

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет
імені Івана Пулюя

Кафедра автоматизації технологічних процесів і виробництв

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДЛЯ ПРАКТИЧНИХ
ЗАНЯТЬ ТА САМОСТІЙНОЇ РОБОТИ

з курсу

**«ТЕХНОЛОГІЯ ПРОЄКТУВАННЯ ТА
АДМІНІСТРУВАННЯ БАЗ ДАНИХ І СХОВИЩ
ДАНИХ»**

для здобувачів першого рівня вищої освіти

спеціальності 051 «Економіка»

Тернопіль
2024

Методичні рекомендації для практичних занять та самостійної роботи з курсу «Технологія проєктування та адміністрування баз даних і сховищ даних» / укл.: Дмитрів Олена Романівна, Дмитрів Дмитро Володимирович, Микулик Петро Миколайович. – Тернопіль: ТНТУ імені Івана Пулюя, 2024. – 59.

У методичних рекомендаціях подано у стислій формі теоретичні відомості, практичні завдання, завдання для самостійного опрацювання, контрольні запитання. Основна мета – отримання і закріплення практичних навичок з проєктування, побудови і застосування реляційних баз даних при розв’язанні задач опрацювання інформації в комп’ютерних системах для прийняття важливих рішень.

Рекомендовано для здобувачів першого рівня вищої освіти спеціальності 051 «Економіка».

Рецензент: Різник Наталія Михайлівна, к.е.н., доцент кафедри економічної кібернетики, ТНТУ імені Івана Пулюя

Розглянуто і затвержено на засіданні кафедри автоматизації технологічних процесів і виробництв. Протокол № 7 від 29 грудня 2023 року.

Рекомендовано науково-методичною комісією факультету прикладних інформаційних технологій та електроінженерії ТНТУ імені Івана Пулюя. Протокол № 5 від 8 лютого 2024 року.

Зміст

Практична робота № 1	4
Концептуальне моделювання предметної області.	
Практична робота № 2	12
Реляційна алгебра. Операції з маніпулювання даними у БД.	
Практична робота № 3	18
Функціональні залежності у відношеннях.	
Практична робота № 4	23
Нормалізація відношень.	
Практична робота № 5	30
Логічне проєктування БД.	
Практична робота № 6	33
Практична реалізація в середовищі Microsoft Access спроектованої БД.	
Практична робота № 7	37
Адміністрування та захист даних в Access.	
Практична робота № 8	38
Визначення об'єктів бази даних за допомогою SQL.	
Практична робота № 9	44
Мова SQL. Вибірка даних.	
Практична робота № 10	54
Мова SQL. Маніпулювання даними.	
Практична робота № 11	57
Визначення прав доступу користувачів до даних.	
Література	59

Практична робота № 1

Тема. Концептуальне моделювання предметної області.

Мета. Набути навичок проєктування баз даних на концептуальному рівні.

Проєктування баз даних – це процес створення проєкту бази даних.

Проєктування баз даних включає чотири етапи:

1. Аналіз предметної області.
2. Побудову концептуальної моделі.
3. Побудову логічної моделі.
4. Побудову фізичної моделі.

На першому етапі проводиться детальний словесний опис об'єктів предметної області і реальних зв'язків, які існують між цими об'єктами. Даний опис має коректно визначити всі взаємозв'язки між об'єктами предметної області.

Головними елементами концептуальної моделі даних є об'єкти та відношення. Об'єкти представляють речі, які користувачі вважають важливими в моделюючій частині реальності. Відношення пов'язують дві об'єктні множини. Відношення саме по собі є об'єктною множиною, що складається з пар об'єктів-елементів, взятих з двох множин, які з'єднують відношення.

Логічне проєктування полягає у визначенні кількості та структури таблиць, формуванні запитів до бази даних, визначенні типів звітних документів, розробці алгоритмів обігу інформації, створенні форм для введення та редагування даних.

Вирішення проблем проєктування на фізичному рівні залежить від використовуваної СКБД.

Завдання 1. Побудувати концептуальну модель даних певної ПО.

Приклад. Предметна область **Ресторан**.

Про кожну **страву** відомі її *назва*, *ціна* та *рецепт*. Кожна страва передбачає певний набір інгредієнтів. Про кожен **інгредієнт** відомі його *назва*, *калорійність* та *міра ваги* (наприклад, штуки, грами, упаковки). Якщо інгредієнт використовується під час приготування певної страви, слід зазначати яка кількість цього інгредієнту необхідна.

Кожна страва належить рівно до однієї **категорії** (наприклад, перші страви, другі страви, закуски, десерти), про яку відомі її *назва* та *опис*. Приготуванням страв кожної з категорій займається окремий **кухар**, про якого відомі його *прізвище*, *ім'я* та *дата народження*.

Назви страв, категорій та інгредієнтів є унікальними. В ресторані не працюють кухарі з одночасно однаковими прізвищем та ім'ям.

Побудова концептуальної моделі бази даних

Аналіз предметної області показує, що основними сутностями є: страва, інгредієнт, категорія, кухар.

Атрибути для кожної сутності виділені в таблиці 1.1.

Атрибути сутностей

Тип сутності	Атрибут
Страва	Назва Ціна Рецепт
Інгредієнт	Назва Калорійність Міра ваги
Категорія	Назва Опис
Кухар	Прізвище Ім'я Дата народження

Типи зв'язків наведені у таблиці 1.2.

Основні типи зв'язків

Тип сутності	Тип зв'язку	Тип сутності
Страва	Має	Інгредієнт
Страва	Належить	Категорія
Кухар	Готує страви	Категорія

Ступінь зв'язку:

- Зв'язок **Має** є зв'язком **багато до багатьох**, необов'язковий з обох боків. Одна і та ж Страва може мати нуль Інгредієнтів, коли страву тільки додано до БД або багато Інгредієнтів. Один

Інгредієнт може належати до нуля Страв, якщо тільки доданий Інгредієнт або до багатьох Страв.

- Зв'язок **Належить** є зв'язком **багато до одного**, обов'язковий з боку Категорія, необов'язковий з боку Страва. Одна Категорія може містити нуль Страв, якщо її тільки створили або багато Страв. Одна Страва належить рівно до однієї Категорії.
- Зв'язок **Готує страви** є зв'язком **один до одного**, обов'язковий з боку Категорія, необов'язковий з боку Кухар. Страву однієї категорії готує один Кухар або нуль, коли Кухар звільнився. Один кухар готує Страви однієї Категорії.

Ключі сутностей наведені у таблиці 1.3.

Таблиця 1.3

Сутності та їх первинні ключі

Сутність	Первинний ключ
Страва	Назва
Інгредієнт	Назва
Категорія	Назва
Кухар	Прізвище, Ім'я

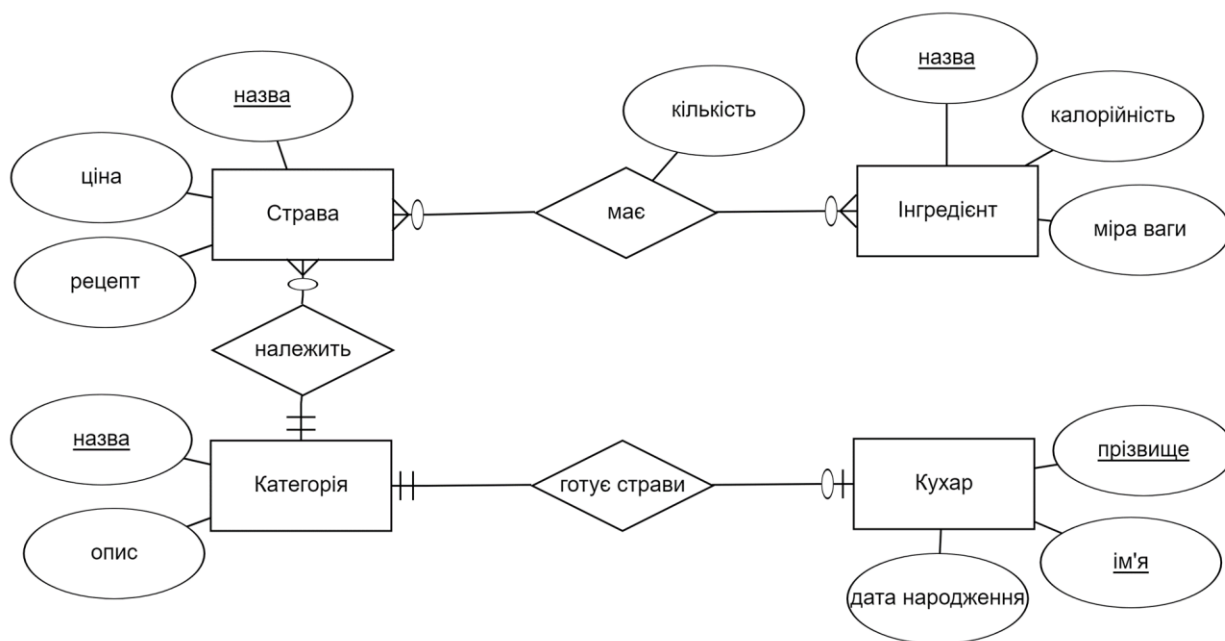


Рис. 1.1. ER-діаграма ПО Ресторан

Створити ER-діаграму можна в середовищі ERDplus <https://er.itolymp.com/>

Завдання для самостійного опрацювання

Завдання 1. Необхідно розробити інформаційну систему для обліку готової продукції, що випускається на підприємстві, база даних якої повинна містити таку інформацію:

- Про готову продукцію: включає унікальний код продукції, назву продукції, собівартість, одиницю вимірювання.
- Про місця зберігання (в якості місць зберігання можуть бути ділянки виробництва, склади). Інформація про місце зберігання включає: його код, назву, відділ чи підрозділ, за яким воно закріплено. Одна і та ж назва готової продукції може зберігатися в різних місцях зберігання.
- Про відділи: включає унікальний номер відділу, назву, прізвище керівника.

- Про співробітників: включає табельний номер, ПІБ співробітника, відділ, в якому він працює, посаду, адресу проживання, оклад.

Готова продукція надходить до місць зберігання відповідно до документа **Накладна на переміщення**. Документ містить: дату виписки і номер, назву підрозділу, який передав і який прийняв продукцію, назву продукції, одиницю вимірювання, кількість переданої продукції.

Дана інформація може бути використана для обліку виробництва готової продукції, заповнення та коригування відомостей про готову продукцію, відслідковування її кількості у всіх місцях зберігання, складання списків співробітників підприємства, коригування даних про відділи підприємства.

Зібрані відомості про дані, які необхідно зберігати в проєктованій базі даних, називають специфікацією вимог підприємства. Виходячи зі специфікації вимог, визначаються основні типи сутностей – об'єкти, дані про які зберігаються в базі даних.

Побудувати концептуальну модель обліку виробництва готової продукції.

Завдання 2. Необхідно розробити інформаційну систему магазину з продажу мобільних телефонів, база даних якої повинна містити таку інформацію:

- Про телефони: включає фірму-виробника, модель телефону, колір корпусу, продажну ціну, наявну кількість, постачальника, закупівельну ціну.

- Про співробітників магазину: включає ПІБ співробітника, посаду, адресу проживання, телефон.
- Про продажі: що, у якій кількості та ким із співробітників продано.
- Про постачальників: назву компанії, юридичну адресу, телефон.

Побудувати концептуальну модель даних магазину.

Завдання 3. Для кожного іменника вибрати є він атрибутом чи сутністю: постачання, індекс постачання, дата постачання, товар, постачальник, ціна, кількість, індекс, індекс постачальника, адреса постачальника, номер складу, назва. Побудувати діаграму моделі ПО Постачання.



Рис. 1.1. Форма постачання

Контрольні запитання

1. Пояснити зміст термінів: база даних (БД), предметна область (ПО), система керування базами даних (СКБД), інформаційна система (ІС), автоматизована інформаційна система (АІС), життєвий цикл бази даних, банк даних (БнД).
2. Охарактеризувати життєвий цикл бази даних. Які основні етапи він включає?
3. Назвати причини використання БД та СКБД.
4. Назвати основні функції СКБД.
5. Назвати основні компоненти інформаційної системи і пояснити їх призначення.
6. Які властивості має мати проєктована БД?
7. Описати компонентний склад середовища СКБД та призначення компонентів.
8. Проаналізувати відомі Вам СКБД з таких точок зору:
 - Яку технологію підтримує система?
 - До якого типу можна віднести систему (персональна, багатокористувацька)?
 - Які мовні засоби має система та до якого типу можна віднести систему за мовою, яку вона використовує?
 - Чи підтримує система функцію ведення розподіленої бази даних?
9. Які основні результати дає етап концептуального моделювання?
10. Пояснити зміст термінів: сутність, атрибут, зв'язок, ступінь зв'язку, ключ, потенційний ключ.

Практична робота № 2

Тема. Реляційна алгебра. Операції з маніпулювання даними у БД.

Мета. Ознайомитись з операціями алгебри відношень.

В реляційних СКБД для виконання операцій над відношеннями використовуються дві групи мов:

1. Реляційна алгебра, яка заснована на теорії множин.
2. Реляційне числення, що базується на математичній логіці – на численні предикатів I-го порядку. Як правило, розглядаються два види реляційного числення – числення доменів і числення предикатів.

Цим механізмам присутня важлива властивість – вони замкнені відносно поняття відношення. Це означає, що вирази реляційної алгебри і формули реляційного числення визначені над відношеннями реляційних БД і результатом також є відношення. В результаті, будь-який вираз чи формула можуть інтерпретуватись як відношення, що дозволяє використовувати їх в інших виразах чи формулах.

Складні запити до баз даних можуть бути виражені за допомогою виразу реляційної алгебри чи формули реляційного числення. Механізми реляційної алгебри і реляційного числення еквівалентні за своїми можливостями і відрізняються на рівні процедурності. Вираз реляційної алгебри будується на основі алгебраїчних операцій високого рівня і має процедурну інтерпретацію. Мови реляційного числення є непроцедурними

(описовими, декларативними) і дозволяють виражати запити за допомогою предиката І-го порядку (висловлювання у вигляді функції), якому повинні задовольняти кортежі чи ділянки) результуючого відношення.

Завдання 1. Дано відношення:

R₁ Виріб 1

Код деталі	Назва	Вага
01	А	1
03	В	2
04	С	3

R₂ Виріб 2

Код деталі	Назва	Вага
02	Д	2
03	В	2
04	С	3

Які типи деталей входять в склад обох виробів?

$$R = R_1 \cup R_2$$

Завдання 2. Дано відношення:

R₁ Потреби

Код деталі	Назва	Вага
01	А	1
02	Д	2
03	В	2
04	С	3
05	Е	1

R₂ Можливості

Код деталі	Назва	Вага
02	Д	2
03	В	2
04	С	3

Відношення R₁ представляє потреби у деяких видах деталей, а відношення R₂ – відомості про ті види деталей, які фірма може виготовити сама. Отримати відношення R, яке містить відомості про ті види деталей, які необхідно придбати.

$$R = R_1 \setminus R_2$$

Завдання 3. Дано відношення:

R₁ Виріб 1

Код деталі	Назва	Вага
01	А	1
02	Д	2
03	В	2
04	С	3
05	Е	1

R₂ Виріб 2

Код деталі	Назва	Вага
02	Д	2
04	С	3
03	В	2
06	К	1

Визначити деталі, які входять в склад обох виробів.

$$R = R_1 \cap R_2$$

Завдання 4. Дано відношення:

R₁ Студенти

Номер залікової книжки	ПІБ
10-001	Бондаренко М.І.
10-002	Коваленко С.М.
10-003	Василенко В.Н.

R₂ Екзамени

Код дисципліни	Дисципліна	Дата	Оцінка
01	математика	16.12.23	
02	фізика	19.12.23	
03	хімія	22.12.23	

Отримати екзаменаційну відомість з усіх дисциплін.

$$R = R_1 \times R_2$$

Із загальної екзаменаційної відомості з усіх дисциплін отримати екзаменаційну відомість з дисципліни математика.

Завдання 5. Дано відношення:

R₁ Відомість

Номер залікової книжки	ПІБ	Дисципліна	Дата
10-001	Бондаренко М.І.	фізика	19.12.23
10-001	Бондаренко М.І.	хімія	22.12.23
10-002	Коваленко С.М.	фізика	19.12.23
10-002	Коваленко С.М.	хімія	22.12.23
10-003	Братченко К.М.	хімія	22.12.23

R₂ Розклад

Дисципліна	Дата
хімія	22.12.23
фізика	19.12.23

Отримати відомості про студентів, які здали всі екзамени.

$$R = R_1 \div R_2$$

Завдання 6. Реалізувати проекцію відношення R₁ на Код товару і Кількість.

Відношення R₁

Код товару	Кількість	Націнка	Дата
1	5	0,2	04.12.23
1	5	0,2	05.12.23
2	2	0,2	04.12.23
3	10	0,2	04.12.23
4	2	0,3	02.12.23

$$R = \pi_{\text{Код товару, Кількість}}(R_1)$$

Завдання 7. Дано відношення:

R₁ Потреби

Код деталі	Назва	Кількість
01	А	1
02	Д	2
03	В	2
04	С	3
05	Е	1

R₂ Склад

Код деталі	Стелаж
01	05
03	10
06	02

Виконати з'єднання відношень R₁ і R₂ за атрибутом Код деталі.

Завдання для самостійного опрацювання

Завдання 1. Дано відношення:

R₁

Прізвище	Вік	Заробітна плата
Богачук	18	20000
Колосюк	20	30000
Павлюк	45	38000

R₂

Прізвище	Вік	Заробітна плата
Панасюк	22	25000
Скрипник	25	32000
Богачук	18	20000

Виконати:

- Об'єднання відношень $R = R_1 \cup R_2$
- Різницю відношень $R = R_1 \setminus R_2$
- Перетин відношень $R = R_1 \cap R_2$
- Проекцію відношення R₁ на атрибути Прізвище і Вік:

$$R = \pi_{\text{Прізвище, Вік}}(R_1)$$

- Селекцію відношення R₁

$$R = \sigma_{(\text{Вік} < 30) \vee (\text{Заробітна плата} > 25000)}(R_1)$$

- Селекцію відношення R_1

$$R = \sigma_{(\text{Вік} < 30) \wedge (\text{Заробітна плата} > 25000)}(R_1);$$

Завдання 2. Дано відношення:

R₁

Прізвище	Населений пункт	Роки
Ліщук	Київ	10
Якименко	Вінниця	15
Ковач	Черкаси	10
Шило	Київ	20
Зелінка	Тернопіль	25

R₂

Місто	Населення
Київ	2950702
Тернопіль	225004

Виконати з'єднання за атрибутом Населений пункт відношення R_1 і

$$\text{Місто відношення } R_2: R = R_1 \underset{N_пункт=Місто}{\bowtie} R_2$$

Завдання 3. Дано відношення:

R₁

Прізвище	Посада	Оклад
Якименко	інженер	10000
Ковач	бухгалтер	30000

R₂

Стаж	Премія
10	10000
20	40000

$$\text{Виконати з'єднання відношень } R_1 \text{ і } R_2: R = R_1 \underset{\text{Оклад} > \text{Премія}}{\bowtie} R_2.$$

Контрольні запитання

1. Що являє собою реляційна алгебра?
2. З яких груп операторів складається реляційна алгебра?
3. Дати визначення теоретико-множинним операціям.
4. Дати визначення спеціальним реляційним операціям.

5. Чи може відношення мати два однакових кортежі після виконання операцій проєкції чи об'єднання?
6. Як змінюється потужність відношень при виконанні операцій реляційної алгебри?

Практична робота № 3

Тема. Функціональні залежності у відношеннях.

Мета. Ознайомитись із концепцією функціональних залежностей.

При проєктуванні схеми реляційної бази даних (логічному моделюванні) використовується базове поняття – функціональна залежність.

Функціональна залежність (ФЗ) є зв'язком типу **багато до одного** між множинами атрибутів усередині даного відношення. Поняття функціональної залежності визначається з поділом функціональних залежностей на виконувани в деяких окремих випадках і виконувани завжди.

Функціональною залежністю між полями X і Y відношення R називається залежність, при якій кожному значенню підмножини X в будь-який момент часу відповідає єдине значення підмножини Y із всіх можливих. Інакше кажучи, якщо два кортежі відношення R співпадають за значенням X , вони також співпадають і за значенням Y .

Алгоритм для перевірки наявності **F**-залежності.

Вхід: відношення R і **F**-залежність $X \rightarrow Y$.

Вихід: істина, якщо R задовольняє $X \rightarrow Y$, хибність – у протилежному випадку.

1. Пересортувати відношення R по X -стовпцям так, щоб зібрати кортежі з рівними X -значеннями разом.
2. Якщо кожна сукупність кортежів з рівними X -значеннями має також рівні Y -значення, то на виході отримуємо істину, в іншому випадку – хибне твердження.

R

A	B	C	D
a1	b1	c1	d1
a2	b2	c1	d1
a1	b1	c2	d2
a3	b1	c2	d3
a2	b2	c3	d2
a1	b1	c3	d4
a4	b3	c4	d2

Рис. 3.1. Відношення R

R

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a1	b1	c3	d4
a2	b2	c1	d1
a2	b2	c3	d2
a3	b1	c2	d3
a4	b3	c4	d2

Рис. 3.2. F-залежність $A \rightarrow B$

R

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a1	b1	c3	d4
a3	b1	c2	d3
a2	b2	c1	d1
a2	b2	c3	d2
a4	b3	c4	d2

Рис. 3.3. $B \not\rightarrow A$

Завдання 1. Дано відношення R з атрибутами A, B, C, D і ФЗ:

$$A \rightarrow BC$$

$$B \rightarrow C$$

$$A \rightarrow B$$

$$AB \rightarrow C$$

$$AC \rightarrow D.$$

Знайти неприведену множину ФЗ, еквівалентну даній множині.

Розв'язок

1. Напишемо ФЗ таким чином, щоб кожна з них мала одноелементну праву частину: $A \rightarrow B$, $A \rightarrow C$, $B \rightarrow C$, $A \rightarrow B$, $AB \rightarrow C$, $AC \rightarrow D$. Залежність $A \rightarrow B$ записана двічі, тому одну з них можна видалити.
2. Атрибут C є надлишковим в лівій частині залежності $AC \rightarrow D$, оскільки є залежність $A \rightarrow C$, і доповненням можна отримати $A \rightarrow AC$. Оскільки дано $AC \rightarrow D$, за допомогою транзитивності отримуємо $A \rightarrow D$.

3. Залежність $AB \rightarrow C$ можна виключити, оскільки є залежність $A \rightarrow C$ і за допомогою доповнення можна отримати $AB \rightarrow CB$, а за допомогою декомпозиції $AB \rightarrow C$.
4. Залежність $A \rightarrow C$ розуміється залежностями $A \rightarrow B$ і $B \rightarrow C$.
5. Отримана неприведена множина залежностей: $A \rightarrow B$ і $B \rightarrow C$, $A \rightarrow D$.

Завдання 2. Дано відношення R з атрибутами (Інд.№, Прізвище, Ім'я, По батькові, Спеціальність, Факультет) і ФЗ:

Інд.№ \rightarrow Прізвище

Інд.№, Прізвище \rightarrow Спеціальність

Спеціальність \rightarrow Факультет

Прізвище, Ім'я \rightarrow Факультет

Не мають місця у відношенні R ФЗ:

Факультет \rightarrow Спеціальність

Прізвище \rightarrow Спеціальність

Застосувати властивості ФЗ для виводу нових на основі заданих.

Завдання для самостійного опрацювання

Завдання 1. Дано відношення Науково-дослідний центр.

Таблиця 3.1. НДЦ

Науково-дослідний центр				
№ співробітника	Назва лабораторії	Назва проекту	Місце розташування (№ корпусу)	Науковий керівник лабораторії
A	B	C	D	E
1	Lab_a	Project_1	8	Петренко
2	Lab_a	Project_1	8	Петренко
3	Lab_a	Project_4	8	Петренко
1	Lab_b	Project_2	15	Кириченко
2	Lab_b	Project_5	15	Кириченко
3	Lab_b	Project_2	15	Кириченко
1	Lab_c	Project_3	8	Петренко
2	Lab_c	Project_3	8	Петренко

Первинний ключ: {A,B}. Виділити всі можливі ФЗ.

Завдання 2. Дано відношення Звіт.

Таблиця 3.2. Звіт

№ звіту	Дата	Час	№ Таб	ФІО	№ авто
A	B	C	D	E	F
ЗВ4	18.03.23	10.00	37	Кристя Н.М.	КА2979
ЗВ4	22.03.23	9.00	15	Верес М.І.	ВО5132
ЗВ4	01.03.23	9.30	15	Верес М.І.	ТО7028
ЗВ16	22.03.23	13.00	15	Верес М.І.	ВО5132
ЗВ16	24.03.23	9.00	37	Кристя Н.М.	ТО7028

Виділити всі можливі ФЗ.

Контрольні запитання

1. Дати визначення функціональної залежності, сформулювати аксіоми, яким такі залежності відповідають.
2. Природа та зміст функціональної залежності у відношеннях.
3. Повні функціональні залежності та їх застосування.
4. Що таке транзитивна залежність? До яких негативних наслідків призводить наявність транзитивної залежності?
5. Що означає, що поля є взаємо незалежними?
6. Навести приклад, де присутні багатозначні залежності.

Практична робота № 4

Тема. Нормалізація відношень.

Мета. Ознайомитись з теорією нормалізації при проєктуванні баз даних.

Концепцію реляційної моделі запропонував американський учений, математик Едгар Франк Кодд у 1970 році. Реляційна модель заснована на математичному понятті відношення (лат. relation), фізичним представленням якого є таблиця (table). Схема реляційного відношення задається ім'ям відношення, іменами стовпців, ключовим атрибутом.

Фундаментальні властивості відношень:

- Відсутність кортежів-дублікатів.
- Відсутність упорядкованості кортежів.
- Відсутність упорядкованості атрибутів.

- Значення атрибута є атомарним.

Нормалізація (normalization) – це процес побудови баз даних за спеціальною методикою, що передбачає дотримання певного набору норм і правил.

Таблиця вважається нормалізованою на певному рівні, якщо вона задовольняє умовам, що накладаються відповідною формою нормалізації. При переході до наступної нормальної форми (НФ) властивості попередніх нормальних форм зберігаються.

Перша нормальна форма є основою реляційної моделі.

Відношення перебуває в **1НФ**, якщо всі його атрибути мають прості (атомарні) значення.

Відношення перебуває в **2НФ**, якщо воно знаходиться в 1НФ, і кожен неключовий атрибут функціонально повно залежить від унікального ідентифікатора (складного первинного ключа) в цілому. Якщо таблиця має простий первинний ключ, який складається лише з одного поля, то вона автоматично знаходиться в 2НФ.

Для приведення таблиці до 2НФ необхідно:

1. Побудувати її проекцію, виключивши атрибути, які не знаходяться в повній функціональній залежності від складного ключа.
2. Побудувати додатково одну або декілька проекцій на частину складеного ключа та атрибути, які функціонально залежать від цієї частини ключа.

Відношення перебуває в **3НФ**, якщо воно знаходиться в 2НФ і перехідні залежності відсутні (всі неключові атрибути залежать лише від унікального ідентифікатора).

Означення 3НФ не підходить для таблиць, в яких:

- Є два і більше потенційних ключа (які є складними).
- Ці потенційні ключі перекриваються, тобто мають хоча б одне спільне поле.

Нормальна форма Бойса-Кодда враховує функціональні залежності, в яких беруть участь всі потенційні ключі таблиці, а не лише її первинний ключ. Для відношення з одним потенційним ключем його 3НФ і НФБК є еквівалентними.

Відношення перебуває в **НФБК** тоді і лише тоді, коли кожен його детермінант є потенційним ключем.

1НФ, 2НФ, 3НФ обмежують залежність непервинних атрибутів від ключів. НФБК обмежує залежність первинних атрибутів.

Алгоритм декомпозиції без втрат, що приводить відношення до НФБК:

1. Визначається ФЗ, яка є причиною того, що відношення не перебуває в НФБК.
2. Виконується проекція відношення на детермінант і залежну частину знайденої ФЗ.
3. Зі схеми вихідного відношення видаляються атрибути, які входять у залежну частину ФЗ.
4. Відношення, отримані в п. 2, п. 3 будуть результатом декомпозиції. Ці відношення перевіряються на відповідність НФБК і, при необхідності, алгоритм повторюється.

Завдання 1. Дано відношення Студент в ненормалізованій формі.

Таблиця 4.1. Студент

Номер залікової книжки	ПІБ студента	Група	ПІБ старости	Куратор
10-001	Іваненко І.С.	ПК-11	Колосюк М.А.	Доц. Ліщук І.В.
10-002	Колосюк М.А.			
10-003	Панасюк В.І.			
10-004	Скрипчук Л.П.	ПК-12	Павлюк О.М.	Доц. Богачук К.В.

1. Перетворити відношення Студент до 1НФ. Який недолік має відношення в 1НФ?
2. Які аномалії присутні у відношенні в 1НФ?
3. Перетворити відношення 1НФ таким чином, щоб позбутись присутніх у ньому аномалій.

При виконанні декомпозиції відношення необхідно знайти такий склад відношень, який би задовольняв наступним вимогам:

- Множина відношень повинна забезпечувати мінімізацію надлишковості представлення інформації.
- Призначені для відношення ключі повинні бути мінімальними.
- При виконанні операцій видалення, додавання, оновлення в БД не повинно бути аномалій.

Завдання 2. Дано відношення Науково-дослідний центр

Таблиця 4.2. НДЦ

Науково-дослідний центр				
№ співробітника	Назва лабораторії	Назва проекту	Місце розташування (№ корпусу)	Науковий керівник лабораторії
A	B	C	D	E
1	Lab_a	Project_1	8	Петренко М.А.
2	Lab_a	Project_1	8	Петренко М.А.
3	Lab_a	Project_4	8	Петренко М.А.
1	Lab_b	Project_2	15	Кириченко Н.К.
2	Lab_b	Project_5	15	Кириченко Н.К.
3	Lab_b	Project_2	15	Кириченко Н.К.
1	Lab_c	Project_3	8	Петренко М.А.
2	Lab_c	Project_3	8	Петренко М.А.

Первинний ключ відношення: {A,B}.

Деякі з функціональних залежностей:

$\{A,B\} \rightarrow C$

$B \rightarrow E$

$B \rightarrow D$

$D \rightarrow E$

Які аномалії присутні у даному відношенні?

Перетворити дане відношення таким чином, щоб позбутись присутніх у ньому аномалій.

Завдання 3. Дано відношення Звіт.

Таблиця 4.3. Звіт

№звіту	Дата	Час	Таб№	ПІБ	№ авто
A	B	C	D	E	F
ЗВ4	18.03.23	10.00	37	Кристя Н.М.	КА2979
ЗВ4	22.03.23	9.00	15	Верес М.І.	ВО5132
ЗВ4	01.03.23	9.30	15	Верес М.І.	ТО7028
ЗВ16	22.03.23	13.00	15	Верес М.І.	ВО5132
ЗВ16	24.03.23	9.00	37	Кристя Н.М.	ТО7028

Функціональні залежності:

$$\{D, B\} \rightarrow F$$

$$\{F, B, C\} \rightarrow \{A, D\}$$

$$\{D, B, C\} \rightarrow \{A\}$$

Які аномалії присутні у даному відношенні?

Перетворити відношення до НФБК.

Завдання для самостійного опрацювання

Завдання 1. Схема реляційного відношення Постачання (Дата постачання, №постачальника, Товар, Адреса постачальника, Кількість товару, Ціна одиниці товару).

Які аномалії присутні у даному відношенні?

Перетворити дане відношення таким чином, щоб позбутись присутніх у ньому аномалій.

Завдання 2. Схема реляційного відношення R (№ лікаря, № пацієнта, Дата, ПІБ пацієнта, Адреса пацієнта, ПІБ лікаря, Лікування, Ліки, Побічний ефект).

Які аномалії присутні у даному відношенні?

Перетворити дане відношення таким чином, щоб позбутись присутніх у ньому аномалій.

Завдання 3. Схема реляційного відношення Консультації дипломників (№ викладача, № залікової книжки, Дата, ПІБ викладача, Посада, ПІБ студента, Тема диплома, Час, Аудиторія, Місткість аудиторії).

Які аномалії присутні у даному відношенні?

Перетворити дане відношення таким чином, щоб позбутись присутніх у ньому аномалій.

Завдання 4. Схема реляційного відношення Постачання (Індекс постачальника, Ім'я постачальника, Індекс товару, Кількість товару). Значення атрибута Ім'я постачальника є унікальним.

Які аномалії присутні у даному відношенні?

Перетворити дане відношення до НФБК.

Контрольні запитання

1. Які є відмінності між математичним і реляційним відношеннями?
2. Що таке ключ у реляційному відношенні?
3. Прості та складені ключі, їх відмінності.
4. Первинні та потенційні ключі, їх особливості.
5. Способи визначення зв'язків між відношеннями за допомогою ключів.
6. Визначити поняття NULL-значення.

7. Пояснити зміст термінів: надлишковість даних; аномалії додавання, оновлення, видалення; нормалізація відношень; декомпозиція; перша нормальна форма; друга нормальна форма; третя нормальна форма; нормальна форма Бойса-Кодда.
8. Навести приклади відношень, які не перебувають в 1НФ.

Практична робота № 5

Тема. Логічне проєктування БД.

Мета. Набути навичок відображення концептуальної моделі предметної області на логічну модель вибраної СКБД.

Завдання 1. Перетворити концептуальну модель Ресторан (створена за завданням в роботі № 1) в логічну для реляційної СКБД.

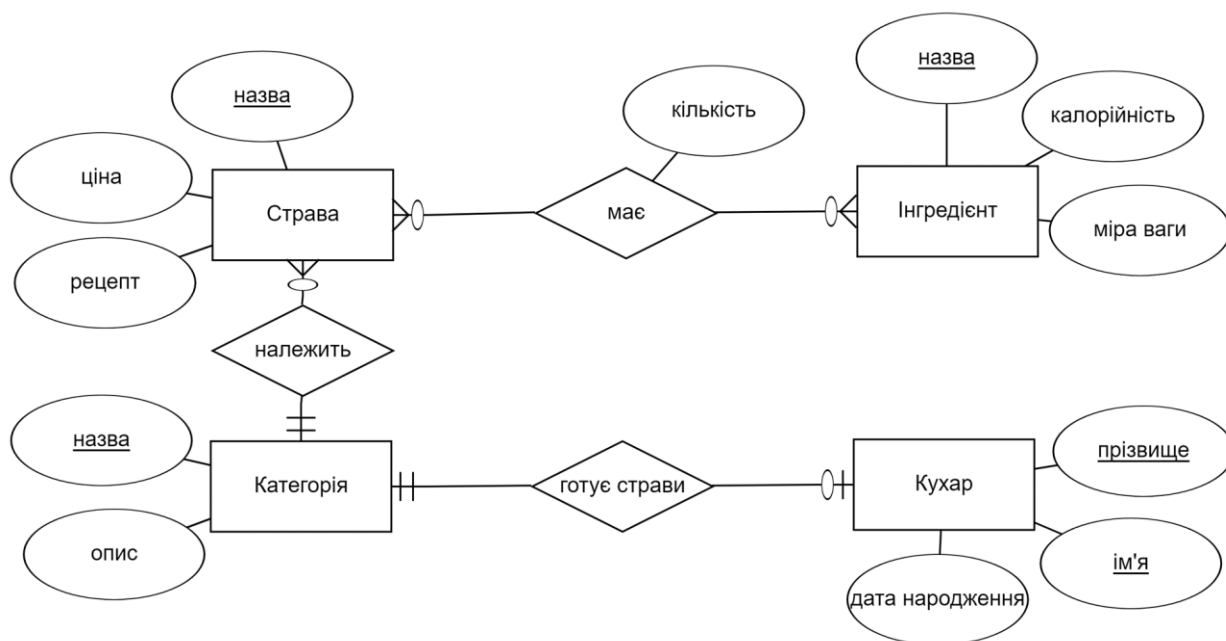


Рис. 5.1. ER-діаграма ПО Ресторан

БД складається з чотирьох інформаційних об'єктів: Страви, Інґредієнти, Категорії, Кухарі. Зв'язок між об'єктами Страви та

Інгредієнти – **багато до багатьох**. Множинні зв'язки ускладнюють керування базою даних. При множинних зв'язках, наприклад, у СКБД Access не можна використовувати механізм каскадного оновлення. Використовувати такі зв'язки небажано. Створимо допоміжний об'єкт зв'язку Інгредієнти страви, який складається з ключових реквізитів об'єктів Страви та Інгредієнти і доповнений описовим реквізитом **Кількість**.

У реляційній базі даних як об'єкти розглядаються відношення, які можна представити у вигляді таблиць. Таблиці між собою зв'язуються за допомогою загальних полів, тобто однакових за форматом і, як правило, за назвою, що є в обох таблицях.

Зв'язаність даних у таблицях Страви та Категорії забезпечує поле **Назва категорії**. Зв'язаність даних у таблицях Страви та Інгредієнти страви забезпечує поле **Назва страви**. Зв'язаність даних у таблицях Інгредієнти Страви та Інгредієнти забезпечує поле **Назва інгредієнта**. Зв'язаність даних у таблицях Категорії та Кухарі забезпечує поле **Назва категорії**.

Схеми відношень:

Страви (**Назва страви**, Ціна, Рецепт, Назва категорії).

Інгредієнти страви (**Назва страви**, **Назва інгредієнта**, **Кількість**).

Інгредієнти (**Назва інгредієнта**, Калорійність, Міра ваги).

Категорії (**Назва категорії**, Опис).

Кухарі (**Прізвище**, **Ім'я**, Дата народження, Назва категорії).

Створені відношення задовольняють вимогам 1НФ, 2НФ, 3НФ.



Рис. 5.2. Логічна модель бази даних

Створити реляційну схему можна в середовищі ERDplus <https://er.itolymp.com/>

Завдання для самостійного опрацювання

Завдання 1. Перетворити концептуальну модель обліку виробництва готової продукції (створена за завданням в роботі № 1) в логічну для реляційної СКБД.

Завдання 2. Перетворити концептуальну модель даних магазину з продажу мобільних телефонів (створена за завданням в роботі № 1) в логічну для реляційної СКБД.

Контрольні запитання

1. Пояснити зміст термінів: реляційна модель даних, реляційна схема БД, відношення, кортеж, атрибут, первинний і зовнішній ключі, домен, кардинальне число, ступінь відношення.
2. Як реалізується підтримка цілісності БД?
3. Що називається каскадним видаленням та оновленням?
4. Описати кроки перетворення ER-моделі на реляційну БД.
5. Чому проєкт бази даних може бути невдалим?

Практична робота № 6

Тема. Практична реалізація в середовищі Microsoft Access спроектованої БД.

Мета. Вивчити концепцію та функціональні можливості реляційних СКБД на прикладі Microsoft Access.

Фізична модель залежить від вибраної СКБД, визначає способи розміщення даних у середовищі зберігання і способи доступу до них, які підтримуються на фізичному рівні.

Microsoft Access – повноцінна реляційна СКБД, у якій може працювати як пересічний користувач, який не володіє основами програмування, так і фахівець-розробник, який створює додатки на мові Visual Basic for Applications (VBA).

Microsoft Access є одним із компонентів пакета Microsoft Office. Основна функція Microsoft Access – робота із структурованою у вигляді таблиць інформацією. Microsoft Access дозволяє забезпечувати введення даних у таблиці БД, їхній супровід та зберігання, а також отримувати із сукупності цієї інформації необхідні дані для прийняття важливих рішень. Access підтримує мови запитів QBE (Query By Example), параметри яких встановлюються у вікні Конструктора запитів, і SQL (Structured Query Language), при створенні яких використовуються оператори і функції мови SQL, та забезпечує перехід між двома формами задання запиту.

Завдання 1. Реалізувати базу даних для ПО Ресторан (створена за завданням в роботі № 5) в середовищі СКБД Access.

Таблиця 6.1.

Таблиці БД Ресторан

Назва	Поля	Тип	Властивості поля	Ключ
Категорії	Назва категорії	Короткий текст	Обов'язкове поле: Так	Первинний К
	Опис	Довгий текст		
Страви	Назва страви	Короткий текст	Обов'язкове поле: Так	К
	Ціна	Число		
	Рецепт	Довгий текст		
	Назва категорії	Майстер підстановок... значення з таблиці Категорії ... Назва категорії		Зовнішній (FK)
Інгредієнти	Назва інгредієнта	Короткий текст	Обов'язкове поле: Так	К
	Калорійність	Число		
	Міра ваги	Короткий текст		
Інгредієнти страви	Назва страви	Майстер підстановок... значення з таблиці	Обов'язкове поле: Так	К

		Страви... Назва страви		
	Назва інгредієнта	Майстер підстановок... значення з таблиці Інгредієнти ... Назва інгредієнта	Обов'язкове поле: Так	К
	Кількість	Число	Правило перевірки: > 0	
Кухарі	Прізвище	Короткий текст	Обов'язкове поле: Так	К
	Ім'я	Короткий текст	Обов'язкове поле: Так	К
	Дата народження	Дата й час		
	Назва категорії	Майстер підстановок... значення з таблиці Категорії ... Назва категорії	Індексовано: Так (Без повторень)	FK, U

- Створити базу даних Ресторан.
- Створити структури таблиць бази даних Ресторан.
- Встановити зв'язки між таблицями.

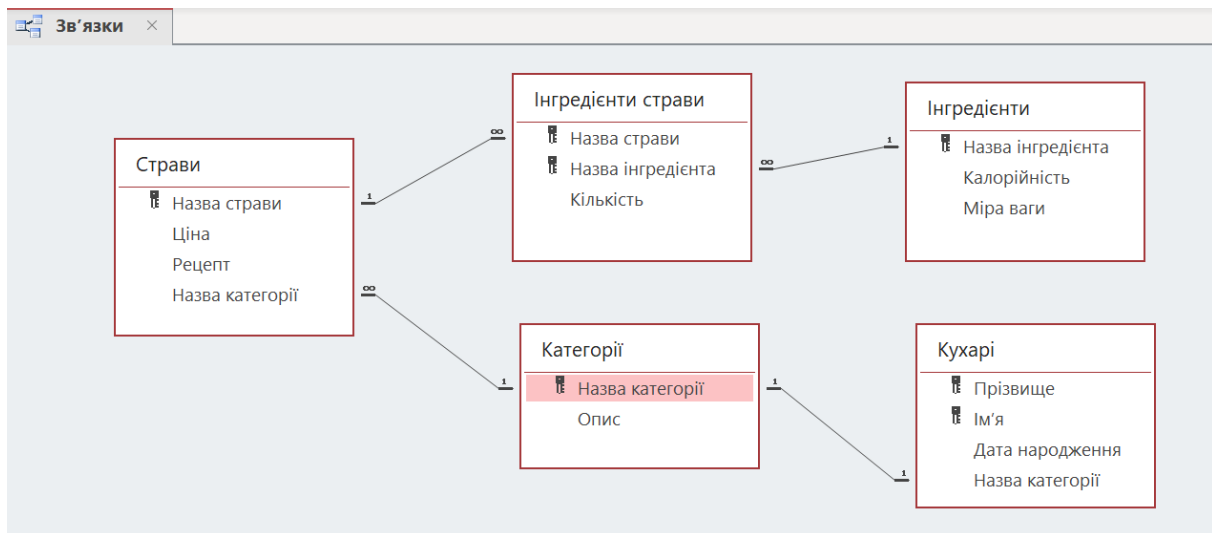


Рис. 6.1. Схема даних із встановленими зв'язками

- Заповнити таблиці конкретними даними.
- Реалізувати типові запити до БД Ресторан. Проаналізувати правильність їх виконання.
- Реалізувати типові форми для БД Ресторан. Проаналізувати правильність їх створення.
- Реалізувати типові звіти для БД Ресторан. Проаналізувати правильність їх виконання.

Завдання для самостійного опрацювання

Завдання 1. Реалізувати базу даних для ПО Облік готової продукції (створена за завданням в роботі № 5) в середовищі СКБД Access.

Завдання 2. Реалізувати базу даних для ПО Магазин (створена за завданням в роботі № 5) в середовищі СКБД Access.

Контрольні запитання

1. Яка модель бази даних залежить від вибраної СКБД?
2. Які параметри слід враховувати при виборі цільової СКБД?

3. Описати роботу з конструктором таблиць. Які типи даних доступні у СКБД Access?
4. Що таке схема даних? Як вона створюється?
5. Які обмеження семантичної цілісності можна задати в Access?
6. Які типи запитів можна створювати в СКБД Access?
7. Чи є мова QBE реляційно повною?
8. Як виконати обчислення у запитах СКБД Access?
9. Екранні форми – види та призначення.
10. Звіти – види та порядок створення.

Практична робота № 7

Тема. Адміністрування та захист даних в Access.

Мета. Розглянути засоби адміністрування баз даних та захисту даних від несанкціонованого доступу.

Створення файлів MDE – це засіб захисту форм, звітів і модулів. Зберігання бази даних як MDE-файлу зробить неможливим виконання таких дій:

- Перегляд, зміна або створення форм, звітів чи модулів у режимі конструктора.
- Зміна назв об'єктів.
- Зміна модулів на Visual Basic.
- Імпорт і експорт форм, звітів чи модулів. Таблиці, запити і макроси можна імпортувати і експортувати в бази даних, які не є MDE-файлами.

Завдання 1. Створити MDE-файл. Перед створенням MDE-файлу зберегти початкову копію БД. Використовується база даних, створена за завданням у роботі № 6.

Завдання 2. Встановити парольний захист БД.

Контрольні запитання

1. Охарактеризувати призначення операцій стиснення та відновлення БД.
2. Назвати основні засоби захисту від несанкціонованого доступу в Access.
3. Які обмеження прав доступу може призначити адміністратор БД в Access?
4. У чому полягає суть захисту на рівні користувача?

Практична робота № 8

Тема. Визначення об'єктів бази даних за допомогою SQL.

Мета. Вміти створювати і модифікувати структуру об'єктів бази даних.

Завдання 1. Створити таблиці за допомогою SQL запиту (СКБД Microsoft Access або СКБД SQLite, середовище розробки баз даних Datagrip).

- Написати команду CREATE TABLE для створення таблиці **Student**.
- Написати команду CREATE TABLE для створення таблиці **Lectures**.

- Написати команду CREATE TABLE для створення таблиці **Subject**.
- Написати команду CREATE TABLE для створення таблиці **University**.
- Написати команду CREATE TABLE для створення таблиці **Exam_Marks**.
- Написати команду CREATE TABLE для створення таблиці **Subj_Lect**.

Таблиця 8.1. Student (Студент)

Student ID	Surname	Name	Stipend	Kurs	City	Univ ID
1	Іваненко	Петро	1400	1	Тернопіль	10
3	Малашко	Олександр	1200	3	Львів	10
6	Панасюк	Вадим	2000	4	Київ	22
10	Кузьменко	Борис	0	2	Тернопіль	10
12	Лебеденко	Ольга	1100	2	Хмельницьк	10
50	Тесленко	Світлана	0	3	Луцьк	10
32	Іваненко	Олександр	1400	5	Львів	14
90	Орищенко	Юлія	2000	3	Тернопіль	10
100	Назаренко	Андрій	2000	4	Івано-Франківськ	22
55	Павленко	Вадим	1200	5	Львів	10
...	

Student_ID – числовий код, який ідентифікує студента.

Surname – прізвище студента.

Name – ім'я студента.

Stipend – стипендія, яку отримує студент.

Kurs – курс, на якому вчиться студент.

City – місто, в якому живе студент.

Birthday – дата народження студента.

Univ_ID – числовий код, який ідентифікує університет, в якому вчиться студент.

Таблиця 8.2. Lecture (Викладач)

Lecture_ID	Surname	Name	City	Univ_ID
24	Кондратенко	Борис	Тернопіль	10
46	Ліщук	Іван	Тернопіль	10
35	Вітенко	Павло	Київ	22
81	Лавренко	Тетяна	Київ	22
92	Марків	Віталій	Львів	5
50	Мілян	Андрій	Львів	5
...

Lecture_ID – числовий код, який ідентифікує викладача.

Surname – прізвище викладача.

Name – ім'я викладача.

City – місто, в якому живе викладач.

Univ_ID – ідентифікатор університету, в якому працює викладач.

Таблиця 8.3. Subject (Навчальний предмет)

Subj_ID	Subj_Name	Hour	Semester
10	Інформатика	56	1
22	Фізика	36	1
43	Філософія	36	2
15	Математика	56	2
20	Історія	36	4
23	Англійська мова	56	3
...

Subj_ID – ідентифікатор навчального предмета.

Subj_Name – навчальний предмет.

Hour– кількість годин, яка відводиться на вивчення предмета.

Semester – семестр, в якому вивчається предмет.

Таблиця 8.4. University (Університет)

Univ_ID	Univ_Name	Rating	City
7	КНУ ім. Т Шевченка	1	Київ
5	ЛНУ ім. І. Франка	4	Львів
10	ТНТУ ім. І. Пулюя	9	Тернопіль
22	Національний університет “Києво-Могилянська академія”	5	Київ
20	ЧНУ ім. Юрія Федьковича	21	Чернівці
...	

Univ_ID – ідентифікатор університету.

Univ_Name – назва університету.

Rating – рейтинг університету.

City – місто, в якому знаходиться університет.

Таблиця 8.5. Exam_Marks (Екзаменаційні оцінки)

Exam_ID	Student_ID	Subj_ID	Mark	Exam_Date
24	12	10	5	12/12/2023
16	8	10	4	23/12/2023
10	9	10	5	05/12/2023
18	25	22	3	22/12/2023
15	32	22	NULL	10/12/2023
2	6	22	4	18/12/2023
...

Exam_ID – ідентифікатор екзамену.

Student_ID – ідентифікатор студента.

Subj_ID – ідентифікатор навчального предмета.

Mark Semester– екзаменаційна оцінка.

Exam_Date – дата здачі екзамену.

Таблиця 8.6. Subj_Lect (Навчальні дисципліни викладачів)

Lecture_ID	Subj_ID
24	10
46	22
35	43
81	5
92	56
100	94
...	...

Lecture_ID – ідентифікатор викладача.

Subj_ID – ідентифікатор навчального предмета.

Створити таблицю Exam_Marks таким чином, щоб не допускалося проведення двох екзаменів по будь-яких предметах в один день.

Створити таблицю Subject таким чином, щоб кількість відведених на предмет годин за замовчуванням була рівною 36, не допускались записи з відсутньою кількістю годин, поле Subj_ID було первинним ключем таблиці і значення семестрів (поле Semester) належали б діапазону від 1 до 12.

Заборонити значення Null в полях, Subj_ID, Student_ID в таблиці Exam_Marks.

Завдання для самостійного опрацювання

Завдання 1. Створити таблицю Subject_1 з тими ж полями, що в таблиці Subject. Поле Subj_ID є первинним ключем.

Завдання 2. Створити таблицю Subj_Lect_1 з полями Lecturer_ID і Subj_ID. Первинним ключем таблиці є пара атрибутів Lecturer_ID і Subj_ID. Поле Lecturer_ID є зовнішнім ключем, який зсилається на таблицю Lecturer_1, аналогічну таблиці Lecturer, а поле Subj_ID є зовнішнім ключем, що зсилається на таблицю Subject_1, аналогічну таблиці Subject.

Завдання 3. Створити таблицю Lecturer_1 з тими ж полями, що в таблиці Lecturer. Первинним ключем таблиці є атрибут Lecturer_ID. Поле Univ_ID є зовнішнім ключем, що зсилається на таблицю University_1. Для цього поля встановити каскадні режими забезпечення цілісності для команд Update і Delete.

Контрольні запитання

1. Призначення та загальна характеристика мови SQL.
2. Які оператори SQL входять до мови опису даних – DDL?
Охарактеризувати призначення операторів та подати їх синтаксис.

Практична робота № 9

Тема. Мова SQL. Вибірка даних.

Мета. Вміти робити вибірку інформації з таблиць БД.

Мова запиту даних (Data Query Language DQL) використовується для вибірки даних із однієї чи кількох таблиць БД. Головною інструкцією цієї мови є команда SELECT. В операторі SELECT обов'язковими є два ключових слова – SELECT і FROM.

Спрощений синтаксис оператора SELECT:

```
SELECT [DISTINCT] <список атрибутів>  
FROM <список таблиць>  
[WHERE <умова вибірки>]  
[ORDER BY <список атрибутів>]  
[GROUP BY <список атрибутів>]  
[HAVING <умова>]  
[UNION <вираз з оператором SELECT>];
```

У квадратних дужках вказані елементи, які можуть бути відсутні у запиті.

За допомогою оператора SELECT можна виконувати три типи операцій реляційної алгебри: селекція, проекція і з'єднання.

Специфікатор WHERE використовується, коли потрібно задати умови на записи, які слід опрацювати. Якщо не задавати WHERE, то результатом запиту будуть всі рядки таблиці. Якщо в запиті беруть участь кілька таблиць і не задано WHERE, то результатом виконання буде скалярний добуток таблиць.

У пропозиції WHERE записується логічна умова, що отримується за допомогою логічних операторів AND, OR, NOT.

У запиті SELECT в склад виразів можуть входити одна або декілька функцій. COUNT – повертає кількість значень у вказаному стовпці, AVG(стовпець) – повертає середнє значення у вказаному стовпці, MAX(стовпець), MIN(стовпець), SUM(стовпець), STDEV(стовпець) – повертає середньоквадратичне відхилення у стовпці, VAR(стовпець) – дисперсія у стовпці, називаються функціями набору і використовуються для отримання зведеної інформації по групі рядків.

В якості оператора даних функцій може використовуватися назва лише одного стовпця, і всі вони повертають єдине значення.

З функціями SUM і AVG можуть використовуватись тільки числові поля. З функціями COUNT, MAX і MIN можуть використовуватись як числові, так і символні поля. При виклику всіх перерахованих вище функцій, крім функцій COUNT(*), здійснюється виключення всіх порожніх значень і тільки після цього операція застосовується до значень стовпця, які залишилися. Функція COUNT(*) виконує підрахунок всіх рядків таблиці незалежно від того, яке значення в них знаходиться.

Агрегатні функції неможна використовувати в пропозиції WHERE, тому що предикати оцінюються в термінах одного рядка, а агрегатні функції – в термінах груп рядків.

Специфікатор GROUP BY використовується при формуванні підсумкових запитів. Об'єднує записи з однаковими значеннями у вказаному списку полів в один запис і видає результуючий запис.

Специфікатор HAVING доповнює GROUP BY за допомогою уточнюючих умов до груп рядків, після їх формування. HAVING визначає критерій, за яким групи необхідно включати у вихідні дані.

Для видачі результатів запиту у відсортованому по якому-небудь стовпцю вигляді, необхідно використати ключове слово ORDER BY. За замовчуванням, для числових даних здійснюється сортування в порядку зростання, а для символічних даних впорядкування буде здійснюватися по буквах алфавіту.

Якщо необхідно здійснити сортування за спаданням, то додається ключове слово DESC:

```
[ORDER BY ім'я_поля1 [ASC|DESC][, ім'я_поля2  
[ASC|DESC]][,...]]
```

Завдання 1. Здійснити вибірку всіх значень полів Name і Surname з таблиці Student. Результатом виконання запиту буде таблиця 9.1., яка містить наступну інформацію:

Таблиця 9.1.

Name	Surname
Петро	Іваненко
Олександр	Малашко
Вадим	Панасюк
Борис	Кузьменко
Ольга	Лебеденко
Світлана	Тесленко
Олександр	Іваненко
Юлія	Орищенко
Андрій	Назаренко
Вадим	Павленко
...	...

Порядок стовпців у результуючій таблиці відповідатиме порядку полів Name і Surname, вказаному у запиті, а не порядку у вхідній таблиці Student.

Завдання 2. Вивести значення всіх стовпців таблиці Student. Результатом виконання запиту буде вся таблиця Student.

Завдання 3. Отримати список назв міст, де проживають студенти, відомості про яких знаходяться в таблиці Student.

Результатом запиту буде таблиця 9.2.

Таблиця 9.2.

City
Тернопіль
Львів
Київ
Тернопіль
Хмельницьк
Луцьк
Львів
Тернопіль
Івано-Франківськ
Львів
...

В таблиці зустрічаються однакові стрічки. Для виключення із результату SELECT-запиту записів, які повторюються, використовується ключове слово DISTINCT (відмінний). Якщо запит SELECT вибирає множину полів, то DISTINCT виключає дублікати рядків, в яких значення всіх вибраних полів є ідентичними. Ключове слово ALL, на відміну від DISTINCT, вибирає всі записи, які задовольняють умову в інструкції SQL. Режим, що задається ключовим словом ALL діє за замовчуванням, тому в реальних запитах він практично не використовується.

Завдання 4. Отримати таблицю, в якій дублікати стрічок (міст) виключені.

Завдання 5. Написати запит, який виконує вибірку імен всіх студентів, з прізвищем Іваненко (таблиця Student).

Результатом запиту буде таблиця 9.3.

Таблиця 9.3.

Surname	Name
Іваненко	Петро
Іваненко	Олександр

Завдання 6. Написати запит, який виконує вибірку імен і прізвищ студентів, які навчаються на третьому курсі і отримують стипендію (розмір стипендії більше нуля).

Завдання 7. Отримати відомості про студентів (таблиця Exam_Marks), у яких екзаменаційні оцінки лише 4 і 5.

Завдання 8. Отримати відомості про студентів (таблиця Exam_Marks), у яких немає жодної екзаменаційної оцінки, рівної 4 або 5.

Завдання 9. Вивести записи про предмети, на вивчення яких відводиться кількість годин в межах від 30 до 40.

Завдання 10. Написати запит, який здійснює вибірку даних з таблиці Student про студентів, прізвища яких починаються на букву "М".

Завдання 11. Визначити середнє значення поля Mark (Оцінки) за всіма записами таблиці Exam_Marks.

Завдання 12. Підрахувати і вивести загальну кількість стрічок в таблиці Exam_Marks.

Завдання 13. Знайти максимальне значення оцінки, отриманої кожним студентом.

Завдання 14. Вивести дані з таблиці Subject у вигляді, відсортованому по назвах предметів за зростанням.

Завдання 15. Вивести дані з таблиці Subject у вигляді, відсортованому по назвах предметів за спаданням.

Завдання 16. Впорядкувати список навчальних предметів за значеннями семестрів, а всередині – за назвами предметів.

Завдання для самостійного опрацювання

Завдання 1. Написати запит, який виводить відомості про номер предмета, його назву, семестр, в якому він читається і кількість на нього відведених годин для всіх стрічок таблиці Subject.

Завдання 2. Написати запит, який виводить всі стрічки таблиці Exam_Marks, для яких предмет навчання (Subj_ID) має номер 10.

Завдання 3. Написати запит, який виводить всі дані з таблиці Student, розташувавши стовпці таблиці в такому порядку: Kurs, Surname, Name, Stipend.

Завдання 4. Написати запит, який виводить назву предмета навчання (Subject_Name) і кількість годин (Hour) для кожного предмета (Subject) в 4-му семестрі (Semester).

Завдання 5. Написати запит, який дозволяє отримати з таблиці Exam_Marks значення стовпця Mark (екзаменаційна оцінка) для всіх студентів, виключивши із списку повторення однакових стрічок.

Завдання 6. Написати запит, який виводить список прізвищ студентів, які навчаються на 3, 4, 5 курсах.

Завдання 7. Написати запит, який виводить дані про прізвище, ім'я та номер курсу для студентів, які отримують стипендію > 1500 грн.

Завдання 8. Написати запит, який виконує вибірку з таблиці Subject назв всіх предметів навчання, на які відводиться більше 30 годин.

Завдання 9. Написати запит, який виводить список університетів, рейтинг яких перевищує 20 балів.

Завдання 10. Написати запит до таблиці Student, який виводить список прізвищ (Surname), імен (Name), номера курсу (Kurs) всіх студентів, які отримують стипендію більше чи рівно 2000, і проживають в Тернополі.

Завдання 11. Написати запит до таблиці Exam_Marks, який виводить номери предметів навчання, з яких здавалися екзамени в період між 20 і 30 грудня 2023 року.

Завдання 12. Написати запит, який виводить інформацію про всі предмети навчання, екзамени з яких здали студенти з ідентифікаторами 12 і 32.

Завдання 13. Написати запит, який виводить навчальні предмети, назви яких починаються на "Ф".

Завдання 14. Написати запит, який виводить відомості про студентів, в яких імена починаються на "П" або "В".

Завдання 15. Написати запит для вибірки з таблиці Exam_Marks записів, у яких відсутні значення оцінок (поле Mark).

Завдання 16. Написати запит, який виводить з таблиці Exam_Marks записи у яких в полі Mark є значення оцінок.

Завдання 17. Підрахувати кількість студентів, які здали екзамен з навчального предмета, номер якого рівний 20.

Завдання 18. Написати запит, який дозволяє підрахувати в таблиці Exam_Marks кількість різних навчальних предметів.

Завдання 19. Написати запит, який виконує вибірку для кожного студента значення його ідентифікатора і мінімальної з отриманих ним оцінок.

Завдання 20. Написати запит, який здійснює вибірку для кожного студента значення його ідентифікатора і максимальної з отриманих ним оцінок.

Завдання 21. Написати запит, який виводить прізвища першого в алфавітному порядку (за прізвищем) студента, прізвище якого починається на “М”.

Завдання 22. Написати запит, який виводить (для кожного навчального предмета) назву предмета і максимальне значення номера семестру, в якому цей предмет викладається.

Завдання 23. Написати запит, який виводить дані для кожного конкретного дня здачі екзамену про кількість студентів, які здавали екзамен у цей день.

Завдання 24. Написати запит для отримання середнього балу для кожного курсу по кожному предмету.

Завдання 25. Написати запит для отримання середнього балу для кожного студента.

Завдання 26. Написати запит для отримання середнього балу для кожного іспиту.

Завдання 27. Написати запит для визначення кількості студентів, які здали кожен екзамен.

Завдання 28. Написати запит для визначення кількості предметів, які вивчаються на кожному курсі.

Завдання 29. Припустимо, що стипендія всім студентам збільшена на 20 %. Написати запит до таблиці Student, який виводить номер студента, прізвище студента і величину збільшеної стипендії. Вихідні дані впорядкувати: а) за значенням останнього стовпця (розмір стипендії); б) в алфавітному порядку за прізвищами студентів.

Завдання 30. Написати запит, який за таблицею Exam_Marks дозволяє знайти а) максимальні і б) мінімальні оцінки кожного студента і який виводить їх разом з ідентифікатором студента.

Завдання 31. Написати запит, який виводить список навчальних предметів у порядку а) спадання семестрів і б) зростання відведених на предмет годин. Поле семестру у вихідних даних має бути першим, за ним повинні слідувати ім'я предмета та ідентифікатор предмета.

Завдання 32. Написати запит, який виводить суму балів усіх студентів для кожної дати здачі екзаменів і надає результати в порядку спадання цих сум.

Завдання 33. Написати запит, який виводить а) середній, б) мінімальний, в) максимальний бал усіх студентів для кожної дати

здачі екзаменів, і який представляє результати в порядку спадання цих значень.

Контрольні запитання

1. Пояснити порядок слідування фраз в операторі SELECT. Які фрази є обов'язковими, а які ні?
2. Пояснити порядок використання в стандарті мови SQL агрегатних функцій. Які обмеження для них задаються?
3. Що таке операція групування?

Практична робота № 10

Тема. Мова SQL. Маніпулювання даними.

Мета. Ознайомитись з операторами мови маніпулювання даними DML.

Мова DML є частиною SQL, призначена для підтримки інформації, що зберігається в реляційних таблицях баз даних. Кожен DML вираз діє на дані лише однієї таблиці.

DML включає три команди SQL:

INSERT – додає нові стрічки в таблицю бази даних.

UPDATE – оновлює існуючі стрічки таблиці бази даних.

DELETE – видаляє стрічки з таблиці бази даних.

DML може покликатися на представлення, яке включає дані із кількох таблиць (представлення, яке містить з'єднання таблиць). В цьому випадку DML-вираз буде відноситись до стовпців лише однієї таблиці в представленні.

Синтаксис запиту INSERT:

```
INSERT INTO <ім'я_таблиці чи представлення>  
[(список полів)]  
VALUES (<список_значень>);
```

Оператор DELETE видаляє одну чи більше стрічок з таблиці, які задовольняють певну умову.

Синтаксис запиту DELETE:

```
DELETE  
FROM ім'я_таблиці  
[WHERE умова];
```

Оператор UPDATE використовується для оновлення даних у вказаних полях таблиці. Всі записи таблиці, які задовольняють задану у фразі WHERE умову змінюються згідно з фразою SET.

Синтаксис запиту UPDATE:

```
UPDATE ім'я_таблиці  
SET ім'я_поля = вираз  
[ім'я_поля = вираз]  
[WHERE умова];
```

Завдання 1. Додати нову стрічку в таблицю Student.

Завдання 2. Додати нову стрічку в таблицю Student з умовою, що в поле Name ввести Null-значення.

Завдання 3. Вибрати значення із таблиці Student і розмістити його в таблиці Student1. Таблиця Student1 попередньо створена командою CREATE TABLE і має структуру, ідентичну таблиці Student.

Завдання 4. Записати запит для видалення всіх стрічок таблиці Exam_Marks1.

Завдання 5. Видалити з таблиці Exam_Marks1 стрічки, які задовольняють умову Student_ID = 50.

Завдання 6. Видалити з таблиці Student1 стрічки, які задовольняють умову City = “Київ”.

Завдання 7. Для всіх університетів, інформація про які знаходиться в таблиці University1, змінити рейтинг на значення 10.

Завдання 8. Для всіх університетів Києва, інформація про які знаходиться в таблиці University1, змінити рейтинг на значення 5.

Завдання 9. Назву навчального предмета **Математика** (Subj_Id = 15) змінити на назву **Вища математика**. Номер зберегти, а у відповідні поля стрічки таблиці ввести нові дані про цей предмет.

Завдання 10. Збільшити в таблиці University1 значення поля Rating в два рази.

Завдання 11. Збільшити в таблиці Student1 значення поля Stipend в два рази для студентів з Києва.

Завдання для самостійного опрацювання

Завдання 1. Написати команду, яка додає в таблицю Subject стрічку для нового навчального предмета з наступними значеннями полів: Semester = 3; Subj_Name = “Алгебра”; Hour = 32; Sub_Id = 35.

Завдання 2. Додати запис для нового студента, якого звати Олещук Микола, навчається на першому курсі ЛНУ, проживає у Львові, відомості про розмір стипендії невідомі.

Завдання 3. Написати команду, яка видаляє з таблиці Exam_Marks записи про всі оцінки студента, номер якого рівний 100.

Завдання 4. Написати команду, яка збільшує на 5 значення рейтингу всіх наявних у базі даних університетів, які розташовані в м. Київ.

Завдання 5. Змінити в таблиці значення міста, в якому проживає студент Назаренко, на “Тернопіль”.

Контрольні запитання

1. Синтаксис оператора DELETE.
2. Синтаксис оператора INSERT INTO.
3. Синтаксис оператора UPDATE.

Практична робота № 11

Тема. Визначення прав доступу користувачів до даних.

Мета. Ознайомитись з операторами мови DCL для виконання адміністративних функцій надання та відміни прав доступу до бази даних.

Завдання 1. Надати користувачам Іваненко та Матвіїв право виконувати Select-запити до таблиці Student.

Завдання 2. Надати користувачеві Іваненко право вводити в таблицю Exam_Marks нові стрічки.

Завдання 3. Надати користувачеві Іваненко весь набір привілеїв у таблиці Student.

Завдання 4. Дозволити будь-якому користувачеві отримувати інформацію з таблиці Exam_Marks.

Завдання 5. Скасувати привілеї INSERT і DELETE у таблиці Student для користувачів Антонюк і Левенко.

Завдання 6. Надати користувачеві Антонюк право доступу лише до стовпців Student_ID і Surname таблиці Student.

Завдання для самостійного опрацювання

Завдання 1. Надати користувачеві Панасюк право на зміну в базі даних значень оцінок для записів про студентів.

Завдання 2. Надати користувачеві Базюк право передавати іншим користувачам права на виконання запитів до таблиці Exam_Marks.

Завдання 3. Скасувати привілей INSERT по відношенню до таблиці Student у користувача Ліщук і у всіх інших користувачів, яким привілей був наданий цим користувачем Ліщук.

Завдання 4. Надати користувачеві Мусяк право виконувати операції вставки чи оновлення для таблиці University, але тільки для записів про університети, значення рейтингів яких в діапазоні від 50 до 100.

Завдання 5. Дозволити користувачеві Гринюк виконувати запити до таблиці Exam_Marks, але забороните йому змінювати в цій таблиці значення оцінок студентам, які мають оцінку незадовільно.

Контрольні запитання

1. Встановлення привілеїв. Команда GRANT.
2. Використання аргументів ALL, PUBLIC.
3. Відміна привілеїв. Команда REVOKE.
4. Привілеї системи CONNECT, RESOURCE, DBA.
5. Створення і видалення користувачів.

Література

1. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань. Книга 1. Організація баз даних та знань: Навчальний посібник. – Львів: “Магнолія 2006”, 2011. – 456 с.
2. Малахов Є.В. Основи проектування баз даних: Навчальний посібник для студентів вищих навчальних закладів. – О.: Наука і техніка, 2011. – 156 с.
3. Ситник Н.В., Краснюк М.Т. Проектування баз і сховищ: Навчально-методичний посібник для самостійного вивчення дисципліни. – К.: КНЕУ, 2005. – 264 с.
4. Черняк О.І., Ставицький А.В., Черноус Г.О. Системи обробки економічної інформації: Підручник. – К.: Знання, 2006. – 447 с. – (Вища освіта ХХІ століття).

Електронні ресурси

1. <https://erdplus.com/> Інструмент моделювання бази даних ERDplus.
2. <https://www.jetbrains.com/datagrip/> Програмне забезпечення JetBrains DataGrip.
3. <https://itolymp.com/access/2/> Відеокурс: Бази даних.