


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра системного проектування

Затверджено

На засіданні кафедри системного
проектування факультету електроніки та
комп'ютерних технологій
Львівського національного університету
імені Івана Франка
(протокол № 1 від 28.08.2023 р.)

Завідувач кафедри:



Роман ШУВАР

Силабус з навчальної дисципліни
„Нереляційні та розподілені бази даних”,
що викладається в межах ОПП
“Інженерія програмного забезпечення”
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності 121 – Інженерія програмного забезпечення

Львів 2023 р.

Назва дисципліни	Нереляційні та розподілені бази даних
Адреса викладання дисципліни	м. Львів, вул. Драгоманова, 50
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій, кафедра системного проектування
Галузь знань, шифр та назва спеціальності	121 Інженерія програмного забезпечення
Викладачі дисципліни	Ляшкевич Марія Юріївна, асистент кафедри системного проектування
Контактна інформація	mariia.liashkevych@lnu.edu.ua , https://electronics.lnu.edu.ua/employee/liashkevych-m-yu
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекційних занять (за попередньою домовленістю). Також можливі он-лайн консультації через MS Teams або систему електронного навчання Moodle. Для погодження часу онлайн консультацій слід писати на електронну пошту викладача.
Сторінка дисципліни	https://moodle.elct.lnu.edu.ua/enrol/index.php?id=96
Інформація про дисципліну	Дисципліна “ Нереляційні та розподілені бази даних ” є дисципліною з циклу професійної та практичної підготовки спеціальності 121 Програмна інженерія для освітньої програми «Інженерія програмного забезпечення», яка викладається в 5 семестрі в обсязі 4 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Навчальну дисципліну розроблено таким чином, щоб оволодіти базовими поняттями, пов'язаними з організацією та використанням розподілених технологій, управлінні ресурсами віддалених розподілених систем, баз та сховищ даних, використанням технологій розподілених обчислень. З цією метою у дисципліні представлено як огляд базових понять та інструментів розподілених технологій, так і засобів, які потрібні для вирішення типових завдань збереження та опрацювання даних таких як розподілені бази та сховища даних, розроблення програм та програмних інтерфейсів для розподілених додатків і баз даних.
Мета та цілі дисципліни	Метою вивчення дисципліни “ Нереляційні та розподілені бази даних ” є надання поглиблених знань та практичних навичок щодо теорії та практики побудови й використання розподілених систем і паралельних обчислень, формування

	<p>системи теоретичних знань і набуття практичних умінь та навичок щодо застосування, налагодження й адміністрування систем на базі технологій розподілених сховищ даних та проектування відповідних надійних та економічно привабливих систем для збереження великих об'ємів даних.</p> <p>Цілями дисципліни “ Нереляційні та розподілені бази даних ” є засвоєння методів створення розподілених систем та технології їх проектування, наповнення даними та підтримання в робочому стані, вивчення методів і засобів паралельних обчислень та їх застосувань на практиці, а також сформулювати знання про:</p> <ul style="list-style-type: none"> - особливості та характерні ознаки звичайного хостингу веб-ресурсів, оренди віртуальних приватних машин та систем хмарних обчислень; - програмні рішення для серверних систем віртуалізації та комплексні рішення, що здатні сформулювати приватне хмарне середовище підприємства чи корпорації; - основні принципи побудови розподілених файлових сховищ даних та загальні властивості масштабування баз даних; - концепції комп'ютерної реалізації моделей предмету дослідження на основі алгоритмічного, структурного, об'єктно-зорієнтованого, компонентного, аспектно-орієнтованого, сервіс-орієнтованого, мультиагентного та інших сучасних підходів, використовувати концепції паралельної обробки інформації; - задачі автоматичного проектування систем управління, створення та випробування автоматичних систем управління.
<p>Література для вивчення дисципліни</p>	<p>Основна література:</p> <ol style="list-style-type: none"> 1. Kai Hwang, Min Chen. Big-Data Analytics for Cloud, IoT and Cognitive Computing: Willey, 2017. - 428 p. 2. Brendan Burns. Designing Distributed System: O'Reilly, 2018. - 165 p. - [Режим доступу]: https://azure.microsoft.com/mediahandler/files/resourcefiles/designing-distributed-systems/Designing_Distributed_Systems.pdf 3. Gerardus Blokdyk. Databricks A complete Guide, 2021. - 205 p. - [Режим доступу]: https://www.everand.com/book/487839900/Databricks-A-Complete-Guide-2021-Edition 4. Tom White. Hadoop: The definitive Guide: O'Reilly, 2015. - 805 p. 5. MySQL Cluster Manager 8.0.31 User Manual. - 2023. - [Електронний ресурс]. - Режим доступу: https://downloads.mysql.com/docs/mysql-cluster-manager-1.4-en.a4.pdf 6. Anthony Molinaro, Robert de Graaf. SQL Cookbook: O'Reilly, 2021. - Режим доступу: https://downloads.yugabyte.com/marketing-assets/O-Reilly-SQL-Cookbook-2nd-Edition-Final.pdf 7. Shakuntala Gupta Edward, Navin Sabharwal. Practical MongoDB: Architecting, Developing, and Administering MongoDB: Apress, 2015. - 263 p. 8. Michael Armbrust. Makeing Apache Spark better with Delta Lake: Databricks, 2020. - 399 p. 9. Apache HBase Team. Apache HBase™ Reference Guide. - 2022. - [Електронний ресурс]. - Режим доступу: https://hbase.apache.org/apache_hbase_reference_guide.pdf 10. Google. Cloud Bigtable Documentation, 2023. - [Електронний ресурс]. - Режим доступу: https://cloud.google.com/bigtable/docs 11. Nagios. 2019. - [Електронний ресурс]. - Режим доступу: https://www.tutorialspoint.com/nagios/nagios_tutorial.pdf 12. Icinga2 open source monitoring. - 2023. - [Електронний ресурс]. - Режим доступу: https://icinga.com/docs/icinga-2/latest/doc/03-monitoring-basics/

Обсяг курсу	Всього 120 годин, з яких: 64 годин аудиторних занять (32 години лекцій, 32 годин лабораторних робіт) та 56 годин самостійної роботи.
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p>ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК02. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК04. Здатність спілкуватися іноземною мовою як усно, так і письмово.</p> <p>ЗК05. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>ФК13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.</p> <p>ФК14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p> <p>ФК15. Здатність розробляти архітектури, модулі та компоненти програмних систем.</p> <p>ФК22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.</p> <p>ФК24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.</p> <p>ФК25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.</p> <p>ФК27. Здатність використовувати для розробки програмного забезпечення перспективні засоби та технології, зокрема, науки про дані, штучного інтелекту, IoT, вбудованих систем тощо.</p> <p>ФК28. Володіння методами сучасних веб-технологій, хмарних технологій, великих даних та засобами розробки веб-застосунків.</p> <p>ФК29. Здатність здійснювати розробку програмного забезпечення використовуючи сучасні парадигми програмування</p> <p>ПРН01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p> <p>ПРН09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.</p> <p>ПРН10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.</p>

	<p>ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.</p> <p>ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p> <p>ПРН18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.</p> <p>ПРН21. Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.</p> <p>ПРН26. Знати та вміти застосовувати засоби інженерії програмного забезпечення для реалізації проєктів з використанням технологій науки про дані та штучного інтелекту.</p> <p>ПРН27. Вміти обирати оптимальні алгоритми та технології розробки програмного забезпечення.</p>
Ключові слова	Розподілені системи, розподілені та паралельні обчислення, розподілені бази даних, кластери даних, бази даних, великі дані, сховища даних.
Формат курсу	Очний. Проведення лекцій, практичних робіт та консультації для кращого розуміння тем.
Теми	Див. СХЕМА КУРСУ
Підсумковий контроль, форма	Залік в кінці семестру
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін: “Паралельні та розподілені обчислення”, “Функціональне програмування”, “Об’єктно-орієнтоване програмування” та “Бази даних”.
Навчальні методи та Презентація, лекції, лабораторні роботи, обговорення, дискусія. техніки, які будуть використовуватися під час викладання курсу	Презентація, лекції, лабораторні роботи, обговорення, дискусія, елементи неформальної освіти.
Необхідне обладнання	Мультимедійне обладнання, комп’ютерний клас, програми та сервіси MS Teams, Moodle, Databricks, Hadoop, Apache Spark, PySpark, Airflow, Apache Kafka, Python
Критерії оцінювання	Оцінювання проводиться упродовж семестру за 100-бальною

(окремо для кожного виду навчальної діяльності)

шкалою, де враховано бали за два контрольні заміри по 35 балів за кожний модуль та 30 балів за складання заліку.

Бали нараховуються за видами робіт з співвідношенням:

- контрольні заміри (2 модулі): 70% семестрової оцінки; максимальна кількість балів 70, а саме:
 - лабораторні роботи: 60% оцінки контрольного заміру; максимальна кількість балів 42 (8 лабораторних робіт).
 - лекційний матеріал: 40% оцінки контрольного заміру; максимальна кількість балів 28 (2 тести по 14 балів кожний).
- залік: 30% семестрової оцінки, максимум 30 балів.

Оцінки за лабораторні заняття розподіляються наступним чином: виконання лабораторних завдань – 60 %, відповіді на запитання викладача по темі заняття – 40 %.

Бали за лабораторними роботами розподіляються так:

- лаб. робота 1 оцінюється в 2 бали, де від 0 до 1 балу за виконання та від 0 до 1 балу за тестування чи опитування.
- лаб. роботи 2-7 оцінюються 4 бали, де від 0 до 3 балів за виконання та від 0 до 1 балу за тестування чи опитування.
- лаб. робота 8 оцінюється в 6 балів, де від 0 до 5 балів за виконання та від 0 до 1 балу за тестування чи опитування.

Оцінювання залікових питань:

10 балів - розглянута тема відтворюється в повному обсязі, правильно, обґрунтовано, логічно, які містять аналіз і систематизацію, аргументовані висновки. Засвідчено глибоке володіння матеріалом. Наведені приклади коду повністю робочі та відповідають темі. Можуть бути присутні несуттєві помилки та невідповідності;

8 балів - відтворюється значна частина розглянутої теми. Виявлено знання і розуміння основних положень навчальної дисципліни, проте присутні неточності та/або невідповідності основній темі. Наведені приклади коду частково робочі, проте в загальному відповідають темі;

5 балів - відстежується загальне розуміння розглянутої теми. Виявлені множинні неточності та невідповідності, пояснення наведеного коду відсутні, код функціонує із значними неточностями (або відсутні приклади запуску коду на виконання взагалі);

3 бали – студент погано розуміє розглянуту тему. Виявлені суттєві неточності та невідповідності. Наведені приклади коду з суттєвими недоліками, або не відповідають темі;

Менше 3 балів – студент взагалі не розуміє розглянуту тему. Тему не розкрито, кількість викладеного матеріалу не відповідає загальним нормам обраного виду роботи. Наведений код не робочий, або відсутній як такий.

Критерії отримання додаткових балів. Нарахування додаткових балів відбувається за написання тез доповідей, наукових статей, участь у наукових семінарах чи конкурсах,

участь у заходах неформальної освіти за отримання сертифікатів про проходження навчання на різних освітніх платформах (Coursera, Prometheus тощо), курсах на провідних ІТ компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни, але не більше 50 балів в навчальних ресурсів (Coursera, UdeMy, Edx та інші), та до 100 балів при отриманні професійних сертифікатів, наприклад від MongoDB, Microsoft, Oracle та ін.

Сертифікація на незалежних ресурсах також є елементом неформальної освіти, тому тільки дозволяє оцінити свої можливості для працевлаштування, але і може бути зарахована з наступними балами:

- [SQL \(Basic\) Skills Certification](#) (20 балів)
- [SQL \(Intermediate\) Skills](#) (30 балів)
- [SQL \(Advanced\) Skills](#) (50 балів).

У випадку незалежних ресурсах, до уваги береться тільки один сертифікат з найвищим балом. Тобто, за умови, що студент отримав три сертифікати, то йому зарахується один з найвищим балом (SQL Advanced Skills з 50 балами).

Академічна доброчесність: Очікується, що лабораторні та контрольні роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.

Відвідання занять є важливою складовою навчання.

Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття.

Студенти зобов'язані дотримуватися усіх термінів визначених для виконання усіх видів робіт, передбачених курсом.

Література. Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.

Політика виставлення балів. Враховуються бали набрані на поточному тестуванні, самостійній роботі та бали підсумкового тестування. При цьому обов'язково враховуються присутність на заняттях та активність студента під час практичного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та

	<p>плагіат; несвоєчасне виконання поставленого завдання. Жодні форми порушення академічної доброчесності не толеруються.</p>
<p>Питання до контрольних робіт</p>	<p>Перелік питань та завдань для проведення підсумкової оцінки знань певних тем до контрольних робіт:</p> <ol style="list-style-type: none"> 1. Концепція розподілених систем (<i>Приклади типових розподілених систем. Компоненти розподілених систем. Інтернет та інтранет. Використання пристроїв у розподілених системах. Обмін ресурсів в WWW. Веб-сервери та веб-браузери. HTTP/HTTPS протоколи. Програмні та апаратні сервісні рівні. Клієнт-серверні моделі. Веб-прохі сервери.</i>) 2. Веб-застосунки (<i>Веб-застосунки. Мобільні агенти. Комп'ютерні мережі. Синхронні та асинхронні розподілені системи. Події в системі. Порядок подій в реальному часі. Канали зв'язку. Питання захисту.</i>) 3. Поняття нереляційних баз даних (<i>Поняття NoSQL. Модель даних. Типи моделей даних. Логічна модель. Створення фізичної моделі даних. Фізична модель даних для реляційних баз даних. Інструменти для моделювання даних.</i>) 4. MongoDB (<i>Гнучке моделювання даних за допомогою MongoDB Atlas. Приклад моделювання бази даних. Запити і агрегування. Детальніше про мову запитів в MongoDB.</i>) 5. MongoDB SQL (<i>Розуміння зіставлення MongoDB та SQL. MongoDB SQL: термінологія. MongoDB SQL: Виконувати файли бази даних. MongoDB SQL: команди. Приклад CRUD функцій в MongoDB. Побудова запитів в MongoDB.</i>) 6. Графові бази даних (<i>Що таке є мова запитів GraphQL. Переваги GraphQL. GraphQL в порівнянні REST. Схеми GraphQL. Схеми визначення. Вирішувачі (Resolvers). Запит даних. Повернення даних. Apollo GraphQL. Приклад з NodeJS та Express. Приклад запиту для React client.</i>) 7. Резидентна система управління базами даних Redis (<i>Розподілене сховище пар ключ-значення. Конфігурації Redis. Типи даних Redis. Команди з використанням ключа. Redis server. Redis on python. Backup. Безпека Redis. Redis benchmark. Транзакції Redis. Redis pipelining. Redis на docker.</i>) 8. Графова база даних Neo4j (<i>Graph Databases. Приклади графових баз даних. Порівняння графових та реляційних баз даних. Графова база даних Neo4j. Neo4j браузер. Схеми Neo4j. Структура зашифрованого запиту. Написання зашифрованих запитів.</i>) 9. Робота з Neo4j (<i>Схеми Neo4j. Структура зашифрованого запиту. Написання зашифрованих запитів. Створення</i>

та запит до вершин. Встановлення взаємозв'язку між вершинами. Використання транзакційного зашифрованого HTTP end-point. Використовувані протоколи.)

10. База даних Hbase (Кластерна архітектура HBase. Відмінності архітектури HBase від інших розподілених файлових систем. Особливості побудови архітектури HBase. Hbase на Python. Операції HBase. Функції операцій HBase. Огляд переваг операцій HBase над іншими файловими системами.)
11. Особливості розроблення веб-додатків і веб-сервісів із застосуванням технологій розподілених сховищ даних (Поняття веб-додатку та веб-сервісу. Розподілені бази даних. Сховища даних. Розподілені технології. Особливості розподілених сховищ даних)
12. Концепція Hadoop (Знайомство з Hadoop. Історія Hadoop. Компоненти Hadoop. Вузли та демони Hadoop. Архітектура Hadoop. Hadoop характеристики. oogle File System (GFS). HDFS. Концепція та архітектура HDFS. Відмінності архітектури HDFS від інших розподілених файлових систем.)
13. Особливості HDFS (Особливості побудови архітектури HDFS. Операції HDFS. Функції операцій HDFS. Огляд переваг операцій HDFS над іншими файловими системами. Типи запитів. Мовна підтримка. MapReduce. Властивості MapReduce. Трекер задач. Hive. HiveQL. Hadoop Fea.)
14. Різновид реляційних мов запиту (Реляційні мови запитів. Запит як приклад (Query-by-Example). Опис запитів мовою QBE. Вибірка даних з умовою. Базові оператори мови SQL та особливості їх запису.)
15. Формування SQL запитів (Формування запитів мовою SQL. Вибірка рядків конструкцією WHERE. Сортування результатів (конструкція ORDER BY). Вкладені запити (підзапити). Запити для кількох зв'язків. Умовний ящик. Microsoft Access. Datalog.)
16. Знайомство з Apache Spark (Виклики та рішення. Що таке Apache Spark? Модель Spark. Потужний стек – гнучка розробка. Компоненти Apache Spark)
17. Spark SQL (Spark SQL. Інтерфейс програмування. Модель даних. Операції DataFrame. Запити рідних наборів даних. Функції, визначені користувачем. Оптимізація та виконання плану. Логічний план. Фізичний план. Фізичний план з предикатом Pushdown і Column Pruning. Генерація коду. Розширення. Spark MLlib конвеєр.)
18. Масштабування даних (Перша зустріч науки з великими даними. Поняття машинного навчання. Дані і знання. Великі масштаби даних. Надвеликі розміри моделей. Класичні алгоритми машинного навчання. Питання

	<i>масштабування. Стратегія паралелізму. Використання MapReduce.)</i>
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

СХЕМА КУРСУ

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в Інтернеті	Завдання, год	Термін виконання, тиж.
1	Паралельні обчислювальні системи. Поняття паралельної обчислювальної системи. Паралельна обробка. Види паралелізму: багатопроцесорна обробка, конвейєрна обробка, векторна обробка, векторно-конвейєрна обробка. Паралельні обчислювальні системи. Типи паралелізму: Паралелізм на рівні бітів. Паралелізм на рівні інструкцій. Паралелізм даних Паралелізм завдань (багатопоточність).	Лекція	1 - 5	2	кінець поточного тижня
1	Планування завантаження багатопроцесорного конвеєра	Лаб. робота	1 - 5	2	кінець поточного тижня
2	Принципи побудови розподіленої системи. Апаратна побудова розподілених систем. Програмна побудова розподілених систем. Види паралельної взаємодії. Класифікація архітектур паралельних обчислювальних систем: Класифікація Флінна. Класифікація Фенга. Класифікація Хокні. Класифікація Скіллікорна. Поняття та склад проміжного програмного забезпечення грид-систем. Промислові грид-системи, засоби доступу та запуск завдань.	Лекція	1 - 5	2	кінець поточного тижня
2	Планування конфліктів по даних у багатопотокових конвеєрах даних	Лаб. робота	1-5	4	кінець 3-го тижня
3	Принципи розробки та етапи проектування паралельних методів. Аналіз обчислювальних систем і декомпозиція їх на частини (підзадачі), які можуть бути реалізовані незалежно одна від одної. Виділення для сформованого	Лекція	1-5	2	кінець поточного тижня

	набору під задач інформаційних взаємодій, що відбуваються в процесі вирішення поставленої задачі. Визначення обчислювальної системи, необхідної для вирішення задачі. Розподілення набору підзадач між процесорами системи.				
4	Моделювання паралельних програм. Представлення паралельного методу на стадії проектування графом «підзадачі – повідомлення». Модель для опису паралельної програми на стадії виконання у вигляді графа «процеси – канали». Особливості застосування моделі «процеси – канали». Синхронізація процесів та даних в паралельних програмах. Паралелізм задач та паралелізм даних. Виділення інформаційних залежностей. Принципи організації оброблення даних в розподілених системах. Концепція машин потоків даних.	Лекція	1-5	2	кінець поточного тижня
4	Планування конвеєрів даних для систем паралельної обробки даних	Лаб. робота	1-5	4	кінець 5-го тижня
5	Архітектура GFS. Відмінності архітектури GFS від інших розподілених файлових систем. Особливості побудови архітектури GFS. Операції GFS Функції операцій GFS. Огляд переваг операцій GFS над іншими файловими системами.	Лекція	6	2	кінець поточного тижня
6	Розподілені файлові сховища даних на базі технологій SAN та NAS. Масштабування систем збереження даних на базі реляційних СУБД. Кластери на базі MySQL та PostgreSQL.	Лекція	7-8	2	кінець поточного тижня
6	Організація MySQL кластеру засобами AWS і DataBricks	Лаб. робота	3, 5	4	кінець 7-го тижня
7	Мови запитів QBE та SQL. Опис запитів мовою QBE. Вибірка даних з умовою. Базові оператори мови SQL	Лекція	8-9	2	кінець поточного тижня

	та особливості їх запису. Формування запитів мовою SQL. Вибірка рядків конструкцією WHERE. Сортування результатів (конструкція ORDER BY). Вкладені запити (підзапити).				
8	Управління транзакціями. Поняття та властивості транзакції. Порядок виконання операції транзакції. Проблеми управління паралельним доступом. Впорядкування та відновлення транзакцій. Методи управління паралельним доступом. Методи блокувань. Методи обробки взаємоблокувань. Методи управління паралельним доступом з використанням часових відміток. Оптимістичні методи впорядкування транзакцій. Механізми відновлення бази даних. Функції СКБД по відновленню бази даних. Журнал транзакцій. Створення контрольних точок. Метод відкладеного оновлення. Метод поточного оновлення. Метод тіньового сторінкового обміну.	Лекція	3, 5, 8	2	кінець поточного тижня
8	Маштабування MongoDB на AWS	Лаб. робота	7-8	4	кінець 9-го тижня
9	Технології нереляційних СУБД у розподілених сховищах даних. Особливості систем MongoDB, CouchDB та Redis. Маштабування MongoDB.	Лекція	7-8, 10	2	кінець поточного тижня
10	Концепція розподілених баз даних. Основні поняття. Відмінності між розподіленими системами баз даних, засобами розподіленої обробки даних та паралельними системами баз даних. Класифікація розподілених баз даних. Переваги та недоліки системи керування розподіленими базами даних. Принципи створення розподілених БД. Проектування розподіленої БД. Фрагментація даних. Реплікація даних. Розміщення даних.	Лекція	5, 11, 12	2	кінець поточного тижня

	Методологія проектування розподілених БД.				
10	Проектування розподіленої бази даних на основі заданих вимог	Лаб. робота	1-5	4	кінець 11-го тижня
11	Управління паралельним доступом у розподільному середовищі. Протоколи блокування для управління паралельним виконанням у розподіленій базі даних. Усунення взаємних блокувань у розподіленому середовищі. Особливості використання часових відміток для управління паралельним виконанням у розподіленій базі даних. Відновлення розподілених баз даних. Особливості відновлення розподілених баз даних. Протокол двофазної фіксації транзакцій (2PC). Неблокуючий протокол трифазної фіксації транзакцій (3PC). Протоколи відновлення. Протоколи аварійного завершення.	Лекція	11, 12	2	кінець поточного тижня
12	Застосування та особливості проектування рішень на базі розподілених сховищ даних. Поняття про великі дані. Архітектура HDFS. Відмінності архітектури HDFS від інших розподілених файлових систем. Особливості побудви архітектури HDFS. Операції HDFS. Функції операцій HDFS. Огляд переваг операцій HDFS над іншими файловими системами.	Лекція	9, 12	2	кінець поточного тижня
12	Робота з MapReduce. Створення та старт роботи	Лаб. робота	4, 8-9	4	кінець 13-го тижня
13	Особливості розроблення веб-додатків і веб-сервісів із застосуванням технологій розподілених сховищ даних. Архітектура HBase. Відмінності архітектури HBase від інших	Лекція	9-11	2	кінець поточного тижня

	розподілених файлових систем. Особливості побудови архітектури HBase. Операції HBase. Функції операцій HBase. Огляд переваг операцій HBase над іншими файловими системами.				
14	Застосування технологій розподілених сховищ даних у хмарних обчисленнях. Місце та значення технологій розподілених сховищ даних під час оброблення великих масивів даних. Перспективи розвитку систем та технологій розподілених сховищ даних.	Лекція	12	2	кінець поточного тижня
14	Робота з Google BigTable	Лаб. робота	10	6	кінець 16-го тижня
15	Архітектура Google BigTable. Відмінності архітектури Google BigTable від інших розподілених файлових систем. Операції та функції операцій Google BigTable. Переваги операцій Google BigTable над іншими файловими системами.	Лекція	10	2	кінець поточного тижня
16	Моніторинг стану розподілених обчислювальних систем і сховищ даних на базі відкритих програмних засобів Nagios та Icinga.	Лекція	11, 12	2	кінець поточного тижня