

Робоча програма дисципліни «Database Development та трьохланкові застосунки»

СПЕЦІАЛЬНІСТЬ 123 · КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

БАКАЛАВР

120

Годин

Загальний обсяг дисципліни

4

Кредити ЄКТС

Відповідно до стандарту

4

Модулі

Змістовних блоки курсу



Загальна інформація про дисципліну

Ідентифікація

Назва: Database Development та трьохланкові застосунки

Спеціальність: 123 Комп'ютерна інженерія

Ступінь: Бакалавр

Обсяг: 120 годин / 4 кредити ЄКТС

Модулів: 4 змістовних блоки

Призначення дисципліни

Дисципліна «Database Development та трьохланкові застосунки» формує у студентів системне розуміння принципів проєктування, розробки та адміністрування реляційних і нереляційних баз даних, а також навички побудови повноцінних трьохланкових клієнт-серверних застосунків. Курс охоплює моделювання даних, мову SQL, серверну логіку, розробку API та інтеграцію клієнтської частини.

Дисципліна є обов'язковою для підготовки бакалаврів зі спеціальності «Комп'ютерна інженерія» та забезпечує формування фахових компетентностей, передбачених освітньо-професійною програмою.

Загальні компетентності

Аналітичне мислення

Здатність до аналізу та синтезу вимог до інформаційних систем: виявлення сутностей, зв'язків та обмежень, формування концептуальних та логічних моделей даних, побудова обґрунтованих архітектурних рішень.

Комунікація

Здатність до технічної комунікації: чітко описувати архітектурні рішення та структури даних, документувати схеми баз даних і API, презентувати розроблені застосунки та обґрунтовувати прийняті рішення.

Самоорганізація

Уміння самостійно планувати роботу над проєктом із розробки бази даних та трьохланкового застосунку, дотримуватись дедлайнів, виконувати лабораторні роботи та завдання самостійної роботи у визначені строки.

Командна робота

Відповідальність, ініціативність та готовність працювати в команді під час виконання командних лабораторних проєктів, розподілу ролей (frontend, backend, DBA), спільного вирішення технічних завдань.

Студент повинен вміти: аналізувати предметну область та формалізувати вимоги до даних; синтезувати ефективні схеми баз даних; планувати власну діяльність і дотримуватись дедлайнів; коректно взаємодіяти з командою розробників; брати участь у розподілі задач при розробці трьохланкових застосунків.

Фахові компетентності

Проектування баз даних

Здатність проектувати реляційні та нереляційні бази даних: розробка концептуальної (ER-діаграми), логічної та фізичної моделей, нормалізація до 3НФ/BCNF, визначення ключів, зв'язків та обмежень цілісності.

Розробка та адміністрування СУБД

Здатність розробляти та виконувати SQL-запити різної складності (DDL, DML, DCL, TCL), створювати збережені процедури, тригери, функції та представлення; адмініструвати СУБД, керувати транзакціями та забезпечувати цілісність даних.

Трьохланкова архітектура

Знання принципів побудови трьохланкових застосунків (Presentation — Business Logic — Data): розробка серверної частини (backend API), інтеграція з базою даних через ORM, побудова RESTful-сервісів та підключення клієнтської частини.

Безпека та оптимізація

Орієнтування у методах захисту баз даних (автентифікація, авторизація, SQL-ін'єкції), оптимізація запитів (індекси, плани виконання), резервне копіювання та відновлення даних; дотримання стандартів безпеки інформаційних систем.

Результати навчання

01

Проектування структури даних

Аналізувати предметну область, будувати ER-діаграми, розробляти нормалізовані реляційні схеми баз даних з урахуванням вимог цілісності та продуктивності для реальних прикладних задач.

03

Побудова трьохланкових застосунків

Проектувати та розробляти повноцінні трьохланкові застосунки: рівень даних (СУБД), рівень логіки (backend API), рівень представлення (клієнт); забезпечувати взаємодію між рівнями через RESTful або GraphQL інтерфейси.

02

Розробка SQL та серверної логіки

Розробляти складні SQL-запити, збережені процедури, тригери та функції; реалізовувати бізнес-логіку на рівні бази даних та серверного застосунку з використанням сучасних фреймворків.

04

Оптимізація та безпека

Застосовувати методи оптимізації запитів та структур даних; реалізовувати механізми захисту баз даних від несанкціонованого доступу; виконувати резервне копіювання та відновлення даних.

Основи баз даних та моделювання даних

1

Тема 1.1

Вступ до баз даних. Класифікація СУБД. Реляційна модель даних та її основні поняття.

2

Тема 1.2

Концептуальне моделювання даних. ER-діаграми: сутності, атрибути, зв'язки. Нотація Чена та ІЕ.

3

Тема 1.3

Логічне та фізичне проектування. Нормалізація до 3НФ та BCNF. Перехід від ER до реляційної схеми.

4

Тема 1.4

Нереляційні бази даних (NoSQL): документні, графові, ключ-значення. Порівняння з реляційними СУБД.

Вступ до баз даних. Реляційна модель

Лекційний матеріал (2 год.)

Лекція 1: Поняття бази даних та системи управління базами даних (СУБД). Еволюція моделей даних: ієрархічна, мережева, реляційна, об'єктно-реляційна, NoSQL. Основні компоненти СУБД: ядро, мова запитів, засоби адміністрування. Огляд популярних СУБД: PostgreSQL, MySQL, Oracle, MS SQL Server, MongoDB.

Реляційна модель: відношення (таблиці), кортежі (рядки), атрибути (стовпці). Первинний та зовнішній ключі. Обмеження цілісності: NOT NULL, UNIQUE, CHECK, FOREIGN KEY. Реляційна алгебра: проєкція, селекція, з'єднання.

Ключові питання лекції

- Поняття бази даних та СУБД, їх роль у сучасних інформаційних системах
- Реляційна модель Кодда: основні принципи та обмеження
- Класифікація СУБД за моделлю даних
- Обмеження цілісності та їх значення для якості даних
- Огляд сучасних комерційних та відкритих СУБД

Лабораторне заняття (2 год.)

Завдання 1. Встановлення та налаштування PostgreSQL та pgAdmin. Огляд інтерфейсу. Створення бази даних та підключення.

Завдання 2. Побудова простої реляційної схеми для предметної області «Інтернет-магазин»: визначення таблиць, ключів та зв'язків.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (7 год.)

- Опрацювання підручника: Баран І. Ю., Боднар І. В. «Бази даних та СУБД» — розділ «Реляційна модель»
- Підготовка реферату: «Порівняльний аналіз СУБД PostgreSQL та MySQL для веб-застосунків»
- Складання таблиці «Класифікація СУБД та сфери їх застосування»



Очікуваний результат: студент вміє характеризувати реляційну модель даних, розрізняти типи СУБД та встановлювати і налаштовувати PostgreSQL.

Концептуальне моделювання. ER-діаграми

1

Предметна область

Аналіз вимог, виявлення сутностей та їх атрибутів. Опис бізнес-правил.

2

ER-діаграма

Побудова сутностей, атрибутів, зв'язків. Кардинальність: 1:1, 1:N, M:N.

3

Логічна схема

Перетворення ER на реляційні таблиці. Розкриття зв'язків M:N через таблицю-асоціацію.

4

Фізична схема

DDL-скрипти CREATE TABLE. Типи даних СУБД, індекси, обмеження.

Лекційний матеріал (2 год.)

Концептуальне проектування: рівні абстракції (концептуальний, логічний, фізичний). ER-модель: сутності, атрибути (прості, складені, багатозначні, похідні), зв'язки. Нотації: Чена, «воронячої лапки» (IE). Кардинальність та участь сутностей у зв'язках. Слабкі сутності. Побудова ER-діаграм для реальних предметних областей.

Лабораторне заняття (2 год.) та СРС (7 год.)

Лабораторна: Розробка ER-діаграми для предметної області за варіантом (університет, бібліотека, лікарня, склад тощо). Використання draw.io або Lucidchart. Перетворення ER у реляційну схему.

СРС: Опрацювання: Олійник А. О., Шацька В. М. «Бази даних: проектування та реалізація» (2023) — розділ «ER-моделювання»; підготовка звіту з побудови ER-діаграми для індивідуальної предметної області.

Нормалізація та NoSQL бази даних

Тема 1.3 — Нормалізація (Лекція 3, 2 год.)

Функціональні залежності та аномалії (вставки, оновлення, видалення). Нормальні форми: 1НФ, 2НФ, 3НФ, BCNF. Алгоритми нормалізації. Денормалізація: доцільність та наслідки. Практика нормалізації реальних схем баз даних. Цілісність даних на рівні схеми.

Лабораторна (2 год.): Нормалізація ненормалізованої схеми бази даних до 3НФ. Виявлення функціональних залежностей. Реалізація нормалізованої схеми у PostgreSQL (DDL).

СРС (7 год.): Олійник А. О., Шацька В. М. «Бази даних: проектування та реалізація» (2023) — розділ «Нормалізація»; підготовка звіту «Нормалізація бази даних індивідуального проекту до BCNF».

Тема 1.4 — NoSQL бази даних (Лекція 4, 2 год.)

Обмеження реляційних СУБД та передумови NoSQL. Теорема CAP. Класифікація NoSQL: документні (MongoDB), ключ-значення (Redis), стовпчасті (Cassandra), графові (Neo4j). Порівняння з реляційними СУБД. Критерії вибору СУБД для різних задач. Гібридні підходи (NewSQL).

Лабораторна (2 год.): Встановлення MongoDB. Операції CRUD у MongoDB (shell та MongoDB Compass). Порівняння документної та реляційної моделі на прикладі однієї предметної області.

СРС (7 год.): Опрацювання: Дем'яненко В. М., Кравченко Г. В. «Нереляційні бази даних у сучасних веб-застосунках» (2023); підготовка порівняльної таблиці «Реляційні vs NoSQL СУБД: переваги, недоліки, сфери застосування».

SQL: мова структурованих запитів та адміністрування СУБД

Тема 2.1

DDL та DML: CREATE, ALTER, DROP, INSERT, UPDATE, DELETE.
Базові SELECT-запити, фільтрація, сортування.

Тема 2.2

Складні SQL-запити: JOIN (INNER, LEFT, RIGHT, FULL),
підзапити, агрегатні функції, GROUP BY, HAVING, вікові функції.

Тема 2.3

Процедурне розширення SQL: збережені процедури, функції,
тригери, представлення (VIEW), індекси.

Тема 2.4

Транзакції, конкурентний доступ, ізоляція. Адміністрування
СУБД: ролі, привілеї, резервне копіювання.

DDL, DML та базові SELECT-запити

Лекційний матеріал (2 год.)

Категорії SQL: DDL (CREATE, ALTER, DROP, TRUNCATE), DML (INSERT, UPDATE, DELETE, MERGE), DCL (GRANT, REVOKE), TCL (COMMIT, ROLLBACK, SAVEPOINT). Типи даних PostgreSQL: числові, рядкові, дата/час, булеві, масиви, JSON. Базовий SELECT: проєкція, WHERE, ORDER BY, LIMIT/OFFSET, DISTINCT. Псевдоніми (AS). Оператори порівняння, логічні оператори, LIKE, IN, BETWEEN, IS NULL.

Ключові питання

- Категорії SQL та їх призначення
- Типи даних PostgreSQL та вибір відповідного типу для атрибута
- Синтаксис та семантика SELECT-запиту
- Фільтрація та сортування результатів
- Операції вставки, оновлення та видалення даних

Лабораторне заняття (2 год.)

Завдання 1. Створення схеми бази даних індивідуального проекту за допомогою DDL-скриптів. Заповнення тестовими даними (INSERT).

Завдання 2. Виконання серії SELECT-запитів різної складності до індивідуальної бази даних: фільтрація, сортування, агрегація без групування.

Завдання 3. Тестування за ключовими термінами (10 питань).

Самостійна робота (7 год.)

- Опрацювання: Баран І. Ю., Боднар І. В. «Бази даних та СУБД» (2022) — розділ «Мова SQL»
- Підготовка звіту: написання та виконання 20 DML-запитів до власної схеми
- Вивчення документації PostgreSQL: типи даних та вбудовані функції



Очікуваний результат: студент вміє створювати схеми баз даних за допомогою DDL та виконувати базові операції DML і SELECT-запити.

Складні SQL-запити: JOIN, підзапити, агрегація



Лекційний матеріал (2 год.)

Операція JOIN: INNER JOIN, LEFT/RIGHT OUTER JOIN, FULL OUTER JOIN, CROSS JOIN, SELF JOIN. Умови з'єднання. Підзапити: у WHERE, FROM (похідні таблиці), у SELECT. Корельовані підзапити. Оператори EXISTS, IN, ANY, ALL. Загальні табличні вирази (CTE) з WITH. Агрегатні функції. GROUP BY та HAVING. Вікові (аналітичні) функції: RANK, DENSE_RANK, ROW_NUMBER, LAG, LEAD, NTILE.

Лабораторне заняття (2 год.) та СРС (7 год.)

Лабораторна: Виконання багатотабличних JOIN-запитів до бази даних індивідуального проєкту (мінімум 5 JOIN-запитів різних типів); написання підзапитів та CTE; застосування агрегатних та вікових функцій.

СРС: Олійник А. О., Шацька В. М. «Бази даних: проєктування та реалізація» (2023) — розділ «Складні запити»; підготовка звіту з 15 складних запитів із поясненням логіки кожного.

Процедурне SQL, транзакції та адміністрування СУБД

Тема 2.3 — Процедурне розширення SQL (Лекція 7, 2 год.)

PL/pgSQL: структура блоку, змінні, умовні оператори, цикли. Збережені процедури (PROCEDURE): параметри, виклик, переваги. Функції (FUNCTION): скалярні, табличні. Тригери: BEFORE/AFTER INSERT/UPDATE/DELETE, INSTEAD OF. Представлення (VIEW): прості та матеріалізовані. Індеси: B-tree, Hash, GIN, GiST; планувальник запитів, EXPLAIN ANALYZE.

Лабораторна (2 год.): Написання збережених процедур та функцій для бізнес-логіки індивідуального проєкту. Створення тригерів для аудиту змін даних. Аналіз планів виконання запитів за допомогою EXPLAIN.

СРС (7 год.): Баран І. Ю., Боднар І. В. «Бази даних та СУБД» (2022) — розділ «PL/pgSQL»; підготовка звіту зі збереженими процедурами та тригерами для власної схеми.

Тема 2.4 — Транзакції та адміністрування (Лекція 8, 2 год.)

Транзакції: властивості ACID (Atomicity, Consistency, Isolation, Durability). TCL: BEGIN, COMMIT, ROLLBACK, SAVEPOINT. Рівні ізоляції: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE. Проблеми конкурентності: брудне читання, неповторне читання, фантомне читання. Блокування. Адміністрування PostgreSQL: ролі та привілеї (GRANT/REVOKE), резервне копіювання (pg_dump, pg_restore), відновлення після збоїв.

Лабораторна (2 год.): Демонстрація рівнів ізоляції транзакцій. Налаштування ролей та привілеїв. Виконання резервного копіювання та відновлення бази даних.

СРС (7 год.): Підготовка аналітичної записки «ACID-властивості транзакцій: практичне значення для розробника».

Серверна розробка та інтеграція з базами даних

Тема 3.1

Серверна розробка на Node.js/Python. Підключення до СУБД. Драйвери та бібліотеки доступу до даних.

Тема 3.2

ORM-фреймворки: Sequelize, SQLAlchemy, Prisma. Міграції схеми. Зіставлення об'єктів і таблиць.

Тема 3.3

Розробка RESTful API: маршрутизація, HTTP-методи, обробка запитів, валідація даних, обробка помилок.

Тема 3.4

Безпека серверних застосунків: автентифікація, авторизація, JWT, захист від SQL-ін'єкцій та XSS.

Серверна розробка. Підключення до СУБД

Лекційний матеріал (2 год.)

Огляд серверних технологій: Node.js (Express.js), Python (FastAPI, Flask), Java (Spring Boot). Архітектура серверного застосунку. Підключення до PostgreSQL: драйвери (node-postgres, psycopg2, JDBC). Пул з'єднань (connection pool): призначення, налаштування. Виконання параметризованих запитів (prepared statements) для захисту від SQL-ін'єкцій. Обробка результатів запитів у серверному коді.

Ключові питання

- Роль серверного рівня у трьохланковій архітектурі
- Принципи підключення серверного застосунку до СУБД
- Пул з'єднань: навіщо він потрібен і як налаштувати
- Параметризовані запити як засіб захисту від ін'єкцій
- Обробка помилок при роботі з базою даних

Лабораторне заняття (2 год.)

Завдання 1. Створення серверного застосунку на Node.js (Express) або Python (FastAPI). Підключення до PostgreSQL через відповідний драйвер. Налаштування пулу з'єднань.

Завдання 2. Реалізація функцій для виконання CRUD-операцій через серверний код (параметризовані запити). Тестування за допомогою Postman.

Завдання 3. Тестування за ключовими термінами теми (10 питань).

Самостійна робота (7 год.)

- Опрацювання: Гладченко М. В., Пасічник В. В. «Серверна розробка та бази даних» (2023) — розділ «Підключення до СУБД»
- Підготовка звіту: реалізація підключення до БД та базових CRUD-функцій
- Вивчення документації до обраного серверного фреймворку



Очікуваний результат: студент вміє створювати серверний застосунок, налаштовувати підключення до СУБД та виконувати параметризовані запити.

ORM-фреймворки та міграції схеми

Лекційний матеріал (2 год.)

Концепція ORM (Object-Relational Mapping): переваги та недоліки порівняно з «сирим» SQL. Огляд популярних ORM: Sequelize (Node.js), SQLAlchemy (Python), Prisma (Node.js/TypeScript), Hibernate (Java). Визначення моделей (сущностей). Асоціації: hasOne, hasMany, belongsToMany. Виконання запитів через ORM: findAll, findOne, create, update, destroy. Ледаче та жадібне завантаження (lazy/eager loading). Міграції схеми: версіонування структури БД, команди up/down.

Лабораторне заняття (2 год.) та СРС (7 год.)

Лабораторна: Підключення ORM (Sequelize або Prisma) до серверного застосунку. Визначення моделей відповідно до схеми індивідуального проєкту. Написання та виконання міграцій. Реалізація CRUD-операцій через ORM з асоціаціями.

СРС: Гладченко М. В., Пасічник В. В. «Серверна розробка та бази даних» (2023) — розділ «ORM та міграції»; підготовка звіту з реалізацією повного набору CRUD-операцій через ORM для індивідуальної бази даних.

❏ **Очікуваний результат:** студент вміє налаштовувати ORM, визначати моделі та асоціації, виконувати запити через ORM та керувати міграціями схеми.

Розробка RESTful API



Маршрутизація та HTTP-методи

GET, POST, PUT, PATCH, DELETE.
Ресурсо-орієнтовані URL.
Параметри шляху та рядка запиту.



Валідація та обробка помилок

Валідація тіла запиту (Joi, Zod, Pydantic). Коды статусу HTTP.
Стандартизовані відповіді про помилки.



Документування API

Swagger/OpenAPI специфікація.
Автоматична генерація документації. Тестування через Swagger UI та Postman.

Лекційний матеріал (2 год.)

Принципи REST: ресурси, представлення, стан, однорідний інтерфейс. HTTP-методи та їх семантика. Формат відповіді JSON. Версіонування API. Пагінація, фільтрація та сортування через параметри запиту. Обробка помилок та HTTP-коди статусу (2xx, 4xx, 5xx). Документування API за допомогою Swagger/OpenAPI. Middleware: логування, обробка CORS, парсинг тіла запиту.

Лабораторне заняття (2 год.) та СРС (7 год.)

Лабораторна: Розробка повноцінного RESTful API для індивідуального проекту: всі CRUD-ендпоінти, валідація вхідних даних, стандартизовані відповіді, обробка помилок. Документування за допомогою Swagger. Тестування у Postman.

СРС: Олексієнко С. І., Сердюк П. В. «Розробка RESTful API та трьохланкових застосунків» (2024) — розділ «REST архітектура»; підготовка звіту з повним набором ендпоінтів та Swagger-документацією.

Безпека серверних застосунків та баз даних

Лекційний матеріал (2 год.)

Загрози безпеці: SQL-ін'єкції, XSS, CSRF, несанкціонований доступ. Методи захисту від SQL-ін'єкцій: параметризовані запити, білі списки. Автентифікація: сесії, токени, JWT (JSON Web Token). Авторизація: RBAC (Role-Based Access Control). Хешування паролів (bcrypt, Argon2). HTTPS та TLS. Захист на рівні СУБД: мінімальні привілеї, шифрування стовпців, аудит доступу. OWASP Top 10 для веб-застосунків.

Ключові питання

- SQL-ін'єкція: механізм атаки та методи захисту
- JWT: структура, видача, перевірка токена
- RBAC: ролі та привілеї у серверному застосунку та СУБД
- Хешування паролів: навіщо і як правильно
- OWASP Top 10: найпоширеніші вразливості

Лабораторне заняття (2 год.)

Завдання 1. Реалізація автентифікації та авторизації у серверному застосунку: реєстрація, вхід, видача JWT, захищені маршрути з перевіркою токена.

Завдання 2. Демонстрація SQL-ін'єкції та захист від неї (параметризовані запити). Налаштування ролей та привілеїв у PostgreSQL для серверного користувача.

Завдання 3. Аудит безпеки власного API за чек-листом OWASP.

Самостійна робота (7 год.)

- Олексієнко С. І., Сердюк П. В. «Розробка RESTful API та трьохланкових застосунків» (2024) — розділ «Безпека»
- Підготовка звіту: реалізація JWT-автентифікації у проєкті
- Аналіз вразливостей OWASP Top 10 з прикладами у контексті баз даних



Очікуваний результат: студент вміє реалізовувати JWT-автентифікацію, налаштовувати авторизацію на основі ролей та захищати застосунок від типових вразливостей.

Трьохланкова архітектура та повноцінні застосунки

Тема 4.1

Архітектура трьохланкових застосунків: рівні даних, логіки та представлення. Патерни проєктування MVC, Repository, Service Layer.

Тема 4.2

Клієнтська частина (Frontend): взаємодія з API, SPA-підхід, React/Vue. Стан застосунку та управління даними на клієнті.

Тема 4.3

Розгортання та DevOps для баз даних: Docker, Docker Compose для СУБД та серверного застосунку. CI/CD пайплайни.

Тема 4.4

Захист та підсумковий проєкт: розробка та презентація повноцінного трьохланкового застосунку з базою даних.

Архітектура трьохланкових застосунків

Лекційний матеріал (2 год.)

Трьохланкова архітектура: рівень даних (Data Tier) — СУБД; рівень логіки (Logic Tier) — серверний застосунок; рівень представлення (Presentation Tier) — клієнт. Патерн MVC (Model-View-Controller): ролі компонентів, потік даних. Патерн Repository: абстракція доступу до даних. Service Layer: інкапсуляція бізнес-логіки. DTO (Data Transfer Object). Принципи SOLID у контексті серверної розробки. Порівняння монолітної та мікросервісної архітектури.

Ключові питання

- Призначення кожного рівня трьохланкової архітектури
- Патерн MVC та його реалізація у серверному фреймворку
- Repository та Service Layer: навіщо та як реалізовувати
- DTO: формування відповідей API без витоку деталей схеми БД
- Моноліт vs мікросервіси: критерії вибору

Лабораторне заняття (2 год.)

Завдання 1. Рефакторинг серверного застосунку індивідуального проєкту відповідно до патернів Repository та Service Layer. Виокремлення шарів: маршрутизатори → контролери → сервіси → репозиторії → СУБД.

Завдання 2. Визначення DTO-класів для всіх ендпоінтів API. Перевірка відповідності патерну MVC.

Завдання 3. Тестування рефакторингу: всі ендпоінти зберігають функціональність.

Самостійна робота (7 год.)

- Олексієнко С. І., Сердюк П. В. «Розробка RESTful API та трьохланкових застосунків» (2024) — розділ «Архітектурні патерни»
- Підготовка звіту з архітектурою проєкту: діаграма компонентів
- Аналіз прикладів реалізації Repository Pattern у відкритих проєктах на GitHub



Очікуваний результат: студент вміє структурувати серверний застосунок відповідно до трьохланкової архітектури та патернів Repository і Service Layer.

Клієнтська частина. Інтеграція Frontend з API



Лекційний матеріал (2 год.)

Рівень представлення у трьохланковій архітектурі. SPA (Single Page Application) та традиційні веб-застосунки. Огляд React/Vue. HTTP-клієнт: Fetch API, Axios — відправка запитів, обробка відповідей, перехоплювачі (interceptors). Управління станом: локальний стан, Context API. Форми та валідація на клієнті. Збереження JWT та реалізація захищених маршрутів. CORS: причини та вирішення.

Лабораторне заняття (2 год.) та СРС (7 год.)

Лабораторна: Розробка клієнтської частини для індивідуального проєкту (React або Vue). Підключення до власного backend API через Axios. Реалізація CRUD-інтерфейсу, форм та таблиць. Інтеграція JWT-автентифікації на клієнті.

СРС: Дем'яненко В. М., Кравченко Г. В. «Нереляційні бази даних у сучасних веб-застосунках» (2023) — розділ «Frontend-інтеграція»; підготовка звіту з клієнтською частиною проєкту.

Розгортання. Docker та DevOps для баз даних

Лекційний матеріал (2 год.)

Контейнеризація: поняття Docker-образу та контейнера. Dockerfile для серверного застосунку. Docker Compose: оркестрація кількох сервісів (СУБД + backend + frontend). Мережі та томи Docker. Змінні середовища (.env) для конфігурації підключення до БД. Хмарні платформи для PostgreSQL: Railway, Supabase, Neon. Основи CI/CD: автоматизовані тести, деплой при push. Моніторинг та логуювання серверного застосунку і СУБД.

Ключові питання

- Docker та контейнеризація: переваги для розробки та розгортання
- Docker Compose для запуску повного стеку локально
- Безпечне зберігання конфігурації підключення до БД
- Хмарні managed-сервіси для PostgreSQL
- CI/CD пайплайн для застосунку з базою даних

Лабораторне заняття (2 год.)

Завдання 1. Написання Dockerfile для backend-застосунку. Створення docker-compose.yml зі службами: PostgreSQL, backend, frontend. Запуск повного стеку через Docker Compose.

Завдання 2. Налаштування змінних середовища для підключення до СУБД. Перевірка persistent volumes для даних PostgreSQL.

Завдання 3. Розгортання застосунку у хмарі (Railway або Render) з хмарною базою даних.

Самостійна робота (7 год.)

- Гладченко М. В., Пасічник В. В. «Серверна розробка та бази даних» (2023) — розділ «Docker та розгортання»
- Підготовка звіту з Docker Compose конфігурацією та посиланням на розгорнутий застосунок
- Налаштування базового CI/CD пайплайну (GitHub Actions)



Очікуваний результат: студент вміє контейнеризувати трьохланковий застосунок за допомогою Docker Compose та розгортати його у хмарному середовищі.

Підсумковий проєкт: повноцінний трьохланковий застосунок

Лекційний матеріал (2 год.)

Огляд повного циклу розробки застосунку з базою даних: від аналізу вимог до розгортання. Типові архітектурні помилки та антипатерни (God Object, N+1 запит, зберігання паролів у відкритому вигляді, відсутність міграцій). Оптимізація продуктивності: кешування (Redis), індекси, пагінація на рівні СУБД. Масштабування: горизонтальне та вертикальне. GraphQL як альтернатива REST. Тестування: unit-тести сервісів, інтеграційні тести API, seed-скрипти для тестових даних.

Ключові питання

- Антипатерн N+1 запит та методи його усунення (eager loading, DataLoader)
- Кешування з Redis: стратегії та інвалідація кешу
- GraphQL vs REST: критерії вибору
- Тестування застосунків із базою даних
- Масштабування бази даних: реплікація, шардинг

Лабораторне заняття (2 год.)

Завдання 1. Завершення та демонстрація індивідуального проєкту: повноцінний трьохланковий застосунок (PostgreSQL + backend API + frontend). Перевірка всіх функціональних вимог.

Завдання 2. Виявлення та усунення N+1 проблеми у власному застосунку (eager loading через ORM).

Завдання 3. Підготовка та захист презентації проєкту: архітектурна діаграма, ER-діаграма, демо застосунку, посилання на репозиторій та розгорнутий застосунок.

Самостійна робота (7 год.)

- Олексієнко С. І., Сердюк П. В. «Розробка RESTful API та трьохланкових застосунків» (2024) — розділ «Оптимізація та тестування»
- Написання unit-тестів для сервісного рівня застосунку
- Підготовка технічної документації проєкту (README.md)



Очікуваний результат: студент розробив і задемонстрував повноцінний трьохланковий застосунок із нормалізованою базою даних, RESTful API та клієнтським інтерфейсом.

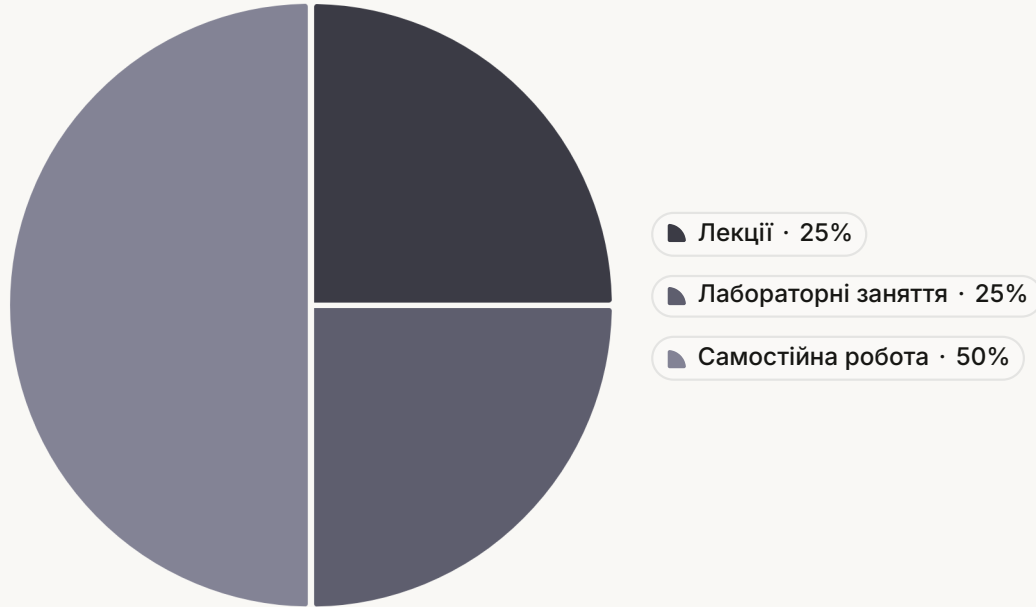
Розподіл навчального часу — 120 годин

Модуль	Тема	Лекції (год.)	Лаборат. (год.)	СРС (год.)
Модуль 1	Тема 1.1 — Вступ до БД. Реляційна модель	2	2	7
	Тема 1.2 — ER-моделювання	2	2	7
	Тема 1.3 — Нормалізація до BCNF	2	2	7
	Тема 1.4 — NoSQL бази даних	2	2	7
Модуль 2	Тема 2.1 — DDL, DML та базові SELECT	2	2	7
	Тема 2.2 — JOIN, підзапити, агрегація	2	2	7
	Тема 2.3 — Процедурне SQL, тригери, індекси	2	2	7
	Тема 2.4 — Транзакції та адміністрування	2	2	7
Модуль 3	Тема 3.1 — Серверна розробка. Підключення до БД	2	2	7
	Тема 3.2 — ORM та міграції	2	2	7
	Тема 3.3 — RESTful API	2	2	7
	Тема 3.4 — Безпека застосунків та БД	2	2	7
Модуль 4	Тема 4.1 — Трьохланкова архітектура	2	2	7
	Тема 4.2 — Frontend. Інтеграція з API	2	2	7
	Тема 4.3 — Docker та розгортання	2	2	7
	Тема 4.4 — Підсумковий проєкт	2	2	4
Разом	16 тем	30	30	60



Загальний обсяг: 120 годин = 4 кредити ЄКТС. Розподіл є орієнтовним і може коригуватися відповідно до специфіки навчального процесу.

Структура навчального часу



Коментар до розподілу

Загальний обсяг — **120 годин (4 кредити ЄКТС)** — розподілено між трьома формами роботи для забезпечення балансу між теорією та практикою.

Лекції — 30 год. (25%)

Системне подання теоретичного матеріалу з проектування баз даних, SQL, серверної розробки та трьохланкової архітектури.

Лабораторні — 30 год. (25%)

Практична розробка бази даних та трьохланкового застосунку: від ER-моделювання до розгортання в Docker. Рівна частка з лекціями підкреслює практичну спрямованість курсу.

Самостійна робота — 60 год. (50%)

Поглиблене вивчення, розробка індивідуального проекту, підготовка звітів та робота з науковою літературою. Самостійна робота становить **50% обсягу дисципліни**.

Самостійна робота студентів — Детальний опис



Опрацювання літератури

Поглиблене вивчення теоретичного матеріалу за підручниками з баз даних та серверної розробки (Баран, Олійник, Гладченко, Олексієнко та ін.), документацією PostgreSQL, MongoDB, Sequelize. Студент конспектує ключові концепції та порівнює підходи різних авторів.



Індивідуальні завдання та проєкт

Виконання практичних завдань (SQL-запити, процедури, API-ендпоінти); поетапна розробка індивідуального проєкту — повноцінного трьохланкового застосунку (БД + backend + frontend), звіти до кожної лабораторної роботи.



Робота з інформаційними ресурсами

Опрацювання офіційної документації: [postgresql.org/docs](https://www.postgresql.org/docs/), [mongodb.com/docs](https://www.mongodb.com/docs/), sequelize.org, fastapi.tiangolo.com. Аналіз відкритих проєктів на GitHub. Пошук наукових статей у НБУВ та Google Scholar.



Підготовка до занять та контролю

Підготовка до лабораторних занять (опрацювання теорії, написання коду заздалегідь); підготовка до тестувань (ключові терміни, SQL-синтаксис); підготовка до захисту індивідуального проєкту та модульних контрольних робіт.

Форми контролю знань

1

Поточний контроль

Завдання: тестування за ключовими термінами та SQL-синтаксисом, перевірка звітів до лабораторних робіт. Оцінюються правильність виконання практичних завдань, якість SQL-коду та рівень засвоєння теоретичного матеріалу. Проводиться після кожної теми у формі тестування (10–15 питань) або захисту лабораторної роботи.

2

Проміжний контроль

Захист поточного стану індивідуального проєкту — демонстрація розробленої схеми БД, SQL-запитів та серверної частини. Оцінюються повнота реалізованого функціоналу, якість коду, відповідність архітектурним вимогам та обґрунтованість прийнятих рішень. Проводиться після завершення кожного змістовного модуля.

3

Підсумковий контроль

Захист індивідуального проєкту (повноцінний трьохланковий застосунок) та іспит відповідно до навчального плану. Форма проведення — тестування, письмові відповіді та демонстрація проєкту. Підсумкова оцінка враховує результати поточного, проміжного та підсумкового контролю: **100-бальна система** з переведенням у національну шкалу та ЄКТС (A, B, C, D, E, FX, F).

Методи викладання

Викладання дисципліни «Database Development та трьохланкові застосунки» поєднує класичні академічні та практико-орієнтовані методи, що забезпечує формування як теоретичної бази з проєктування та розробки баз даних, так і практичних навичок розробки повноцінних застосунків, необхідних для майбутньої професійної діяльності у сфері комп'ютерної інженерії.



Лекції

Системне подання теоретичного матеріалу з проєктування БД, SQL та архітектури застосунків з використанням діаграм, схем та реальних прикладів.



Лабораторні роботи

Практична реалізація завдань: від написання SQL-запитів до розгортання повноцінного трьохланкового застосунку. Кожна лабораторна оформлюється звітом.



Проектна робота

Розробка наскрізного індивідуального проєкту протягом усього курсу: від ER-діаграми та схеми БД до розгорнутого застосунку з frontend-інтерфейсом.



Технічні дискусії

Обговорення архітектурних рішень: реляційна vs NoSQL, REST vs GraphQL, ORM vs «сирий» SQL. Розвиток критичного мислення та аргументації.



Code Review

Взаємна перевірка SQL-скриптів та серверного коду; аналіз типових помилок (N+1, відсутність індексів, небезпечні запити); формування навичок рецензування.



Архітектурне моделювання

Побудова ER-діаграм, діаграм компонентів та архітектурних схем; формування навичок візуалізації та документування технічних рішень.



Усі методи спрямовані на формування фахових компетентностей, передбачених освітньо-професійною програмою спеціальності 123 «Комп'ютерна інженерія».

Рекомендована література — Підручники та посібники (2022–2024 рр.)

1. Баран І. Ю., Боднар І. В.

Бази даних та системи управління базами даних: навч. посіб. для студентів ЗВО. — Львів : Видавництво Львівської політехніки, 2022. — 380 с.

2. Олійник А. О., Шацька В. М.

Бази даних: проектування та реалізація: навч. посіб. — Київ : Академвидав, 2023. — 344 с.

3. Гладченко М. В., Пасічник В. В.

Серверна розробка та бази даних: навч. посіб. — Львів : Видавництво Львівської політехніки, 2023. — 296 с.

4. Олексієнко С. І., Сердюк П. В.

Розробка RESTful API та трьохланкових застосунків: навч. посіб. для студентів спеціальності 123. — Харків : ХНУРЕ, 2024. — 312 с.

5. Дем'яненко В. М., Кравченко Г. В.

Нереляційні бази даних у сучасних веб-застосунках: навч. посіб. — Київ : КПІ ім. Ігоря Сікорського, 2023. — 258 с.

6. Сергієнко І. В., Задірака В. К.

Комп'ютерні технології та бази даних: підручник для ЗВО. — 2-ге вид., перероб. — Київ : Наукова думка, 2022. — 420 с.

7. Висоцька В. А., Чирун Л. Б.

Технології розробки клієнт-серверних застосунків: навч. посіб. — Львів : Видавництво Львівської політехніки, 2023. — 276 с.

8. Петрик М. Р., Гарматюк О. О.

Проектування та розробка інформаційних систем: навч. посіб. — Тернопіль : ТНТУ ім. Івана Пулюя, 2024. — 318 с.

Додаткова література та наукові статті

9. Лупенко С. А., Луців М. В.

Оптимізація запитів у реляційних базах даних. Вісник Тернопільського національного технічного університету. — 2023. — № 1(109). — С. 112–121. *Рекомендується для тем 2.2–2.3 (оптимізація SQL).*

10. Куцаченко О. П., Марченко О. І.

Порівняльний аналіз ORM-фреймворків для Node.js при розробці серверних застосунків. Комп'ютерні системи та мережі. — 2023. — № 3. — С. 45–54. *Рекомендується для теми 3.2.*

11. Ткачук Р. Л., Шевченко Д. В.

Методи захисту баз даних від несанкціонованого доступу в умовах трьохланкових архітектур. Захист інформації. — 2022. — Т. 24, № 4. — С. 213–224. *Рекомендується для теми 3.4.*

12. Іванченко О. В., Білоус Н. В.

Контейнеризація баз даних та серверних застосунків засобами Docker: навч.-метод. матеріали. — Київ : НТУ «КПІ ім. Ігоря Сікорського», 2024. — 128 с. *Рекомендується для теми 4.3.*

Нормативно-правова база

→ Закон України «Про вищу освіту» від 01.07.2014 № 1556-VII

Визначає засади вищої освіти в Україні, вимоги до освітніх програм та кредитної системи ЄКТС. zakon.rada.gov.ua

→ Стандарт вищої освіти України за спеціальністю 123 «Комп'ютерна інженерія» (бакалавр)

Визначає компетентності та результати навчання для освітньо-кваліфікаційного рівня бакалавра. mon.gov.ua

→ Закон України «Про захист персональних даних» від 01.06.2010 № 2297-VI

Регулює обробку та захист персональних даних в інформаційних системах — актуально для розробки застосунків.
zakon.rada.gov.ua

→ Закон України «Про електронні документи та електронний документообіг» від 22.05.2003 № 851-IV

Встановлює правові засади електронного документообігу та зберігання даних. zakon.rada.gov.ua

→ Закон України «Про інформацію» від 02.10.1992 № 2657-XII (зі змінами)

Визначає правовий режим інформації та інформаційних ресурсів, актуально для проектування баз даних. zakon.rada.gov.ua

Офіційні ресурси та фахові видання

Документація PostgreSQL

[postgresql.org/docs](https://www.postgresql.org/docs/) — офіційна документація PostgreSQL: SQL-синтаксис, PL/pgSQL, адміністрування, оптимізація.

Документація MongoDB

[mongodb.com/docs](https://www.mongodb.com/docs/) — офіційна документація MongoDB: CRUD, агрегація, індекси, Atlas (хмарний сервіс).

Національна бібліотека України ім. В. І. Вернадського

nbuv.gov.ua — електронні ресурси, наукові статті та монографії з інформаційних технологій та баз даних.

Репозиторій МОН України

mon.gov.ua — освітні стандарти, навчальні програми, методичні матеріали для спеціальності 123.

Журнал «Комп'ютерні системи та мережі»

csn.lpnu.ua — фахове видання Львівської політехніки з комп'ютерних систем, мереж та баз даних.

Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка

ela.kpi.ua — наукові публікації з розробки програмного забезпечення та інформаційних систем.

OWASP Foundation

owasp.org — відкрита база знань з безпеки застосунків: OWASP Top 10, чек-листи, методики тестування.

GitHub — відкриті навчальні проєкти

github.com — відкриті проєкти трьохланкових застосунків на Node.js/Python для аналізу архітектурних рішень.

Зв'язок дисципліни зі спеціальністю 123 «Комп'ютерна інженерія»

Чому Database Development для інженерів?

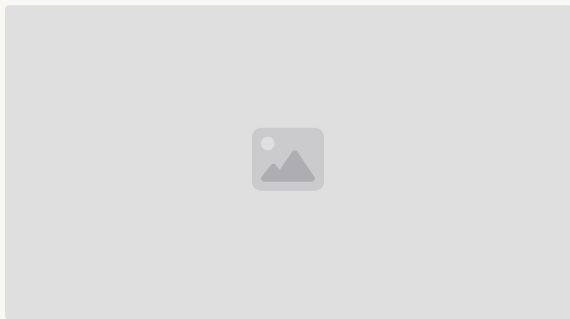
Дисципліна «Database Development та трьохланкові застосунки» є фундаментальною для підготовки бакалаврів зі спеціальністю 123 «Комп'ютерна інженерія». Практично кожна сучасна інформаційна система — від мобільного застосунку до корпоративної ERP — використовує базу даних як основу зберігання та обробки інформації.

Сучасний фахівець у сфері комп'ютерної інженерії, незалежно від спеціалізації (системне програмування, вбудовані системи, IoT, хмарні технології), неодмінно взаємодіє з базами даних та багаторівневими архітектурами.

Практичне значення для майбутньої кар'єри

- Проєктування та розробка баз даних — одна з найзатребуваніших навичок на ринку праці в IT
- Знання SQL є обов'язковою вимогою для більшості позицій backend-розробника та системного інженера
- Навички побудови трьохланкових застосунків дозволяють реалізовувати повний цикл розробки продукту
- Розуміння архітектурних патернів (MVC, Repository) підвищує якість коду та полегшує командну роботу
- Знання Docker та основ DevOps відповідає сучасним вимогам до інженерів у хмарній розробці
- Навички безпеки застосунків та баз даних критичні для захисту інформаційних систем України

Ключові компетентності курсу — Підсумок



Курс формує у студентів повний технологічний стек для розробки сучасних застосунків — від проєктування нормалізованої бази даних та написання складних SQL-запитів до побудови RESTful API, інтеграції клієнтської частини та розгортання у контейнерному середовищі.

Бажаємо успіхів у вивченні Database Development!

120

Годин

Загальний обсяг дисципліни

4

Кредити ЄКТС

Відповідно до стандарту

16

Тем

У 4 змістовних модулях

12

Джерел

Рекомендованої літератури 2022–2024 рр.

«База даних — це серце будь-якої інформаційної системи. Навчись проектувати її правильно, і твої застосунки будуть надійними, швидкими та масштабованими.»

Спеціальність 123 · Комп'ютерна інженерія · Бакалавр · 4 кредити ЄКТС