

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи ступеня

магістра

(назва освітньо-кваліфікаційного рівня)

студента *Жежель Антон Костянтинович*

(ПІБ)

академічної групи *121М-21-1*

(шифр)

спеціальності *121 Інженерія програмного забезпечення*

(код і назва спеціальності)

освітньої програми *«121 Інженерія програмного забезпечення»*

(назва освітньої програми)

на тему: *Розробка програмного забезпечення для*

*реалізації методів та алгоритмів трансформації*

*баз даних з ER-моделі в реляційну модель*

*А.К. Жежель*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинг овою	інституційною	
розділів кваліфікаційної роботи				
спеціальний	<i>Проф. Алексєєв М.О.</i>			

Рецензент				
-----------	--	--	--	--

Нормоконтролер	<i>проф. Лактіонов І.С.</i>			
----------------	-----------------------------	--	--	--

Дніпро  
2022

**Міністерство освіти і науки України**  
**Національний технічний університет**  
**«Дніпровська політехніка»**

---

**ЗАТВЕРДЖЕНО:**

Завідувач кафедри

Програмного забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(прізвище, ініціали)

(підпис)

«    »

20 22 року

**ЗАВДАННЯ**  
**на виконання кваліфікаційної роботи**

спеціальності 121 Інженерія програмного забезпечення  
 (код і назва спеціальності)

студенту 121м-21-1 Жежель Антон Костянтинович  
 (група) (прізвище та ініціали)

Тема кваліфікаційної роботи Розробка програмного забезпечення для  
реалізації методів та алгоритмів трансформації  
баз даних з ER-моделі в реляційну модель

**1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ**

Наказ ректора НТУ «Дніпровська політехніка» від 31.10.2022 р. № 1200 -с

**2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ**

**Об'єкт досліджень** – бази даних різного типу, включно з реляційними та ER-моделями баз даних.

**Предмет досліджень** – методи та алгоритми обробки даних у реляційних та ER-моделей баз даних.

**Мета НДР** – виявлення тенденцій формування та проблем розвитку цифрової економіки в Україні в умовах розвитку четвертої індустріальної революції.

**Вихідні дані для проведення роботи** – теоретико-математичні дослідження, основи бази даних різного типу при вирішенні задач трансформації баз даних з ER-моделі в реляційну модель

### 3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

**Новизна запропонованих рішень** полягає у:

- реалізації ефективного алгоритму трансформації баз даних з ER-моделі в реляційну модель;
- консолідації текстових даних з різних незалежних джерел

**Практична цінність** результатів полягає у розробці програми, яка реалізує методи та алгоритми трансформації баз даних з ER-моделі в реляційну модель, що в значній мірі забезпечує економію часу та ресурсів.

### 4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє безпосереднє використання методики формування класифікаційних ознак інформаційних процесів управління при їх класифікації з метою оцінки функціонального стану об'єкту. В результаті роботи повинен бути розроблений програмний комплекс для реалізації методів та алгоритмів трансформації баз даних з ER-моделі в реляційну модель.

### 5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Розробка концепції трансформації баз даних з ER-моделі в реляційну модель	01.10.2022-20.10.2022
Розробка алгоритмів трансформації баз даних з ER-моделі в реляційну модель	21.10.2022-30.10.2022
Розробка програмного забезпечення на основі обраної концепції та алгоритму з тестуванням	01.11.2022-16.12.2022

Завдання видав

\_\_\_\_\_

(підпис)

*Алексєєв М.О.*

\_\_\_\_\_

(прізвище, ініціали)

Завдання прийняв до виконання

\_\_\_\_\_

(підпис)

*Жежель А.К.*

\_\_\_\_\_

(прізвище, ініціали)

Дата видачі завдання: 01.10.2022 р.

Термін подання кваліфікаційної роботи до ЕК 01.12.2022

## РЕФЕРАТ

Пояснювальна записка: 74 стор., 17 рис., 5 додатка, 56 джерел, 1 таблиця.

Об'єкт дослідження: бази даних різного типу, включно з реляційними та ER-моделями баз даних.

Предмет дослідження: методи та алгоритми обробки даних у реляційних та ER-моделей баз даних.

Мета роботи: розробити програмне забезпечення для реалізації методів та алгоритмів трансформації баз даних з ER-моделі в реляційну модель та виявити тенденції формування та проблеми розвитку цифрової економіки в Україні в умовах розвитку четвертої індустріальної революції.

Методи дослідження. Виявлення тенденцій формування та проблем розвитку цифрової економіки в Україні в умовах розвитку четвертої індустріальної революції. У переліку завдань - аналіз глобальних процесів, що розгортаються, які ведуть до зміни технологічного укладу, і моделей економічного зростання; узагальнення теоретичних підходів до понять "інтелектуальна економіка" і "цифрова економіка"; аналіз стану та визначення перспектив розвитку цифрової економіки в Україні.

Новизна отриманих результатів полягає у збільшенні структури та функціональності ядра CASE-системи. Створено динамічну бібліотеку, яка містить класи для трансформації схем з ERM-моделі в реляційну, розроблений виконуваний прототип модуля в середовищі Data Atlas Modeler.

Практична цінність результатів полягає у розробці прототип модуля який може легко вбудовуватись в середовище Data Atlas Modeler для отримання реляційних схем, на базі яких можна створювати SQL-скрипти для побудови різних баз даних.

Область застосування. Розроблена програма для реалізації методів та алгоритмів трансформації баз даних з ER-моделі в реляційну модель яка може застосовуватися для вирішення моделі яка будується методом послідовних уточнень початкових діаграм.

Значення роботи та висновки. Зроблено проектування інформаційного забезпечення, як інфологічне, так і даталогічне. Спроектовано як логічну ER-модель, так і фізична.

Прогнози щодо розвитку досліджень. Подальший розвиток проекту може виділити завдання, пов'язані з доповненням інструменту механізмами для генерації реляційних схем та інтеграцією інструменту з графічними редакторами для ER-схем.

Список ключових слів: модель даних, трансляція схем, ER-модель, реляційна модель.

## ABSTRACT

Explanatory note: 74 pages, 17 figures, 5 applications, 56 sources, 1 table.

Research object: databases of various types, including relational and ER database models.

Research subject: methods and algorithms of data processing in relational and ER database models.

The purpose of the work: to develop software for the implementation of methods and algorithms for the transformation of databases from the ER model to the relational model and to identify the trends in the formation and problems of the development of the digital economy in Ukraine in the context of the development of the fourth industrial revolution.

Research methods. Identification of trends in the formation and development of the digital economy in Ukraine under the conditions of the development of the fourth industrial revolution. The list of tasks includes the analysis of the unfolding global processes that lead to changes in the technological structure and models of economic growth; generalization of theoretical approaches to the concepts of "intellectual economy" and "digital economy"; analysis of the state and determination of prospects for the development of the digital economy in Ukraine.

The novelty of the obtained results lies in the increase in the structure and functionality of the core of the CASE system. A dynamic library was created, which contains classes for transforming schemes from the ERM model into a relational one, an executable prototype of the module was developed in the Data Atlas Modeler environment.

The practical value of the results lies in the development of a prototype module that can be easily integrated into the Data Atlas Modeler environment to obtain relational schemas on the basis of which SQL scripts can be created to build various databases.

Field of application. A program has been developed for the implementation of methods and algorithms for the transformation of databases from the ER-model into a relational model, which can be used to solve a model that is built by the method of successive refinements of initial diagrams.

Value of work and conclusions. Designing of information support, both infological and datalogical, has been done. Both a logical ER model and a physical one are designed.

Forecasts regarding the development of research. Further development of the project may highlight tasks related to supplementing the tool with mechanisms for generating relational schemas and integrating the tool with graphic editors for ER schemas.

List of keywords: data model, schema translation, ER-model, relational model.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – бази даних;

ОУ –об’єкт управління;

ПЗ – програмне забезпечення;

РСУБД - Реляційна система керування базами даних;

СУБД - Система управління базами даних;

ООСУБД - Об’єктно-орієнтована система управління базами даних;

НФБК - Нормальна форма Бойса-Кодда

ЗНФ - Третя нормальна форма

ЕОМ - Комплекс пристроїв для обробки інформації

СП – спектральне перетворення;

ШП – швидке перетворення;

РС – personal computer.

## ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. МЕТОДИ ТА АЛГОРИТМИ ТРАНСФОРМАЦІЇ БАЗ ДАНИХ	3
ER-МОДЕЛІ В РЕЛЯЦІЙНУ МОДЕЛЬ	10
1.1. Введення та класифікації баз даних	10
1.2. Модель даних ER, основні поняття	19
1.3. Об'єктно-орієнтовані бази даних	23
1.4. Висновки до першого розділу	27
РОЗДІЛ 2. DATA ATLAS MODELER ВИКОРИСТАННЯ ER-МОДЕЛІ ДЛЯ СТРУКТУРИ ДАНИХ	29
2.1. Проектування інформаційного забезпечення. Створення ER-моделі	29
2.2. ER–метод проектування реляційних баз даних	37
2.3. Перетворення ER-моделі на реляційну модель	43
2.4. Висновки до другого розділу	50
РОЗДІЛ 3. ПРОЕКТУВАННЯ ER-ДІАГРАМИ	51
3.1. Інфологічна модель даних, приклад побудування ER-діаграми	51
3.2. Модель «сутність – зв'язок» ER-моделі	55
3.2.2 Приклад	57
3.3. Висновки до третього розділу	60
ВИСНОВКИ	62
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
Додаток А. КОД ПРОГРАМИ	69
Додаток Б. ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ	71
Додаток В. ПОНЯТТЯ ER-МОДЕЛІ. ПОНЯТТЯ СУТНОСТІ (ENTITY). АТРИБУТИ. ВИДИ АТРИБУТІВ	72
Додаток Г. ER-МОДЕЛЬ. ПОНЯТТЯ ЗВ'ЯЗКУ. ПОТУЖНІСТЬ ЗВ'ЯЗКУ. ТИПИ ЗВ'ЯЗКІВ	73
Додаток Д. ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ. ПОНЯТТЯ ПРЕДМЕТНОЇ ОБЛАСТІ	74

## ВСТУП

**Актуальність теми.** Отримання достовірної інформації в сучасну епоху є серйозною проблемою, з якою стикаються організації. Це завдання вимагає швидкого доступу або до єдиного загального джерела інформації, або до добре організованої системи збирання даних з різних джерел. В останньому випадку проблемою є те, що кожне з джерел інформації зазвичай дає змогу отримати вузькоспеціалізовану конкретну інформацію, що зберігається в ньому, і це, як наслідок, тягне за собою втрату уявлення про запитуваний об'єкт загалом або спотворення інформації шляхом, наприклад, неякісної синхронізації даних. Тому завдання консолідації даних, особливо текстових, з різних незалежних джерел є актуальним.

**Об'єкт досліджень:** бази даних різного типу, включно з реляційними та ER-моделями баз даних.

**Предмет досліджень:** методи та алгоритми обробки даних у реляційних та ER-моделей баз даних.

**Мета дослідження:** виявлення тенденцій формування та проблем розвитку цифрової економіки в Україні в умовах розвитку четвертої індустріальної революції. У переліку завдань - аналіз глобальних процесів, що розгортаються, які ведуть до зміни технологічного укладу, і моделей економічного зростання; узагальнення теоретичних підходів до понять "інтелектуальна економіка" і "цифрова економіка"; аналіз стану та визначення перспектив розвитку цифрової економіки в Україні.

**Методи дослідження.** Метод теоретичного дослідження і Експеримент з використанням комп'ютера. Методологія основи дослідження є основними фундаментальним положенням для розробки програмного забезпечення, наукового дослідження вітчизняних та закордонних компаній та науковців у галузі комп'ютеризованих систем.



**Новизна запропонованих рішень.** Запропоновано визначення ситуації як усвідомленого знання ER-модель яка являє собою формальну конструкцію, сама по собі не наказує ніяких графічних засобів її візуалізації. Як стандартна графічної нотації, за допомогою якої можна візуалізувати ER-модель, була запропонована діаграма сутність-зв'язок (ER-діаграма) є основою для побудови семантичної моделі взаємозв'язків між сутностями зв'язку. Для формалізації знань у сфері ситуаційного управління запропоновані класифікація ситуацій на основі їх типів і модальностей, онтологічна модель представлення такої класифікації.

**Практичне значення.** Розроблено інформаційну технологію, метод проектування інформаційно-аналітичних систем, що базуються на основі об'єктно-реляційної моделі даних. Результати дипломної роботи використано та впроваджено при розробці інформаційної системи, при розробці та реалізації інформаційно-аналітичної системи документообігу. Також результати дипломної роботи було використано та впроваджено в навчальному процесі кафедри штучного інтелекту національного університету радіоелектроніки в курсах лекцій «Організація баз даних і знань», «Інформаційні технології в корпоративних мережах», а також у роботі Центру інформаційних систем і технологій національного університету радіоелектроніки.

**Особистий внесок автора.** Особистий внесок автора. У роботах, опублікованих зі співавторами, здобувачі належать: у – визначення нового підкласу для постреляційних моделей даних, запропонований підхід формування формально-логічних моделей даних; у – визначення структурної компоненти для об'єктно-реляційної моделі даних на базі проведеного аналізу структурної специфікації реляційної моделі даних; у – аналіз структурної специфікації інфологічної ER-моделі «сутність зв'язок», розширення операційної специфікації та визначення особливостей відображення компонент моделі в об'єктно-реляційну модель даних; у – аналіз структурної специфікації реляційної бази даних; у – визначення формального підходу до проектування інформаційних систем різного призначення та їхнє використання для

розв'язання практичних задач управління складними організаційними системами.

**Структура та обсяг дипломної роботи.** Робота складається з вступу, трьох розділів і висновків. Містить 74 сторінок друкованого тексту, в тому числі 51 сторінок тексту основної частини, 17 рисунків, перелік використаних джерел, 5 додатків на 6 сторінках.

## РОЗДІЛ 1

### МЕТОДИ ТА АЛГОРИТМИ ТРАНСФОРМАЦІЇ БАЗ ДАНИХ З ER-МОДЕЛІ В РЕЛЯЦІЙНУ МОДЕЛЬ

#### 1.1. Введення та класифікації баз даних

База даних (БД) - сукупність даних, організована за певними правилами і призначена для тривалого зберігання зовнішньої пам'яті комп'ютера, постійного оновлення та використання.

БД є інформаційною моделлю певної предметної області. Бази даних забезпечують зберігання інформації, а також зручний та швидкий доступ до даних. Інформація в БД має бути несуперечливою, надмірною та цілісною.

Для створення БД та маніпулювання їх даними використовують системи управління базою даних (СУБД). Вони являють собою сукупність мовних та програмних засобів, призначених для створення, ведення та використання БД. Мовні засоби - мови опису та маніпулювання даними, мови запитів до БД. Програмні засоби - прикладні програми для обробки даних, що містяться у БД.

Таким чином, саме СУБД дозволяє створювати бази даних, оновлювати та доповнювати інформацію, забезпечувати гнучкий доступ до даних. СУБД створює на екрані комп'ютера певне середовище для роботи користувача (інтерфейс БД) і має певні режими роботи та систему команд.

Основні функції СУБД:

- створення структури БД;
- введення даних у БД та їх зберігання;

- зміна (редагування) структури БД та даних;
- пошук даних у БД;
- сортування та угруповання даних;
- захист БД;
- перевірка цілісності даних БД.

Приклади сучасних СУБД: Oracle, Firebird, Interbase, SQL Server, MySQL, MS Access.

Стрімке поширення нових технологій у всіх сферах людської діяльності призводить до швидких і глибоких змін структури сучасного промислового виробництва, глобальних ринків, а також економічної та соціальної сфери. У сучасних розвинених країнах розпочався активний пошук нових джерел зростання на основі розвитку науково-технологічного потенціалу, що формується на базі нових інформаційних, цифрових і промислових технологій. За розвитком цих галузей слідує нова технологічна революція й активне зростання продуктивності праці. Сукупно ці зміни оцінюються як "нова промислова революція". У зв'язку з цим наукове осмислення організаційних і методичних проблем формування технологічної основи для економічного зростання російської економіки в умовах глобальної трансформації світової системи поділу праці під впливом широкомасштабного впровадження інноваційних технологій четвертої промислової революції дасть змогу розробити парадигму та методичний інструментарій для подальшого впровадження й успішної реалізації в країні цифрової економіки, орієнтованої на підвищення ефективності промислового виробництва за рахунок використання нових технологій.

Залежно від характеру даних розрізняють фактографічні та документальні бази даних.

Фактографічні БД містять інформацію, що відноситься до певної предметної області та представлені у строго визначеному форматі. Наприклад, фактографічними є бази даних кадрового складу установи, транспортних систем, наукових даних, відомостей про населення, про природні ресурси тощо.

Документальні БД служать для зберігання та роботи з документами. Часто це документи природною мовою - монографії, публікації в періодиці, тексти законодавчих актів і т. д. Найчастіше документальні БД містять відео-і звукові документи.

Сучасні БД дедалі частіше поєднують у собі ознаки документальних та фактографічних баз.

За способом зберігання даних розрізняють централізовані та розподілені БД. У централізованих БД усі дані зберігаються одному комп'ютері (локальному чи мережевому). У розподілених БД частини бази зберігаються на багатьох комп'ютерах, об'єднаних між собою в мережу.

Залежно від виду організації даних розрізняють такі основні моделі подання даних у базі:

- ієрархічну;
- мережеву;
- реляційну.

Найбільш поширеною з цих моделей є реляційна. Сьогодні з'являються нові види організації БД, які, як правило, мають у своїй основі реляційний підхід.

В ієрархічній моделі дані подають у вигляді деревоподібної (ієрархічної) структури. У ньому існує строга підпорядкованість елементів, одне із яких головний, інші підлегли. Така організація даних зручна до роботи з інформацією, впорядкованої за рівнями, проте даних зі складними логічними зв'язками вона виявляється занадто громіздкою. Графічне представлення моделі дерево.

У мережевих моделях дані організовані у вигляді довільних графів. Тобто зв'язки між елементами довільні. Він більш гнучкий, не має шарнірних ключових елементів і має потенціал для встановлення бічних зв'язків. Його недолік в тому, що він дуже складний у виконанні.

Реляційна модель є сукупністю таблиць, які також називають відносинами (звідси і назва: англ. relation - Відношення). Ця модель була

запропонована в 70-х роках ХХ століття співробітником фірми ІВМ Едгар Коддом. Вона заснована на теорії множин та математичній логіці.

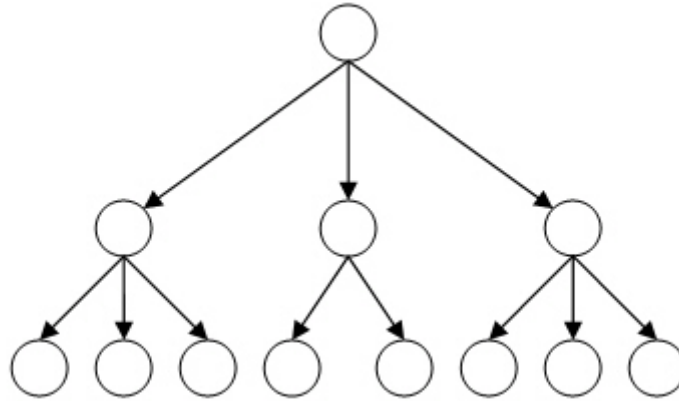


Рис. 1.1. Схема ієрархічної моделі

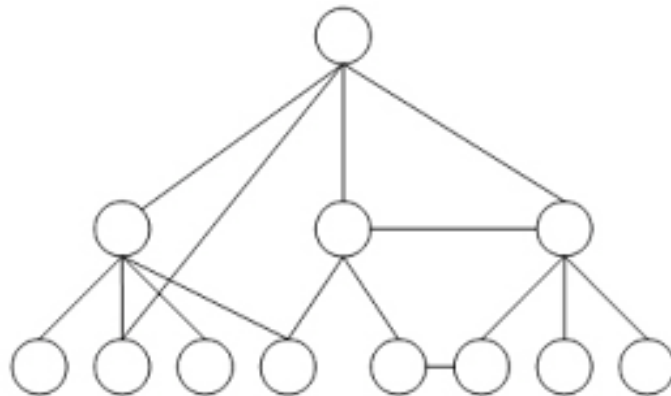


Рис. 1.2. Схема мережевої моделі

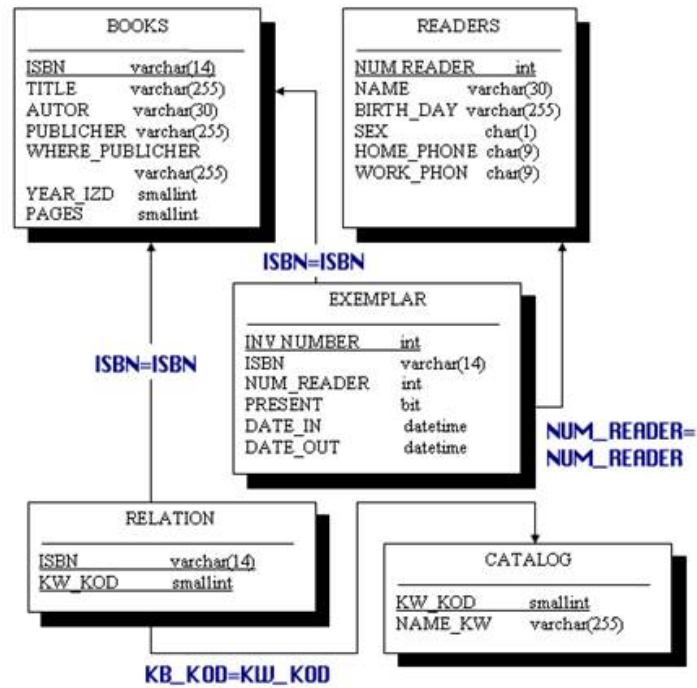


Рис. 1.3. Схема реляційної моделі

Доведено, що будь-яку структуру даних можна звести до табличної форми.

Перевагами реляційної моделі даних є простота, гнучкість структури, зручність комп'ютерної реалізації, наявність теоретичного опису. Більшість сучасних БД для персональних комп'ютерів є реляційними.

У реляційній БД ефективно реалізуються такі операції:

- сортування даних (наприклад, учнів за абеткою, сеансів за часом);
- вибірка даних за групами (наприклад, усі маршрути за напрямками, учні за класами);
- пошук записів (наприклад, пошук книг за їх автором, товарів за найменуванням) і т.д.

Як правило, реляційна БД складається із набору взаємозалежних таблиць, деякі з них є головними, інші - підлеглими. Організація зв'язку (відносин) між таблицями називається зв'язуванням чи з'єднанням таблиць.

Зв'язки між таблицями встановлюються під час створення БД. Поля, що використовуються зв'язування таблиць, називаються полями зв'язку. Поле зв'язку підлеглої таблиці називається зовнішнім ключем.

Існують такі види зв'язків між таблицями:

- відношення «один до одного»;
- відносини «один-до-багатьох» і «багато-до-одного»;
- відношення «багато-багатьом».

Ставлення «один-к-одному» означає, що одного запису в головній таблиці відповідає не більше одного запису в підпорядкованій таблиці. Полями зв'язку у разі є ключові поля обох таблиць.

Ставлення «один до одного» зазвичай використовують, щоб розбити таблицю з великим числом полів на кілька таблиць. У цьому випадку в першій таблиці залишаються поля з найбільш важливою інформацією, що часто використовується, а інші поля переносяться в іншу таблицю. Наприклад, у базі даних бібліотеки у першій таблиці — поля з назвою книги, прізвищем її автора, роком видання, а у другій — поля з датою придбання книги, місцем її зберігання тощо.



Рис. 1.4. Зв'язки типу «Один-до-Одного»

Відношення «один-ко-многим» означає, що одного запису головної таблиці може відповідати кілька записів підпорядкованої таблиці, а кожного запису підлеглої - лише один запис головної таблиці. Це найпоширеніший вид відносин.

Наприклад, у головній таблиці зберігаються дані про книги, а підлеглий - дані про видачу книг читачам. Одну книгу можуть брати в бібліотеці для читання багаторазово, тому одного запису в головній таблиці може відповідати кілька записів підпорядкованої таблиці.

Ставлення «багато-до-одного» відрізняється від відношення «один-до-багатьох» тільки напрямом. Якщо на відношення «один-до-багатьох» подивитися з боку підлеглої таблиці, а не головної, воно перетворюється на ставлення «багато-до-одного».

Відношення «багато-багатьох» означає, що кожен запис однієї таблиці може відповідати кілька записів іншої таблиці, і в той же час кожного запису другої таблиці - кілька записів першої.

Подібне відношення реалізується, наприклад, у базі даних бібліотеки між таблицями книг та читачів. Будь-яку книгу можуть взяти для читання (по черзі) багато читачів і будь-який читач може взяти кілька книг.

Для відношення «багато-багатьох» складно організувати зв'язки між таблицями та взаємодію між їх записами. Багато СУБД, у тому числі MS Access, не підтримують організацію такого відношення. У разі його реалізують через ставлення «один-ко-многим». Наприклад, ставленням «один-багатьох» пов'язані таблиця книг і таблиця видачі книг читачам, і таким самим ставленням пов'язані таблиця читачів та таблиця видачі книг читачам.

До способів пошуку даних у базі даних можна віднести:

- 1) Використання команди Пошук,
- 2) Фільтрування таблиці,
- 3) Сортування записів,
- 4) Використання запитів та їх створення.



Для пошуку даних у таблиці існують різні методи. Наприклад, можна скористатися автоматичним пошуком за вмістом таблиці та переходити від одного знайденого значення до іншого. Інший спосіб - упорядкувати (відсортувати) записи таблиці певним чином, що полегшить пошук потрібної інформації. Нарешті, можна використовувати механізм відбору даних – фільтрацію та запити.

У MS Access для пошуку значення достатньо ввести фрагмент значення, що шукається в полі Пошук рядка стану таблиці. Вже в процесі введення курсор переміститься у відповідний запис. Щоб перейти до наступного результату, потрібно натиснути клавішу Enter.

Сортування даних у таблиці – це впорядкування її записів. Для сортування треба вказати поля сортування та порядок (за зростанням або зменшенням значень). Записи у таблиці сортуються повністю.

Наприклад, при сортуванні таблиці абітурієнтів за датою надходження запису з ранньою датою будуть повністю переміщені вгору стосовно записів з пізньою датою.

У MS Access для сортування служать команди «Зростання» та «Спадання» у заголовку кожного поля таблиці та на вкладці стрічки Головна.

Сортування полів типу "дата" враховує календарну послідовність. Порядок сортування полів символічного типу (рядків) визначається таблицею кодування символів. Рядки порівнюються посимвольно зліва направо до першого символу, що не збігається. Символ, що має менший код, визначить найменше значення всього поля. Наприклад, значення рядка "березень" більше значення "травень", тому що код символу "р" більше коду символу "й". Коротке значення вважається меншим за довге, якщо коротке повністю збігається з початком довгого. Так, значення "комп'ютер" менше значення "комп'ютерний".

Якщо рядки мають однакову довжину і містять одну й ту саму послідовність символів, вони рівні.

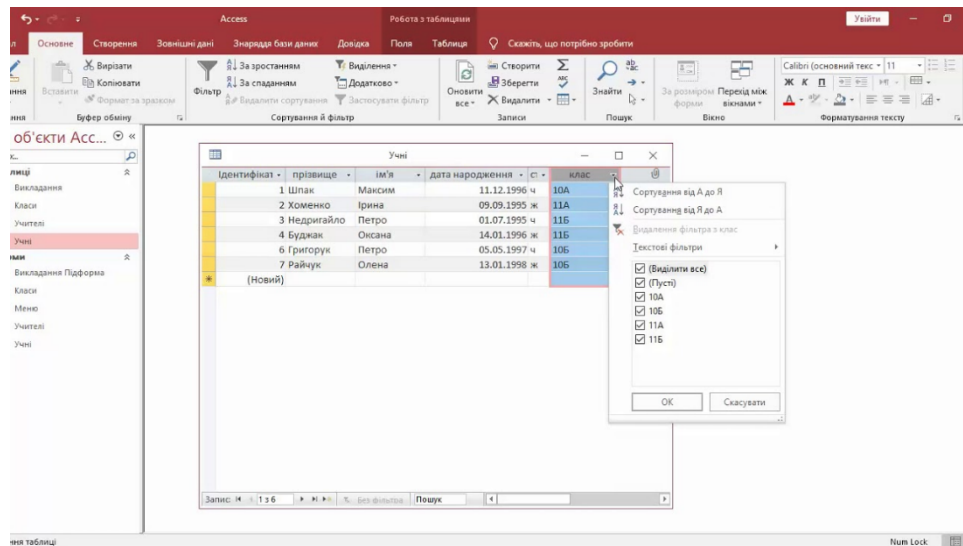


Рис. 1.5. Сортування записів таблиці баз даних

Найчастіше потрібно відібрати частину записів таблиці, які задовольняють певну умову. Цю умову називають критерієм відбору.

Відбір записів у самій таблиці за допомогою умови відбору називається фільтрацією таблиці. Фільтрування дозволяє вибрати записи з однієї таблиці для простих умов. Якщо потрібно вибрати дані відразу з кількох таблиць або умова є складною, створюють спеціальні об'єкти бази даних - запити. Вони є засоби отримання даних з БД відповідно до вимог користувача.

І в тому і в іншому випадку умова відбору має бути записана у вигляді логічного виразу. Логічне вираз - вираз, що містить логічні та арифметичні операції, а також операції порівняння. Результатом логічного висловлювання є логічна величина («істина» чи «брехня»). До логічних виразів застосовуються правила обчислення результатів логічних операцій.

Найпростіша форма логічного висловлювання за умови відбору — зазначення імені будь-якого поля логічного типу. Оскільки такі поля мають значення "істина" або "брехня", то результат виразу збігатиметься зі значенням поля.

Прості логічні вирази можуть містити операції порівняння (їх ще називають відносинами): більше (>), менше (<), більше або одно (>=), менше

або одно ( $\leq$ ), дорівнює ( $=$ ), не дорівнює ( $\neq$  або  $\diamond$ ). Порівняння даних здійснюється за тими самими правилами, що і при сортуванні.

Операції порівняння накладають вимоги до значення поля. Наприклад, вираз Кількість  $\leq 100$  вимагає знайти всі записи таблиці, у яких значення поля Кількість не перевищує 100.

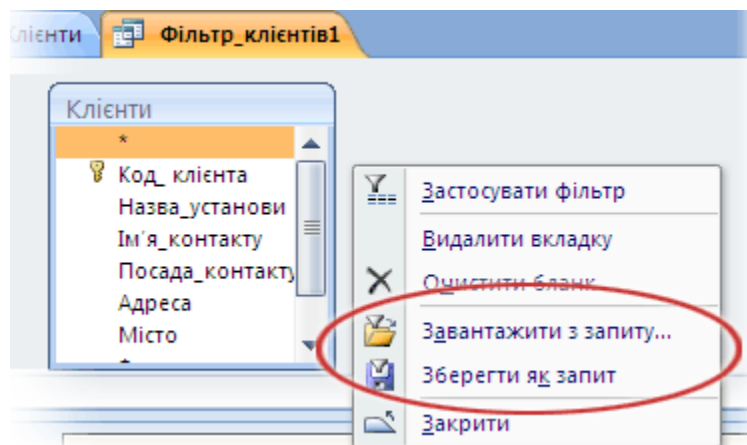
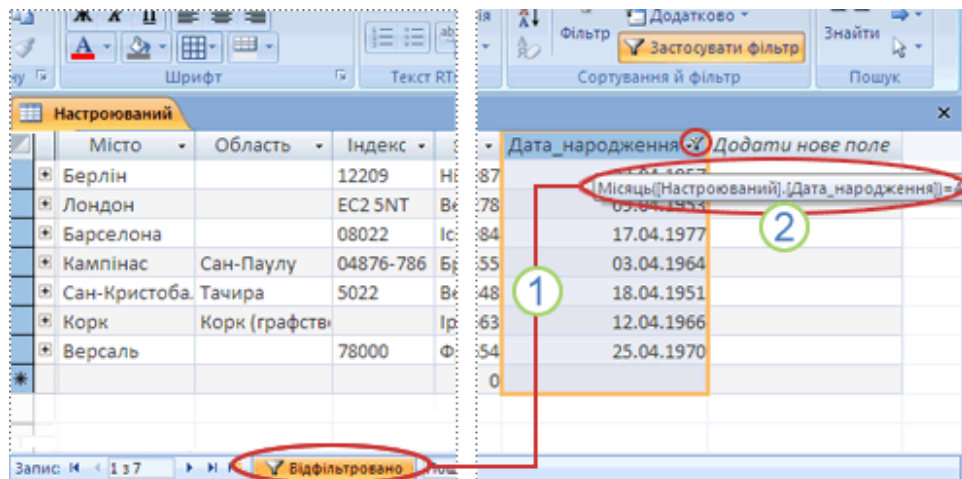


Рис. 1.6. Фільтрування таблиці БД

## 1.2. Модель даних ER, основні поняття

ER-модель використовується під час високорівневого (концептуального) проектування баз даних. З її допомогою можна виокремити ключові сутності та позначити зв'язки, які можуть встановлюватися між цими сутностями.

Під час проектування баз даних відбувається перетворення ER-моделі на конкретну схему бази даних на основі обраної моделі даних (реляційної, об'єктної, мережевої або ін.).

ER-модель являє собою формальну конструкцію, яка сама по собі не передбачає жодних графічних засобів її візуалізації. Як стандартну графічну нотацію, за допомогою якої можна візуалізувати ER-модель, було запропоновано діаграму сутність-зв'язок (ER-діаграма)

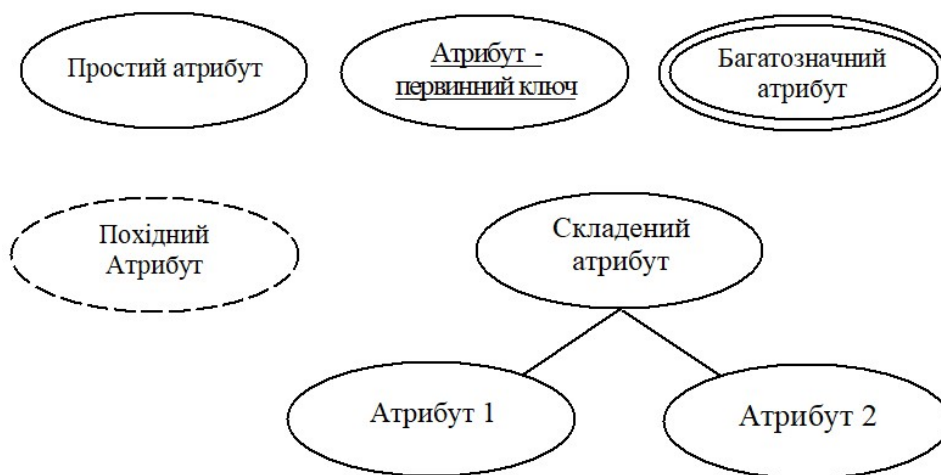


Рис. 1.7. Бази даних. Поняття ER-моделі та її сутності

Зв'язок представляється у вигляді неспрямованої лінії, що з'єднує дві сутності або веде від сутності до неї самої. При цьому в місці стикування зв'язку з сутністю використовуються:

- Трьох-крапковий вхід у прямокутник сутності, якщо для цієї сутності у зв'язку можуть (або повинні) використовуватися багато екземплярів сутності;
- Одно-крапковий вхід, якщо у зв'язку може (або повинен) брати участь лише один екземпляр сутності.

Обов'язковий кінець зв'язку є суцільною лінією, а необов'язковий – переривчастою лінією.

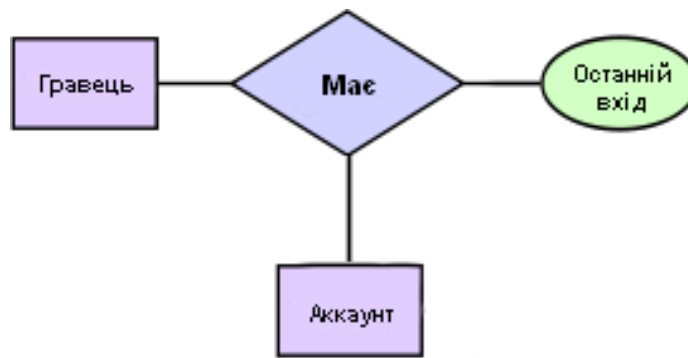


Рис. 1.8. Приклад типу зв'язку

Кожен атрибут стає стовпцем таблиці з тим самим ім'ям; може вибиратися точніший формат представлення даних. Стовпці, які відповідають необов'язковим атрибутам, можуть містити невизначені значення; стовпці, що відповідають обов'язковим атрибутам, не можуть.

Необов'язкові зв'язки відповідають стовпцям зовнішнього ключа, що припускає наявність невизначених значень; обов'язкові зв'язки – стовпцям, які допускають невизначених значень. Якщо між двома типами сутності A і B є зв'язок "один до одного", то відповідний зовнішній ключ за бажанням проектувальника може бути оголошений як у таблиці A, так і таблиці B. Щоб відобразити у визначенні таблиці обмеження, яке полягає в тому, що ступінь кінця зв'язку має дорівнювати одиниці, відповідний (можливо, складовий) стовпець повинен бути додатково специфікований як можливий ключ таблиці (у разі використання мови SQL для цього є специфікація UNIQUE).

За для первинного ключа створюються індекси (індекс унікальний) атрибутів та зовнішніх ключів, запити базувати переважно передбачається.

Концептуальні схеми ER-діаграми, можливі два способи, підтипи присутні, реляційних схем подання їх у двох способах:

- 1) В одній таблиці зібрати всі підтипи;
- 2) Утворити окрему таблицю для кожного підтипу.

Застосовуючи метод (1) таблиця створюється для максимального супер типу, уявлення можуть створюватися для підтипів. Містяться стовпці таблиць, котрі кожному атрибуту відповідають підтипи. Один стовпець додається у таблицю котрий видає код типу; ключ первинної частини. Тип сутності визначається для кожного рядка таблиці. Зв'язкам та атрибутам які відповідають стовпці цього рядка, значення повинні містити невизначені.

При використанні методу (2) кожного підтипу першого рівня (для глибших рівнів застосовується метод (1)) супертип відтворюється з допомогою представлення UNION (з усіх таблиць підтипів вибираються загальні стовпці – стовпці супер типу).

Кожен спосіб має свої переваги і недоліки. Перший спосіб переваг (одна таблиця для супер типу і всіх його підтипів) можна віднести наступне:

- відповідність логіці супер типів та підтипів; оскільки будь-який екземпляр будь-якого підтипу є екземпляром супер типу, логічно зберігати разом усі рядки, що відповідають екземплярам супер типу;
- забезпечення простого доступу до екземплярів супер типу та не надто складний доступ до екземплярів підтипів;
- можливість обійтися невеликим числом таблиць.

Недоліки методу (1):

- прикладна програма, що має справу з однією таблицею супер типу, повинна включати додаткову логіку роботи з різними наборами стовпців (залежно від значення стовпця ТИП) та різними обмеженнями цілісності (залежно від особливостей зв'язків, визначених для підтипу);
- загальна для всіх підтипів таблиця потенційно може стати вузьким місцем при розрахованому на багато користувачів доступі через можливість блокування таблиці цілком;
- для індивідуальних стовпців підтипів має допускатися можливість утримувати невизначені значення; таким чином, потенційно в загальній таблиці буде багато невизначених значень, що при використанні деяких РСУБД може вимагати значного обсягу зовнішньої пам'яті.

Переваги методу (2) полягають у наступному:

- діють зрозуміліші правила роботи з підтипами (кожному підтипу відповідає однойменна таблиця);
- спрощується логіка додатків; кожна програма працює лише з потрібною таблицею.

Недоліки методу (2):

- у випадку потрібно занадто багато окремих таблиць;
- робота з екземплярами супер типу на основі уявлення, що поєднує таблиці супер типів, може виявитися недостатньо ефективною;
- оскільки безліч екземплярів супер типу є об'єднанням множин екземплярів підтипів, не всі РСУБД можуть забезпечити виконання операцій модифікації екземплярів супер типу.

### **1.3. Об'єктно-орієнтовані бази даних**

Об'єктно-орієнтована база даних (ООСУБД, але іноді просто звана "об'єкт бази даних") - це СУБД, яка зберігає дані в логічній моделі, що тісно пов'язана із застосуванням програми об'єктної моделі, що підтримує створення та моделювання даних як об'єкти. ООСУБД також включає підтримку класів і об'єктів, успадкування властивостей класів, і включає в себе методи, підкласи та їхні об'єкти. Звісно, ООСУБД матиме фізичні дані моделі, які оптимізовані для вигляду логічної моделі даних, які вона очікує.

Об'єктно-орієнтована система керування базами даних (ООСУБД) допомагає програмістам створювати об'єкти, створені на мові програмування, які поведуться як об'єкт бази даних. Об'єктно-орієнтована система управління базами даних (ООСУБД) - це система управління базами даних, і як і більшість із баз даних об'єктів, також може запропонувати якусь мову запитів і дозволів об'єктів, які будуть знайдені через декларативний підхід до програмування.

Об'єктно-орієнтоване програмування базується на серії робочих об'єктів. Кожен об'єкт являє собою самостійно функціонуючі додатки та програми, і їм призначено певні завдання або ролі для виконання. Об'єктно-орієнтована система управління базами даних - це реляційна база даних, призначена для управління всіма цими незалежними програмами, з використанням даних, отриманих для того, щоб оперативно реагувати на запити про надання інформації від більших застосунків.

Щоб проілюструвати ООСУБД, ми можемо використати просту торговельну кухню з трьома співробітниками: шеф-кухар, кухар, що варить, та третій, що готує. Шеф-кухар несе відповідальність за стейк, а другий кухар відповідає за фрі і салат. Обидва зайняті роботою над їхніми функціями, навіть без жодних наказів. Шеф-кухар - це об'єктно-орієнтована система управління базами даних, кухар і другий кухар обидва об'єкти. Замовник розміщує замовлення на стейк, картоплю і салат, яке офіціант вручає шеф-кухарю. Шеф-кухар виголошує наказ. Кухар швидко забезпечує приготований стейк на тарілку, водночас другий кухар додає картоплю та салат і каже шеф-кухарю, що замовлення зроблено. Вони обидва здатні надати саме те, що потрібно негайно, тому що вони виконують свої індивідуальні завдання заздалегідь.

Кожен елемент можна було зробити з одним додатком або шеф-кухарем, але це зайняло б більше часу, а поділ ресурсів у кількох напрямках скорочує час відгуку. Об'єкти або кухарі можуть використовуватися як окремі програми, але час відгуку швидший, а інформація в єдиному пакеті за злагодженої роботи з шеф-кухарем або ООСУБД.

Збільшення використання об'єктно-орієнтованих мов програмування, як-от Python, java, C#, Visual Basic, .Net, C++, Objective-C, і у всіх у них зросла популярність в ООСУБД.

Об'єктно-орієнтовану систему управління базами даних, як правило, найкраще використовувати в бізнес-додатках, де потрібна висока продуктивність у складних умовах. Галузі, які мають високий попит на цей вид



програмування, як правило, в машинобудуванні, телекомунікаціях, спеціалізовані фінансові послуги та в наукових дослідженнях.

Об'єкт управління даними групи - це група об'єктів бази даних і відображення торговців, науковців та інших людей, які зібралися разом, щоб створити набір стандартної специфікації для об'єктно-орієнтованих програм. Такий стандарт може потенційно поліпшити переносимість додатків, написаних для об'єктно-орієнтованої системи управління базами даних, і тим самим знизити витрати на створення щоразу нового коду. Група розпалася 2001 року, але різні групи та ініціатори, як і раніше, роблять спроби визначити стандарт для забезпечення багатofункціональних додатків.

До всіх ранніх реалізацій ООСУБД застосовані зауваження. Системи були майже не застосовні для практичного використання, оскільки вони дуже повільно працювали, підтримували тільки однокористувацький режим роботи і були вкрай ненадійні. У них не підтримувалися розподілені дані, безпека і можливість відновлення після збоїв. Майже у всіх системах були відсутні розвинені механізми формулювання запитів. Під час побудови користувацьких інтерфейсів не використовували навіть уже наявні результати, отримані групами з галузі людино-машинних взаємодій.

Як зазначалося вище, ООСУБД GemStone була однією з перших комерційно доступних ООСУБД. Система була розроблена компанією Servio-Logic спільно з OGI. У вихідному варіанті системи розробники GemStone спиралися на мову Smalltalk. Хоча в перших випусках системи її основна мова називалася Oral, одразу було видно, що насправді це всього лише Smalltalk з підтримкою стабільного зберігання об'єктів, і незабаром назву мови було замінено на GemStone Smalltalk. Згодом у GemStone було забезпечено підтримку мов C і C++, але в усі часи базовою мовою залишався Smalltalk, а всі інші інтерфейси будувалися поверх базової. І серверна, і клієнтська частини системи можуть працювати під керуванням усіх основних гілок ОС UNIX і всіх розвинених варіантів Windows. Наразі продукт підтримується, розвивається і поширюється компанією GemStone Systems Inc.

Система ґрунтується на триланковій архітектурі клієнт-сервер, у якій прикладне опрацювання даних відбувається на середньому рівні між процесом взаємодії з користувачем і процесом, що підтримує сховище даних. Важливість цього підходу полягає в тому, що, якщо в застосунку використовується багато даних, то код застосунку доцільно розташувати на боці сховища даних, а якщо в застосунку відбувається багато змін над невеликим об'ємом даних, то є сенс розмістити код застосунку на боці користувача. Тим самим, архітектура дає змогу зменшити обсяг мережевого трафіку без перевантаження сервера, що підвищує швидкість обробки даних.

Для управління мульти доступом використовується механізм транзакцій. Механізм ґрунтується на так званому оптимістичному підході, за якого кожна сесія працює з власною локальною копією сховища об'єктів, і злиття зроблених у сесіях змін сховища відбувається під час завершення транзакції. Якщо під час завершення транзакції виявляється, що здійснені в ній зміни конфліктують зі змінами інших раніше зафіксованих транзакцій, то фіксація транзакції не здійснюється, транзакція не завершується, і розв'язання проблеми покладається на користувача. Автоматичне блокування об'єктів не здійснюється, але користувач може явно запросити блокування, що підвищує шанси на успішну фіксацію транзакції.

Для забезпечення безпеки даних підтримується механізм авторизації доступу на рівні власника об'єкта і його групи користувачів, так що може бути обмежений доступ до деяких об'єктів або деяких методів об'єктів. До кожного об'єкта приписують авторизаційний об'єкт, що містить дані про те, які користувачі і в якому режимі (читання або зміни) мають доступ до об'єкта.

У GemStone підтримується динамічне збирання сміття (garbage collection). Процес-"сміттяр" автоматично звільняє пам'ять, займану об'єктами, на які відсутні посилання.

У середовищі GemStone можна використовувати різні реалізації Smaltalk, а також мови C і C++. Класи та об'єкти можна створювати з використанням будь-якої з цих мов, і об'єкти, створені однією мовою, можна використовувати в

додатках, написаних будь-якою іншою мовою. Реалізація мови C являє собою набір функцій і набір компонентів, що перетворюють об'єкти GemStone на покажчики та літерали C і навпаки. Реалізація C ++ включає препроцесор у чистий C і бібліотеку класів.

GemStone можна також використовувати для управління даними, що відповідають стандартам OLE і CORBA. Для роботи з даними в реляційному стилі підтримуються стандарти SQL і ODBC .

Об'єктно-орієнтовані системи управління базами даних забезпечують спосіб організації даних, відмінних від цифр і тексту. Дизайнери використовують їх, щоб розмістити елементи мультимедіа, такі як фотографії, музику та відео. Ця система управління базами даних використовує для кожного елемента два ідентифікатори. Перший - це описове ім'я об'єкта, а другий - мініатюрна програма з інструкцією або методом, які комп'ютер виконує в процесі зберігання і вилучення. Ці дві частини стають об'єктом, тож користувачі можуть організувати свою базу даних із текстом або цифрами.

#### **1.4. Висновки до першого розділу**

Проаналізовано існуючі підходи до трансляції баз даних одного формату в інший, а також підходи до оптимізації структури NOSQL БД та оптимізації запитів до них.

В результаті аналізу існуючих типів баз даних встановлено, що реляційні БД є найпоширенішими формами зберігання даних. За останні десятиліття все більше використовуються на практиці ER-моделі та ER-моделі. При переході від реляційних баз даних до ER-моделі актуальною є проблема автоматичного перетворення форматів даних та збору даних із баз даних різного типу. Більшість досліджень у цій галузі присвячено вирішенню завдань консолідації реляційних баз даних та не торкаються проблеми трансляції реляційної бази даних у форматі ER-моделі. Тому проблеми консолідації реляційних баз даних з

ER-моделі та трансляції даних з БД одного формату в інший, а також запити до них є актуальною.

Встановлено, що для оптимізації структур баз даних є різні підходи. Зокрема, для проектування схеми та оптимізації структури реляційної бази даних застосовуються методи нормалізації та денормалізації.

Існують підходи до оптимізації розподілених баз даних. Але відсутні методи оптимізації структури ER-моделі баз даних, таких як «ключ-документ» або «родина стовпців». Проаналізовано методи оптимізації запитів до баз даних. Встановлено, що як механізми прискорення виконання запитів використовуються індексування та кешування, а також паралельне виконання запитів. Для оптимізації структури запитів використовуються методи, що базуються на аналізі плану виконання запиту. Формалізованих методів оптимізації запитів з урахуванням метаданих про їх структуру та схеми

БД немає. Вирішення цієї проблеми є актуальним.

Аналіз існуючих підходів до формалізованого подання запитів показав, що найбільш ефективними є методи теорії графів та теорії множин.

## РОЗДІЛ 2

### DATA ATLAS MODELER ВИКОРИСТАННЯ ER-МОДЕЛІ ДЛЯ СТРУКТУРИ ДАНИХ

#### 2.1. Проектування інформаційного забезпечення. Створення ER-моделі

IBM VisualAge DataAtlas складається з трьох основних модулів:

1) DataAtlas Modeler призначений для побудови концептуальної, так і логічної моделі даних.

Modeler використовує діаграми "сутність-зв'язок" (ER) для представлення моделі даних, що розробляється. З їхньою допомогою здійснюється деталізація сховищ даних моделюється системи, документуються сутності системи та способи їх взаємодії, включаючи ідентифікацію об'єктів предметної області (сутностей), властивостей цих об'єктів (атрибутів) та їх відносин з іншими об'єктами (зв'язків). DataAtlas Modeler забезпечує побудову як концептуальної, і логічної моделі даних;

2) DataAtlas Dictionary забезпечує роботу з визначеннями даних.

Поряд з реляційними системами керування базами даних (DB2 та Oracle) підтримуються ієрархічні (IMS) та програми на мовах розробки високого рівня. DataAtlas Dictionary використовує IBM VisualAge TeamConnection для зберігання, обробки та поділу об'єктів баз даних з модулями DataAtlas Designer та Modeler та іншими засобами розробки (наприклад, IBM VisualAge Generator).

3) DataAtlas Designer - інструмент для роботи з реальною базою даних та імпорту/експорту до неї абстрактної моделі даних. Він надає засоби для створення та підтримки оптимального фізичного визначення даних на основі побудованої ER-моделі.

DataAtlas Designer допомагає розробнику або адміністратору бази даних у складних процедурах оптимізації зберігання даних, що часто зустрічаються.

До них належать такі завдання:

- оцінка продуктивності для програми, що розробляється;
- схема фізичного розміщення даних;
- визначення та зв'язування додаткових ресурсів (таких як індекси, області таблиць, простору буферів);
- рішення про фізичне вирівнювання даних (таких як кластеризація індексів).

ІВМ пропонує широкий вибір засобу розробки програмного забезпечення, маючи у своєму розпорядженні програмні продукти для вирішення різних завдань, що виникають при розробці. Деякі з цих засобів пропонують рішення окремих, вузькоспеціальних завдань (наприклад, VisualAge Exchange), інші (як VisualAge PackBase) пропонують комплексне рішення, що охоплює всі етапи життєвого циклу програми. Відкриті технології та відповідність стандартам дозволяє використовувати продукти ІВМ спільно з програмними засобами інших виробників, а широка гамма пропонованих ІВМ засобів розробки програмного забезпечення дозволяє вибрати інструмент, що максимально відповідає поставленій задачі.

Проектування БД логічно є важливим кроком процесу проектування. Якщо відображення концептуального моделювання застосовується ER-діаграма, вона є і логічною моделлю. У нашому випадку використовується ER-модель, отже, концептуальна модель не потребує трансформації у логічну. Тому достатньо створити ER-діаграму.

Спочатку необхідно визначити сутність предметної області. Сутність являє собою реальний або уявний об'єкт, що має істотне значення для предметної області, інформація про який підлягає зберіганню. Для реалізації програмного засобу необхідно розробити сутності, які зберігали б у відповідності між собою всю необхідну інформацію про навчальні заклади.

Основними сховищами даних будуть сутності «vuz» та «suz». Вони зберігатимуться всі необхідні записи для отримання різноманітних інформації про заклади ВПО і СПО. Далі виникає необхідність створення сутностей fvuz і

fsuz, які будуть пов'язані відповідно з первинними сутностями. Сутності «fvuz» та «fsuz» необхідні для зберігання даних спеціальностей закладів ВПО та СПО відповідно. У результаті виходить чотири сутності, а саме:

- 1) vuz;
- 2) suz;
- 3) fvuz;
- 4) fsuz.

Далі розглянемо зв'язки сутностей. Зв'язок є поіменованою асоціацією між двома сутностями, значущою для розгляду даної предметної галузі. Зв'язку може даватися ім'я, виражене граматичним оборотом дієслова і розміщене біля лінії зв'язку. У нашій моделі є два зв'язки:

- 1) vuz – fvuz;
- 2) suz – fsuz.

В обох випадках визначимо тип зв'язку як одним-багатьом. Це означає, що екземпляр першої сутності пов'язаний із кількома екземплярами другої сутності. Це найчастіше використовуваний тип зв'язку. Модальність зв'язку – «повинен». Це означає, що екземпляр однієї сутності може бути пов'язаний щонайменше ніж із одним екземпляром іншої сутності, тобто. зв'язок обов'язковий.

Тепер розглянемо докладніше всі описані вище сутності. Для цього нам необхідно визначити, які атрибути в них будуть утримуватися. Атрибутом є будь-яка характеристика сутності, що розглядається в даній предметній галузі та призначена для кваліфікації, ідентифікації, класифікації, кількісної характеристики або вираження стану сутності. Атрибут є типом характеристик або властивостей, асоційованих з безліччю реальних або абстрактних об'єктів.

Розглянемо атрибути сутності «vuz»:

- 1) idvuz – зберігає унікальний порядковий номер закладу ВПО;
- 2) Ім'я – зберігає дані про назву закладу ВПО;
- 3) Адреса – зберігає адресу закладу;
- 4) Контакт - зберігає номери контактних телефонів та факсів;

5) Сайт – зберігає адреси офіційних сайтів навчального закладу;

6) e-mail- зберігає адреси електронної пошти закладу ВПО.

Розглянемо атрибути сутності "suz":

1) isvuz - зберігає унікальний порядковий номер закладу ППО;

2) Ім'я - зберігає дані про назву закладу СПО;

3) Адреса – зберігає адресу закладу;

4) Контакт - зберігає номери контактних телефонів та факсів;

5) Сайт – зберігає адреси офіційних сайтів навчального закладу;

6) e-mail- зберігає адреси електронної пошти закладу ППО.

Розглянемо атрибути сутності fvuz:

1) idfvuz – зберігає унікальний порядковий номер спеціальності ВПО;

2) Факультет – зберігає назву спеціальності закладу ВПО;

3) idvuz – зберігає порядковий номер закладу ВПО, якому належить

спеціальність.

Розглянемо атрибути сутності fsuz:

4) idfsuz – зберігає унікальний порядковий номер спеціальності СПО;

5) Факультет - зберігає назву спеціальності закладу СПО;

6) idsuz – зберігає порядковий номер закладу СПО, якому належить

спеціальність.



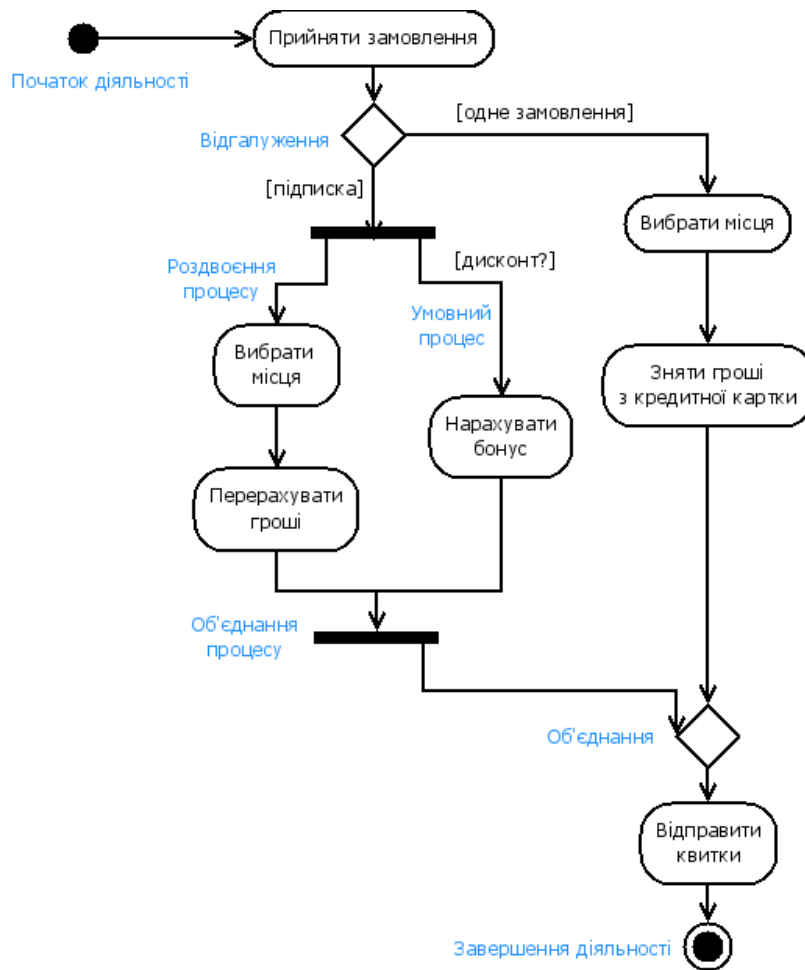


Рис. 2.1. Моделі й методи проектування інформаційних систем

Для побудови ER-моделі необхідно вказати значення для кожного атрибута. Атрибут може входити до складу первинного ключа (позначається символом «#») або описовим, тобто. звичайним дескриптором сутності. Значення атрибута може бути обов'язковим (позначається символом «») або необов'язковим (позначається символом «\*»). Подаємо список сутностей предметної області, список атрибутів сутностей. (Сутність, Атрибут, Значення, vuz, idvuz, #Ім'я, \_Адреса, \_Контакт, \_Сайт, \*e-mail, \*fvuz, idfvuz, #Факультет, \_idvuz, \_suz, idsuz, #Ім'я, \_Адреса, \_Контакт, \_Сайт, \*e-mail, \*fsuz, idfsuz, #Факультет, \_Idsuz)

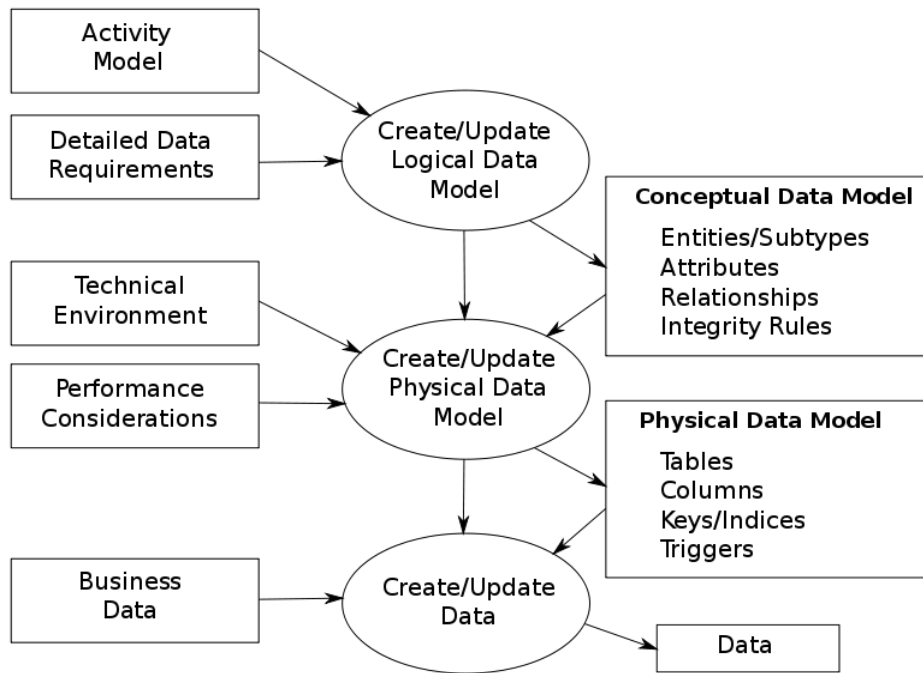


Рис. 2.2. Моделювання даних

Основною метою інфологічного проектування є отримання семантичних моделей, які повною мірою відображають предметну область та потреби користувачів. Головним інструментом для створення таких моделей є ER-модель (Entity-Relationship), що в перекладі означає "Сутність-зв'язок". В основі цієї моделі лежать графічні діаграми, що складаються з різних по суті компонентів.

Головними компонентами ER-моделей є сутності, їх атрибути та зв'язки між сутностями.

Сутність це якийсь об'єкт предметної галузі, інформація про який повинна зберігатися і бути доступною в різні часові проміжки. Розрізняють також такі поняття як атрибут та екземпляр сутності.

Атрибут сутності - це іменована характеристика, що є деякою властивістю сутності. Свого часу ж екземпляр сутності це конкретний представник сутності.

Примірники мають бути помітні, тобто сутності повинні мати деякі властивості є унікальними для кожного екземпляра цієї сутності.

Найменування атрибуту має бути виражене ім'ям іменником в однині. Зв'язок - це деяка асоціація між сутностями. Одна сутність може бути пов'язана з іншою сутністю або навіть сама із собою.

Сам процес моделювання інформаційної моделі складається з 7 етапів:

- визначення сутностей;
- визначення залежностей (зв'язків);
- завдання первинних та альтернативних ключів;
- визначення атрибутів сутностей;
- приведення моделі до необхідного рівня нормальної форми;
- формування фізичного опису моделі;
- генерація бази даних.

Атрибути сутності бувають своєю чергою двох типів – атрибути, включають первинний ключ і не ключові атрибути.

Потужність – це відношення кількості екземплярів у батьківській сутності до кількості цих у дочірній. У IDEFIX представлено всього 4 види потужності.

- . - Від нуля і до нескінченності;
- r – від одного до нескінченності;
- z – від нуля до нескінченності;
- n – рівно вказане число.

При цьому допускається наявність порожніх значень у батьківській сутності.

До ключових понять проектування баз даних належать:

1) CASE-технології - програмна основа CASE-засобів, що застосовується для розроблення та підтримки процесів життєвих циклів ПЗ, які використовуються в моделюванні даних. Найчастіше програмні коди в CASE-технологіях пишуться мовою SQL;

2) концептуальне проектування - побудова узагальненої, що не має конкретики, моделі бази даних;

3) фізичне проектування - побудова схеми бази даних під конкретну СУБД. Під час такого проектування враховуються обмеження на іменування об'єктів бази даних, обмеження на певні типи даних, фізичні умови зберігання даних у БД (поділ за файлами та пристроями), можливість доступу.

Під час проектування баз даних за допомогою CASE-засобів виокремлюють і аналізують певні бізнес-процеси, для яких створюють БД, визначають взаємозв'язки їхніх елементів, оптимізують їхню інфраструктуру. CASE-засоби дають змогу суттєво скоротити час на розробку БД і зменшити кількість помилок у них.

Для створення баз даних під найпоширеніші СУБД найчастіше використовують такі CASE-засоби:

1) ERwin (Logic Works) - CASE-інструмент для створення концептуальних і логічних. Він дає змогу редагувати різні набори даних, представляючи їх у вигляді електронних таблиць, розробляти структури баз даних, синхронізувати моделі, скрипти і БД, налаштовувати шаблони, виводити робочу інформацію у вигляді звітів, будувати зручні та зрозумілі діаграми, що відображають різні процеси в системі;

2) S-Designor (SDP) - графічний CASE-інструмент для проектування структури реляційних БД. Він створює моделі баз даних у два етапи - вибудовуючи концептуальну модель і потім перетворюючи її на фізичну, причому в даному процесі розробки можливий як прямий, так і зворотний перехід між моделями. Цей інструмент дає змогу проектувати бази даних під різні СУБД, зокрема під Oracle і MySQL;

3) DataBase Designer (ORACLE) - інтегроване CASE-середовище, що дає змогу аналізувати предметну галузь створення БД, виконувати програмування та проектування, проводити оцінювання й тестування, здійснювати супровід, забезпечувати якість, керувати конфігурацією та проектом, розробляти й аналізувати вимоги до інформаційної системи.

Наукометрія сьогодні - ефективний інструмент для аналізу та оцінки діяльності наукових організацій, а також для побудови стратегій і прогнозів розвитку досліджень. Для проведення наукометричних досліджень потрібні фахівці, які володіють знаннями та навичками роботи з наукометричними даними. З цією роллю якнайкраще впораються бібліотечні співробітники. Мета дослідження - визначити роль наукової бібліотеки в проведенні наукометричних досліджень в інтересах наукових організацій. Описано досвід бібліометричного аналізу та розвиток наукометрії в Центральній науковій бібліотеці Київського відділення Української академії наук. Проведено дослідження стану наукометрії в наукових організаціях методом анкетування їхніх представників. Виявлено, що найчастіше роботу в наукометричних базах даних ведуть вчені секретарі та заступники директорів з наукової роботи. У деяких відповідях спостерігається обмежене використання можливостей наукометричних баз даних. На основі отриманої інформації встановлено потребу у фахівці з питань наукометрії для кожної наукової організації. Окреслено переваги бібліотечних фахівців під час проведення якісного наукометричного аналізу, підкреслено значущість взаємодії наукової організації та бібліотеки.

## **2.2. ER–метод проектування реляційних баз даних**

Метод сутність-зв'язок називають також методом "ER-діаграм" (англ. "Entity-Relationship model"). Ця модель була вперше запропонована Пітером Ченом у 1976 році. Її призначення – концептуальне моделювання предметної галузі. Модель "сутність-зв'язок" (об'єктно-орієнтована модель предметної області) ґрунтується на якійсь важливій семантичній інформації про реальний світ і призначена для логічного представлення даних. Вона визначає значення даних у тих взаємозв'язку з іншими даними. Важливим є той факт, що з моделі "сутність-зв'язок" можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна), тому вона є найбільш загальною. ER-модель зазвичай представляється у графічній формі.

Основні елементи ER-моделей:

- об'єкти (сутності);
- атрибути об'єктів;
- зв'язки між об'єктами.

Основні переваги ER-моделей:

наочність;

проектування бази даних з великою кількістю об'єктів та атрибутів;

ER-моделі реалізовані у багатьох системах автоматизованого проектування баз даних.

Процес проектування бази даних є ітераційним - що допускає повернення до попередніх етапів для перегляду раніше прийнятих рішень і включає наступні етапи:

- 1) Виділення сутностей та зв'язків між ними;
- 2) Побудова діаграм ER-типу з урахуванням усіх сутностей та їх зв'язків;
- 3) Формування набору попередніх відносин із зазначенням передбачуваного первинного ключа для кожного відношення та використанням діаграм ER-типу;
- 4) Додавання не ключових атрибутів у відносини.

Поряд із методом "ER-діаграм" у проектуванні баз даних застосовується метод нормальних форм. Процедуру нормалізації можна охарактеризувати як послідовне приведення заданого набору змінних відношення до деякої більш бажаної форми. Можливість виконання зворотного перетворення є дуже важливою характеристикою, оскільки це означає, що у нормалізації зберігається інформація. Кожній нормальній формі відповідає певний набір обмежень, і відношення перебуває у певній нормальній формі, якщо задовольняє властивому їй набору обмежень. Нормалізація призначена для приведення структури бази даних до виду, що забезпечує мінімальну надмірність. Кінцевою метою нормалізації є зменшення потенційної суперечливості інформації, що зберігається в БД.

Теоретично реляційних баз даних зазвичай виділяється наступна послідовність нормальних форм:

- перша нормальна форма (1NF або 1НФ);
- друга нормальна форма (2NF або 2НФ);
- третя нормальна форма (3NF або 3НФ);
- нормальна форма (BCNF або НФБК);
- четверта нормальна форма (4NF або 4НФ);
- п'ята нормальна форма або нормальна форма проєкції-з'єднання (5NF або PJ/NF).

Функціонально повна залежність означає, що атрибут функціонально залежить від всього складового ключа, але при цьому не знаходиться у функціональній залежності від будь-якої з атрибутів (частин), що входять до нього. Відношення знаходиться у третій нормальній формі, якщо воно знаходиться у другій нормальній формі та між атрибутами немає транзитивних залежностей. Транзитивна залежність - це залежність між не ключовими атрибутами. Транзитивні залежності вилучаються також з допомогою декомпозиції відносини інші два чи більше відносин, які містять транзитивних відносин і об'єднання яких дасть початкове ставлення. Змінна стосунки перебуває у нормальній формі Бойса-Кодда і тоді, коли детермінанти всіх її функціональних залежностей є потенційними ключами. Відношення знаходиться в 4NF, якщо воно знаходиться в НФБК і не містить багатозначних нетривіальних залежностей. Тобто, всі багатозначні залежності є функціональними залежностями від ключів відношення.

Таким чином, нормалізація - це розбиття таблиці на дві або більше, які мають кращі властивості при включенні, зміні та видаленні даних. Остаточна мета нормалізації зводиться до одержання проєкту бази даних, у якому виключено надмірність інформації. Це робиться з метою економії пам'яті, а також для виключення можливої суперечливості даних, що зберігаються. При проєктуванні досить простих баз даних необхідний низький рівень надмірності дає відношення до 3НФ.

Одним з найбільш зрозумілих та практично використовуваних методів проектування реляційних баз даних є метод, в основу якого покладено модель «сутність-зв'язок».

Ключовими елементами моделі «сутність-зв'язок» є сутності, їх властивості (атрибути) та зв'язки між об'єктами.

Сутність визначається як деякий об'єкт предметної області, що розглядається, інформація про яку має бути відображена в базі даних.

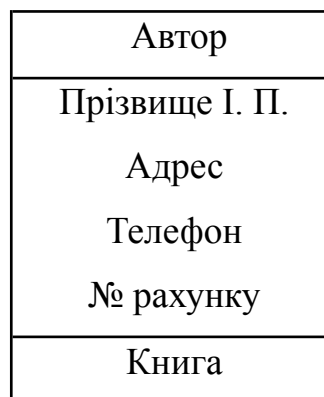
Взаємини сутностей виражаються зв'язками з-поміж них.

Атрибут – це властивість сутності.

Сутність - це, як правило, іменник; зв'язок найчастіше виражається дієсловом.

Наприклад, проектується база даних видавництва, призначена для зберігання інформації про книги та авторів, які їх написали. Тоді два головні об'єкти (дві сутності, інформація про які має бути відображена в базі даних) – це книга та автор. Ці сутності змістовно з'єднані з допомогою зв'язку пише. Атрибути сутності книга – назва, кількість сторінок, тираж, дата виходу сигнального примірника, ціна тощо. Атрибути сутності автор - це прізвище, адреса, телефон, №рахунки і т.д.

Сутності зображуються у вигляді прямокутника, атрибути вписуються всередину прямокутника, що зображує сутність:





Назва
Кількість сторінок
Тираж
Дата виходу

Рис. 2.3. Таблиця Автор/Книга

Дані визначення не є формальними, однак вони є прийнятними для використання при проектуванні бази даних. Особливість ER-методу полягає в тому, що різні проектувальники можуть розглядати ту саму предметну область з різних точок зору, отримуючи в результаті різні набори сутностей та зв'язків. Визначення найкращого з кількох можливих наборів є питанням особистої переваги.

Атрибут або набір атрибутів, що використовується для ідентифікації екземпляра сутності, називається ключем сутності. Таким чином, ключ сутності повинен бути унікальним для кожного екземпляра цієї сутності. Ключ кожної сутності має бути надлишковим, тобто видалення будь-якого атрибуту з цього набору порушуватиме його унікальність. Ключові атрибути будь-яким чином виділяються на діаграмі (наприклад, підкресленням або більше жирним шрифтом).

У нашому прикладі як ключовий атрибут сутності АВТОР було вирішено взяти прізвище, а як ключовий атрибут сутності КНИГА взяти її назву. Перше рішення свідомо не є безперечним: можлива поява авторів-однофамільців (тоді атрибут Прізвище І.П. втрачає унікальність і не може бути використаний як ключ). В принципі, припустима поява книг з однаковими назвами.

Якщо є хоча б мінімальна підозра, що атрибут, що вибирається як ключовий, може втратити свою унікальність, потрібно відмовитися від його використання як ключа та спробувати підібрати на цю роль інший атрибут.

Якщо виявиться, що жоден змістовий атрибут не може бути використаний як ключовий, то існує (щонайменше) два способи вирішення цієї проблеми:

- підібрати набір атрибутів, значення яких будуть унікальними для кожного екземпляра сутності;
- ввести ще один атрибут, який не відображатиме будь-якої властивості сутності, але буде придатний як ключовий. Зазвичай таким атрибутом стає номер екземпляра аналізованої сутності.

У результаті розвитку концепцій БД виокремлено три рівні представлення інформації: інфологічний, даталогічний і фізичний. На кожному рівні проводять структурування інформації таким чином, щоб на третьому рівні інформацію можна було подати у вигляді структур даних, які реалізуються в пам'яті ЕОМ. На інфологічному рівні визначають, яка інформація про предметну область буде зберігатися й оброблятися в комп'ютері, а в результаті дослідження предметної області будують її інфологічну модель, яку описують у термінах класів об'єктів та їхніх взаємодій. В інфологічній моделі інформація представляється незалежно від того, що являють собою дані і які технічні засоби будуть використані надалі для її зберігання та обробки. Даталогічна та фізична моделі безпосередньо реалізуються в системах управління БД (СУБД), а фізична модель у свою чергу визначає структуру збереження даних на фізичних носіях. Мета інфологічного проектування полягає в поданні семантики (сенсу) предметної області. Для опису предметної області найчастіше використовують модель "сутність-зв'язок", яку скорочено називають ER-моделлю від англійської назви "Entity - Relationship" ("Сутність - зв'язок"). ER-діаграма моделі має лексикографічну структуру і містить у собі текст та елементи графіки. На практиці інфологічна (семантична) модель використовується на першій стадії проектування БД. При цьому в термінах ER-моделі описується концептуальна (понятійна) схема БД, яка потім перетворюється до реляційної або іншої схеми. ER-моделі набули поширення в CASE-системах, що підтримують проектування реляційних БД. Слід зазначити,

що під час підготовки вчителів та студентів інженерних спеціальностей вищих навчальних закладів слід виокремлювати ключові поняття теорії БД і застосовувати їх під час проектування БД.

У даній роботі ми пропонуємо розгляд побудови інфологічної (семантичної) моделі даних і побудови ER-діаграми під час проектування бази даних "Кафедра".

Створимо інфологічну (семантичну) модель предметної області, що міститиме відомості про студентів, успішність студентів з предметів і викладачів, які викладають ці предмети в термінах ER-моделі з побудовою ER-діаграми.

ER-модель спирається на поняття сутність, атрибут, зв'язок, і предметна область має бути подана як сукупність сутностей з атрибутами, між якими встановлено зв'язки. Тип сутності означає множину однорідних об'єктів, а "екземпляр сутності" - конкретний об'єкт із множини об'єктів. Наприклад, сутністю є Студент, який являє собою всіх студентів вишу, а один із них, Петров Петро Петрович, є не сутністю, а конкретною реалізацією цієї сутності (екземпляром). Атрибутами сутності Студент є: ПІБ, номер групи, номер залікової книжки, дата зарахування та ін.

### **2.3. Перетворення ER-моделі на реляційну модель**

"Ставлення" (relation) - безліч атрибутів: унікальних імен з уточненням типу даних; плюс безліч "наборів величин" ("рядів"), відповідних атрибутам. Величини в наборах можуть бути представлені лише одиничними значеннями відповідних атрибутів типів, тобто бути скалярами ("1 нормальна форма").

"Ключ" (key) - група атрибутів, значення яких у всіх наборах відносно різні, але жодна підгрупа цих атрибутів такою властивістю вже не має (властивість "мінімальності" ключа). Зокрема група може складатися з єдиного

атрибути. Ключ щодо повинен бути завжди, і якщо їх кілька, одне їх має бути призначений "первинним" (primary).

"Зовнішній ключ" (foreign key) — група атрибутів, значення яких у кожному наборі величин відносини повинні збігатися зі значеннями ключа можливо іншого відношення. Зовнішні ключі не обов'язкові і проголошуються за потребами моделювання.

"Операції" (operation) - безліч спільних дій над відносинами, що дають в результаті знову-таки відносини ("замкненість операцій"). Використовуються для отримання нових відносин у потребах подальшого моделювання або під час вилучення з бази необхідних даних. Перелік операцій можна визначати по-різному; у перших пропозиціях моделі наводилося вісім операцій (проекції, з'єднання, відбору та ін.), вже не мінімальний набір, як компроміс між відсутністю надмірності та зручністю вживання.

"Реляційна база даних" (relational database) - набір відносин.

"Тип даних" іноді називають "доменом" (domain), але іноді під "доменом" розуміють лише "область визначення" величин. "Набір величин" (tuple) російською інакше називають "кортежем" або "n-кою".

При переході від ER-моделі до даталогічної реляційної моделі, як правило кожна сутність описується окремою таблицею;

— атрибути стають полями таблиць, їм задаються відповідні типи даних, які у СУБД;

— у таблицях визначаються первинні ключі, при необхідності вводяться сурогатні;

— для зв'язків "багато до багатьох" вводяться відповідні таблиці, забезпечені, можливо, необхідними атрибутами;

— при необхідності проводиться нормалізація таблиць до заданої нормальної форми (як правило - до 3НФ).

Після розробки макета БД його рекомендується проаналізувати з погляду зручності та ефективності виконання типових запитів. У цьому етапі іноді доводиться як змінювати даталогічну модель, і навіть денормалізувати БД,

перекладаючи ряд функцій до рівня докладання. Наведемо типовий приклад. Допустимо, в БД зберігається інформація про файли, що включає шлях, ім'я, опис (звичайне завдання для множини ІВ). Всі ми знаємо, що при роботі з подібними файловими колекціями в Інтернеті користувачеві зазвичай повідомляється розмір файлу в байтах.

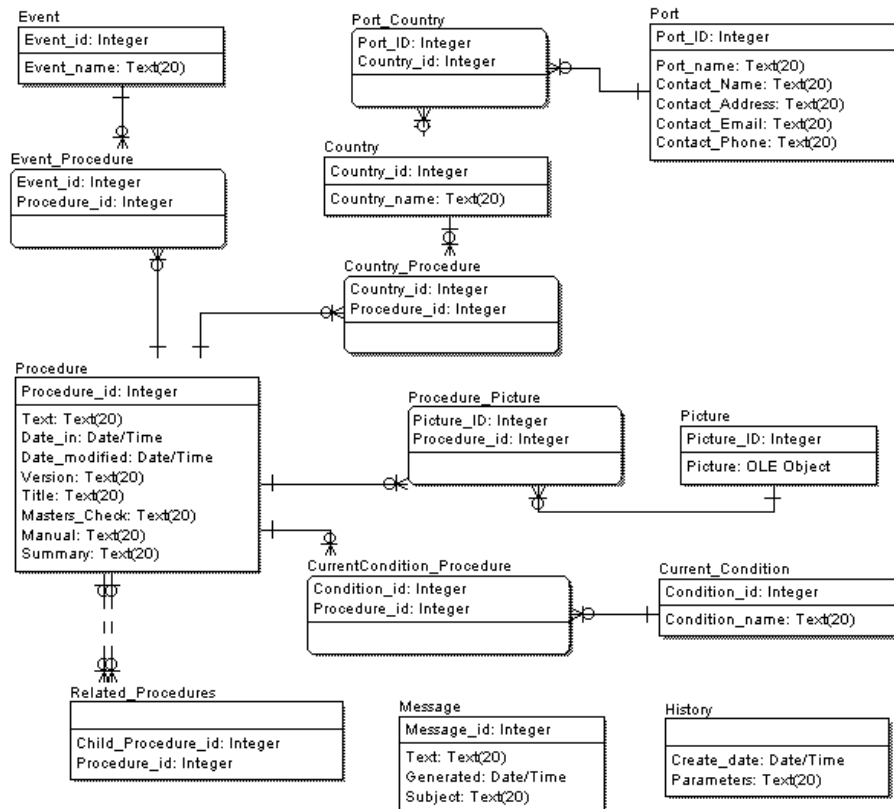


Рис. 2.4. Моделювання даних №1

Зрозуміло, що розмір можна обчислити у будь-який момент, однак оскільки решта інформації береться з БД, а обчислення розміру вимагає виконання повільної операції — звернення до файлової системи, нерідко приймають рішення зберігати розмір у БД і лише на рівні програми обчислювати і оновлювати його за кожної операції із файлом. Зрозуміло, що в цьому випадку таблиця не задовольнятиме навіть другої нормальної форми.

Розробники баз даних загалом дотримуються наведеної вище схеми. Разом про те вони, наприклад, часто передбачають первинні ключі ще етапі побудови ER-моделі, а таблиці відразу намагаються робити нормалізованими. Проте спроектувати БД "відразу" можна лише на невеликому проєкті. Як правило, всі описані вище етапи однаково доводиться проходити. І чим кваліфікованіший спеціаліст, тим ретельніше він ставиться до всіх кроків розробки.

Важко уявити, щоб у рамках базового курсу вчитель мав можливість продемонструвати всі етапи низхідного проєктування БД. Однак такого завдання у базовому курсі і не ставиться. Головне, щоб учні усвідомили основні поняття, пов'язані з реляційними БД - таблиця, ключ, зв'язок.

Продемонструємо, як можна розумно мінімізувати процес демонстраційної розробки БД, відразу спустившись до потрібного рівня абстракції, не використовуючи понять, не підтриманих часом, необхідним для засвоєння та закріплення, і акцентуючи увагу лише на найважливіших питаннях. Розробимо БД "Сторінка класного журналу". Дуже важливо те, що типова сторінка класного журналу добре знайома учням.

Починаючи з проєктування бази даних, передусім слід визначитися з тим, які дані потрібно зберігати. Іноді, щоб отримати відповідь це питання, доводиться довго працювати із замовником (експертом) чи структурувати свої власні думки. У нашому випадку ситуація полегшується тим, що ми знаємо, як виглядає журнал і що зберігають у ньому. Оскільки ми не збираємося робити краще кращого, постараємося просто покласти паперовий класний журнал на музику реляційної бази даних. На сторінці журналу, що відноситься до одного предмета, є список учнів (зазвичай для кожного учня вказуються ім'я та прізвище), дати уроків та оцінки. Крім того, вчителі зазвичай позначають олівцем типи деяких уроків (і, відповідно, походження отриманих на них оцінок: "с/р", "тест" тощо). Усі?.. Ні, не всі. Крім оцінок, у журналі зазвичай відзначаються перепустки уроків. Це також слід враховувати під час проєктування структури БД. Нарешті, введемо дуже важливе обмеження:

вважатимемо, що у одній клітині журналу зберігається лише оцінка, тобто. клітин виду 2/5 у журналі немає.

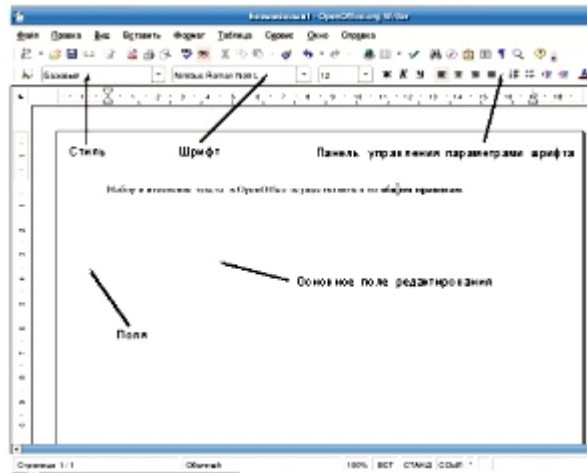


Рис. 2.5. Приклад

Відразу зазначимо, оскільки йдеться про одну сторінку журналу, предмет і вчитель є властивостями самої БД, і зберігати їх у БД немає сенсу. Ходімо далі.

Взагалі, сторінка сама по собі вже є таблицею, але є одна неприємність: кількість стовпців у ній не фіксована. Хоча в кожному конкретному журналі воно те саме, але й журнали різні бувають, і не всі стовпці завжди заповнені (наприклад, нову чверть нерідко починають на новій сторінці). Тому перше, що зробимо, створимо макет таблиці, що містить всі необхідні дані та має фіксоване число стовпців.

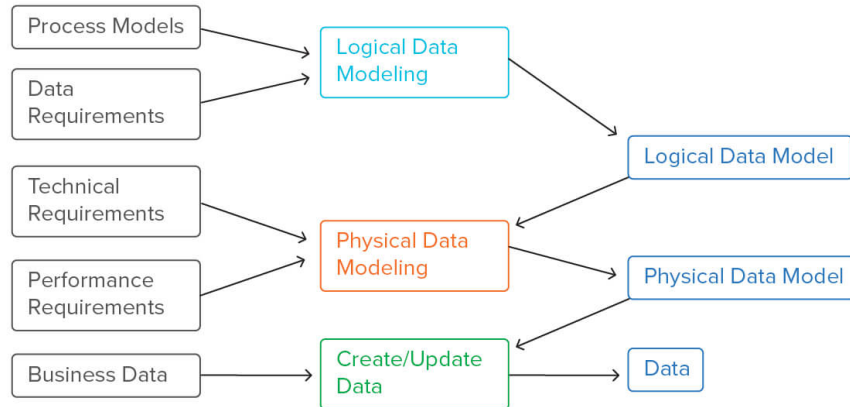


Рис. 2.6. Моделювання даних №2

Перерахуємо ті недоліки обраного представлення даних, що лежать лежить на поверхні.

1. Навіщо ми багато разів повторили, що 8.09 була самостійна робота? Зрозуміло, що коли вона була, то у всіх. І взагалі, якщо вже був урок, то у всіх (але не слід забувати, що учень міг бути відсутнім на уроці або просто не отримати за нього оцінку).

2. Як ми збираємося зберігати значення у полі “Вміст клітинки журналу”? Яке число? Але в цьому стовпці трапляються і символи "н". Як символ? Можна, звичайно, але тоді при різних обчисленнях (наприклад, при обчисленні середньої оцінки) доведеться щось із цим символом робити, щоб перетворити його на цифру.

3. Раптом у нас у класі є два Бананові Борі? Як ми їх збираємося розрізняти?

4. Нарешті, навіщо нам рядки з порожніми клітинами у полі “Вміст клітини журналу”? Це в журналі є порожні клітини, нам вони навіщо?

Розглянемо ці питання порядку, починаючи з останнього. Такі рядки нам не потрібні. Можна просто викинути їх із таблиці.



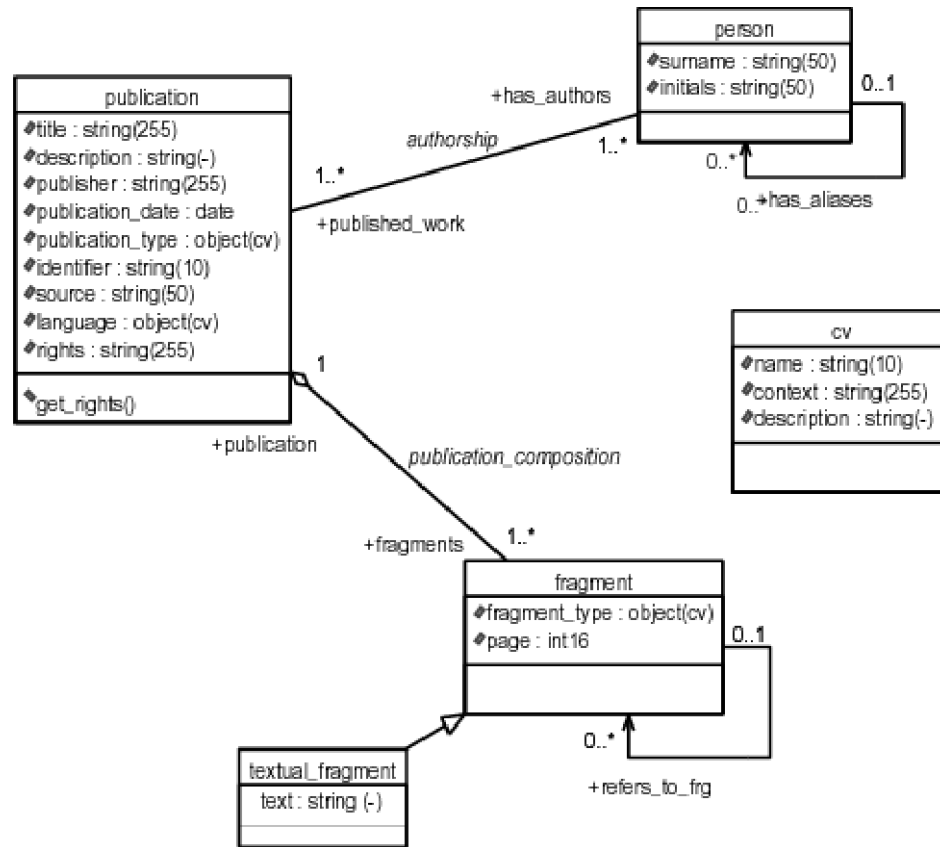


Рис. 2.7. Моделювання даних №3

Відповідаючи на третє питання, варто ще раз згадати, як виглядає сторінка журналу. Чи всі дані ми перенесли до таблиці бази даних? Адже питання, яке постало перед нами, не є специфічним саме для бази даних, його якось доводиться вирішувати і у звичайному журналі. Виявляється, ми просто забули про те, що список учнів у журналі пронумерований, а номер є тим унікальним полем, яке однозначно визначає кожного учня. Має сенс завести окрему таблицю для зберігання номерів (унікальних ідентифікаторів) учнів та їхніх імен та прізвищ. Тоді саме номери можна використовувати в інших таблицях.

Символи "н" нам заважають. То, можливо, й у фіксації перепусток завести окрему таблицю? І відзначати у цій таблиці, хто і в який день був відсутній.

І для уроків також можна завести окрему таблицю. Достатньо один раз позначити в цій таблиці дату уроку і, можливо, тут же записати додаткову інформацію.

## **2.4. Висновки до другого розділу**

Розроблено методи оптимізації структури бази даних DataAtlas Modeler для моделі "зовнішній ключ".

Розроблено формалізований метод, заснований на теорії множин і що дозволяє автоматизувати процес побудови за заданою сукупністю властивостей об'єктів та з урахуванням їх входження до запитів до бази даних визначити оптимальну структуру бази даних та програми DataAtlas Modeler .

Розроблено метод визначення структури вкладених документів у БД типу DataAtlas Modeler. Цей метод враховує тип зв'язків між таблицями реляційної моделі. Залежно від типу зв'язку виїдено правила для створення структури колекції із вкладеними документами.

Протестовано методи оптимізації структури бази даних DataAtlas Modeler з різними тестовими базами даних (база даних) та різними обсягами даних. Результати тестування показали, що розроблені методи є ефективними для прискорення обробки даних та скорочення, що використовується для зберігання даних пам'яті.

Розроблено методику перетворення реляційної бази даних у формат DataAtlas Modeler типу «зовнішній ключ» залежно від вихідних даних, таких як наявність інформації про існуючу реляційну базу даних, наявність необхідності трансляції вихідної бази даних у новий формат, належності реляційної бази даних до нормальної форми.

## РОЗДІЛ 3

### ПРОЕКТУВАННЯ ER-ДІАГРАМИ

#### 3.1. Інфологічна модель даних, приклад побудування ER-діаграми

У діловій або особистій сфері часто доводиться працювати з даними з різних джерел, кожне з яких пов'язане з певним видом діяльності. Нині завдяки величезним можливостям комп'ютерів, пов'язаним зі зберіганням і обробленням великих масивів інформації, комп'ютер застосовують для розв'язання широкого кола завдань буквально в усіх сферах людської діяльності. Одночасно з розвитком комп'ютерної техніки розвивалася і теорія баз даних (БД), які являють собою набори взаємопов'язаних даних про деяку предметну область. Такі набори мають певну структуру і постійно зберігаються в пам'яті комп'ютера.

У результаті розвитку концепцій БД виокремлено три рівні представлення інформації: інфологічний, даталогічний і фізичний. На кожному рівні проводять структурування інформації таким чином, щоб на третьому рівні інформацію можна було подати у вигляді структур даних, які реалізуються в пам'яті ЕОМ. На інфологічному рівні визначають, яка інформація про предметну область буде зберігатися й оброблятися в комп'ютері, а в результаті дослідження предметної області будують її інфологічну модель, яку описують у термінах класів об'єктів та їхніх взаємодій. В інфологічній моделі інформація представляється незалежно від того, що являють собою дані і які технічні засоби будуть використані надалі для її зберігання та обробки. Даталогічна та фізична моделі безпосередньо реалізуються в системах управління БД (СУБД), а фізична модель у свою чергу визначає структуру зберігання даних на фізичних носіях. Мета інфологічного проектування полягає в поданні семантики (сенсу) предметної області. Для опису предметної області найчастіше використовують модель "сутність-зв'язок", яку скорочено називають ER-моделлю від англійської назви "Entity - Relationship" ("Сутність - зв'язок").

ER-діаграма моделі має лексикографічну структуру і містить у собі текст та елементи графіки. На практиці інфологічна (семантична) модель використовується на першій стадії проектування БД. При цьому в термінах ER-моделі описується концептуальна (понятійна) схема БД, яка потім перетворюється до реляційної або іншої схеми. ER-моделі набули поширення в CASE-системах, що підтримують проектування реляційних БД. Слід зазначити, що під час підготовки вчителів та студентів інженерних спеціальностей вищих навчальних закладів слід виокремлювати ключові поняття теорії БД і застосовувати їх під час проектування БД.

Процес проектування БД являє собою послідовність переходів від неформального словесного опису інформаційної структури предметної області до формалізованого опису об'єктів предметної області в термінах деякої моделі. При цьому можна виділити такі етапи проектування [4]:

- системний аналіз і словесний опис інформаційних об'єктів предметної області;
- проектування інфологічної моделі предметної області - частково формалізований опис об'єктів предметної області в термінах деякої семантичної моделі, наприклад, у термінах ER-моделі;
  - даталогічне або логічне проектування БД;
  - фізичне проектування БД.

Створимо інфологічну (семантичну) модель предметної області, що міститиме відомості про студентів, успішність студентів з предметів і викладачів, які викладають ці предмети в термінах ER-моделі з побудовою ER-діаграми.

ER-модель спирається на поняття сутність, атрибут, зв'язок, і предметна область має бути подана як сукупність сутностей з атрибутами, між якими встановлено зв'язки. З об'єктами ER-моделі пов'язані поняття: тип - набір однорідних предметів, явищ, які виступають як єдине ціле; примірник - конкретний елемент набору, який (набір) означає деякий тип; множина - конкретний набір примірників типу. Сутність - це об'єкт, який може бути

ідентифікований деяким способом, що відрізняє його від інших об'єктів. Тип сутності означає набір однорідних сутностей деякого типу. Набір сутностей - множина сутностей одного типу. Сутність фактично є множиною атрибутів, які описують властивості всіх членів даного набору сутності. Тип сутності означає множину однорідних об'єктів, а "екземпляр сутності" - конкретний об'єкт із множини об'єктів.

Зв'язок - це поійменоване відношення, що має місце між двома сутностями. Такий зв'язок є бінарним у тому сенсі, що він має місце між двома поійменованими сутностями або ж має вигляд відношення сутності до самої себе. Кожен зв'язок має два кінці, кожен з яких має: ім'я, ступінь (потужність), ознаку обов'язковості. Ці властивості використовуються для характеристики зв'язку стосовно кожної зі сторін, що беруть участь у ньому.

Кожна сутність повинна мати унікальний ідентифікатор (ключ), тобто має бути унікально визначена: кожен екземпляр сутності має мати чітке і неоднозначне визначення, що дає змогу відрізнити його від інших екземплярів тієї самої сутності. Ключем може бути атрибут, комбінація атрибутів, комбінація зв'язків або атрибутів і зв'язків. Наприклад, ключем для сутності Студент є атрибут № залікової книжки.

На сьогодні існують кілька нотацій для представлення ER-діаграм:

1. Нотація Чена: сутність зображується прямокутником, атрибут - овалом, сполученим зі своєю сутністю (атрибут, що ідентифікує, підкреслено), а зв'язок - ромбом, сполученим зі зв'язаними сутностями. Вид лінії в місці з'єднання із сутністю визначає кардинальність зв'язку ("вороняча лапка" - М, "хрест" - 1). Імена сутності, атрибута і зв'язку розташовуються всередині їхніх зображень (Рис. 3.1).



Рис. 3.1. Приклад ER-діаграми в нотації Чена

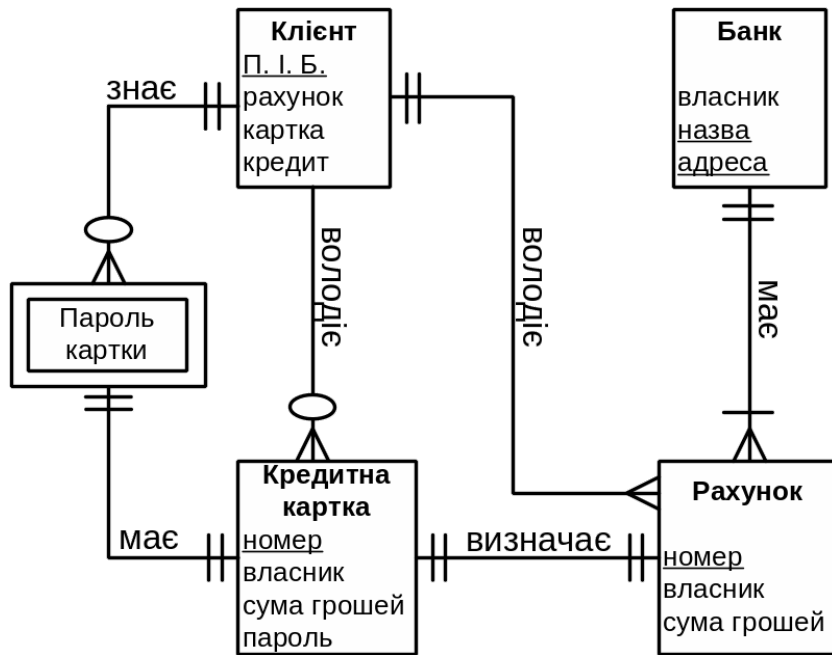


Рис. 3.2. Приклад ER-діаграми в нотації Мартіна

## Слабкі (залежні) сутності

- Нехай існування сутності А залежить від існування сутності В (кажуть, що сутність А "слабка", а сутність В - "сильна").

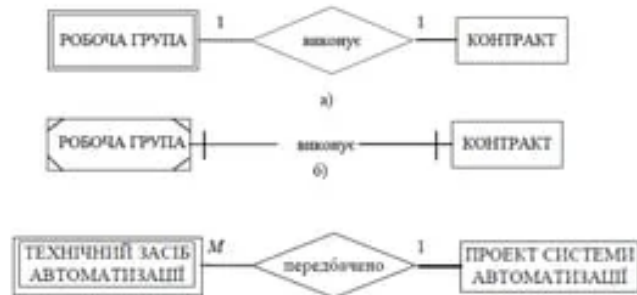


Рис. 3.2. Моделі даних БД. ER-діаграми

І хоча існує усвідомлює, що більшість людей воліє вчитися на власних помилках, він усе ж таки ще раз радить недосвідченим проектувальникам баз даних:

- чітко розмежовувати такі поняття як запит на дані та введення даних (введення, зміна та видалення);
- пам'ятати, що, як правило, база даних є інформаційною основою не одного, а кількох додатків, частина з яких з'явиться в майбутньому;
- поганий проект бази даних не може бути виправлений за допомогою будь-яких (навіть найвитонченіших) додатків.

### 3.2. Модель «сутність – зв'язок» ER-моделі

ER-модель є однією з найпростіших візуальних моделей даних (графічних нотацій). Вона дає змогу позначити структуру "великими мазками", у загальних рисах. Цей загальний опис структури називається ER-діаграмою або онтологією обраної предметної області (area of interest).

Прикладом ER-моделі є банківські реквізити - явна характеристика покупця. Найменування товару - явна характеристика товару.

Ціна товару - схоже, що це характеристика товару. Чи відрізняється ця характеристика від ціни в накладній? Одиниця виміру - явна характеристика товару. Номер накладної - явна унікальна характеристика накладної. Дата накладної - явна характеристика накладної.

Список товарів у накладній - список не може бути атрибутом. Імовірно, потрібно виділити цей список в окрему сутність.

Кількість товару в накладній - це явна характеристика, але характеристика чого? Це характеристика не просто "товару", а "товару в накладній".

Ціна товару в накладній - знову ж таки це має бути не просто характеристика товару, а характеристика товару в накладній. Але ціна товару вже зустрічалася вище - це одне й те саме? Сума накладної - явна характеристика накладної. Ця характеристика не є незалежною. Сума накладної дорівнює сумі вартостей усіх товарів, що входять до накладної. Найменування складу - явна характеристика складу.

На етапі переходу до реалізації цієї ER-діаграми у вигляді реальної інформаційної системи або програми відбувається відображення ER-моделі в детальнішу модель даних реляційної (об'єктної, мережевої, логічної, або ін.) бази даних, яку називають фізичною моделлю даних по відношенню до вихідної ER-діаграми.

Будь-який фрагмент предметної області можна подати як множину сутностей, між якими існує деяка множина зв'язків. Дамо визначення:

Сутність (entity) - це об'єкт, який можна ідентифікувати якимось способом, що відрізняє його від інших об'єктів. Приклади: конкретна людина, підприємство, подія тощо.

Набір сутностей (entity set) - безліч сутностей одного типу (що мають однакові властивості). Приклади: всі люди, підприємства, свята тощо. Набори



сутностей не обов'язково мають бути такими, що не перетинаються. Наприклад, сутність, що належить до набору ЧОЛОВІКИ, також належить набору ЛЮДИ.

Сутність фактично являє собою множину атрибутів, які описують властивості всіх членів цього набору сутностей.

### 3.2.2. Приклад

Розглянемо безліч працівників якогось підприємства. Кожного з них можна описати за допомогою характеристик табельний номер, ім'я, вік. Тому, сутність ПРАЦІВНИК має атрибути ТАБЕЛЬНИЙ\_НОМЕР, ІМ'Я, ВІК. Використовуючи нотацію мови Pascal цей факт можна представити як:

```
type employe = record
    number : string[6];
    name : string[50];
    age : integer;
end;
```

Надалі для визначення сутності та її атрибутів будемо використовувати позначення вигляду.

СПІВРОБІТНИК (ТАБЕЛЬНИЙ\_НОМЕР, ІМ'Я, ВІК).

Наприклад відділи, на які підрозділяється підприємство, і в яких працюють співробітники, можна описати як ВІДДІЛ(НОМЕР\_ВІДДІЛУ, НАЗВАННЯ).

Множина значень (область визначення) атрибута називається доменом. Наприклад, для атрибута ВІК домен (назвемо його ЧИСЛО\_РОКІВ) задано інтервалом цілих чисел більших за нуль, оскільки людей з від'ємним віком не буває.

Звідси визначається ключ сутності - група атрибутів, така, що відображення набору сутностей у відповідну групу наборів значень є взаємно однозначним відображенням. Інакше кажучи: ключ сутності - це один або

більше атрибутів, які унікально визначають цю сутність. У нашому прикладі ключем сутності СПІВРОБІТНИК є атрибут ТАБЕЛЬНИЙ\_НОМЕР (звісно, тільки в тому разі, якщо всі табельні номери на підприємстві унікальні).

Зв'язок (relationship) - це асоціація, встановлена між кількома сутностями. Приклади:

- оскільки кожен співробітник працює в якомусь відділі, між сутностями СПІВПРАЦІВНИК і ВІДДІЛ існує зв'язок "працює в" або ВІДДІЛ-працівник;

- оскільки один із працівників відділу є його керівником, то між сутностями СПІВПРАЦІВНИК і ВІДДІЛ існує зв'язок "керує" або ВІДДІЛ-КВАЛІВНИК;

- можуть існувати і зв'язки між сутностями одного типу, наприклад зв'язок БАТЬКІВ - СВІТОМОК між двома сутностями ЛЮДИНА;

(У дужках тут слід зазначити, що в методиці проектування даних є своєрідне правило хорошого тону, згідно з яким сутності позначаються за допомогою іменників, а зв'язки - дієслівними формами. Це правило, однак, не є обов'язковим).

На жаль, не існує загальних правил визначення, що вважати сутністю, а що зв'язком. У розглянутому вище прикладі ми поклали, що "керує" - це зв'язок. Однак, можна розглядати сутність "керівник", яка має зв'язки "керує" із сутністю "відділ" і "є" із сутністю "співробітник".

Зв'язок також може мати атрибути. Наприклад, для зв'язку ВІДДІЛ-працівник можна задати атрибут СТАЖ\_РОБОТИ\_У\_В\_ВІДДІЛІ.

Роль сутності у зв'язку - функція, яку виконує сутність у цьому зв'язку. Наприклад, у зв'язку БАТЬК-ПОТОМОК сутності ЛЮДИНА можуть мати ролі "батько" і "нащадок". Зазначення ролей у моделі "сутність-зв'язок" не є обов'язковим і слугує для уточнення семантики зв'язку.

Набір зв'язків (relationship set) - це відношення між  $n$  (причому  $n$  не менше 2 сутностями, кожна з яких відноситься до деякого набору сутностей).

Приклад:

сутності набори сутностей

-----

e1 належить E1

e2 належить E2

...

en належить En

тоді  $[e1, e2, \dots, en]$  - набір зв'язків R

Хоча, відверто кажучи, поняття "зв'язок" і "набір зв'язків" є різними (перший є елементом другого), їх, тим не менш, дуже часто змішують. Тому, ми, не претендуючи на академічну строгість, надалі також будемо часто користуватися термінами "зв'язок", маючи на увазі "набір зв'язків", і "сутність", маючи на увазі "набір сутностей".

У випадку  $n=2$ , тобто коли зв'язок об'єднує дві сутності, він називається бінарним. Доведено, що  $n$ -арний набір зв'язків ( $n>2$ ) завжди можна замінити множиною бінарних, проте перші краще відображають семантику предметної області.

Модель "сутність - зв'язок" (або ER-модель) являє собою спосіб логічного уніфікованого представлення даних деякої предметної області. Хоча, як ми побачимо далі, ця модель дуже нагадує систему пов'язаних одна з одною таблиць, насправді це абсолютно загальне уявлення. Ця модель може бути перетворена до будь-якої з існуючих конкретних моделей даних: ієрархічної, мережевої, реляційної, об'єктної. Суттєво, що ER-модель дає змогу уявляти лише дані, але не дії, які з ними можуть виконуватися, тому вона використовується лише для проектування структури збережених даних. Оскільки багато понять, які ми будемо розбирати у зв'язку з моделлю "сутність - зв'язок", були нами розглянуті в основах реляційних баз даних, будемо спиратися на ці знання.

Перевагами цієї моделі є:

— простота;

- наочність;
- однозначність;
- використання природної мови.

Наприклад, сутність Факультет позначає приналежність студента до даного підрозділу інституту, але є цілком самостійною.

Будь-який фрагмент предметної області може бути представлений деяким набором сутностей і зв'язками між ними. Наприклад, розглядаючи предметну область ФАКУЛЬТЕТ, можна виділити такі основні сутності: СТУДЕНТ, КАФЕДРА, СПЕЦІАЛЬНІСТЬ, ДЕКАНАТ, ГРУПА, ВИКЛАДАЧ, ЕКЗАМЕН. На першому етапі створення ER-моделі даних слід виокремити всі сутності, які передбачається описувати з огляду на постановку завдання. Зайвий раз підкреслимо, що сутністю може бути не тільки деякий матеріальний об'єкт, а й деякий процес, наприклад, ЕКЗАМЕН, ЛЕКЦІЯ. Сутністю може бути і деяка кількісна та якісні характеристики об'єкта: ВЧЕНЕ ЗВАЖЕННЯ, СТАЖ тощо. Усе насправді залежить від постановки задачі та від нашого аналізу предметної області.

### **3.3. Висновки до третього розділу**

Розроблено метод побудови схеми бази даних «сутність – зв'язок» ER-моделі та типу побудування ER-діаграми з урахуванням інформаційного графа запиту та розподіленої структури реляційної бази даних. Метод заснований на теорії графів та теорії множин і дозволяє побудувати на шардах та репліках оптимальну за швидкістю виконання запитів структуру колекцій. Метод заснований на розробленому методі побудови інформаційного графа запиту для розподіленої бази даних ER-діаграми базується на теорії графів. Надалі до цього графа можуть бути застосовані методи розпаралелювання класичних алгоритмів з метою оптимізації процесу виконання запиту за різними параметрами, такими як час виконання, обсяг обчислювальних ресурсів, мережевих параметрів.

Розроблено метод трансляції запитів із ER-діаграми з урахуванням структури бази даних. Розроблений метод складається із чотирьох кроків: парсинг вихідного запиту, формування словника речень запиту ER-моделі, переформування речень словника з урахуванням структури бази даних

ER-моделі та синтез вихідного запиту. Розроблений метод грає важливу участь у загальному вирішенні проблеми трансляції та консолідації баз даних різного типу.

Розглянуто процес моделювання предметної області. Визначено основні поняття, що використовуються при цьому (сутність, атрибут, ідентифікатор, зв'язок, типи зв'язків, ER-діаграма).

## ВИСНОВКИ

1. Розглянуто процес моделювання предметної області. Визначено основні поняття, що використовуються при цьому (сутність, атрибут, ідентифікатор, зв'язок, типи зв'язків, ER-діаграма).

2. Розглянуто основні етапи моделювання сутностей і зв'язків (моделювання локальних уявлень, об'єднання локальних моделей з використанням понять ідентичність, агрегація, узагальнення).

3. Дано поняття обмежень цілісності, що мають безпосередній стосунок до предметної області (зовнішні обмеження; обмеження, описані за допомогою спеціальних конструкцій).

4. Реальним засобом моделювання даних є не формальний метод нормалізації відносин, а так зване семантичне моделювання.

5. У разі правильного визначення сутностей, отримані таблиці відразу перебуватимуть у 3НФ. Основна перевага методу полягає в тому, що модель будується методом послідовних уточнень початкових діаграм.

6. Реферативна база даних (частковий доступ до тексту) - база даних, що включає анотацію, реферат або інші вказівки про зміст документа в неповному його доступі.

7. Зроблено проєктування інформаційного забезпечення, як інфологічне, так і даталогічне. Спроєктовано як логічну ER-модель, так і фізична.

8. Проведено програмування на стороні SQL-сервера, виведено список мінімальних вимог до системи, а також проведено детальний опис користувацького інтерфейсу системи.

Під час виконання випускної кваліфікаційної роботи було отримано такі результати:

- проаналізовано предметну область;
- складено опис бізнес-процесів в обраних нотаціях;
- спроєктовано алгоритм роботи і структуру системи;

- здійснено проектування інформаційного забезпечення;
- розроблено та протестовано інформаційну систему.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Атре, Ш. Структурний підхід до організації бази даних / Ш. Атре. Атре - М.: Фінанси і статистика, 2019. - 312 с.
2. Ахтирченко, К.В. Розподілені об'єктні технології в інформаційних системах / К.В. Ахтирченко, В.В. Лесик // № 5-6 – Київ, 2019. - С. 52-64.
3. Бейлі, Р. Принципи корпоративних фінансів / Р. Бейлі, С. Майерс - М.: Олімп - Бізнес, 2020. - 356 с.
4. Бойко, В.В. Проектування баз даних інформаційних систем / В.В. Бойко, В.М. Северін - М.: Фінанси і статистика, 2019. - 352 с.
5. Гірський, А. Microsoft SQL. Server 6.5 для професіоналів / А. Гортинський, Ю. Стругайло - СПб.: Київ, 2020. - 374 с.
6. Гусенко, Т.І., Проектування баз даних у прикладах і задачах / Т.І. Гусенко. Т.І. Гусенко, Ю.Б. Берашатський - М.: Радіо і зв'язок, 2018. - 160 с., іл.
7. Двуреченський, С.В. Концептуальне проектування баз даних до АСУ / С.В. Двуреченський - М.: , 2019. - 121 с.
8. Дейт, К. Введення в системи баз даних: Пер. з англ / К. Дейт - М.: Наука, 2019. - 464 с.
9. Джексон, Г. Проектування реляційних баз даних для використання з мікро-ЕОМ / Г. Джексон - М.: Мир, 2020. - 416 с.
10. Дуванов, А.А. Web-конструювання / А.А. Дуванов - Київ: 2021. - 384 с.
11. Едомський, Ю. Є. Техніка Web-дизайну для студента / Ю.Є. Едомський. Єдомський - Київ: 2019. - 491 с.
12. Ендовицький, Д.А. Інвестиційний аналіз у реальному секторі економіки. / Д.А. Ендовицький, за ред. Л.Т. Солом'янської. - М.: Фінанси і статистика, 2021. - 352 с.
13. Зіндер, Є.З. Проектування баз даних: нові вимоги, нові підходи / Є.З. Зіндер - М.: Фінанси і статистика, 2019. - 182 с.



14. Карпатський, Т.С. Бази даних: моделі, розробка / Т.С. Карпатський - СПб.: Київ, 2020. - 304 с.
15. Кириленко, В.В. Основи проектування баз даних / В.В. Кириленко - М.: Фінанси і статистика, 2019. - 204 с.
16. Коцюбинський, В.В. Методичні рекомендації з оцінки ефективності інвестиційних проектів: (Друга редакція) / ГК по буд-ву, архіт. і жит. політиці; авткол / В.В. Комаренко, В.Н. Лівшицький, М.: "Видавництво Економіка", 2019. - 421 с.
17. Крамер, Е. HTML: наочний курс Web-дизайну: Пер. з англ. / Е. Крамер. Крамер - М.: Діалектика, 2020. - 304 с.
18. Кренке, Д. Теорія і практика побудови баз даних: Пер. з англ. / Д. Кренке. - 9 - е изд. - СПб.: Київ, 2019. - 858 с.
19. Криниченко, Е.І. Аналіз ефективності інвестиційної та інноваційної діяльності підприємства / Е.І. Криниченко, І.В. Жеруйко - М.: Фінанси і статистика, 2019. М.: Фінанси і статистика, 2019. - 384 с.
20. Кучеренко, П.А. Практичне керівництво з використання "ProjectExpert" в економічних розрахунках / П.А. Кучеренко, Є.В. Косинський - Львів: 2019. - 29 с.
21. Ладиженко, Г. Системи управління базами даних - коротко про головне / Г. Ладиженко // СУБД № 2. - Київ, 2019, С. 34-39.
22. Левицький, Б. Web-дизайн. Посібник користувача / Б. Левицький - Київ: 2019. - 384 с.
23. Мейєр, Д. Теорія реляційних баз даних / Д. Мейєр - М.: Світ, 2020. - 608 с.
24. Мацяшек, Лешек А. Аналіз і проектування інформаційних систем за допомогою UML 2.0, 3-е изд.: Пер. з англ. - М.: ТОВ "І. Д. Вільямс", 2020. - 816 с.
25. Винниченко В.Н. Інформаційні системи / В.Н. Винниченко - СПб: Київ, 2019. - 688 с.

26. Перепілка, В.М. Створення Web-сторінок і Web-сайтів / В.М. Перепілка.- М.: Тріумф: 2020. - 370 с.
27. Роббінс, Д. Web-дизайн. Довідник / Д. Роббінс - М.: "": 2019. - 816 с.
28. Сестринський, І.Є. Початки web-дизайну / І.Є. Сестринський - Київ: 2020. - 491 с.
29. Тіорі, Т. Проектування структур баз даних: У 2-х кн. Пер. з англ. / Т. Тіорі, Дж. Фрай - М.: Мир, 2019. - 287 с., іл.
30. Северін, Ю.М. Аналіз даних на комп'ютері / Під. ред. В.Е. Гунько, Ю.М. Северін, А.А. Макаренко - М.: ІНФРА-М, Фінанси і статистика, 2020. - 384 с.
31. Стерненко, Дж. Введення в системи баз даних / Дж. Ульман, Дж. Відом - М.: Лорі, 2019. - 374 с.
32. Хансен, Г. Бази даних: розробка та управління / Г. Хансен, Д. Хансен - М.: , 2020. - 704 с.
33. Хомоненко, А.Д. Бази даних: Підручник для вищих навчальних закладів / Під ред. проф. А.Д. Хомоненко, В.М. Цицерон, М.Г. Миколайчук СПб.: КОРОНА принт, 2019. - 416 с.
34. Цикритзис, Д. Моделі даних / Д. Цикритзис, Ф. Лоховські - М.: Фінанси і статистика, 2019. - 214 с.
35. Гайдамак, В.М. Бази та банки даних / В.М. Гайдамак - М.: Вищ. шк., 2019. - 245 с.
36. HTML і Web дизайн для початківців [Електронний ресурс]: - Режим доступу
37. Мак-Дональд М. Access 2017. Відсутній посібник: Пер. з англ. - М.: Видавництво "Правда"; Спб: "Київ", 2021. 784 с.
38. Абламейко М. С. Електронний документообіг у країнах євразійського економічного союзу: порівняльний аналіз та шляхи уніфікації / М. С. Абламейко, А. А. Андрончик // Журнал зарубіжного законодавства і порівняльного правознавства. - 2020. - № 2. - С. 36-47.

39. Абраменко М. О. Деякі аспекти трансформації функцій сучасних грошей в умовах діджиталізації економіки / Абраменко М. О. // Банківські послуги. - 2020. - N 1. - С. 12-16.
40. Бойченко І. С. Правове регулювання електронної взаємодії в умовах цифрової економіки: проблеми та вектори розвитку / І. С. Бойченко // Право і держава: теорія і практика. - 2019.
41. Булавко О. А. Оцінка інвестиційного та інноваційного потенціалу в період розвитку цифрової економіки / О. А. Булавко Питання інноваційної економіки. - 2020. - Т. 10, № 1. - С. 103-110.
42. Василенко І. А. "Розумне місто" в цифровому суспільстві 5.0: соціально-політичні та гуманітарні ризики цифровізації. гуманітарні ризики цифровізації суспільного простору / І. А. Василенко // Влада. - 2019. - Т. 27, № 5. - С. 67-73.
43. Веретено А. А. Розумний блогінг і влогінг. Платні способи просування в Instagram / О. О. Веретено // Маркетингові комунікації. - 2020. - № 1. - С. 44-54.
44. Лірник Н. А. Вимірювання впливу цифрової трансформації сфери послуг на якість життя населення / Н. А. Лірник // Державне управління. Електронний вісник. - 2019.
45. Герасименко В. В. Напрями розвитку інструментів маркетингу в умовах цифровізації ринків / В. В. Герасименко // Маркетинг в Україні та за кордоном. - 2019. - № 6. - С. 44-51.
46. Гончаренко, Л. П. Цифровізація національної економіки / Л. П. Гончаренко, // Вісник університету. - 2019.
47. Мартін Джеймс. Мови четвертого покоління. - Том. 1. - New Jersey: Prentice-Hall, Inc, 2019.
48. Hansen Gary W., Hansen James V. Database Management and Design. - New Jersey: Prentice-Hall, Inc., 2020.
49. Загальногалузеві керівні матеріали зі створення банків даних. - М.: ДКНТ, 2021.

50. Саймон А.Р. Стратегічні технології баз даних: менеджмент на 2021 рік. - М.: Фінанси і статистика, 2021.

51. Фаулер М., Скотт К. UML у короткому викладі. Застосування стандартної мови об'єктного моделювання / Пер. з англ. - М.:, 2019.

52. Фаулер М. UML. Основи / Пер. з англ. - 3-е изд. - СПб: Символ-Плюс, 2021. - 192 с., іл.

53. Хансен Г., Хансен Дж. Бази даних. Розробка та управління. - М.: Біном, 2020.

54. Мюллер, Дж. Бази даних і UML. Проектування / Дж. Мюллер. - М.: Лорі, 2019. - 356 с.

55. Конноллі Т., Бегг К. Бази даних: проектування, реалізація, супровід. Теорія і практика, 3-е изд. : Пер. з англ. : Навч. Пос. - М.: Вид. дім "Вільямс", 2019. - 1440 с.

56. Метод. вказівки до дипломного проектування з курсу "Бази даних" / Київський державний інститут; Укл: Карпенко І.П. - М., 2020. - 28 с.

## ДОДАТОК А

## КОД ПРОГРАМИ

- На MySQL:

```
select
l_returnflag ,
l_linestatus ,
sum ( l_q u a n t i t y ) a s s u m_q t y ,
sum ( l_e x t e n d e d_p r i c e ) a s s u m_b a s e_p r i c e ,
sum ( l_e x t e n d e d_p r i c e * ( 1 - l_d i s c o u n t ) ) a s s u m_d i s c_p r i c e ,
sum ( l_e x t e n d e d_p r i c e * ( 1 - l_d i s c o u n t ) * ( 1 + l_t a x ) ) a s s u m_c h a r g e ,
avg ( l_q u a n t i t y ) a s a v g_q t y ,
avg ( l_e x t e n d e d_p r i c e ) a s a v g_p r i c e ,
avg ( l_d i s c o n t ) a s a v g_d i s c ,
count ( * ) a s c o u n t_o r d e r
from
l i n e i t e m
where
l_s h i p_d a t e <= '1998-09-02'
groupby
l_returnflag ,
l_linestatus
orderby
l_returnflag ,
l_linestatus
```

- Для варианта 1:

```
db.getCollection("V").aggregate([
{
"$match": { "L_SHIPDATE": { "$lte": newDate(1998, 12, 1) } }
},
{
"$group": {
"_id": { "L_RETURNFLAG": "$L_RETURNFLAG", "L_LINESTATUS": "$L_LINESTATUS" },
"sum_qty": { "$sum": "$L_QUANTITY" },
"sum_base_price": { "$sum": "$L_EXTENDEDPRI" },
"sum_disc_price": { "$sum":
{ "$multiply": [ "$L_EXTENDEDPRI",
{ "$subtract": [ 1, "$L_DISCOUNT" ] } ] } }
},
"sum_charge": { "$sum":
164
{ "$multiply": [ "$L_EXTENDEDPRI",
"$subtract": [ 1, "$L_DISCOUNT" ] } , { "$add": [ 1, "$L_TAX" ] } ] } } ,
"avg_qty": { "$avg": "$L_QUANTITY" },
"avg_price": { "$avg": "$L_EXTENDEDPRI" },
"avg_disc": { "$avg": "$L_DISCOUNT" },
"count_order": { "$sum": 1 }
}
},
{ $sort: { "_id.L_RETURNFLAG": 1, "_id.L_LINESTATUS": 1 } } ] )
```

- Для варианта 2:

```
db.getCollection("lineitem").aggregate([
{
"$match": { "L_SHIPDATE": { "$lte": newDate(1998, 12, 1) } }
},
{
"$group": {
"_id": { "L_RETURNFLAG": "$L_RETURNFLAG", "L_LINESTATUS": "$L_LINESTATUS" },
"sum_qty": { "$sum": "$L_QUANTITY" },
"sum_base_price": { "$sum": "$L_EXTENDEDPRI" },
"sum_disc_price": { "$sum":
{ "$multiply": [ "$L_EXTENDEDPRI",
{ "$subtract": [ 1, "$L_DISCOUNT" ] } ] } }
},
"sum_charge": { "$sum":
{ "$multiply": [ "$L_EXTENDEDPRI",
"$subtract": [ 1, "$L_DISCOUNT" ] } , { "$add": [ 1, "$L_TAX" ] } ] } } ,
"avg_qty": { "$avg": "$L_QUANTITY" },
"avg_price": { "$avg": "$L_EXTENDEDPRI" },
"avg_disc": { "$avg": "$L_DISCOUNT" },
"count_order": { "$sum": 1 }
}
},
{ $sort: { "_id.L_RETURNFLAG": 1, "_id.L_LINESTATUS": 1 } } ] )
```

- Для варіанта 3 без вкладених документів:

```
db.getCollection("V3").aggregate([
{
"$match": { "L_SHIPDATE" : { "$lte" : newDate (1998 , 12 , 1 ) }}
},
{
"$group" : {
"_id" : { "L_RETURNFLAG" : "$L_RETURNFLAG" , "L_LINESTATUS" : "
$L_LINESTATUS" } ,
"sum_qty" : { "$sum" : "$L_QUANTITY" } ,
"sum_base_price" : { "$sum" : "$L_EXTENDEDPRI" } ,
165
"sum_disc_price" : { "$sum" :
{ "$multiply" : [ "$L_EXTENDEDPRI
" , { "$subtract" : [ 1 , "$L_DISCOUNT" ] ] } ] }
} ,
"sum_charge" : { "$sum" :
{ "$multiply" : [ "$L_extendedprice" , {
"$subtract" : [ 1 , "$L_DISCOUNT" ] } , { "$add" : [ 1 , "$L_TAX" ] } ] } } ,
"avg_qty" : { "$avg" : "$L_QUANTITY" } ,
"avg_price" : { "$avg" : "$L_EXTENDEDPRI" } ,
"avg_disc" : { "$avg" : "$L_DISCOUNT" } ,
"count_order" : { "$sum" : 1 }
}
} ,
{ $sort : { "_id.L_RETURNFLAG" : 1 , "_id.L_LINESTATUS" : 1 } } ] )
```

- Для варіанта 3 із вкладеними документами:

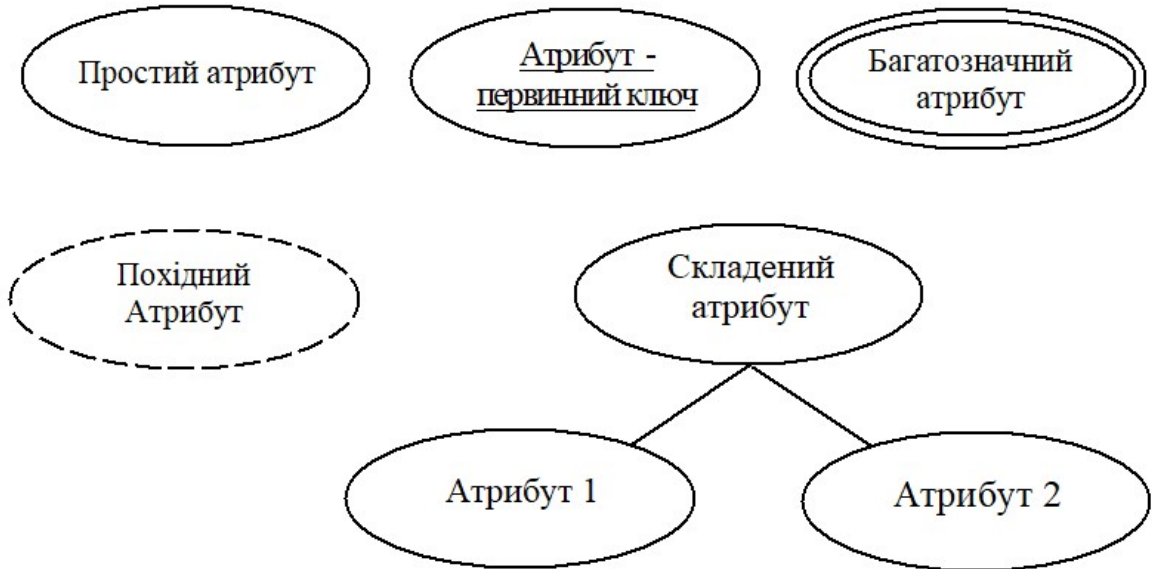
```
db.getCollection("V3").aggregate([
{
"$match" : { "Orders .
"}
},
{
"$group" : {
"_id" : { "L_RETURNFLAG" : "$Orders .
L_LINESTATUS" : "$Orders .lineitem.L_LINESTATUS" } ,
"sum_qty" : { "$sum" : "$Orders .lineitem.L_QUANTITY" } ,
"sum_base_price" : { "$sum" : "$Orders .lineitem.L_EXTENDEDPRI
" } ,
"sum_disc_price" : { "$sum" :
```

## ДОДАТОК Б

## ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

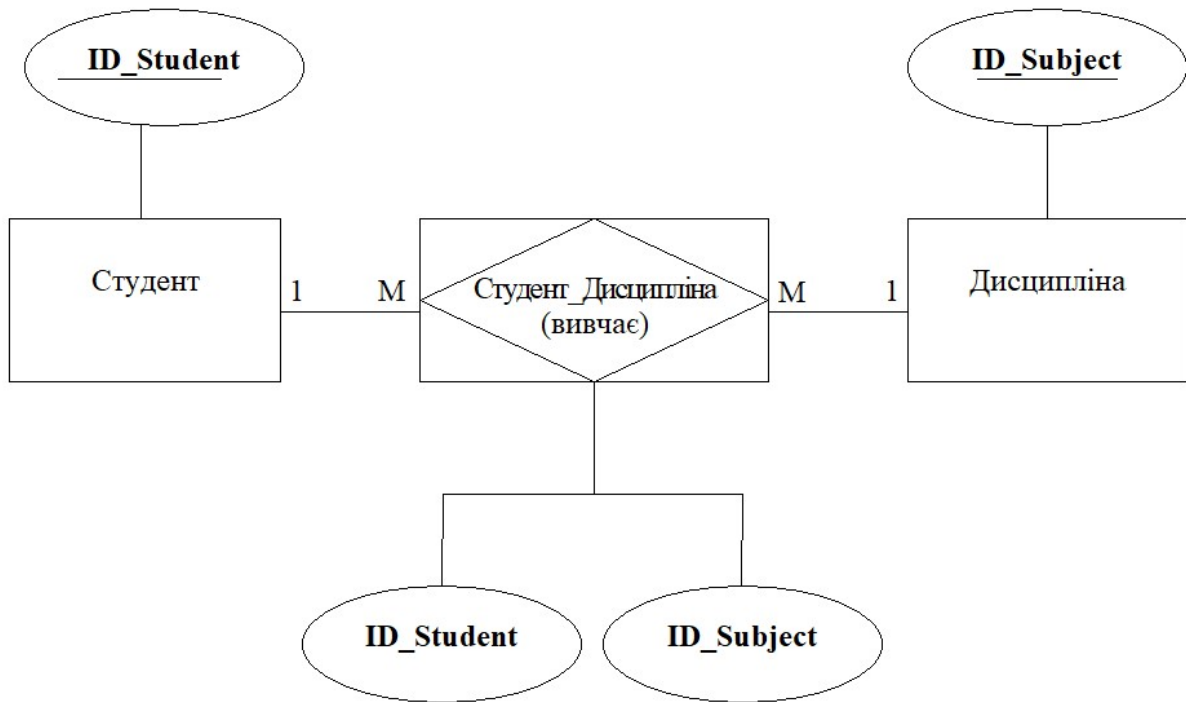
Ім'я файлу	Опис
Пояснювальні документи	
Магістерська_робота_ЖежельАК_12 1м-21-1.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Магістерська_робота_ЖежельАК_12 1м-21-1.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF.
Програма	
Program.zip	Архів. Містить код програми і відкомпільовану програму.
Презентація	
Презентація_ЖежельАК_121м-21-1. ppt	Презентація кваліфікаційної роботи.

ДОДАТОК В  
ПОНЯТТЯ ER-МОДЕЛІ. ПОНЯТТЯ СУТНОСТІ (ENTITY).  
АТРИБУТИ. ВИДИ АТРИБУТІВ.





## ДОДАТОК Г

**ER-МОДЕЛЬ. ПОНЯТТЯ ЗВ'ЯЗКУ. ПОТУЖНІСТЬ ЗВ'ЯЗКУ.  
ТИПИ ЗВ'ЯЗКІВ.**

**ДОДАТОК Д**  
**ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ. ПОНЯТТЯ ПРЕДМЕТНОЇ**  
**ОБЛАСТІ.**

