data models. Data integration based on ETL technology. Distributed data processing.

Laboratory classes (computer workshop)

The main tasks of the cycle of laboratory classes (computer workshop) are the students' acquisition of the necessary practical skills related to the design, implementation and use of databases using the MySQL database and the Java programming language. The topics of laboratory works cover separate stages of designing and implementing databases within the limits of individual tasks for the development of application programs for various branches of industrial activity.

No. z/p	Name of laboratory work (computer workshop)	Number of aud. hours
1	2	3
1	Identification of requirements for the functionality of the program. Development of requests from interested parties.	2
2	Development of Use case model.	2
3	Development of a model of business objects. Development of the entity-relationship model	2
5	Development of a relational scheme. Database implementation using MySQL.	2
6	Database filling, implementation of queries using the SQL language.	2
7	Implementation of an object-oriented model of the subject area using the Java language.	2
8	Implementation of object-relational mapping using the Java language.	2
9	Development of a software application using the Java language.	2
	Together:	18

6. Self-study of a student/graduate student

In the process of completing individual tasks, students must consolidate knowledge acquired during lectures and independent work, independently study specific topics, deepen their knowledge for further study. Students' independent work consists of the following:

- In preparation for lecture classes on the study of previous lecture material;
- In the performance of lecture assignments at self-study of a student;
- In preparation for laboratory works with the study of the theory of a laboratory session with an oral answer to the given questions of the section;
- In execution with the preparation of the protocol for each laboratory session on the previous topic.

Policy and control

7. Course policy (educational component)

In the process of completing individual tasks, students must consolidate knowledge acquired during lectures and independent work, independently study specific topics, and deepen their knowledge for further study. Students' independent work consists of the following:

- preparation for lecture classes by studying the previous lecture material;
- performance of lecture self-study tasks
- preparation for laboratory work with the study of the theory of laboratory work with an oral answer to the given questions of the section;
- implementation and preparation of the protocol for each laboratory session on the previous topic.

8. Types of control and rating system for evaluating learning outcomes

The student's rating consists of the points he receives for the performance and defense of laboratory work

System of rating points

1. Performance and protection of laboratory work.

During the semester, students do 10 laboratory works. Weighted score - 9.
The maximum number of points for each work:

- For performing and protecting laboratory work on time - 9 points.
- For early completion and defense of all laboratory work - 10 points.

Penalty points:

- absence from laboratory work without a valid reason - 0.25 points;
- being late for laboratory work by more than 5 minutes - 0.25 points;
- performance or defense of laboratory work not on time - 3 points

The maximum number of points for laboratory works R_1 = 90 points: (9 * 10 laboratory works + 10 for early defense of all works).

The minimum number of points for one laboratory work is 6 points. That is, a student who completed all laboratory work can receive 60 points.

Calculation of the size (R) of the student rating:

The sum of the weighted points of control measures during the semester is:
$R_c = R_1$, where

- R_1 - the sum of weighted points of control measures during the semester for laboratory works,

The minimum value of R_c under the condition of completion of all laboratory works is $R_c = 60$.

The size of the credit module rating scale is 100 points.
A necessary condition for a student's admission to credit is the absence of laboratory work arrears.
To receive credit from this credit module “automatically” student must have a rating of at least 60 points.
Students, who have a rating of less than 60 points, as well as those who want to increase their rating in the ECTS system, undergo an interview, as a result of which they can receive additional points. (Maximum value $r_d = 20$). A student's additional points r_d are added to his semester rating R.

$$RD = R_c + r_d$$

The grade (ESTS and traditional) is assigned according to the RD points scored. The point scored by the student (total rating of the student) is RD according to the table.

Rating value from the RD credit module	ECTS assessment	Traditional credit assessment
95-100	A	Enrolled
85-94	B	
75-85	C	
65-75	D	
60-65	E	
< 60	Fx	Not counted
There are arrears for laboratory work	F	Not allowed

Working program of the Course (syllabus):

Compiled by: Associate Professor of the Computer Engineering Department, Candidate of Technology. Sciences, associate professor, Andrii BOLDAK

Approved by the Department of Computing (Protocol No. 10 dated 05/25/2022)

Agreed by the Methodical Commission of the faculty [1](protocol No. 10 dated 06/09/2022)