KHMELNYTSKYI NATIONAL UNIVERSITY

APPROVED BY

Dean of the Faculty of Information Technologies

Tetiana HOVORUSHCHENKO

Signature Name, SURNAME

1 09 2025

WORK PROGRAM OF THE ACADEMIC DISCIPLINE

Design of Vector and Distributed Databases

Discipline title

Purpose of the Work Program	For educational programs of various specialties
Level of Higher Education	Second (Master's)
Language of Instruction	English
Discipline Volume, ECTS Credits	8
Discipline Status	Selective
Faculty	Information Technologies
Department	Computer Science

Form of Education	Discipline Volume		Number of Hours						Form of semester control
			Classroom Sessions					Independent work (including IT)	
	ECTS Credits	Hours	Total	Lectures	Laboratory Session	Practical Sessions	Seminar Sessions		Credit
Full-time	8	240	68	32	36			172	+

The work program is based on the educational programs of training and the standard of higher education of the specialty

Work program compiled by ____________ Ph.D. in E. S., As. Prof. Oleksandr MAZURETS
Signature Academic degree, academic title, Name, SURNAME

Work program compiled by ____________ Marina MOLCHANOVA
Signature Academic degree, academic title, Name, SURNAME

Approved at the meeting of the Department of Computer Science

Protocol from 29.08.2025 #1.

Head of the Department of Computer Science ____________ Oleksandr BARMAK
Title Signature Name, SURNAME

Khmelnytskyi 2025

3. Explanatory note

Course Description. The academic discipline **"**Design of Vector and Distributed Databases" belongs to the cycle of elective courses and aims to deepen students' professional competencies in the design, implementation, and optimization of high-performance data storage systems that combine vector and distributed paradigms. The course contributes to mastering theoretical foundations and modern technologies for vector-based data modeling, approximate similarity search, distributed data management, and integration with machine learning pipelines. Through lectures and laboratory sessions, students acquire both technical and analytical skills necessary for developing scalable, fault-tolerant, and semantically aware database systems. The study of this discipline enhances students' ability to work with technical documentation, research methodologies, and industrial data management standards, while also fostering essential soft skills such as teamwork, communication, initiative, and responsibility in solving complex engineering problems.

The purpose of the discipline. The purpose of the discipline is to provide students with a comprehensive understanding of the fundamental concepts, principles, and practical skills required for professional activity within the chosen field of study. It aims to develop both theoretical knowledge and applied competencies necessary for solving specialized tasks, conducting independent research, and implementing innovative solutions using modern technologies and methodologies relevant to the discipline.

Subject of the discipline. The subject of the discipline is the study and application of methods, models, and technologies for designing, implementing, and managing vector and distributed databases. It focuses on representing complex multidimensional data, organizing vector data structures, performing similarity search and indexing in high-dimensional spaces, and integrating distributed database systems with modern software tools and machine learning pipelines.

Discipline Tasks. To develop practical skills in designing vector entity structures and distributed database architectures; to master approximate similarity search methods and indexing optimization techniques; to study and apply principles of consistency, fault tolerance, and data replication; to acquire experience in implementing distributed and vector database systems using open-source tools; to integrate vector search mechanisms into machine learning workflows; and to strengthen analytical, teamwork, and problem-solving skills relevant to database engineering.

Learning results. Upon successful completion of the course, students should:understand the fundamental principles of vector data representation, similarity search, and distributed data storage; know the architectures and functionalities of modern vector database frameworks (FAISS, ScaNN, Milvus, Qdrant) and distributed DBMS (Cassandra, CockroachDB); be able to design datalogical and physical models, configure indexing structures, and optimize performance for high-dimensional data retrieval; apply knowledge to integrate databases with ML-based systems for semantic and recommendation tasks; justify engineering decisions based on performance analysis and theoretical understanding; demonstrate initiative, independent learning, effective communication, teamwork, and professional responsibility in the development of intelligent data management systems.

4. Structure of credits for the discipline

Title of section (topic)	Number of hours allocated to:		
	lectures	**laboratory classes**	**ISW**
Topic 1. Introduction to Vector and Distributed Databases	2	4	16
Topic 2. Fundamentals of Vector Data Representation	2		
Topic 3. Designing Vector Entity Structures in Databases	2	4	16
Topic 4. Relational Models Extended for Vector Representations	2		
Topic 5. Architecture and Organization of Local Vector Databases	2	4	16
Topic 6. Similarity Search in Vector Spaces	2		
Topic 7. Facebook AI Similarity Search (FAISS): Architecture and Methods	2	4	20
Topic 8. Optimization and Indexing Strategies in High-Dimensional Spaces	2		
Topic 9. Scalable Search and Hybrid Filtering Methods	2	4	20
Topic 10. Fundamentals of Distributed Databases	2		
Topic 11. Designing Datalogical Models and Node Diagrams for Distributed Systems	2	4	20
Topic 12. Consistency, Fault Tolerance, and Replication Mechanisms	2		
Topic 13. Practical Implementation of Distributed Databases Using DBMS Tools	2	4	20
Topic 14. Security, Integrity, and Access Control in Distributed Systems	2		
Topic 15. Integration of Vector Databases with Machine Learning Pipelines	2	8	44
Topic 16. Emerging Trends in Vector and Distributed Databases	2		
Total per semester:	32	36	172

5. Academic discipline program

5.1 Lecture course content

Lecture number	List of lecture topics, their annotations	Number of hours
	Topic 1. Introduction to Vector and Distributed Databases	**2**
1	Concept of vector data representation in modern information systems. Evolution from relational to vector and hybrid storage systems. The role of vector databases in semantic search and recommender systems. Course overview and linkage between lectures and lab sessions. Ref: [1], [2], [4], [5]	2
2	**Topic 2. Fundamentals of Vector Data Representation**	**2**
	Vectorization of textual, visual, and audio data. Feature embedding methods: Word2Vec, FastText, Sentence Transformers, CLIP. Similarity metrics such as cosine, Euclidean, Manhattan, and dot product. Spatial characteristics of vectors and the curse of dimensionality. Ref: [1], [2], [8]	2
	Topic 3. Designing Vector Entity Structures in Databases	**2**
3	Modeling entities with vector attributes. Schema design for vector storage. Mapping between relational and vector-based structures. Implementation examples using Milvus, Pinecone, and Weaviate. Ref: [1], [2], [3], [8]	2
	Topic 4. Relational Models Extended for Vector Representations	**2**
4	Transition from traditional ER models to hybrid vector-relational models. Representing semantic relationships in vector space. Implementation with PostgreSQL + pgvector and Qdrant. Ref: [1], [2], [4], [5]	2

	Topic 5. Architecture and Organization of Local Vector Databases	**2**
5	Core components of vector DBMS: storage, indexing, query engine, and API layer. Mechanisms of vector insertion, update, and deletion. Performance bottlenecks and optimization strategies for local deployments. Ref: [2], [3]	2
	Topic 6. Similarity Search in Vector Spaces	**2**
6	Concept and theory of Nearest Neighbor Search (NNS). Exact versus approximate search methods. Fundamentals of Approximate Nearest Neighbor Search (ANNS) and their use cases. Ref: [1], [2], [3]	2
	Topic 7. Facebook AI Similarity Search (FAISS): Architecture and Methods	**2**
7	Overview of FAISS components, data structures, and processing flow. Indexing algorithms such as Flat, IVFPQ, HNSW, and LSH. Optimization and parameter tuning for large-scale similarity search. Ref: [1], [2], [10]	2
	Topic 8. Optimization and Indexing Strategies in High-Dimensional Spaces	**2**
8	Analysis of indexing strategies including HNSW, IVF, PQ, and scalar quantization. Trade-offs between accuracy, latency, and memory consumption. Selection of optimal index configurations based on dataset characteristics. Ref: [1], [2]	2
	Topic 9. Scalable Search and Hybrid Filtering Methods	**2**
9	Google ScaNN architecture and algorithmic innovations. Principles of hybrid filtering combining metadata filtering and vector similarity. Application of scalable vector search for image and multimodal retrieval. Ref: [1], [3]	2
	Topic 10. Fundamentals of Distributed Databases	**2**
10	Concepts of distribution, fragmentation, and replication. Benefits and challenges of distributed storage systems. Comparative analysis of shared-nothing, shared-disk, and shared-memory architectures. Ref: [4], [5], [6]	2
	Topic 11. Designing Datalogical Models and Node Diagrams for Distributed Systems	**2**
11	Logical and physical modeling of distributed database components. Mapping data fragments to nodes. Design examples using Cassandra, MongoDB, and CockroachDB. Ref: [4], [5]	2
	Topic 12. Consistency, Fault Tolerance, and Replication Mechanisms	**2**
12	CAP theorem and consistency models: strong, eventual, and causal consistency. Data replication, load balancing, and high availability. Node failure scenarios, recovery strategies, and resilience testing. Ref: [4], [5]	2
	Topic 13. Practical Implementation of Distributed Databases Using DBMS Tools	**2**
13	Integration of datalogical models with DBMS infrastructure. Configuration and deployment of distributed clusters. Monitoring, scaling, and performance evaluation. Ref: [4], [6]	2
	Topic 14. Security, Integrity, and Access Control in Distributed Systems	**2**
14	Access control policies and authentication in vector and distributed databases. Data confidentiality and encryption mechanisms. Transaction management and concurrency control in distributed environments. Ref: [4], [5]	2
	Topic 15. Integration of Vector Databases with Machine Learning Pipelines	**2**
15	Embedding generation and feature extraction workflows. Integration with TensorFlow, PyTorch, and HuggingFace Transformers. Applications in recommendation, clustering, and retrieval-augmented generation (RAG) systems. Ref: [1], [2]	2
	Topic 16. Emerging Trends in Vector and Distributed Databases	**2**
16	Comparative analysis of Milvus, Pinecone, Weaviate, Qdrant, and Vespa. Hybrid vector-relational and vector-graph systems. Future directions including edge vector search, federated vector storage, and multimodal retrieval architectures. Ref: [5], [6], [7]	2
	Total per semester:	**32**

5.2 Laboratory content

№	Laboratory lesson topic	Number of hours
1	Designing vector entity structures in vector databases. Designing relational data structures to vector data representations. Ref: [1], [2]	4
2	Organizing work with local vector databases. Using «Facebook AI Similarity Search» for associated search of vector similarity using ANNS method. Ref: [10]	4
3	Optimization and indexing strategies in high-dimensional vector databases. Comparative analysis of indexing methods (HNSW, IVF-PQ, Flat Index, PQ) in FAISS or Milvus for improving efficiency of similarity search. Ref: [8], [10]	4
4	Scalable search for image vectors similarity in multidimensional vector spaces. Using the software extension «Google ScaNN» for scalable search for vectors using similarity the hybrid filtering method. Ref: [2], [3]	4
5	Designing datalogical models and node diagrams of distributed databases. Ref: [4], [5]	4
6	Consistency, fault tolerance and replication in distributed databases. Modeling node failures, replication, and recovery in distributed systems (e.g., Cassandra, CockroachDB) to evaluate data consistency and system reliability. Ref: [4], [5]	4
7	Applied implementation of distributed database using DBMS tools. Software integration of distributed database datalogical model. Ref: [5], [6]	4
8	Integration of vector databases with machine learning pipelines. Implementing vectorization, indexing and semantic search workflows with FAISS/Milvus and ML models (e.g., Sentence Transformers, CLIP). Ref: [8], [10]	8
	Total per semester:	**36**

5.3 Content of independent (including individual) work of a higher education student

Independent work of students of all forms of education consists of the systematic study of program material from relevant sources of information, preparation for laboratory work, tests, and completion of individual tasks. Students are provided with the department's page in the Modular Learning Environment, where the Working Program of the discipline and the necessary materials for its educational and methodological support and control of learning results are posted.

Week number	Type of independent work	Number of hours
1-2	Studying lecture material. Preparing for laboratory work No. 1. Preparing for the defense of laboratory work No. 1.	16
3-4	Studying lecture material. Preparing for laboratory work No. 2. Preparing for the defense of laboratory work No. 2.	16
5-6	Studying lecture material. Preparing for laboratory work No. 3. Preparing for the defense of laboratory work No. 3.	16
7-8	Studying lecture material. Preparing for laboratory work No. 4. Preparing for the defense of laboratory work No. 4. Beginning of individual task No. 1 of independent work.	20
9-10	Studying lecture material. Preparing for laboratory work No. 5. Preparing for the defense of laboratory work No. 5. Completing individual task No. 1 of independent work.	20
11-12	Studying lecture material. Preparing for laboratory work No. 6. Preparing for the defense of laboratory work No. 6. Beginning of individual task No. 2 of independent work.	20
13-14	Studying lecture material. Preparing for laboratory work No. 7. Preparing for the defense of laboratory work No. 7.	20
15-16	Studying lecture material. Preparing for laboratory work No. 8. Preparing for the defense of laboratory work No. 8. Completing individual task No. 2 of independent work.	20
17	Studying lecture material. Preparing a report on independent work. Drawing up a report on independent work. Presentation of the results of the individual task.	24
	Total per semester:	**172**

Students are assigned individual homework assignments (IDAs) on relevant topics, as specified in the methodological recommendations for laboratory classes and independent work. The guidance of independent work and control over the implementation of individual assignments is carried out by the teacher according to the schedule of consultations outside of class time.

The requirements for completing individual homework are outlined in the Modular Learning Environment on the subject page.

6. Technologies and teaching methods

The process of learning in the discipline is based on the use of traditional and modern technologies and teaching methods, in particular: lectures (using visualization methods, problem-based and interactive learning, motivational techniques, information and communication technologies); practical classes (using instruction, demonstration, solving typical and applied problems, case analysis, situational tasks, discussion elements, etc.); independent work (studying theoretical material, preparing for practical work, current and final control, performing individual and homework), using information and computer technologies and distance learning technologies.

7. Control methods

Current control is carried out during classroom laboratory classes, as well as on the days of control events established by the work program and schedule of the educational process, including using the Modular Learning Environment. The following methods of current control are used:

- oral interview before admission to the laboratory session;
- evaluation of the results of laboratory work defense;
- performing control work;
- presentation and defense of individual tasks.

The percent of the maximum score in any type of academic work is considered ***to have*** academic debt. Academic debt from semester control is eliminated during the examination session or according to the schedule established by the dean's office in accordance with the "Regulations on Control and Evaluation of Learning Results of Higher Education Applicants at KhNU".

8. Discipline Policy

The policy of the academic discipline is generally determined by the system of requirements for a higher education applicant, which are stipulated by the current regulations of the University on the organization and educational and methodological support of the educational process. In particular, passing a safety briefing; attending classes on the discipline is mandatory. For objective reasons (confirmed by documentation), theoretical training, in agreement with the lecturer, can take place online. Successful mastery of the discipline and the formation of professional competencies and program learning outcomes requires preparation for a laboratory class (studying theoretical material on the topic of the work, preparation for an oral interview for admission to the class (given in the Methodological recommendations for laboratory classes)), active work in the class, qualitatively prepare a report, defend the results of the work performed, participate in discussions regarding decisions made when performing laboratory work by applicants, etc.

Higher education students must adhere to the established deadlines for completing all types of academic work in accordance with the work program of the academic discipline. The deadline for defending laboratory work is considered timely if the student defends it at the next lesson after completing the work. The student is obliged to complete the missed laboratory lesson within the deadline established by the teacher, but no later than two weeks before the end of theoretical classes in the semester.

The student's assimilation of theoretical material in the discipline is checked in the process of current control during laboratory classes. The completion of an individual task is completed by its presentation within the time limits established by the schedule of independent work.

A higher education applicant, when performing individual work on a discipline, must adhere to the integrity policy (copying, plagiarism (including using mobile devices) are prohibited). If a violation of the academic integrity policy is detected in any type of academic work, the higher education applicant receives an unsatisfactory grade and must re-perform the tasks on the relevant topic (type of work) provided for in the work program. Any forms of violation of academic integrity ***are not allowed.***

Within the framework of studying the academic discipline, higher education applicants are provided with the recognition and enrollment of learning outcomes acquired through informal education, which are posted on accessible platforms that contribute to the formation of competencies and deepening of learning outcomes determined by the work program of the discipline, or ensure the study of the relevant topic and / or type of work from the program of the academic discipline (for more details, see the Regulation on the procedure for recognizing and enrolling learning outcomes of higher education applicants at KhNU).

9. Evaluation of student learning outcomes in the semester

Assessment of academic achievements of a higher education applicant is carried out in accordance with the "Regulations on monitoring and evaluating the results of higher education applicants at KhNU". During the ongoing assessment of the work performed by the applicant in each structural unit and the results obtained by him, the teacher assigns him a certain number of points from the Work Program established for this type of work. In this case, each structural unit of educational work can be credited if the applicant has scored at least 60 percent (the minimum level for a positive assessment) of the maximum possible amount of points assigned to the structural unit.

When assessing the learning outcomes of higher education applicants in any type of educational work (structural unit), it is recommended to use the following generalized criteria:

Table – Criteria for evaluating the academic achievements of a higher education applicant

Assessment and level of achievement by the applicant of planned PND and developed competencies	Generalized content of the evaluation criterion
Excellent (high)	The higher education applicant has deeply and fully mastered the content of the educational material, easily navigates in it and skillfully uses the conceptual apparatus; is able to connect theory with practice, solve practical tasks, confidently express and justify their judgments. An excellent grade assumes a logical presentation of the answer in the language of instruction (oral or written), demonstrates high-quality design of the work and mastery of special devices and tools, applied programs. The applicant does not hesitate when modifying the question, is able to make detailed and generalizing conclusions, demonstrates practical skills in solving professional tasks. When answering, he made two or three insignificant ***errors***.
Good (average)	The higher education applicant has demonstrated complete mastery of the educational material, possesses a conceptual apparatus, is oriented in the studied material; consciously uses theoretical knowledge to solve practical problems; the presentation of the answer is competent, but the content and form of the answer may contain individual inaccuracies, unclear formulation of rules, patterns, etc. The higher education applicant's answer is based on independent thinking. The higher education applicant made two or three ***insignificant errors in the answer***.
Satisfactorily (sufficient)	The higher education applicant has demonstrated knowledge of the basic program material in the amount necessary for further study and practical activity in the profession, copes with the implementation of practical tasks provided by the program. As a rule, the higher education applicant's answer is built at the level of reproductive thinking, the higher education applicant has weak knowledge of the structure of the academic discipline, makes inaccuracies and ***significant errors*** in the answer, hesitates when answering a modified question. At the same time, he has acquired the skills necessary to perform simple practical tasks that meet the minimum assessment criteria and has the knowledge that allows him to eliminate inaccuracies in the answer under the guidance of a teacher.
Unsatisfactorily (insufficient)	The higher education applicant has demonstrated disparate, unsystematic knowledge, is unable to distinguish between the main and secondary, makes mistakes in defining concepts, distorts their meaning, presents the material chaotically and uncertainly, and cannot use knowledge when solving practical tasks. As a rule, an “unsatisfactory” grade is given to a higher education applicant who cannot continue their studies without additional work on studying the academic discipline.

Structuring the discipline by types of academic work and evaluating the learning outcomes of full-time students studying in a semester

Classroom work								**Control measures**		**Independent work**		**Semester control**
Laboratory work No.:								Control work		IDW		Credit
1*	2	3	4	5	6	7	8	1	2	1	2	
Number of points for each type of academic work (minimum-maximum)												By rating
6-10	6-10	6-10	6-10	6-10	6-10	6-10	6-10	3-5	3-5	3-5	3-5	
48-80								**6-10**		**6-10**		**60-100****

Notes: *For any type of academic work in the discipline, the number of points scored below the established minimum, the applicant receives an unsatisfactory grade and must retake it within the period established by the teacher (dean). The institutional grade is established in accordance with the table "**Correlation of the institutional assessment scale and the ECTS assessment scale**".

Assessment of the results of the defense of laboratory work

The laboratory work performed and designed in accordance with the requirements established by the Methodological Recommendations is comprehensively assessed by the teacher during its defense, taking into account the following criteria: independence and correctness of execution; completeness of the answer and knowledge.

The result of the performance and defense by the applicant of each laboratory work is assessed in accordance with the table of Criteria for assessing the academic achievements of the applicant for higher education.

If the applicant's level of knowledge is lower than 60 percent of the maximum score established by the Work Program for each structural unit, the laboratory work is not credited to him and to defend it, he must study the material on the topic of the work in more detail, the methodology for its implementation, correct gross errors and re-appear for its defense at the time designated by the teacher.

Assessment of the control work

The control work involves the performance of five tasks. When assessing the test work, the following are taken into account: the completeness of the answer and the quality of the performance. Each task is assessed with 1 point, the total amount of points for a positive assessment is from 3 to 5.

The distribution of points in the assessment of control work tasks

Number of correct answers	1	2	3	4	5
Percentage of correct answers	**0-59**		**60**	**80**	**100**
Number of points received	0		3	4	5

If you receive a negative grade, you must retake the test before the **next** test date.

Evaluating the results of individual tasks

An individual task (IT) completed and designed in accordance with the requirements specified in the Methodological Recommendations is comprehensively evaluated by the teacher, taking into account the following criteria: independence of execution; correctness of solving the tasks; reasonableness of the choice of solution methods; completeness of explanations and justification of answers; quality of design and compliance with the requirements for the structure and content of the work.

The result of the higher education student's performance of each IZ is assessed in accordance with the table of **Criteria for assessing the academic achievements** of a higher education student, taking into account the level of achievement of the planned program learning outcomes and formed competencies. Based on the results of the defense, the appropriate amount of points is assigned (minimum positive score - 3 points, maximum - 5 points).

If the higher education student has revealed a level of knowledge and performance of the IZ that is lower than 60 percent of the maximum number of points established by the Work Program for this structural unit, the task is not counted. In this case, the student must rework the content of the task, eliminate errors and submit the revised IZ for verification within the terms agreed with the teacher.

The final semester grade on the institutional scale and the ECTS scale is determined in an automated mode after the teacher enters the assessment results in points for all types of educational work into the electronic journal. The correlation between the institutional grading scale and the ECTS grading scale is given below in the table “Correlation”.

A semester grade is issued at the last lesson, provided that the total amount of points accumulated by the applicant in the discipline (other educational component) according to the results of the current control is within the range of 60 to 100 points. In this case, the institutional scale is marked with the grade “passed”, and the ECTS scale is marked with the letter designation of the grade corresponding to the number of points earned by the student in accordance with the Correlation table. The applicant’s presence in this case is not mandatory.

Table – Correlation between institutional grading scale and ECTS grading scale

<table>
<tr><th rowspan="2">ECTS assessment</th><th rowspan="2">Rating scale points</th><th colspan="2">Institutional assessment (level of achievement by a higher education student of the planned learning outcomes in the academic discipline)</th></tr>
<tr><th>Test</th><th>Exam/differentiated test</th></tr>
<tr><td>A</td><td>90-100</td><td rowspan="5">Enrolled</td><td>Excellent – a high level of achievement of the planned learning outcomes in the academic discipline, which indicates the applicant’s unconditional readiness for further study and/or professional activity in the specialty</td></tr>
<tr><td>B</td><td>83-89</td><td rowspan="4">Good – average (maximum sufficient) level of achievement of planned learning outcomes in the academic discipline and readiness for further study and/or professional activity in the specialty

Satisfactory – There are minimally sufficient learning outcomes in the academic discipline for further study and/or professional activity in the specialty.</td></tr>
<tr><td>C</td><td>73-82</td></tr>
<tr><td>D</td><td>66-72</td></tr>
<tr><td>E</td><td>60-65</td></tr>
<tr><td>FX</td><td>40-59</td><td rowspan="2">Not included</td><td>Unsatisfactory / Fail – A number of planned learning outcomes in the academic discipline are missing. The level of acquired learning outcomes is insufficient for further study and/or professional activity in the specialty</td></tr>
<tr><td>F</td><td>0-39</td><td>Unsatisfactory / Fail – No learning outcomes</td></tr>
</table>

10. Questions for self-monitoring learning outcomes

1. What are the main differences between relational and non-relational database models?
2. How does vector data representation improve data retrieval and analysis?
3. What are the main benefits of normalization in database design?
4. How do distributed databases maintain consistency across multiple nodes?
5. What challenges arise when implementing replication in distributed systems?
6. How does the CAP theorem define trade-offs in distributed database design?
7. What is data partitioning and why is it important for scalability?
8. How does sharding enhance query performance?
9. What factors should be considered when selecting a partition key?
10. How is concurrency control implemented in distributed databases?
11. What distinguishes optimistic from pessimistic locking?
12. How does the two-phase commit protocol ensure transactional reliability?

13. What are vector embeddings and how are they generated?
14. How do vector databases differ from traditional key-value or relational databases?
15. In which applications are vector databases most effective?
16. How does similarity search work in vector databases?
17. What is the difference between cosine similarity and Euclidean distance?
18. How does indexing accelerate similarity queries?
19. What are Approximate Nearest Neighbor (ANN) algorithms?
20. How do HNSW and FAISS differ in terms of performance and scalability?
21. Why are ANN algorithms essential for large-scale vector search?
22. How is query optimization performed in distributed databases?
23. How does a query planner distribute tasks across nodes?
24. What factors influence the cost of distributed query execution?
25. What consistency models exist in distributed systems?
26. How do strong, eventual, and causal consistency differ?
27. When is causal consistency preferable?
28. How are failures detected and recovered in distributed databases?
29. What is the role of checkpointing in recovery mechanisms?
30. How does replication enhance fault tolerance?
31. How can data security be ensured in distributed environments?
32. What encryption methods are commonly used for data at rest and in transit?
33. How do access control mechanisms differ between centralized and distributed systems?
34. What indexing strategies are used in vector databases?
35. How do distributed indexing structures handle updates and deletions?
36. How does indexing affect the trade-off between performance and storage?
37. Which modern technologies support distributed and vector databases?
38. How do systems like Cassandra, MongoDB, and Pinecone implement distributed processing?
39. What distinguishes vector-oriented systems from conventional databases?
40. How are machine learning models integrated with vector databases?
41. What is the role of embeddings in semantic search and recommendation systems?
42. How can hybrid architectures combine structured and unstructured data?
43. What are the benefits of cloud-based distributed databases?
44. How do scalability and elasticity differ in cloud environments?
45. What are the trade-offs between managed and self-hosted distributed systems?
46. How does FAISS implement approximate similarity search?
47. What are the main indexing types in FAISS and their use cases?
48. How can FAISS indexes be optimized for large datasets?
49. What is the architecture of Google ScaNN?
50. How does hybrid filtering improve search accuracy and efficiency?
51. How can ScaNN be applied to multimedia or multimodal datasets?
52. What is the role of node diagrams in distributed database modeling?
53. How are data fragments assigned to nodes?
54. How do node diagrams support system scalability and fault tolerance?
55. How can vector databases be integrated into a machine learning pipeline?
56. What preprocessing steps are necessary for vector embeddings?
57. How does semantic search improve retrieval quality in AI applications?
58. What are the main performance metrics for vector and distributed databases?
59. How can throughput and latency be measured and optimized?
60. How do hardware resources affect database performance?
61. What are emerging trends in vector and distributed database research?
62. How is AI transforming database design and operation?
63. What future challenges are anticipated regarding consistency, scalability, and efficiency?

11. Educational and methodological support

The educational process in the discipline "Design of Vector and Distributed Databases" is provided with the necessary educational and methodological materials, which are placed in the modular learning environment MOODLE: Course " Design of Vector and Distributed Databases" https://msn.khmnu.edu.ua/course/view.php?id=7933

12. Material, technical and software support of the discipline

Information and computer support: PC with Windows Education 10 or Windows Education 11 operating system installed, projector. Software: Microsoft Office LTSC Professional Plus 2021 and Microsoft Office LTSC Professional Plus 2024 programs, Internet access.

Studying the academic discipline requires the use of specialized software applications for vector and distributed database design and experimentation, in addition to standard programs and operating systems. These include FAISS, Google ScaNN, Milvus, Cassandra, CockroachDB, Qdrant, and Python programming environment with libraries for data processing and machine learning (e.g., NumPy, Pandas, PyTorch, Transformers). All listed software products are either free, open-source, or available to students under academic licenses.

13. Recommended literature:

Main

1. Nexla. Vector Databases: Tutorial, Best Practices & Examples [Electronic resource]. – 2024. – Available at: https://nexla.com/ai-infrastructure/vector-databases/ (accessed 07 October 2025).
2. Chandeepa, S. A Beginner's Guide to Implementing Vector Databases [Electronic resource]. – 2024. – Available at: https://zilliz.com/learn/beginner-guide-to-implementing-vector-databases (accessed 07 October 2025).
3. Microsoft. Understanding Vector Databases [Electronic resource]. – 2024. – Available at: https://learn.microsoft.com/en-us/data-engineering/playbook/solutions/vector-database/ (accessed 07 October 2025).
4. Jiang, R. How to Build a Dynamic Distributed Database with DistSQL [Electronic resource]. – 2022. – Available at: https://opensource.com/article/22/9/dynamic-distributed-database-distsql (accessed 07 October 2025).
5. Hire, A. What Is a Distributed Database? A Complete Guide [Electronic resource]. – 2025. – Available at: https://www.pingcap.com/article/what-is-a-distributed-database-a-complete-guide/ (accessed 07 October 2025).
6. Leangroup. Distributed Database Management Systems [Electronic resource]. – 2024. – Available at: https://webweb.ams3.cdn.digitaloceanspaces.com/data/simmcdev.webweb.ai.in/MCAdigitalbook/DBMS-20240629T071926Z-001/Distributed_Database_Management_Systems_Leangroup_org.pdf (accessed 07 October 2025).

Supporting

7. PingCAP. Modern Distributed Database Fundamentals [Electronic resource]. – 2023. – Available at: https://static.pingcap.com/files/2023/08/16173906/PingCAP-eBook-Modern-Distributed-Database-Fundamentals.pdf (accessed 07 October 2025).
8. Milvus. Milvus Open-Source Vector Database Documentation [Electronic resource]. – 2024. – Available at: https://milvus.io/docs/v2.3.0/milvus_overview.md (accessed 07 October 2025).
9. Zilliz. Chroma Vector Database Tutorial [Electronic resource]. – 2024. – Available at: https://zilliz.com/blog/chroma-tutorial-vector-database (accessed 07 October 2025).
10. FAISS. Facebook AI Similarity Search Library [Electronic resource]. – 2024. – Available at: https://github.com/facebookresearch/faiss (accessed 07 October 2025).

14. Information resources

1. Modular learning environment. URL: https://msn.khmnu.edu.ua/course/view.php?id=7933
2. University electronic library. URL: http://library.khmnu.edu.ua/
3. KhNU Repository. URL: https://elar.khmnu.edu.ua/home

DESIGN OF VECTOR AND DISTRIBUTED DATABASES

Discipline Status	Selective
Level of Higher Education	Second (master's)
Language of Instruction	English
Семестр	–
Discipline Volume, ECTS Credits	8,0
Forms of education for which the discipline is taught	Full-time (day)

Learning results. Upon successful completion of the course, students should:understand the fundamental principles of vector data representation, similarity search, and distributed data storage; know the architectures and functionalities of modern vector database frameworks (FAISS, ScaNN, Milvus, Qdrant) and distributed DBMS (Cassandra, CockroachDB); be able to design datalogical and physical models, configure indexing structures, and optimize performance for high-dimensional data retrieval; apply knowledge to integrate databases with ML-based systems for semantic and recommendation tasks; justify engineering decisions based on performance analysis and theoretical understanding; demonstrate initiative, independent learning, effective communication, teamwork, and professional responsibility in the development of intelligent data management systems.

Content of the academic discipline. Introduction to Vector and Distributed Databases. Fundamentals of Vector Data Representation. Designing Vector Entity Structures in Databases. Relational Models Extended for Vector Representations. Architecture and Organization of Local Vector Databases. Similarity Search in Vector Spaces. Facebook AI Similarity Search (FAISS): Architecture and Methods. Optimization and Indexing Strategies in High-Dimensional Spaces. Scalable Search and Hybrid Filtering Methods. Fundamentals of Distributed Databases. Designing Datalogical Models and Node Diagrams for Distributed Systems. Consistency, Fault Tolerance, and Replication Mechanisms. Practical Implementation of Distributed Databases Using DBMS Tools. Security, Integrity, and Access Control in Distributed Systems. Integration of Vector Databases with Machine Learning Pipelines. Emerging Trends in Vector and Distributed Databases.

Planned learning activities: The minimum amount of study hours in one ECTS credit of an academic discipline for the second (master's) level of higher education in full-time education is 8 hours per 1 ECTS credit.

Forms (methods) of learning: lectures (using visualization methods, problem-based and interactive learning, motivational techniques, information and communication technologies); practical classes (using instruction, demonstration, solving typical and applied problems, case analysis, situational tasks, discussion elements, etc.); independent work (studying theoretical material, preparing for practical work, current and final control, performing individual and homework), using information and computer technologies and distance learning technologies.

Forms of assessment of learning outcomes: laboratory work, current control work, presentation of the results of individual tasks.

Type of semester control: credit.

Educational resources:

1. Nexla. Vector Databases: Tutorial, Best Practices & Examples [Electronic resource]. – 2024. – Available at: https://nexla.com/ai-infrastructure/vector-databases/ (accessed 07 October 2025).
2. Chandeepa, S. A Beginner's Guide to Implementing Vector Databases [Electronic resource]. – 2024. – Available at: https://zilliz.com/learn/beginner-guide-to-implementing-vector-databases (accessed 07 October 2025).
3. Microsoft. Understanding Vector Databases [Electronic resource]. – 2024. – Available at: https://learn.microsoft.com/en-us/data-engineering/playbook/solutions/vector-database/ (accessed 07 October 2025).
4. Jiang, R. How to Build a Dynamic Distributed Database with DistSQL [Electronic resource]. – 2022. – Available at: https://opensource.com/article/22/9/dynamic-distributed-database-distsql (accessed 07 October 2025).
5. Hire, A. What Is a Distributed Database? A Complete Guide [Electronic resource]. – 2025. – Available at: https://www.pingcap.com/article/what-is-a-distributed-database-a-complete-guide/ (accessed 07 October 2025).
6. Leangroup. Distributed Database Management Systems [Electronic resource]. – 2024. – Available at: https://webweb.ams3.cdn.digitaloceanspaces.com/data/simmcdev.webweb.ai.in/MCAdigitalbook/DBMS-20240629T071926Z-001/Distributed_Database_Management_Systems_Leangroup_org.pdf (accessed 07 October 2025).
7. PingCAP. Modern Distributed Database Fundamentals [Electronic resource]. – 2023. – Available at: https://static.pingcap.com/files/2023/08/16173906/PingCAP-eBook-Modern-Distributed-Database-Fundamentals.pdf (accessed 07 October 2025).
8. Milvus. Milvus Open-Source Vector Database Documentation [Electronic resource]. – 2024. – Available at: https://milvus.io/docs/v2.3.0/milvus_overview.md (accessed 07 October 2025).
9. Zilliz. Chroma Vector Database Tutorial [Electronic resource]. – 2024. – Available at: https://zilliz.com/blog/chroma-tutorial-vector-database (accessed 07 October 2025).
10. FAISS. Facebook AI Similarity Search Library [Electronic resource]. – 2024. – Available at: https://github.com/facebookresearch/faiss (accessed 07 October 2025).
11. Modular learning environment. URL: https://msn.khmnu.edu.ua/course/view.php?id=7933
12. University electronic library. URL: http://library.khmnu.edu.ua/
13. KhNU Repository. URL: https://elar.khmnu.edu.ua/home

Teachers: Ph.D. in E. S., As. Prof. Mazurets O.V.,
Ph.D., Senior lecturer of the CS Department Molchanova M.O.