

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАКЛАД ВИЩОЇ ОСВІТИ
«ПОДІЛЬСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ»
ФАКУЛЬТЕТ ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

*Кафедра інформаційних технологій,
фізико-математичних та безпекових дисциплін*

**СИСТЕМНИЙ АНАЛІЗ ТА ПРОЄКТУВАННЯ
КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ СИСТЕМ**

методичні рекомендації
до виконання лабораторних робіт
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю 126 Інформаційні системи та технології

Кам'янець-Подільський

2026

Укладачі:

ДРАГАН Євгеній Вікторович - кандидат педагогічних наук, асистент кафедри інформаційних технологій, фізико-математичних та безпекових дисциплін;

СЕМЕНИШИНА Ірина Віталіївна - кандидат фізико-математичних наук, доцент, доцент кафедри інформаційних технологій, фізико-математичних та безпекових дисциплін.

*Рекомендовано до друку науково-методичною радою
Закладу вищої освіти «Подільський державний університет»
протокол № 5 від 27.05. 2026 р.*

Рецензенти:

ГУДИМА Уляна Василівна, кандидат фізико-математичних наук, доцент, доцент кафедри математики Кам'янець-Подільського національного університету імені Івана Огієнка.

ГАРАСИМЧУК Ігор Дмитрович, кандидат технічних наук, доцент, доцент кафедри енергетики та електротехнічних систем, Заклад вищої освіти «Подільський державний університет».

Системний аналіз та проектування комп'ютерних інформаційних систем. Методичні рекомендації до виконання лабораторних робіт для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 126 Інформаційні системи та технології/ С.В. ДРАГАН, І.В.СЕМЕНИШИНА. Кам'янець-Подільський: ЗВО «ПДУ», 2026. 43 с.

Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни «Системний аналіз та проектування комп'ютерних інформаційних систем» для здобувачів вищої освіти першого (бакалаврського) рівня спеціальності 126 Інформаційні системи та технології.

Зміст

Передмова	4
Критерії оцінювання	4
Загальні вимоги до оформлення звіту	5
Лабораторна робота № 1. Аналіз предметної області	6
Лабораторна робота № 2. Специфікація вимог	8
Лабораторна робота № 3. Базова Use Case діаграма	10
Лабораторна робота № 4. Відношення include та extend	12
Лабораторна робота № 5. Специфікація варіантів використання	14
Лабораторна робота № 6. Побудова ER-діаграми	16
Лабораторна робота № 7. Нормалізація бази даних	19
Лабораторна робота № 8. Завершення концептуальної моделі	21
Лабораторна робота № 9. MySQL Workbench. Створення бази даних ...	23
Лабораторна робота № 10. Типи даних MySQL	26
Лабораторна робота № 11. CREATE TABLE, обмеження, ключі	29
Лабораторна робота № 12. ALTER TABLE, TRUNCATE, DROP	31
Лабораторна робота № 13. INSERT INTO, UPDATE	33
Лабораторна робота № 14. DELETE, транзакції	35
Лабораторна робота № 15. SELECT, WHERE, ORDER BY, JOIN	37
Лабораторна робота № 16. Агрегатні функції та звіти	39
Рекомендована література	41

Передмова

Методичні рекомендації призначені для студентів спеціальності 126 Інформаційні системи та технології, які вивчають дисципліну «Системний аналіз та проєктування комп'ютерних інформаційних систем». Видання охоплює повний цикл проєктування інформаційної системи — від аналізу вимог до фізичної реалізації реляційної бази даних.

Наскрізний проєкт «ПіцаЕкспрес» (система замовлення та доставки їжі) використовується в усіх 16 лабораторних роботах як єдина предметна область, що дозволяє студентам поступово нарощувати складність моделі.

Для виконання робіт необхідні:

- комп'ютер з доступом до Інтернету;
- draw.io (<https://app.diagrams.net>);
- MySQL Workbench 8.x (<https://dev.mysql.com/downloads/workbench/>);
- текстовий редактор (Google Docs або Microsoft Word, LibreOffice Writer).
-

Критерії оцінювання

Оцінювання лабораторних робіт здійснюється відповідно до робочої програми дисципліни. Максимальна оцінка за одну лабораторну роботу — 6 балів.

Рівень виконання	Балів	Характеристика
Високий	6	Всі завдання виконані правильно і в повному обсязі, відповіді на контрольні питання вичерпні, висновки обґрунтовані.
Достатній	4–5	Завдання виконані з незначними недоліками або неточностями, контрольні питання розкриті в цілому правильно.
Середній	2–3	Завдання виконані частково або з суттєвими помилками, відповіді на питання поверхневі.
Низький	1	Завдання виконані не повністю, практичні уміння відсутні.
Не зараховано	0	Роботу не виконано або не подано.

Детальний розподіл балів за модулями та критерії оцінювання екзамену наведені в робочій програмі навчальної дисципліни.

Загальні вимоги до оформлення звіту

Звіт з кожної лабораторної роботи оформлюється у форматі .docx та подається викладачеві у встановлений термін. Вимоги до форматування:

- шрифт: Times New Roman, 14 пт;
- міжрядковий інтервал: 1,5;
- поля: верхнє та нижнє — 2 см, лівє — 3 см, правє — 1,5 см;
- нумерація сторінок — у нижньому правому куті;
- діаграми вставляються як зображення PNG з підписом «Рис. N. Назва діаграми»;
- таблиці нумеруються: «Таблиця N. Назва таблиці».

Тема 1. Основи системного аналізу

Лабораторна робота № 1. Аналіз предметної області

Тема: Аналіз предметної області та визначення вимог до ІС.

1.1. Мета роботи

Навчитися аналізувати предметну область ІС: виявляти бізнес-процеси, визначати ключові об'єкти та вимоги, документувати результати аналізу.

1.2. Теоретичні відомості

Системний аналіз — методологія дослідження складних об'єктів з метою побудови їх формалізованого опису. Аналіз бізнес-процесів виконується за алгоритмом:

1. Визначити мету підприємства.
2. Виявити основні бізнес-процеси («Що робить підприємство?»).
3. Описати кожен процес: вхід → дії → вихід → учасники.
4. Виявити зв'язки між процесами.
5. Визначити об'єкти, з якими працюють процеси.

Вимоги поділяються на функціональні (що система робить) та нефункціональні (надійність, продуктивність, зручність).

1.3. Хід роботи

Предметна область: «ПіцаЕкспрес» — система замовлення та доставки їжі.

Завдання 1. Аналіз бізнес-процесів

Ознайомтеся з описом системи та заповніть таблицю (визначте не менше 4 бізнес-процесів):

Бізнес-процес	Вхід	Дії (коротко)	Результат
Прийом замовлення			
Приготування їжі			
Доставка замовлення			
[Додати самостійно]			

Завдання 2. Визначення об'єктів системи

Виявіть іменники у описах бізнес-процесів — це кандидати на об'єкти. Визначте не менше 5 об'єктів:

№	Об'єкт	Що описує	Приклад атрибутів
1	Клієнт	Особа, що замовляє їжу	ім'я, телефон, email
2			
3			
4			
5			

Завдання 3. Формулювання вимог

Сформулюйте не менше 6 функціональних та 3 нефункціональних вимог до системи «ПіцаЕкспрес».

Функціональні вимоги (приклади — продовжіть самостійно):

1. Система повинна дозволяти клієнту переглядати меню.
2. Система повинна дозволяти клієнту оформлювати замовлення.
3. Система повинна надсилати клієнту підтвердження замовлення.
4. [Продовжіть: не менше 3 вимог]

Нефункціональні вимоги (приклади — продовжіть самостійно):

- Продуктивність: обробка замовлення не довше 3 секунд.
- [Надійність: ...]
- [Зручність: ...]

1.4. Вимоги до звіту

- Титульний аркуш (назва дисципліни, ЛР, прізвище, група, дата).
- Мета роботи.
- Заповнена таблиця бізнес-процесів (Завдання 1).
- Таблиця об'єктів системи (Завдання 2).
- Перелік функціональних та нефункціональних вимог (Завдання 3).
- Відповіді на контрольні питання. Висновки.

1.5. Контрольні питання

1. Що таке системний аналіз? Яка його мета?
2. Що таке бізнес-процес? Назвіть його обов'язкові елементи.
3. Яка різниця між функціональними та нефункціональними вимогами?
4. Як виявляються об'єкти предметної області?
5. Що таке предметна область інформаційної системи?

Лабораторна робота № 2. Специфікація вимог

Тема: Специфікація вимог до ІС. Документ SRS.

2.1. Мета роботи

Навчитися складати специфікацію вимог (SRS) у спрощеному форматі, систематизувати вимоги та визначати пріоритети методом MoSCoW.

2.2. Теоретичні відомості

Специфікація вимог (SRS) — документ, що описує призначення системи, її функції та обмеження. Метод MoSCoW розподіляє вимоги на:

- Must have — обов'язкові (без них система не функціонує);
- Should have — важливі (суттєво покращують систему);
- Could have — бажані (реалізуються за наявності ресурсів);
- Won't have — не плануються в поточній версії.
-

2.3. Хід роботи

Завдання 1. Таблиця вимог з пріоритетами MoSCoW

№	Вимога	Користувач	MoSCoW
1	Перегляд меню без реєстрації	Клієнт	Must have
2	Реєстрація та авторизація	Клієнт	Must have
3	Оформлення замовлення	Клієнт	Must have
4	Підтвердження замовлення	Оператор	Must have
5–10	[Визначте самостійно — не менше 6 вимог]		

Завдання 2. Спрощений SRS-документ

Складіть SRS за структурою:

1. Загальний огляд системи (2–3 речення: призначення, основні користувачі).
2. Функціональні вимоги (таблиця з Завдання 1).
3. Нефункціональні вимоги (з ЛР №1 + доповнення).
4. Обмеження системи (технічні, бізнес-обмеження, юридичні).
5. Словник термінів (5–7 ключових понять із визначеннями).

2.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- Таблиця вимог з пріоритетами MoSCoW.

- Спрощений SRS-документ.
- Відповіді на контрольні питання. Висновки.

2.5. Контрольні питання

1. Що таке SRS? Яка її структура?
2. Що означає кожна категорія MoSCoW?
3. Яка різниця між обмеженням та нефункціональною вимогою?
4. Що таке словник термінів і навіщо він потрібен?
5. Як перевірити, що вимога є «перевірною»?

Тема 2. Діаграми варіантів використання

Лабораторна робота № 3. Базова Use Case діаграма

Тема: Визначення акторів та варіантів використання.

3.1. Мета роботи

Навчитися визначати акторів та варіанти використання; побудувати базову Use Case діаграму в draw.io або PlantUML.

3.2. Теоретичні відомості

Елемент	Позначення	Опис
Актор	Стікмен	Зовнішня сутність, що взаємодіє із системою
Варіант використання	Еліпс (овал)	Функція системи. Формат: «Дієслово + Іменник»
Межа системи	Прямокутник	Охоплює всі Use Cases; актори — поза межею
Асоціація	Суцільна лінія	Зв'язок між актором і Use Case

Джерело: Fowler, M. (2004). UML Distilled (3rd ed.). Addison-Wesley.

3.3. Хід роботи

Завдання 1. Визначення акторів

№	Актор	Роль	Приклад взаємодії
1	Клієнт	Замовляє їжу	Переглядає меню, оформлює замовлення
2	Оператор		
3	Кур'єр		

Завдання 2. Варіанти використання

ID	Актор	Варіант використання	Результат
UC-01	Клієнт	Переглянути меню	Клієнт бачить страви
UC-02	Клієнт	Оформити замовлення	Замовлення прийнято
UC-03	Клієнт	Відстежити статус замовлення	Клієнт знає де замовлення
UC-04	Клієнт	<i>[Визначте самостійно]</i>	
UC-05	Оператор	<i>[Визначте самостійно]</i>	
UC-06	Оператор		

UC-07	Кур'єр	<i>[Визначте самостійно]</i>	
UC-08	Кур'єр		

Завдання 3. Use Case діаграма в draw.io

1. Відкрийте draw.io → Create New Diagram.
2. Намалюйте прямокутник системи «ПіцаЕкспрес».
3. Додайте акторів ліворуч/праворуч, Use Cases — всередині.
4. З'єднайте акторів з їхніми Use Cases лініями асоціації.
5. Збережіть у форматах .drawio та .png.

3.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- Таблиця акторів; таблиця Use Cases.
- Use Case діаграма (PNG, підпис «Рис. 1.»).
- Відповіді на контрольні питання. Висновки.

3.5. Контрольні питання

1. Що таке Use Case діаграма?
2. Що таке актор? Чи може ним бути зовнішня система?
3. За якою формулою формуються Use Cases?
4. Що позначає межа системи?
5. Як з'єднуються актори з Use Cases?

Лабораторна робота № 4. Відношення include та extend

Тема: Розширена Use Case діаграма: include, extend, узагальнення.

4.1. Мета роботи

Навчитися використовувати відношення <<include>> та <<extend>>; доповнити базову діаграму «ПіцаЕкспрес».

4.2. Теоретичні відомості

Відношення	Напрямок стрілки	Коли	Приклад
<<include>>	Базовий → Включений	Завжди виконується	«Оформити» включає «Авторизуватися»
<<extend>>	Розширювальний → Базовий	За певних умов	«Оформити» розширюється «Промокод»
Узагальнення	Нащадок → Батько	Спадкування ролі	«VIP-клієнт» — різновид «Клієнта»

Ключова різниця: стрілка <<include>> — від базового ДО включеного; <<extend>> — від розширювального ДО базового.

4.3. Хід роботи

Завдання 1. Виявлення відношень

Базовий Use Case	Пов'язаний	Тип	Умова (для extend)
Оформити замовлення	Авторизуватися	<<include>>	—
Оформити замовлення	Застосувати промокод	<<extend>>	Є промокод
[Додати самостійно]			
[Додати самостійно]			

Завдання 2. Оновлення діаграми

1. Додайте нові Use Cases (Авторизуватися, Перевірити наявність, Застосувати промокод).
2. Позначте <<include>> пунктирними стрілками.
3. Позначте <<extend>> пунктирними стрілками.
4. Збережіть оновлену діаграму (.drawio та .png).

4.4. Вимоги до звіту

- Титульний аркуш; мета роботи.

- Таблиця відношень з поясненням.
- Оновлена Use Case діаграма PNG.
- Відповіді на контрольні питання. Висновки.

4.5. Контрольні питання

1. Яка різниця між <<include>> та <<extend>>?
2. У якому напрямку стрілка для <<include>>? Для <<extend>>?
3. Коли доцільно узагальнення між акторами?
4. Чи може один Use Case мати і <<include>>, і <<extend>>?
5. Як <<include>> допомагає уникнути дублювання?

Лабораторна робота № 5. Специфікація варіантів використання

Тема: Специфікація Use Case: основний та альтернативні потоки.

5.1. Мета роботи

Навчитися складати специфікацію варіанту використання у табличній формі: описувати основний потік подій, альтернативні потоки, передумови та постумови.

5.2. Теоретичні відомості

Поле	Зміст
Назва	«Дієслово + Іменник»
Ідентифікатор	UC-02
Актор	Хто ініціює
Мета	Що хоче досягти актор
Передумови	Що має бути виконано ДО початку
Основний потік	Покрокова послідовність при успішному виконанні
Альтернативні потоки	Що відбувається при помилках або відмовах
Постумови	Стан системи після завершення

5.3. Хід роботи

Завдання 1. Специфікація «Оформити замовлення»

Назва	Оформити замовлення
Ідентифікатор	UC-02
Актор	Клієнт
Мета	Клієнт хоче замовити їжу з доставкою
Передумови	Клієнт авторизований. Меню завантажено.
Основний потік	1. Клієнт відкриває кошик. 2. Система відображає вміст та суму. 3. Клієнт вводить адресу доставки. 4. Клієнт обирає спосіб оплати. 5. Клієнт підтверджує замовлення. 6. Система зберігає замовлення і надсилає підтвердження.
Альт. потік А: кошик порожній	Крок 1: система повідомляє і пропонує перейти до меню.
Альт. потік Б: помилка оплати	<i>[Опишіть самостійно]</i>
Постумови	Замовлення зі статусом «Нове». Оператор отримав сповіщення.

Завдання 2. Самостійна специфікація

Складіть повну специфікацію для одного Use Case на вибір:

- «Підтвердити замовлення» (Оператор) — не менше 5 кроків основного потоку, 2 альтернативних.
- «Призначити кур'єра» (Оператор).
- «Змінити статус доставки» (Кур'єр).

5.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- Повна специфікація «Оформити замовлення» з заповненим альт. потоком Б.
- Самостійно складена специфікація.
- Відповіді на контрольні питання. Висновки.

5.5. Контрольні питання

1. Що таке специфікація Use Case? Яка її структура?
2. Що таке основний потік і чим він відрізняється від альтернативного?
3. Що таке передумова і постумова?
4. Як специфікації пов'язані з Use Case діаграмою?
5. Навіщо описувати альтернативні потоки?

Тема 3. ER-діаграми та концептуальне проєктування БД

Лабораторні роботи № 6–8 присвячені концептуальному проєктуванню бази даних системи «ПіцаЕкспрес».

Лабораторна робота № 6. Побудова ER-діаграми

Тема: Визначення сутностей, атрибутів та зв'язків. Нотація Crow's Foot.

6.1. Мета роботи

Навчитися будувати концептуальну ER-модель: визначати сутності, атрибути, зв'язки; застосовувати нотацію Crow's Foot у draw.io.

6.2. Теоретичні відомості

Елемент	Crow's Foot	Опис
Сутність	Прямокутник	Реальний об'єкт, про який зберігається інформація
РК (первинний ключ)	Позначений РК	Унікальний ідентифікатор запису
FK (зовнішній ключ)	Позначений FK	Посилання на РК іншої сутності
Зв'язок 1:N	———<	Один запис пов'язаний з багатьма
Зв'язок M:N	> ———<	Реалізується через проміжну таблицю

6.3. Хід роботи

Завдання 1. Сутності

Визначте сутності системи (не менше 6). Перші чотири подані як приклад:

№	Сутність (EN)	Сутність (UA)	Що описує
1	Customer	Клієнт	Особа, що здійснює замовлення
2	Order	Замовлення	Факт замовлення: статус, сума, дата
3	OrderItem	Позиція замовлення	Страва у замовленні та кількість (проміжна таблиця M:N)
4	Product	Продукт	Страва або напій: назва, ціна, опис
5	[Визначте]		
6	[Визначте]		

Завдання 2. Атрибути та ключі

Сутність	Атрибут	Тип	Примітка
----------	---------	-----	----------

Customer	customer_id (PK), name, phone, email, created_at	INT/VARCHAR/DATETIME	phone — UNIQUE
Order	order_id (PK), customer_id (FK), status, total_price, created_at	INT/ENUM/DECIMAL/DATETIME	status: new/confirmed/delivering/done/cancelled
OrderItem	item_id (PK), order_id (FK), product_id (FK), quantity, unit_price	INT/DECIMAL	unit_price фіксується на момент замовлення
Product	product_id (PK), name, description, price, is_available	INT/VARCHAR/TEXT/DECIMAL/TINYINT	is_available: 1/0
<i>[Ваші сутності]</i>			

Завдання 3. Зв'язки

Зв'язок	Тип	Обов'язковість	Пояснення
Customer → Order	1 : N	0..* Order	Клієнт має 0 або багато замовлень
Order → OrderItem	1 : N	1..* OrderItem	Замовлення містить ≥ 1 позицію
Product → OrderItem	1 : N	0..* OrderItem	Продукт фігурує в 0 або багатьох позиціях
<i>[Ваші зв'язки]</i>			

Завдання 4. ER-діаграма в draw.io

1. File → New → Entity Relationship.
2. Для кожної сутності — блок з атрибутами (PK зверху).
3. З'єднайте лініями Crow's Foot з відповідною кардинальністю.
4. Збережіть у форматах .drawio та .png.

6.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- Таблиця сутностей; таблиця атрибутів; таблиця зв'язків.
- ER-діаграма PNG (Рис. 1. ER-діаграма системи «ПіцаЕкспрес»).
- Відповіді на контрольні питання. Висновки.

6.5. Контрольні питання

1. Що таке ER-діаграма?
2. Яка різниця між РК та FK?
3. Що означає зв'язок M:N і як він реалізується?
4. Навіщо OrderItem є окремою сутністю?
5. Чому unit_price зберігається в OrderItem, а не в Product?

Лабораторна робота № 7. Нормалізація бази даних

Тема: 1НФ, 2НФ, 3НФ. Перевірка та вдосконалення ER-моделі.

7.1. Мета роботи

Ознайомитися з нормалізацією реляційних БД; перевірити ER-модель на 1НФ, 2НФ, 3НФ; усунути аномалії.

7.2. Теоретичні відомості

НФ	Вимога	Аномалія, що усувається
1НФ	Атрибути атомарні. Немає повторюваних груп. Є РК.	Список значень в одному полі (напр. 'піца,напій')
2НФ	1НФ + повна залежність від усього РК	Часткова залежність атрибута від частини ключа
3НФ	2НФ + немає транзитивних залежностей між неключовими атрибутами	Транзитивна залежність (місто залежить від поштового індексу)

7.3. Хід роботи

Завдання 1. Перевірка на 1НФ

Перевірте кожну сутність з ЛР №6. Для кожного порушення запишіть: який атрибут порушує 1НФ і як усунути порушення.

Приклад порушення: атрибут tags VARCHAR зберігає 'піца,гостра,сирна' — порушення 1НФ. Рішення: виділити таблицю ProductTag.

Завдання 2. Перевірка на 2НФ та 3НФ

Таблиця	Залежність	НФ	Дія
[Ваші таблиці]			

Завдання 3. Оновлення ER-діаграми

1. Виправте порушення, якщо виявлені.
2. Якщо модель вже нормалізована — обґрунтуйте у висновках.
3. Збережіть оновлену діаграму.

7.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- Таблиця перевірки на 1НФ.
- Таблиця функціональних залежностей.

- Оновлена ER-діаграма або обґрунтування відсутності змін.
- Відповіді на контрольні питання. Висновки.

7.5. Контрольні питання

1. Що таке нормалізація і навіщо вона потрібна?
2. Яка різниця між 1НФ та 2НФ?
3. Що таке транзитивна залежність?
4. Чи завжди потрібно доводити БД до 3НФ?
5. Як нормалізація впливає на ER-діаграму?

Лабораторна робота № 8. Завершення концептуальної моделі

Тема: Повна ER-модель. Словник даних. Документування рішень.

8.1. Мета роботи

Завершити концептуальне проектування БД «ПіцаЕкспрес»; скласти словник даних; задокументувати проєктні рішення.

8.2. Теоретичні відомості

Словник даних (data dictionary) — структурований опис атрибутів: назва, тип, обмеження, значення за замовчуванням, опис. Є єдиним джерелом правди для розробника БД.

8.3. Хід роботи

Завдання 1. Словник даних — зразок для таблиці orders

Атрибут	Тип	Обмеження	За замовч.	Опис
order_id	INT	PK, NOT NULL, AUTO_INCREMENT	—	Унікальний ідентифікатор замовлення
customer_id	INT	FK, NOT NULL	—	Посилання на customers
status	ENUM	NOT NULL	'new'	new/confirmed/delivering/done/cancelled
total_price	DECIMAL(10,2)	NOT NULL	0.00	Загальна сума замовлення
created_at	DATETIME	NOT NULL	NOW()	Дата і час створення

Повторіть таблицю для всіх сутностей вашої ER-моделі.

Завдання 2. Обґрунтування проєктних рішень

- Чому OrderItem є окремою сутністю, а не частиною Order?
- Чому unit_price зберігається в OrderItem, а не береться з Product?
- Поясніть 1–2 власних нетривіальних рішення.

Завдання 3. Фінальна ER-діаграма

Переконайтеся, що діаграма містить усі сутності, атрибути (PK/FK) та зв'язки з кардинальністю. Збережіть фінальний PNG.

8.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- Повний словник даних (всі сутності).

- Обґрунтування проєктних рішень.
- Фінальна ER-діаграма PNG.
- Відповіді на контрольні питання. Висновки.

8.5. Контрольні питання

1. Що таке словник даних?
2. Які типи MySQL для рядків, чисел, дат?
3. Що таке AUTO_INCREMENT?
4. Чим концептуальна модель відрізняється від логічної?
5. Як з ER-діаграми перейти до CREATE TABLE?

Змістовий модуль 2. MySQL та реалізація бази даних

Лабораторні роботи № 9–16 присвячені реалізації бази даних у СУБД MySQL з використанням MySQL Workbench.

Інструмент: MySQL Workbench 8.x

(<https://dev.mysql.com/downloads/workbench/>). Підключення: Host — localhost, Port — 3306.

Тема 4. Основи MySQL та створення БД

Лабораторна робота № 9. MySQL Workbench. Створення бази даних

Тема: Інтерфейс MySQL Workbench. CREATE DATABASE, USE, SHOW.

9.1. Мета роботи

Ознайомитися з інтерфейсом MySQL Workbench; створити базу даних pizza_express; виконати базові команди перегляду.

9.2. Теоретичні відомості

```
SHOW DATABASES;  
CREATE DATABASE IF NOT EXISTS pizza_express  
    CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;  
USE pizza_express;  
SHOW TABLES;  
SELECT DATABASE();
```

9.3. Хід роботи

Завдання 1. Підключення та перегляд

1. Запустіть MySQL Workbench → підключіться до localhost.
2. Виконайте SHOW DATABASES; — запишіть результат у звіт.
3. Виконайте CREATE DATABASE pizza_express...
4. Переконайтеся: pizza_express з'явилась у Navigator → Schemas.

Завдання 2. Дослідження інтерфейсу

- Клацніть правою кнопкою на pizza_express → Set as Default Schema.
- Відкрийте нову вкладку: File → New Query Tab.
- Спробуйте автодоповнення: введіть SHOW і натисніть Ctrl+Space.

Завдання 3. Дослідження метаданих бази даних

MySQL зберігає інформацію про структуру всіх баз даних у системній схемі information_schema. Це «база даних про бази даних» — довідник, що дозволяє отримати метадані без перегляду вихідного коду.

Крок 1. Перегляньте системні бази даних та змінні сервера:

```
-- Показати всі бази даних на сервері:
SHOW DATABASES;

-- Показати поточні системні змінні MySQL:
SHOW VARIABLES LIKE 'character%';
SHOW VARIABLES LIKE 'collation%';

-- Версія сервера та час роботи:
SELECT VERSION();
SHOW STATUS LIKE 'Uptime';
```

У звіті зафіксуйте: яке кодування використовує ваш сервер? Що означає значення `collation_server`?

Крок 2. Дослідіть схему `information_schema`:

```
-- Скільки таблиць існує в pizza_express?
SELECT TABLE_NAME, TABLE_ROWS, TABLE_TYPE
FROM information_schema.TABLES
WHERE TABLE_SCHEMA = 'pizza_express';

-- Усі стовпці таблиці customers з типами та обмеженнями:
SELECT COLUMN_NAME, COLUMN_TYPE, IS_NULLABLE,
       COLUMN_DEFAULT, COLUMN_KEY, EXTRA
FROM information_schema.COLUMNS
WHERE TABLE_SCHEMA = 'pizza_express'
      AND TABLE_NAME = 'customers'
ORDER BY ORDINAL_POSITION;
```

У звіті зафіксуйте: Яка додаткова інформація з'явилась?

Крок 3. Перегляньте зовнішні ключі бази:

```
-- Усі зовнішні ключі у pizza_express:
SELECT TABLE_NAME,
       COLUMN_NAME,
       REFERENCED_TABLE_NAME,
       REFERENCED_COLUMN_NAME
FROM information_schema.KEY_COLUMN_USAGE
WHERE TABLE_SCHEMA = 'pizza_express'
      AND REFERENCED_TABLE_NAME IS NOT NULL
ORDER BY TABLE_NAME;
```

У звіті зафіксуйте: скільки зовнішніх ключів у базі? Чи збігається результат з вашою ER-діаграмою?

Крок 4. Порівняльна таблиця (заповнити у звіті):

Спосіб отримання інформації	Команда	Що показує додатково
DESCRIBE	DESCRIBE customers;	
information_schema.COLUMNS	SELECT ... FROM ...	

SHOW CREATE TABLE	SHOW CREATE TABLE customers;	
-------------------	---------------------------------	--

Виконайте SHOW CREATE TABLE customers; і порівняйте результат з попередніми способами. Внесіть спостереження до таблиці.

9.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- Скріншот підключення до Workbench.
- Результат SHOW DATABASES (скріншот).
- SQL-код створення БД.
- Скріншот: pizza_express у Navigator.
- Відповіді на контрольні питання. Висновки.
- Скріншоти результатів SHOW VARIABLES, SELECT VERSION(), SHOW STATUS.
- Скріншот результату запиту до information_schema.TABLES.
- Скріншот результату запиту до information_schema.COLUMNS для таблиці customers.
- Скріншот результату запиту до information_schema.KEY_COLUMN_USAGE.
- Заповнена порівняльна таблиця (Крок 4).
- Письмовий висновок (5–7 речень): що таке information_schema, навіщо вона потрібна розробнику, чим запити до неї корисніші за DESCRIBE.

9.5. Контрольні питання

1. Що таке СУБД? Різниця між СУБД та базою даних?
2. Яка команда вибирає активну БД?
3. Навіщо CHARACTER SET utf8mb4?
4. Що робить SHOW TABLES?
5. Як видалити БД? Які наслідки?

Лабораторна робота № 10. Типи даних MySQL

Тема: Типи даних: INT, VARCHAR, DECIMAL, ENUM, DATETIME.

10.1. Мета роботи

Ознайомитися з основними типами даних MySQL; навчитися обирати відповідні типи для атрибутів системи «ПіцаЕкспрес».

10.2. Теоретичні відомості

Тип	Опис	Використання
INT	Ціле число	ID, лічильники (до ~2 млрд)
DECIMAL(p,s)	Точне дробове	Фінансові суми: price, total_price
VARCHAR(n)	Рядок змінної довжини	Імена, адреси, email
TEXT	Довгий рядок	Описи продуктів, нотатки
ENUM(...)	Обмежений набір	Статуси: 'new','done',...
DATETIME	Дата і час	created_at, delivered_at
TINYINT(1)	Булеве значення	is_available: 0/1

10.3. Хід роботи

Завдання 1. Таблиця типів даних

Таблиця	Атрибут	Тип MySQL	Обмеження
customers	customer_id	INT	PK, NOT NULL, AUTO_INCREMENT
customers	name	VARCHAR(100)	NOT NULL
orders	status	ENUM('new', 'confirmed', 'delivering', 'done', 'cancelled')	NOT NULL, DEFAULT 'new'
<i>[Всі інші атрибути]</i>			

Завдання 2. Обґрунтування вибору

- Де використали DECIMAL замість FLOAT і чому?
- Де використали ENUM і які значення включили?
- Чи є атрибути, де вагалися у виборі? Обґрунтуйте.

Завдання 3. Практична перевірка типів даних через SQL

Теорія типів даних стає зрозумілою через живі приклади. У цьому завданні ви перевірите, як MySQL насправді зберігає і обробляє різні типи — і побачите кілька несподіваних результатів.

Крок 1. Точність числових типів:

```
-- Класична пастка з FLOAT:
SELECT 0.1 + 0.2;
SELECT 0.1 + 0.2 = 0.3;

-- Те саме з DECIMAL (правильний спосіб для грошей):
SELECT CAST(0.1 AS DECIMAL(10,2)) + CAST(0.2 AS DECIMAL(10,2));
SELECT CAST(0.1 AS DECIMAL(10,2)) + CAST(0.2 AS DECIMAL(10,2)) =
0.3;
```

У звіті поясніть: чому перший результат не дорівнює 0.3? Що це означає для зберігання цін у таблиці products?

Крок 2. Рядки та кодування:

```
-- Різниця між LENGTH (байти) та CHAR_LENGTH (символи):
SELECT LENGTH('Hello'), CHAR_LENGTH('Hello');
SELECT LENGTH('Привіт'), CHAR_LENGTH('Привіт');
SELECT LENGTH('ПіцаЕкспрес'), CHAR_LENGTH('ПіцаЕкспрес');

-- Порівняння VARCHAR та CHAR за пробілами:
SELECT CHAR_LENGTH(CONCAT('abc', ' ')),
       CHAR_LENGTH(TRIM('abc '));
```

У звіті поясніть: чому LENGTH і CHAR_LENGTH дають різні результати для українського тексту? Що це означає при плануванні VARCHAR(n)?

Крок 3. Дати та час:

```
-- Поточний момент різними функціями:
SELECT NOW(), CURDATE(), CURTIME();

-- Різниця між датами (скільки днів від початку семестру):
SELECT DATEDIFF(NOW(), '2026-02-09') AS
днів_від_початку_семестру;

-- Форматування дати:
SELECT DATE_FORMAT(NOW(), '%d.%m.%Y') AS сьогодні_укр;
SELECT DATE_FORMAT(NOW(), '%Y-%m-%d %H:%i') AS timestamp_iso;
```

У звіті поясніть: яку функцію використати при збереженні часу замовлення в таблиці orders — NOW() чи CURDATE()? Чому?

Крок 4. ENUM на практиці:

```
-- Створимо тестову таблицю з ENUM:
CREATE TABLE test_status (
  id      INT AUTO_INCREMENT PRIMARY KEY,
  status  ENUM('new', 'confirmed', 'delivering', 'done', 'cancelled')
          NOT NULL DEFAULT 'new'
);

-- Вставляємо валідне і невалідне значення:
INSERT INTO test_status (status) VALUES ('new');
INSERT INTO test_status (status) VALUES ('done');
INSERT INTO test_status () VALUES ();           -- DEFAULT
INSERT INTO test_status (status) VALUES ('unknown'); -- помилка!

SELECT * FROM test_status;
DROP TABLE test_status;
```

У звіті зафіксуйте: яке повідомлення про помилку видає MySQL при спробі вставити 'unknown'? Чому ENUM є кращим вибором для поля status, ніж VARCHAR?

10.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- Повна таблиця типів для всіх атрибутів.
- Письмовий аналіз вибору типів.
- Відповіді на контрольні питання. Висновки.
- Скріншоти результатів усіх 4 кроків.
- Письмові пояснення до кожного кроку (за питаннями у завданні).
- Загальний висновок (5–7 речень): як практичні результати змінили або підтвердили ваш вибір типів даних у таблиці з Завдання 1?

10.5. Контрольні питання

1. Яка різниця між VARCHAR та CHAR?
2. Коли DECIMAL, а коли FLOAT?
3. Що таке ENUM?
4. Чому для фінансових сум не FLOAT?
5. Що таке AUTO_INCREMENT?

Тема 5. DDL — структура бази даних

Лабораторна робота № 11. CREATE TABLE, обмеження, ключі Тема: DDL: CREATE TABLE, PK, FK, NOT NULL, UNIQUE, DEFAULT.

11.1. Мета роботи

Навчитися створювати таблиці з обмеженнями; реалізувати структуру БД «ПіцаЕкспрес».

11.2. Теоретичні відомості

```
CREATE TABLE customers (  
    customer_id INT NOT NULL AUTO_INCREMENT,  
    name VARCHAR(100) NOT NULL,  
    phone VARCHAR(20) NOT NULL,  
    email VARCHAR(150),  
    created_at DATETIME NOT NULL DEFAULT NOW(),  
    PRIMARY KEY (customer_id),  
    UNIQUE KEY uq_phone (phone)  
);
```

11.3. Хід роботи

Завдання 1. Базові таблиці

```
USE pizza_express;  
CREATE TABLE products (  
    product_id INT NOT NULL AUTO_INCREMENT,  
    name VARCHAR(150) NOT NULL,  
    description TEXT,  
    price DECIMAL(10,2) NOT NULL,  
    is_available TINYINT(1) NOT NULL DEFAULT 1,  
    PRIMARY KEY (product_id)  
);  
CREATE TABLE orders (  
    order_id INT NOT NULL AUTO_INCREMENT,  
    customer_id INT NOT NULL,  
    status  
ENUM('new','confirmed','delivering','done','cancelled')  
NOT NULL DEFAULT 'new',  
    total_price DECIMAL(10,2) NOT NULL DEFAULT 0.00,  
    created_at DATETIME NOT NULL DEFAULT NOW(),  
    PRIMARY KEY (order_id),  
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);  
CREATE TABLE order_items (  
    item_id INT NOT NULL AUTO_INCREMENT,  
    order_id INT NOT NULL,  
    product_id INT NOT NULL,  
    quantity INT NOT NULL DEFAULT 1,  
    unit_price DECIMAL(10,2) NOT NULL,  
    PRIMARY KEY (item_id),
```

```
FOREIGN KEY (order_id) REFERENCES orders(order_id),  
FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```

Завдання 2. Перевірка структури

```
SHOW TABLES;  
DESCRIBE customers;  
DESCRIBE orders;  
DESCRIBE order_items;
```

Завдання 3. Самостійне створення

Створіть таблиці для решти сутностей з вашої ER-моделі (категорії, кур'єри, адреси доставки тощо).

11.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- SQL-код CREATE TABLE для всіх таблиць.
- Скріншоти DESCRIBE для кожної таблиці.
- Скріншот Navigator → pizza_express → Tables.
- Відповіді на контрольні питання. Висновки.

11.5. Контрольні питання

1. Яка команда DDL для створення таблиці?
2. Що таке FOREIGN KEY?
3. Різниця PRIMARY KEY та UNIQUE?
4. Що якщо FK посилається на неіснуючу таблицю?
5. Що показує DESCRIBE?

Лабораторна робота № 12. ALTER TABLE, TRUNCATE, DROP

Тема: DDL: ALTER TABLE (ADD, MODIFY, RENAME, DROP COLUMN), TRUNCATE TABLE, DROP TABLE.

12.1. Мета роботи

Навчитися змінювати структуру існуючих таблиць; розуміти різницю між TRUNCATE та DROP.

12.2. Теоретичні відомості

```
ALTER TABLE customers ADD COLUMN loyalty_points INT DEFAULT 0;
ALTER TABLE customers MODIFY COLUMN phone VARCHAR(30) NOT NULL;
ALTER TABLE customers RENAME COLUMN name TO full_name;
ALTER TABLE customers DROP COLUMN loyalty_points;
TRUNCATE TABLE orders;          -- видаляє дані, зберігає структуру
DROP TABLE IF EXISTS orders;    -- видаляє таблицю повністю
```

12.3. Хід роботи

Завдання 1. Модифікація таблиць

```
-- 1. Додати поле нотаток до замовлень:
ALTER TABLE orders ADD COLUMN notes TEXT;
-- 2. Додати бонусні бали для клієнтів:
ALTER TABLE customers ADD COLUMN loyalty_points INT NOT NULL
DEFAULT 0;
-- 3. Розширити довжину phone:
ALTER TABLE customers MODIFY COLUMN phone VARCHAR(30) NOT NULL;
-- 4. Додати поле оновлення до products:
ALTER TABLE products ADD COLUMN updated_at DATETIME
    DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP;
DESCRIBE customers;
DESCRIBE orders;
DESCRIBE products;
```

Завдання 2. TRUNCATE vs DROP

```
CREATE TABLE test_table (
    id INT AUTO_INCREMENT PRIMARY KEY,
    value VARCHAR(50)
);
INSERT INTO test_table (value) VALUES ('рядок 1'), ('рядок 2');
SELECT * FROM test_table;    -- бачимо 2 рядки
TRUNCATE TABLE test_table;  -- 0 рядків, AUTO_INCREMENT скинуто
SELECT * FROM test_table;
DROP TABLE IF EXISTS test_table; -- таблиця зникає
SHOW TABLES;                -- test_table більше немає
```

12.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- SQL-код ALTER TABLE. Скріншоти DESCRIBE до і після змін.
- Пояснення різниці між TRUNCATE та DROP (своїми словами).

– Відповіді на контрольні питання. Висновки.

12.5. Контрольні питання

1. Різниця DROP TABLE і TRUNCATE TABLE?
2. Як додати стовпець?
3. Що якщо DROP TABLE з FK?
4. Що означає ON UPDATE CURRENT_TIMESTAMP?
5. Як перейменувати стовпець у MySQL 8?

Тема 6. DML — операції з даними

Лабораторна робота № 13. INSERT INTO, UPDATE

Тема: DML: INSERT INTO та UPDATE. Заповнення тестовими даними.

13.1. Мета роботи

Навчитися додавати (INSERT) та оновлювати (UPDATE) записи; заповнити таблиці «ПіцаЕкспрес» тестовими даними.

13.2. Теоретичні відомості

```
-- INSERT: один рядок
INSERT INTO customers (name, phone, email)
VALUES ('Іванченко Олег', '+380501234567', 'oleg@ex.com');
-- INSERT: кілька рядків за один запит
INSERT INTO products (name, price) VALUES
    ('Маргарита', 220.00), ('Пепероні', 260.00);
-- UPDATE з WHERE (ОБОВ'ЯЗКОВО!)
UPDATE products SET price = 245.00 WHERE product_id = 1;
```

Увага: UPDATE без WHERE змінить УСІ рядки таблиці!

13.3. Хід роботи

Завдання 1. Довідникові таблиці

```
USE pizza_express;
INSERT INTO customers (name, phone, email) VALUES
    ('Мельник Ірина', '+380671111111', 'melnik@gmail.com'),
    ('Коваль Андрій', '+380672222222', 'koval@ukr.net'),
    ('Петренко Наталя', '+380673333333', NULL),
    ('Бондар Олексій', '+380674444444', 'bondar@gmail.com'),
    ('Шевченко Марія', '+380675555555', 'maria@ukr.net');
INSERT INTO products (name, description, price) VALUES
    ('Маргарита', 'Томат, моцарела, базилік', 220.00),
    ('Пепероні', 'Томат, моцарела, пепероні', 260.00),
    ('Чотири сири', 'Моцарела, дор блю, гауда, пармезан', 280.00),
    ('Гаваї', 'Томат, моцарела, шинка, ананас', 250.00),
    ('Кола 0.5л', NULL, 45.00),
    ('Фанта 0.5л', NULL, 45.00);
```

Завдання 2. Транзакційні таблиці

```
INSERT INTO orders (customer_id, status, total_price) VALUES
    (1, 'done', 480.00),
    (2, 'confirmed', 260.00),
    (3, 'delivering', 280.00),
    (4, 'new', 340.00);
INSERT INTO order_items (order_id, product_id, quantity,
unit_price) VALUES
    (1, 1, 1, 220.00), -- Маргарита в замовленні #1
    (1, 2, 1, 260.00), -- Пепероні в замовленні #1
    (2, 2, 1, 260.00), -- Пепероні в замовленні #2
    (3, 3, 1, 280.00), -- Чотири сири в замовленні #3
```

```
(4, 4, 1, 250.00), -- Гаваї в замовленні #4
(4, 5, 1, 45.00), -- Кола в замовленні #4
(4, 6, 1, 45.00); -- Фанта в замовленні #4
SELECT * FROM orders;
SELECT * FROM order_items;
```

Завдання 3. Оновлення даних

```
UPDATE orders SET status = 'delivering' WHERE order_id = 2;
UPDATE products SET price = 230.00 WHERE name = 'Маргарита';
SELECT order_id, status FROM orders WHERE order_id = 2;
SELECT name, price FROM products WHERE name = 'Маргарита';
```

13.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- SQL-код INSERT та UPDATE.
- Скріншоти SELECT * для кожної таблиці після заповнення.
- Відповіді на контрольні питання. Висновки.

13.5. Контрольні питання

1. Яка команда DML для додавання даних?
2. Як вставити кілька рядків одним запитом?
3. Що станеться якщо UPDATE без WHERE?
4. Чи можна вставити NULL у NOT NULL стовпець?
5. Яка різниця між INSERT та UPDATE?

Лабораторна робота № 14. DELETE, транзакції

Тема: DELETE, START TRANSACTION, COMMIT, ROLLBACK.

14.1. Мета роботи

Навчитися безпечно видаляти дані; використовувати транзакції для захисту від помилок.

14.2. Теоретичні відомості

```
-- DELETE з умовою (завжди використовуйте WHERE!):  
DELETE FROM order_items WHERE order_id = 3;  
DELETE FROM orders      WHERE order_id = 3;  
-- Транзакція: або всі операції успішні, або жодна:  
START TRANSACTION;  
    UPDATE orders SET status = 'cancelled' WHERE order_id = 4;  
    DELETE FROM order_items WHERE order_id = 4;  
COMMIT;    -- зберегти зміни  
-- або ROLLBACK; -- скасувати всі зміни
```

Каскадне видалення: при ON DELETE CASCADE дочірні записи видаляються автоматично.

14.3. Хід роботи

Завдання 1. Безпечне видалення

Порядок важливий: спочатку видаляємо дочірні записи (order_items), потім батьківські (orders).

```
DELETE FROM order_items WHERE order_id = 3;  
DELETE FROM orders      WHERE order_id = 3;  
SELECT * FROM orders;  
SELECT * FROM order_items;
```

Завдання 2. Транзакції — скасування замовлення

```
-- Успішна транзакція: скасовуємо замовлення #4  
START TRANSACTION;  
    UPDATE orders SET status = 'cancelled' WHERE order_id = 4;  
    DELETE FROM order_items WHERE order_id = 4;  
COMMIT;  
SELECT order_id, status FROM orders WHERE order_id = 4;  
SELECT * FROM order_items WHERE order_id = 4;  
-- Демонстрація ROLLBACK:  
START TRANSACTION;  
    INSERT INTO orders (customer_id, status, total_price)  
    VALUES (5, 'new', 500.00);  
    SELECT * FROM orders;    -- тимчасово бачимо новий рядок  
ROLLBACK;  
SELECT * FROM orders;    -- нового рядка немає
```

14.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- SQL-код DELETE та транзакцій.

- Скріншоти до і після видалення.
- Скріншот, що демонструє відкат ROLLBACK.
- Пояснення: коли потрібна транзакція?
- Відповіді на контрольні питання. Висновки.

14.5. Контрольні питання

1. Яка різниця між DELETE та TRUNCATE?
2. Що таке транзакція?
3. Що таке ROLLBACK і коли застосовується?
4. Чому при видаленні з FK треба спочатку видаляти дочірні записи?
5. Що таке ON DELETE CASCADE?

Тема 7. Прості SELECT-запити та звітність

Лабораторна робота № 15. SELECT, WHERE, ORDER BY, JOIN

Тема: Вибірка даних: фільтрація, сортування, з'єднання таблиць.

15.1. Мета роботи

Навчитися будувати SELECT-запити з WHERE, ORDER BY, LIMIT та JOIN.

15.2. Теоретичні відомості

```
-- INNER JOIN: лише рядки з відповідністю в обох таблицях:
SELECT o.order_id, c.name, o.status
FROM orders o
INNER JOIN customers c ON o.customer_id = c.customer_id;
-- LEFT JOIN: всі рядки лівої + NULL якщо немає відповідності:
SELECT c.name, o.order_id
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id;
```

15.3. Хід роботи

Завдання 1. Прості SELECT-запити

```
SELECT name, price FROM products ORDER BY price DESC;
SELECT name, price FROM products WHERE price > 250.00;
SELECT name, phone FROM customers ORDER BY name ASC LIMIT 3;
SELECT order_id, total_price FROM orders WHERE status = 'done';
SELECT name, phone FROM customers WHERE email IS NULL;
```

Завдання 2. Запити з JOIN

```
-- Замовлення з іменами клієнтів:
SELECT o.order_id, c.name, o.status, o.total_price
FROM orders o
INNER JOIN customers c ON o.customer_id = c.customer_id
ORDER BY o.created_at DESC;
-- Деталі замовлень (три таблиці):
SELECT o.order_id, c.name AS клієнт,
       p.name AS продукт, oi.quantity, oi.unit_price
FROM order_items oi
INNER JOIN orders o ON oi.order_id = o.order_id
INNER JOIN customers c ON o.customer_id = c.customer_id
INNER JOIN products p ON oi.product_id = p.product_id
ORDER BY o.order_id;
-- Клієнти та кількість замовлень (включно з тими, хто не
замовляв):
SELECT c.name, COUNT(o.order_id) AS кількість_замовлень
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name;
```

15.4. Вимоги до звіту

- Титульний аркуш; мета роботи.

- SQL-код усіх запитів. Скріншоти Result Grid.
- Пояснення різниці між INNER JOIN та LEFT JOIN.
- Відповіді на контрольні питання. Висновки.

15.5. Контрольні питання

1. Яке призначення WHERE?
2. Різниця INNER JOIN та LEFT JOIN?
3. Що таке аліас таблиці (AS)?
4. Як фільтрувати IS NULL?
5. Що робить LIMIT?

Лабораторна робота № 16. Агрегатні функції та звіти

Тема: COUNT, SUM, AVG, MIN, MAX; GROUP BY, HAVING. Бізнес-звіти.

16.1. Мета роботи

Навчитися використовувати агрегатні функції та GROUP BY для формування аналітичних звітів.

16.2. Теоретичні відомості

Функція	Що повертає	Приклад
COUNT(*)	Кількість рядків	COUNT(*) — всі рядки
SUM(col)	Сума значень	SUM(total_price) — виручка
AVG(col)	Середнє арифметичне	AVG(price) — середня ціна
MIN/MAX(col)	Мін/Макс значення	MIN(price), MAX(total_price)

```
-- GROUP BY: групує та застосовує агрегат:  
SELECT status, COUNT(*) AS кількість  
FROM orders GROUP BY status;  
-- HAVING: фільтр після GROUP BY (аналог WHERE для груп):  
SELECT customer_id, COUNT(*) AS замовлень  
FROM orders GROUP BY customer_id  
HAVING COUNT(*) > 1;
```

16.3. Хід роботи

Завдання 1. Агрегатні запити

```
SELECT COUNT(*) AS всього, SUM(total_price) AS виручка,  
       AVG(total_price) AS середній_чек FROM orders;  
SELECT status, COUNT(*) AS кількість FROM orders  
GROUP BY status ORDER BY кількість DESC;  
SELECT MAX(price) AS макс, MIN(price) AS мін FROM products;  
SELECT p.name, SUM(oi.quantity) AS замовлено  
FROM order_items oi  
INNER JOIN products p ON oi.product_id = p.product_id  
GROUP BY p.product_id ORDER BY замовлено DESC;
```

Завдання 2. Бізнес-звіти

```
-- Звіт 1: Топ-клієнти (з 2+ замовленнями):  
SELECT c.name, COUNT(o.order_id) AS замовлень,  
       SUM(o.total_price) AS сума  
FROM orders o  
INNER JOIN customers c ON o.customer_id = c.customer_id  
GROUP BY o.customer_id, c.name  
HAVING COUNT(o.order_id) >= 2  
ORDER BY сума DESC;  
-- Звіт 2: Склад замовлень з підсумком:  
SELECT o.order_id, c.name AS клієнт,
```

```
        COUNT(oi.item_id) AS позицій,  
        SUM(oi.quantity * oi.unit_price) AS сума  
FROM order_items oi  
INNER JOIN orders o ON oi.order_id = o.order_id  
INNER JOIN customers c ON o.customer_id = c.customer_id  
GROUP BY o.order_id, c.name  
ORDER BY o.order_id;
```

16.4. Вимоги до звіту

- Титульний аркуш; мета роботи.
- SQL-код усіх запитів. Скріншоти Result Grid.
- Пояснення різниці між WHERE та HAVING.
- Відповіді на контрольні питання. Висновки.

16.5. Контрольні питання

1. Яка функція для підрахунку рядків?
2. Різниця WHERE та HAVING?
3. Що таке GROUP BY?
4. Що повертає AVG якщо є NULL?
5. Як знайти найбільш популярний продукт?

Рекомендована література

Основна:

1. Авраменко В. С., Авраменко А. С. Проектування інформаційних систем: навч. посіб. Черкаси: ЧНУ ім. Б. Хмельницького, 2017. 434 с.
2. Коваленко О. С., Добровська Л. М. Проектування інформаційних систем. Київ: КПІ ім. Ігоря Сікорського, 2020. 192 с.
3. Ізмайлова О. В. Проектування інформаційних систем: навч. посіб. Київ: КНУБА, 2023. 88 с.
4. Ушенко Ю. О. та ін. Методологія інформаційних систем та баз даних. Чернівці: ЧНУ, 2021. 240 с.
5. Прокопенко Т. О. Теорія систем і системний аналіз. Черкаси: ЧДТУ, 2019. 139 с.

Допоміжна:

1. Fowler, M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. Addison-Wesley, 2004. 208 p.
2. Sommerville, I. Software Engineering. 10th ed. Pearson, 2016. 816 p.
3. Silberschatz, A., Korth, H., Sudarshan, S. Database System Concepts. 7th ed. McGraw-Hill, 2020. 1376 p.
4. Codd, E.F. Further normalization of the data base relational model. Data Base Systems, 1972. Vol. 6, pp. 33–64.
5. Chen, P.P. The entity-relationship model — toward a unified view of data. ACM TODS, 1976. 1(1), 9–36.

Інтернет-ресурси:

- MySQL 8.0 Reference Manual: <https://dev.mysql.com/doc/refman/8.0/en/>
- MySQL Workbench: <https://dev.mysql.com/downloads/workbench/>
- draw.io: <https://app.diagrams.net>
- PlantUML: <https://www.plantuml.com/plantuml>
- ОПІ 126 ПДУ: <https://pdatu.edu.ua/images/navchalna-robota/opp/opp23/opp-b-ieee-5-infst.pdf>

ДРАГАН Євгеній Вікторович
СЕМЕНИШИНА Ірина Віталіївна

ОПЕРАЦІЙНЕ ТА ВАРІАЦІЙНЕ ЧИСЛЕННЯ

методичні рекомендації
до виконання лабораторних робіт
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю 126 Інформаційні системи та технології/

