МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ
БАЗИ ДАНИХ

друге видання, доповнене

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів, які навчаються
за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані
технології»*

Київ
КПІ ім. Ігоря Сікорського
2022

Проєктування інформаційних систем : Бази даних: [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / КПІ ім. Ігоря Сікорського ; уклад.: А. І. Жученко, Л. Д. Ярощук. – 2-ге вид., допов. – Електронні текстові данні (1 файл: 2,8 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 166 с.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 3 від 01.12.2022 р..)
за поданням Вченої ради інженерно-хімічного факультету (протокол № 9 від 26.09.2022 р.)

Електронне мережне навчальне видання

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

БАЗИ ДАНИХ

друге видання, доповнене

Укладачі: *Жученко Анатолій Іванович,* д-р техн. наук., проф.
Ярощук Людмила Дем'янівна, канд. техн. наук, доц.

Відповідальний редактор *Панов Є. М..,* д-р техн. наук., проф.

Рецензент: *Волощук В. А.* д-р техн. наук., проф.

Запропоноване навчальне видання розкриває основні положення інформаційних систем, зокрема баз даних: описує їх види, моделі даних, розглядає способи створення *ER* – діаграм та їхню роль в інфологічному моделюванні, властивості реляційних баз даних та алгоритм їх створення. У цьому виданні наведені завдання для самостійної роботи студентів та приклади тестів для перевірки їх знань. Посібник містить багато прикладів моделювання даних та застосування баз даних для предметної області «Автоматизація технологічних процесів».

Посібник є другим, доповненим виданням навчального посібника:

Жученко, А. І. Основи проектування баз даних [Електронний ресурс] : навчальний посібник для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / А. І. Жученко, Л. Д. Ярощук ; НТУУ «КПІ». – Електронні текстові дані (1 файл: 2, 50 Мбайт). – Київ : НТУУ «КПІ», 2015. – 158 с.

Посібник призначений для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» усіх форм навчання.

© А. І. Жученко, Л. Д. Ярощук, 2022
© КПІ ім. Ігоря Сікорського, 2022

ПЕРЕДМОВА

Інформаційні системи пов’язані з різними аспектами життя кожної людини. Їх застосовували вже тоді, коли комп’ютерів ще не було. Розвиток цих систем і перетворення їх в перспективні об’єкти проектування та дослідження залежить від ефективності технічних комп’ютерних засобів і програмного забезпечення.

Існують різноманітні визначення для поняття “Інформаційні системи”, зокрема згідно ДСТУ 2392-94, “Інформаційна система – комунікаційна система, що забезпечує збирання, пошук, оброблення та пересилання інформації”.

При розробці сучасної інформаційної системи треба передбачити створення і використання баз даних.

Бази даних – один з найбільш поширених способів зберігання інформації. Базуючись на новітніх засобах обчислювальної техніки, вони є могутнім засобом розв’язання науково-технічних задач, тому їх широко використовують у сучасних системах керування технологічними процесами і виробництвами.

Однією з важливих складових систем штучного інтелекту є база знань. Її формування потребує навичок у створенні спеціальних комп’ютерних баз даних.

Дисципліна “Проєктування інформаційних систем-1. Бази даних” дає змогу студентам ознайомитися з теорією та практикою створення і використання баз даних та з системою керування базами даних *MS Access*. Дисципліна належить до циклу нормативних освітніх компонентів професійної підготовки.

При вивченні дисципліни особливу увагу приділено розгляду тих предметних областей, які притаманні саме спеціальності “Автоматизація та комп’ютерно-інтегровані технології”. Бази даних відіграють зараз роль засобу для обслуговування систем керування.

Їх використовують, зокрема, для визначення режимних параметрів технологічних об’єктів керування при зміні властивостей сировини, для створення специфікацій на технічні засоби автоматизації (ТЗА) при виконанні проектних робіт, для обліку технічних засобів автоматизації на виробництві тощо. Бази даних для зазначеної спеціальності не передбачають поки що складної структури і значної кількості записів, тому для засвоєння теорії їхнього проєктування і отримання навичок у їх створенні розглянуто систему *MS Access*, як найбільш доступну і поширену. Знання, отримані при вивченні цієї дисципліни, допоможуть майбутньому фахівцю у разі потреби перейти до більш потужних систем керування базами даних.

Метою вивчення дисципліни є отримання необхідних знань та вмінь для роботи з базами даних та базами знань.

В результаті вивчення цієї дисципліни студенти повинні знати:

- типи моделей знань та даних;
- способи створення інформаційно-логічних моделей предметних областей;
- можливості *MS Access*.

Після вивчення цієї дисципліни студенти повинні вміти:

- досліджувати предметні області і вибирати відповідні моделі даних;
- створювати та використовувати реляційні бази даних і різноманітні їх об′єкти.

У запропонованому навчальному посібнику розглянуто питання, пов’язані з етапами проєктування баз даних.

Для прикладів застосування теоретичних положень підібрано матеріал, пов’язаний як з поточною діяльністю студента – навчанням так і з майбутньою спеціальністю – ”Автоматизація та комп’ютерно-інтегровані технології”.

Посібник є другим, доповненим виданням навчального посібника

Жученко А. І. Основи проектування баз даних [Електронний ресурс] : навчальний посібник для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / А. І. Жученко, Л. Д. Ярощук ; НТУУ «КПІ». – Електронні текстові дані (1 файл: 2, 50 Мбайт). – Київ : НТУУ «КПІ», 2015. – 158 с.

У доповненому виданні автори більше уваги приділили поняттю інформаційних систем і їхньому місцю в предметній області "Комп'ютерні технології", розширили матеріал з нормалізації відношень та створення зв'язків між сутностями типу «багато до багатьох».

1. ОСНОВНІ ПОНЯТТЯ ПРО БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ

1.1. Поняття про інформацію, дані та базу даних

Розглянемо спочатку поняття “***інформація***” та “***дані***“, які лежать в основі будь-якого курсу з інформаційних технологій і курсу ”Бази даних“, зокрема. У даний час не існує єдиного визначення інформації як наукового терміну, наведемо найбільш уживане визначення.

Інформація – це будь-які відомості про подію, процес чи об’єкт незалежно від форми їх подання (лат. *Informatio* – роз'яснення, виклад, обізнаність).

Відомості можуть бути невпорядкованими, тоді їх важко аналізувати для прийняття рішень. Опрацювання таких відомостей дозволяє їх структурувати і отримати так звані ”*дані*”.

Дані (*Data***)** – це інформація, подана у формі, зручній для зберігання, передачі та обробки людиною чи технічними засобами.

Структурування – це укладання угод про способи подання даних. Неструктурованими даними називають такі, що записані, наприклад, у текстовому файлі.

Приблизне уявлення про те, що таке база даних, можна одержати, звернувшись до записника, якщо у ньому розташовані однотипні записи, що містять відомості про прізвища, імена, телефони, адреси людей. Наведемо визначення бази даних.

База даних (БД, *DB* – *Database*) – це пойменована структурована сукупність даних, що відображає стан об'єктів і їх відношення в даній предметній області.

Предметна область (ПО, *Subject Domain*) – досліджувана частина (велика чи мала) реального світу. Назвемо декотрі предметні області:

- **АВТОМАТИЗАЦІЯ,**
- **СИСТЕМА КЕРУВАННЯ,**
- **ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ,**
- **ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД,**
- **ПЕДАГОГІЧНІ КАДРИ**.

Бачимо, що предметна область може бути більш і менш широка (порівняйте перші три, а потім —дві останні).

Можна дати ще й інше визначення БД.

База даних – це сукупність взаємопов'язаних даних, які зберігаються разом. Ця сукупність має таку мінімальну надмірність, що припускає використання даних оптимальним чином для однієї або декількох задач.

Створюючи БД, розробники намагаються впорядкувати (структурувати) інформацію за різноманітними ознаками для того, щоб швидко отримувати вибірку з довільним сполученням ознак.

Бази даних, що створюються в пристроях пам'яті комп'ютера, можуть містити мільйони однотипних записів, які зберігають сукупність взаємопов'язаних відомостей про ті чи інші об'єкти.

Уперше термін “база даних” з'явився у 1962 р. Не кожен набір даних може бути названий базою даних.

База даних має такі властивості:

- інтегрованість, спрямована на розв'язування загальних задач;
- модельність (тобто структурованість), що відображає вибрану ПО;
- взаємопов'язаність даних;
- незалежність опису даних від прикладних програм.

Робота з базами даних обумовила створення нової інформаційної технології, що ґрунтується на таких двох принципах:

- ***інтеграція даних***, коли всі дані накопичують і зберігають централізовано, створюючи динамічно (у реальному часі) поновлювану модель предметної області;

- ***незалежність прикладних програм від даних***, тобто відділення логічної уяви користувачів БД про предметну область від фізичного подання даних у пам'яті ЕОМ (так звані *логічна* та *фізична незалежність даних*).

1.2. Система керування базами даних. Автоматизовані інформаційні системи

Згідно з цією технологією створюють один для всіх завдань блок даних (*база даних*) і використовують одну програму – *систему керування базами даних* (СКБД), призначену для маніпулювання даними на фізичному рівні і спілкування з користувачем [1 – 11].

Уведення СКБД відокремлює логічну структуру даних (тобто уявлення користувача про організацію даних) від фізичної структури даних у пам'яті ЕОМ.

Таким чином, **СКБД** – це сукупність мовних і програмних засобів загального чи спеціального призначення для створення, ведення і сумісного використання БД багатьма користувачами. Відповідні абревіатури:

англійська – ***DBMS*** – "*Database Management System*",

німецька – ***DBVS*** – "*Datenbankverwaltungssystem*".

У сучасному світі основними є ***автоматизовані*** (комп'ютерні) БД**,** які називають також ***автоматизовані інформаційні системи***, ***AIC*** (*Automated information System*).

Швидкий пошук необхідних записів і подання їх у зручній формі, проведення необхідних розрахунків, поповнення, видалення і корекцію даних виконують у них за допомогою ЕОМ.

Узагальнену структуру АІС наведено на рис. 1.1.

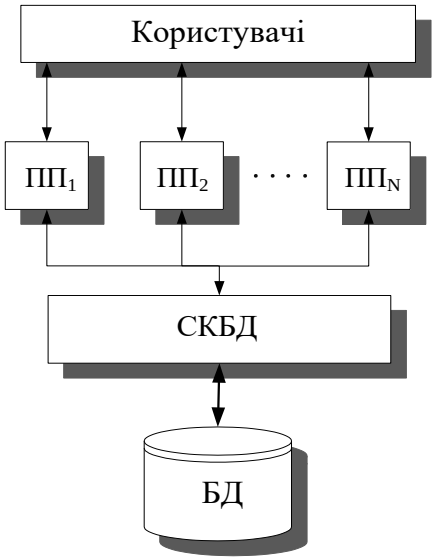


Рис.1.1. Узагальнена структурна схема АІС
($ПП_1$-$ПП_N$ – прикладні програми, які виконує АІС)

Користувачами БД можуть бути різноманітні прикладні програми, програмні комплекси, а також спеціалісти різних предметних областей.

Користувачів БД часто розділяють на дві групи – внутрішні та кінцеві.

Внутрішніми користувачами вважають фахівців, які створюють та обслуговують АІС. До них належать адміністратори БД, адміністратори функціональних підсистем, системні та прикладні програмісти.

Такі користувачі створюють і розвивають інформаційну систему, а також підтримують правильне її функціонування.

Кінцеві користувачі – це фахівці предметної області, заради задач яких створюють базу даних.

Адміністратор БД очолює її розробку і експлуатацію. На етапі проектування він розробляє концепцію БД, керує створенням її програмного супроводу. На етапі експлуатації БД він відповідає за захист даних, забезпечення коректного групового доступу, ефективне використання ресурсів інформаційної системи.

Адміністратори функціональних підсистем разом з адміністратором БД визначають алгоритми обробки даних.

Системні програмісти забезпечують функціонування і (або) розширення СКБД – генерують СКБД, аналізують її роботу, створюють до неї нові модулі за замовленням адміністратора.

Прикладні програмісти розробляють на мові програмування СКБД прикладні програми для кінцевих користувачів АІС.

Кінцеві користувачі бувають двох типів – опосередковані та безпосередні.

Опосередковані кінцеві користувачі не звертаються до комп'ютера безпосередньо, вони формулюють свої задачі службі адміністратора. Відповіді для таких користувачів надходять у паперовому вигляді іноді з результатами аналізу певних експертів.

Безпосередні кінцеві користувачі працюють з АІС в інтерактивному режимі.

Саме кінцевих користувачів будемо далі називати *користувачами*.

При роботі з персональним комп'ютером (ПК) користувач може бути адміністратором, системним і прикладним програмістом в одній особі.

Програми, за допомогою яких працює користувач, називають ***прикладними програмами*** (ПП, *Application Program*) або ***додатками***.

Зміни фізичної організації даних сприймаються СКБД і не впливають на прикладну програму. Зміна логіки прикладної програми не вимагає реорганізації і зміни механізму доступу до фізичних даних.

Відмінність роботи користувачів з АІС і просто з набором файлів показано на рис. 1.2. Нехай у межах однієї установи існують різні відділи, які повинні розв'язувати притаманні їм задачі. У той же час, ці задачі пов'язані загальною предметною областю, складом працівників, приміщеннями, матеріальними цінностями тощо.

За відсутності СКБД кожний користувач (підрозділ організації) мав би використовувати свій набір даних і свої прикладні програми.

Отже, зміна даних одних підрозділів не призводить автоматично до змін тих самих даних інших підрозділів. Застосування СКБД дозволяє створити один набір даних, з яким можуть працювати усі зацікавлені фахівці. Його легше оновлювати і обслуговувати.

СКБД для персональних комп'ютерів (ПК), задовольняючи всім вимогам теорії БД, мають порівняно не складну архітектуру, ними легко оволодіти, забезпечені вбудованою системною інтерактивною допомогою. Сьогодні на ринку програмного забезпечення представлено значну кількість СКБД для ПК.

Автоматизованим інформаційним системам дають різне тлумачення, у цьому посібнику використано наступне: "***АІС – це сукупність програмних та апаратних засобів для зберігання та/або керування даними та інформацією та виконання обчислень***". Воно дуже близьке до визначення СКБД, тому у багатьох джерелах наводять класифікацію таких систем саме як класифікацію СКБД.

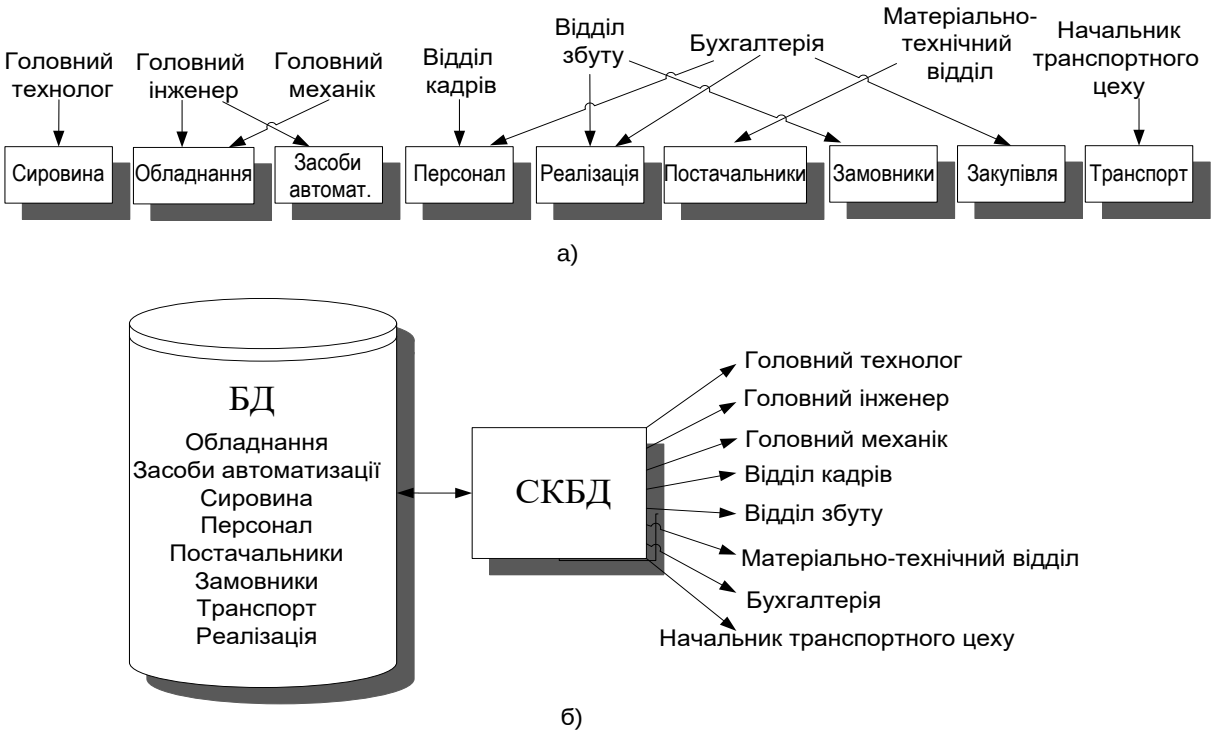


Рис.1.2. Схеми інформаційних систем
а) без централізованого зберігання даних (система файлів); б) з централізованим зберіганням даних (система бази даних)

1.3. Структура та властивості СКБД

СКБД, як пакет програм, повинна мати такі функції:

- забезпечувати операції створення і маніпулювання логічними даними (вибір, вставлення, оновлення, видалення і т. ін.) і одночасне виконання цих же операцій над фізичними даними;

- забезпечувати захист і цілісність (узгодженість) даних при колективному режимі користування базою даних (питання розглянуто у п. 3.4).

У зв’язку з цим СКБД містить звичайно компоненти, які виконують наступні функції:

- визначення та модифікація структури БД;
- діалогова взаємодія з кінцевим користувачем (для введення і корекції інформації, введення запитів і отримання відповідей):
    - генерування звітів;
    - створення додатків:
    - створення екранних форм;
    - створення і підключення програм;
- настроювання параметрів системи, її генерація і обслуговування;
- зв'язок із іншими СКБД і пакетами.

СКБД надає прикладній програмі інтерфейс з базою даних та засоби безпосереднього доступу до неї. Таким чином, стає зрозумілим, що СКБД є центральною складовою у автоматизованій інформаційній системі.

СКБД використовує два типу мов доступу до бази даних:

- мову опису даних (МОД) – високорівневу не процедурну мову декларативного типу, призначена для опису логічної структури даних;

- мову маніпулювання даними (ММД) – сукупність конструкцій, які забезпечують виконання основних операцій з даними у прикладних програмах: уведення, модифікування і вибірку даних по запитах.

Ці мови у різних СКБД можуть різнитися. Найбільше поширення отримали дві стандартизовані мови доступу до даних:

- мова запитів за зразком, *QBE – Query By Example*, характеризується властивостями мови маніпулювання даними;

- структурована не процедурна мова з відносно невеликим (близько 30) набором команд, *SQL – Structured Query Language*, має властивості мов обох типів.

На рис. 1.3. наведено архітектуру АІС. За допомогою мови опису даних створюють описи елементів (не саму БД, а її опис), груп та записів даних, а також взаємозв'язки між ними, які, як правило, задаються у вигляді таблиць. У залежності від конкретної реалізації СКБД мову опису даних підрозділяють на мову опису схеми бази даних (МОС) та мову опису підсхем бази даних (МОП).

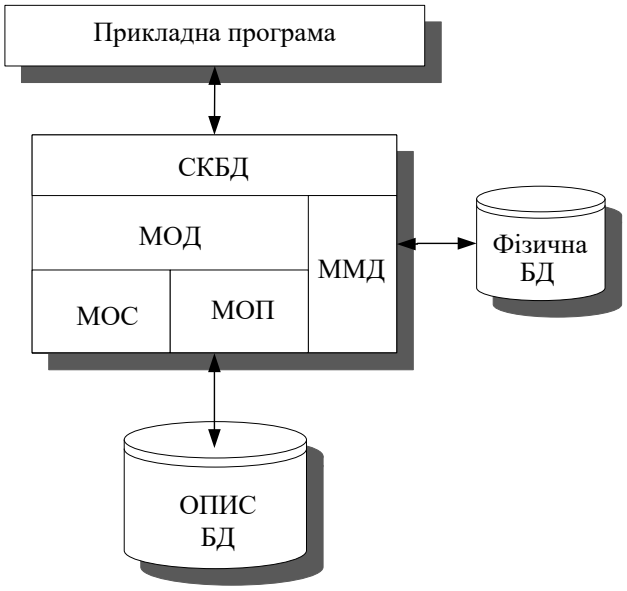


Рис. 1.3. Архітектура АІС

Фактична структура фізичного зберігання даних відома тільки СКБД.

З метою забезпечення зв'язків між СКБД і програмами користувачів (що особливо важливо при мультипрограмному режимі роботи операційної системи) система містить специфічну складову - резидентний модуль системи керування базами даних. Цей модуль значно менший від інших складових системи за обсягом, тому на час функціонування АІС він може постійно знаходитись в основній пам'яті ЕОМ та забезпечувати взаємодію всіх складових СКБД і програм, які до неї звертаються.

Описану структуру мають усі СКБД, а різняться вони обмеженнями та можливостями по виконанню відповідних функцій. Отже, процес порівняння і оцінки таких систем для одного конкретного застосування зводиться до співставлення можливостей наявних СКБД з вимогами користувачів.

Ефективність конкретної СКБД визначається наявністю і зручністю засобів для виконання необхідних операцій.

1.4. Класифікація АІС

У спеціальній літературі існують різні ***класифікації автоматизованих баз даних***. У залежності від задач та точок зору фахівці виділяють ті чи інші критерії і розподіляють пріоритети між ними. Зупинимось на одній з класифікацій, згідно з якою АІС (а також СКБД) розрізняють за такими критеріями:

1) за призначенням;

2) за мовами;

3) за локалізацією;

4) за доступом до даних;

5) за схемою додаткової обробки;

6) за структурами даних.

Розглянемо детальніше кожний з цих критеріїв.

За **призначенням** АІС розділяють на *документальні, фактографічні* та *змішані***.**

***Документальні* (*дескрипторні*,** ДАІС) – це системи зорієнтовані на обробку та зберігання документа (порівняно великої за розміром послідовності символів), внутрішню структуру якого система майже повністю ігнорує, тобто він неподільний (атомарний) з точки зору системи. Споживачем результатів пошуку виступає, як правило, кінцевий користувач.

Дескрипторні АІС історично були першими. Спочатку їх мовою була нічим не обмежена природна мова. Перші ДАІС призначалися для пошуку книг та документів у бібліотеках і великих сховищах, тому їх називали ще *документографічними*. Основним елементом інформаційного простору ДАІС була анотація або реферат книги, документа, явища чи об'єкта. Реферат повинен відображувати ті риси, які цікавлять користувача. У рефераті треба було виділити слова чи словосполучення, які повинні однозначно відповідати повному опису об'єкта. Таких слів повинно бути небагато, їх називають *ключовими словами* або *дескрипторами.* Запит для ДАІС можна сформулювати у вигляді переліку дескрипторів, який характеризує потрібний реферат, тобто об’єкт.

Фактографічні системи оперують фактами (даними) різних типів, що зв'язані у більш чи менш складні структури.

Змішані системи мають риси обох вищеназваних варіантів. Переважну більшість сучасних систем для ПЕОМ слід віднести до категорії змішаних.

За мовами розрізняють *замкнені* системи, системи *з базовою мовою* та *змішані.*

Замкнені системи самостійно забезпечують користувача всіма необхідними засобами як для локалізації даних, так і для їх постпошукової чи передпошукової обробки. Недоліком таких систем є те, що в них відсутні або малоефективні засоби для розробки надбудов – проблемно-орієнтованих комплексів.

Замкнена СКБД є інтерпретатором (або компілятором) спеціальної мови запитів до даних, здатним у діалоговому або програмному режимі виконувати команди керування БД. Наведемо назви деяких відомих фірм та їхніх розробок (жирне – назва фірми): ***Borland***: *dBase* (*xBase*), *Foxbase*, *Paradox*, *Interbase*; ***IBM***: *DB*2, *IMS*, *Informix* (колишньої фірми ***Informix***), ***MicroSoft***: *VisioFoxPro*, *Access*; ***Oracle***: *Oracle*, *MySQL*; ***Sybase***: *ASA*, *ASE*, *Sybase System* і т. ін.

З переліком СКБД можна ознайомитися за посиланням https://uk.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85

Системи з базовою мовою передбачають взаємодію користувача з СКБД із середовища якоїсь іншої універсальної мови програмування, де і виконують більшість перетворень даних. Для цього в універсальну мову програмування (*C++*, *C#*, *Java*, *JavaScript*, *Python*, *PHP*, *Ruby*, *Perl*, *Pascal*, *Delphi*, *LISP* і т. ін.) уводять засоби опису і маніпулювання даними для створення прикладних програм.

Такий підхід зручний для розробки різного роду систем як надбудов над СКБД, оскільки дає можливість створювати високоефективні програми постпошукової обробки даних.

Змішані системи передбачають наявність обох попередніх підходів і є найбільш поширеними на сьогодні.

За **локалізацією** системи розділяють на *локальні* (*local Database, LDB*) та *розподілені* (РБД, *distributed Database, DDB*).

Локальність передбачає розташування всього програмного забезпечення і даних на одному комп'ютері. Комп'ютер з локальною (***централізованою***) СКБД і даними може знаходитись як окремо, так і в складі певної мережі.

У централізованій АІС легше організувати безпеку, цілісність і несуперечливість інформації при оновленні даних. Однак при нарощуванні структури БД і при експлуатації її організаціями, розташованими на різних територіях, виникають наступні складнощі:

- значний обсяг даних, що передаються;
- недостатня надійність передачі інформації;
- зниження продуктивності програмних засобів;
- зростання витрат на супровід та модернізацію робіт.

Розподілена АІС складається з декількох частин (файлів, відношень), які перетинаються або дублюються та зберігаються у пристроях пам'яті різних ЕОМ комп'ютерної мережі. Існують різні стратегії рознесення по мережі програмного забезпечення і даних. Зв'язок між складовими цієї бази може бути функціональними або через копії одного і того самого файлу. У користувачів повинно бути відчуття роботи з централізованою АІС. Робота з такими БД забезпечується системами керування розподіленою базою даних (СКРБД).

Децентралізація дозволяє досягти наступних результатів:

-прискореної обробки інформації внаслідок розподілення навантаження на різні ЕОМ;

-оптимального використання даних на місцях при виконанні віддалених запитів;

-зменшення витрат на розробку та супровід БД;

-спрощення управління базою даних.

Сіткові варіанти СКБД мають високу потужність та працюють на всіх типах персональних комп'ютерів. Розмір і складність баз даних обмежуються тільки можливостями операційної системи комп'ютера і обсягом доступної пам'яті.

За ***способом доступу до даних*** (***за архітектурою***) сіткові БД розділяють на такі, як *файл - сервер* та *клієнт – сервер*.

Сервером (*Server*) певного ресурсу у комп'ютерній мережі називають комп'ютер (програму), який керує ресурсом, а *клієнтом* – комп'ютер (програму), який використовує цей ресурс. Ресурсом мережі можуть бути бази даних, файлові системи, служби друку, поштові служби. Тип сервера визначається видом ресурсу, наприклад, якщо ресурсом виступають бази даних, то сервер називають *сервером бази даних*.

Файл – сервер (*File – Server*) передбачає наявність однієї ЕОМ як центральної (сервер), у якій зберігають БД. Решта ЕОМ мережі виконують функції робочих станцій (*Workstation*), з якими працюють користувачі. Файли БД після запиту користувачів передають на робочі станції на обробку. Ця система історично з'явилася першою.

Недоліком такого варіанта архітектури є висока інтенсивність передачі даних. Оскільки файли БД передаються повністю, то дуже часто користувачу надходять надлишкові дані, незалежно від того, скільки записів з бази йому треба.

На рис. 1.4 подано схему сіткової БД архітектури файл – сервер.

Клієнт – сервер (*Client – Server*) передбачає використання центральної ЕОМ не тільки для зберігання, але і для основної обробки даних. Запит користувача викликає дії самого сервера. Сам сервер виконує пошук і отримання потрібних даних, які потім передають користувачу. Отже, вибрані дані, а не файли (як у першому типі архітектури), надходять від сервера до користувача (клієнта). Специфікою цієї архітектури є використання мови запитів *SQL*. Позитивним у такому способі є помітно менший обсяг даних, що передаються по мережі.

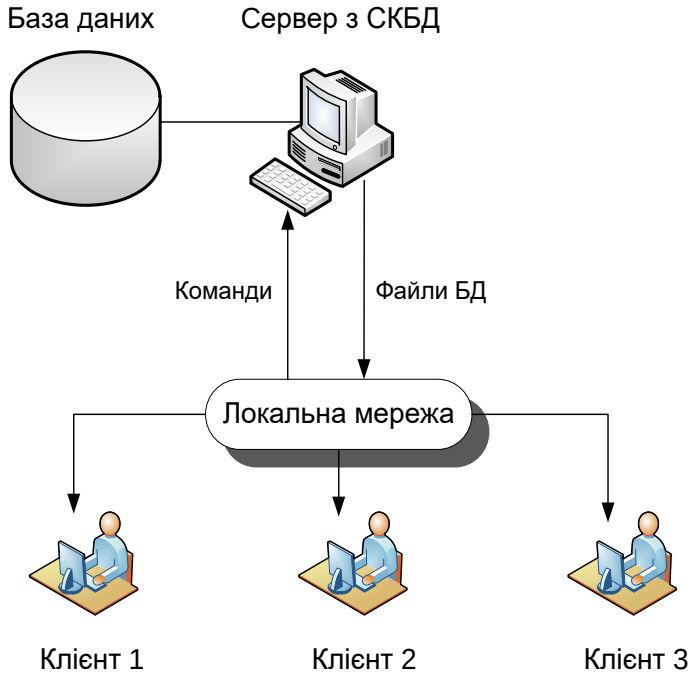


Рис. 1.4. Схема сіткової БД архітектури файл – сервер

На рис. 1.5 подано схему сіткової БД архітектури клієнт – сервер.

Рис. 1.5. Схема сіткової БД архітектури клієнт – сервер

Класифікації автоматизованих БД за останніми двома пунктами добре ілюструє рис. 1.6 [8].

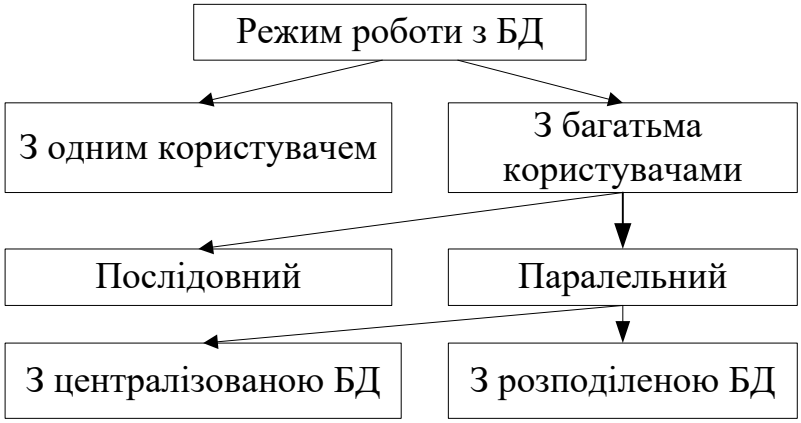


Рис.1.6. Режими роботи з базами даних

За **схемою додаткової обробки** системи розділяють на системи з *попередньою* та *постобробкою* обробкою, що означає виконання певних математичних операцій з даними до або після їх зберігання у БД.

Головним призначенням будь-якої системи баз даних є підтримка функцій локалізації даних, що зберігаються, але дуже важливою властивістю, що може значно підняти інтерфейсний рівень системи, є наявність постобробки даних після їх локалізації в базі даних, чи попередньої обробки.

За **структурами даних** розглядають системи *ієрархічні, сіткового типу* та *реляційні***.**

Структура даних, яку підтримує СКБД, це важливий фактор, що впливає як на смислову потужність, так і на ефективність її функціонування. Існуючі структури детально розглянуто у розділі «Моделі даних».

1.5. Інформаційні системи. Основні типи, функції та компоненти

Інформаційна система (**ІС,** *informative System*)– це система, що включає інформаційні, програмні, технічні, лінгвістичні, організаційні засоби збору, накопичення, передачі, пошуку, обробки даних. ІС призначені для збору, накопичення, передачі, пошуку, обробки даних.

В українському законодавстві використано декілька тлумачень ІС [https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0#cite_note-3],

наведемо їх:

- Інформаційна система — автоматизована система, комп'ютерна мережа або система зв'язку;

- Інформаційна система — організаційно-технічна система обробки інформації за допомогою технічних і програмних засобів;
- Інформаційна система — система, призначена для одержання, обробки, зберігання, відображення та/або реєстрації даних про технічний стан конструкцій, систем, елементів, їх властивості та/або функціонування.

Інформаційні системи є складовими частинами мереж мобільного зв'язку та телебачення, вони мають місце у цифрових відеокамерах то фотоапаратах тощо. До інформаційної системи дані надходять від джерела інформації, їх надсилають на зберігання чи певного опрацювання у системі й потім передають споживачеві.

Типи інформаційних систем по значущості виконуваних функцій наступні:

- інформаційно – пошукові системи;

- системи обробки даних.

Тип інформаційної системи багато в чому зумовлює вимоги до організації її інформаційної складової. У табл. 1.1 наведено загальні властивості інформаційних систем.

Таблиця 1.1. Загальні властивості інформаційних систем

Назва	Особливості інформаційної складової	Функції
OLTP (*Online Transaction Processing*) Системи оперативної обробки даних	Інформаційну компоненту (базу даних) *OLTP* - систем дуже часто оновлюють. Тому вона не повинна містити надлишкового дублювання даних. Більшість інформаційних систем – *OLTP*-системи.	- оновлення; - обробка даних
OLAP (*Online Analytical Processing*) Системи оперативного аналізу даних	Інформаційну компоненту *OLAP* - систем, як правило, не оновлюють (її називають сховищем даних). Тому тут можливе надлишкове дублювання даних. Системи прийняття рішень будують у вигляді *OLAP* – систем.	- накопичення; - аналіз і обробка даних

Для побудови інформаційних систем застосовують різні інструментальні засоби, принципи і методології. Якщо при створенні і функціонуванні інформаційної системи використовують БД та СКБД, то до такої ІС інколи застосовують термін “банк даних”.

Банк даних (БнД, *Bank of Data, Data Bank*) – це система спеціальним чином організованих даних – баз даних (однією чи декількома), програмних, технічних, мовних та організаційно-методичних засобів, які призначені для забезпечення централізованого накопичення і колективного багатоцільового використання даних.

Програмні засоби містять систему управління базами даних. Технічні засоби – це персональний комп'ютер або мережа.

Лінгвістичні засоби представлені мовними засобами СКБД: – мовою опису даних та мовою маніпулювання даними.

Організаційні засоби – засоби адміністрування бази даних – (забезпечення цілісності даних, супровід програмних засобів і т. ін.)

Поняття інформаційної системи ширше, ніж поняття банку даних База даних – це лише одна з компонент банку даних.

На рис. 1.7 наведено структурну схему банку даних [2]. Словник (або каталог) даних використовують для централізованого накопичення і описи ресурсу даних. Він містить опис ПО, відомості про структуру БД, про зв'язки між елементами БД. Його можна розглядати як частку самої бази даних.

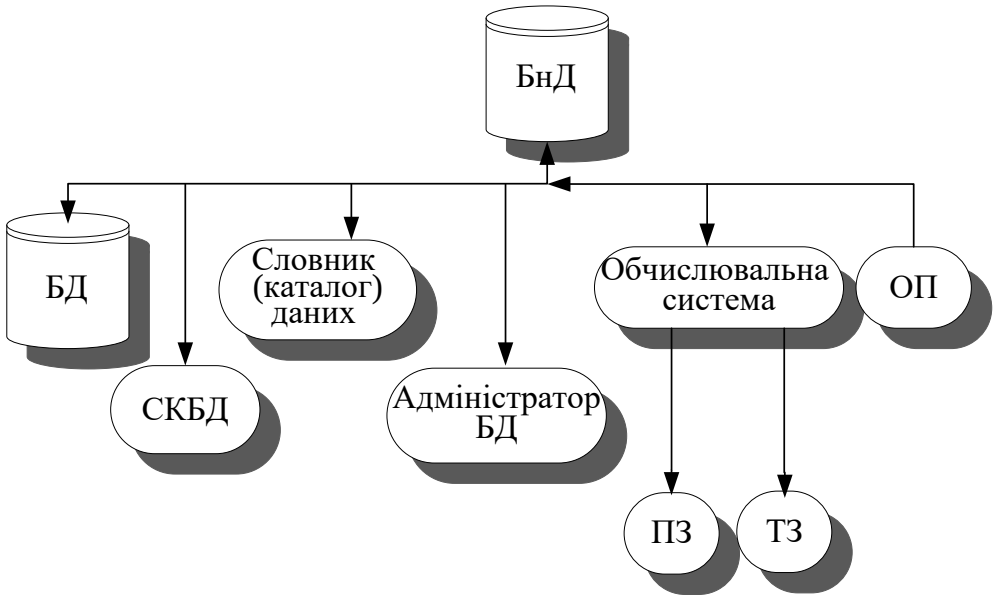


Рис. 1.7. Структурна схема банку даних:
БД – одна чи декілька БД; ПЗ, ТЗ – програмні та технічні засоби обчислювальної системи; ОП – обслуговуючий персонал

Сховище даних (*Data Warehouse*) — спеціально створена предметно – орієнтована база даних призначена для підготовки звітів і бізнес-аналізу з метою підтримки ухвалення рішень в організації. Сховище даних будують на основі систем керування базами даних і систем підтримки ухвалення рішень.

Дані, що надходять у сховище даних, як правило, доступні тільки для читання. Дані з *OLTP* – системи копіюють у сховище даних так, щоб створення звітів і *OLAP* – аналіз не використовував ресурси транзакційної системи і не порушував її стабільності. Зазвичай, дані завантажують у сховище з певною періодичністю, тому актуальність даних може декілька відставати від *OLTP* – системи.

Сховища даних організовують за такими принципами**:**

- проблемно – предметна орієнтація, тобто дані об’єднують у категорії і зберігають відповідно до областей, які вони описують, а не за додатками, які вони використовують;
- інтегрованість, коли. дані об’єднані так, щоб вони задовольняли всім вимогам підприємства в цілому, а не одній тільки функції бізнесу;
- некоректованість, вона означає, що дані в сховищі даних не створюють, а вони надходять із зовнішніх джерел, їх не коректують і не видаляють;
- залежність від часу, дані в сховищі точні і коректні тільки у тому випадку, коли вони відповідають деякому проміжку або моменту часу.

Існують два архітектурні напрями дизайну сховищ – нормалізовані сховища даних і сховища з вимірами.

У *нормалізованих сховищах* дані знаходяться в предметно – орієнтованих таблицях третьої нормальної форми (детальніше – у п. 3.6).

Нормалізовані сховища характеризуються як прості у створенні й управлінні, недоліки нормалізованих сховищ – велика кількість таблиць як наслідок нормалізації, тому для отримання певної інформації потрібно робити вибірку з багатьох таблиць одночасно. Це погіршує продуктивність системи.

Сховища *з вимірами* використовують схему “зірка” або схему “сніжинка”. При цьому в центрі «зірки» знаходяться дані (**Таблиця фактів**), а виміри утворюють промені зірки.

Різні таблиці фактів спільно використовують таблиці вимірів, що значно полегшує операції об'єднання даних з декількох наочних таблиць фактів (приклад – факти продажів та постачань товару). Таблиці даних з відповідними вимірами утворюють архітектуру “шина”.

Виміри часто створюються в третій нормальній формі, зокрема, для протоколювання зміни у вимірах. Основною перевагою сховищ з вимірами є простота і зрозумілість для розробників і користувачів, також, завдяки ефективнішому зберіганню даних і формалізованим вимірам, полегшується і прискорюється доступ до даних, особливо при складних аналізах. Основним недоліком є складніші процедури підготовки і завантаження даних, а також управління і зміна вимірів даних.

Джерелами даних можуть бути: традиційні системи реєстрації операцій, окремі документи, набори даних.

При роботі зі сховищами виконують такі операції з даними:

- витягання – переміщення інформації від джерел даних до окремої БД, приведення їх до єдиного формату;
- перетворення – підготовка інформації до зберігання в оптимальній формі для реалізації запиту, необхідного для ухвалення рішень;
- завантаження – розташування даних у сховищі виконується атомарно, шляхом додавання нових фактів або коректуванням тих, що існують;
- аналіз – *OLAP*, *Data Mining*, зведені звіти;
- подання результатів аналізу.

Запитання та завдання для самостійної роботи

1. Що таке база даних ?

2. Назвіть властивості БД

3. Що таке система керування базами даних ?

3. Зобразіть узагальнену структуру АІС.

5. Які можливості повинна мати АІС?

6. Яку загальну архітектуру має АІС?

7. Наведіть загальну класифікацію АІС.

8. Як класифікують АІС за технологією обробки даних?

9. Як класифікують АІС за способами доступу до даних?

10. Наведіть 6 прикладів предметних областей, які відповідають Вашій спеціальності.

11. Зробіть пошук в *Internet* і визначте, які фірми створюють СКБД, запишіть назви цих СКБД.

2. МОДЕЛІ ДАНИХ

2.1. Поняття про модель даних

Моделлю даних (*Model of Data*) називають спосіб відображення об'єктів предметної області, їх властивостей і взаємозв'язків.

Предметна область може складатися з декількох частин (систем, фрагментів тощо). Наприклад, ПО ”**ПРОМИСЛОВЕ ПІДПРИЄМСТВО**” можна розділити на такі частини:

- виробництво (технологічна частина);
- працівники;
- приміщення;
- транспорт.

Наступним прикладом ПО може бути ”**ТЕХНОЛОГІЧНИЙ ОБ’ЄКТ**”, її можна розділити наступним чином:

- обладнання;
- технологічний процес;
- вхідні матеріальні потоки;
- вихідні матеріальні потоки;
- режимні параметри.

У п. 1.1 було сказано, що предметні області бувають більш і менш об’ємними (широкими). Наведені приклади ілюструють це ще раз.

У свою чергу, кожна з частин предметних областей складається з множини певних об’єктів та процесів, до яких причетні численні користувачі з власними виробничими задачами.

Якщо модель даних не буде правильною, то створена на її основі БД, не зможе ефективно обробляти інформацію.

2.2. Класифікація моделей даних. Трирівнева схема подання даних

У процесі створення БД прийнято розглядати предметну область послідовно у вигляді таких подань:

- у тому виді, як ПО реально існує;
- у тому виді, як цю ПО сприймає проектувальник БД;
- у виді певних символів.

Про цей перелік подань кажуть, що розглядають саму *реальність*, *опис реальності* та *дані*, які відображають цей опис. Названим поданням відповідають певні моделі даних.

Існують різноманітні підходи до класифікації моделей даних.

Згідно з однією розглядають дві категорії: ***концептуальні*** моделі та моделі ***реалізації***.

Концептуальна (понятійна, смислова) модель (*conceptual Model*) відбиває логічну природу даних, тобто уявлення про них основних користувачів. Модель цього типу має подавати інформацію про предметну область у вигляді, незалежному від СКБД. Таку модель використовують на початку проєктування автоматизованої інформаційної системи – при створенні її концепції.

Головне призначення концептуальної моделі – смисловий опис предметної області, подання інформації про неї в зручній для користувача формі, а подробиці організації фізичного збереження даних він може і не знати. Дуже часто замість слова ”смисловий” вживають слово ”*семантичний*”, від грецького слова *semantikos* – ”той, що визначає (означає)”.

Ця модель розглядає основні логічні об'єкти (сутності) ПО та зв'язки між ними, необхідні та достатні для ефективного застосування інформації з точки зору користувача. Фахівці називають її *інформаційно – логічною* моделлю, або просто ***інфологічною*** (*infological Model*).

Модель реалізації, на відміну від концептуальної, спрямована на відбиття способу представлення (синтаксису) даних у БД, тобто пов’язана з типом СКБД. Вона відображає те, яким чином мають бути реалізовані структури даних для адекватного відбиття семантики моделі. До моделей реалізації БД відносять ієрархічну, сіткову, реляційну та об'єктно-орієнтовану. Моделі реалізації за рівнем абстракції розділяють на *зовнішні*, *внутрішні* та *фізичні*.

Таку класифікацію можна відобразити схемою, поданою на рис. 2.1.

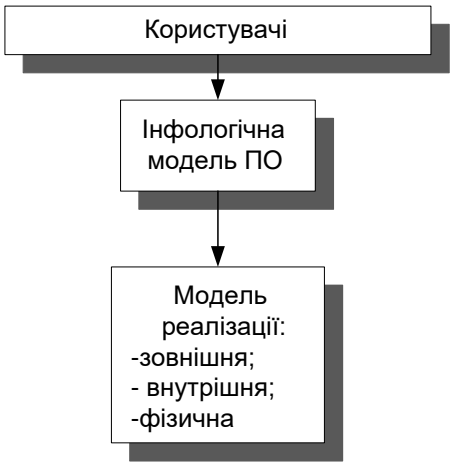


Рис.2.1. Схема моделювання зі структурою ”інфологічна модель – модель реалізації”

Розглянемо також іншу класифікацію моделей даних, яка використовує схожу термінологію з дещо зміненим смислом.

Комітет стандартів і норм (*Standards Planning and Requirements Committee – SPARC*) Національного інституту стандартизації США (*American National Standards Institute – ANSI*) запропонував трирівневий підхід до подання даних при побудові СКБД.

Так звана модель *ANSI — SPARC* не стала стандартом, але її використовують для розуміння функціональних особливостей СКБД.

Модель *ANSI — SPARC* виділяє три моделі у відповідності з рівнями абстракції: *зовнішню, концептуальну* та *внутрішню* (див. рис. 2.2.). Рівень, на якому дані сприймає користувач – зовнішній (*external Level*), операційна система і СКБД сприймають дані на внутрішньому рівні (*internal Level*). Концептуальний рівень (*conceptual Level*) подання даних призначений для відображення зовнішнього рівня на внутрішній (тому його іноді називають допоміжним). Для відбиття проблем реалізації БД, специфічних для обраної СКБД у внутрішній моделі, до цих рівнів абстракції додається ще один – *фізичний*.

Загальний опис БД називають *схемою* БД. Тому зазначені у *ANSI — SPARC* моделі три рівні іноді називають трьома *схемами.*

Детальніше розглянемо визначені *ANSI* та наведені на рис. 2.2 рівні абстракції подання даних [2].

Зовнішній рівень – це рівень розуміння даних користувачем (*зовнішнє подання даних*). Кожна група користувачів виділяє в модельованій предметній області, спільній для всієї установи, необхідні саме для неї дані. Опис БД, необхідний окремій групі користувачів або орієнтований на певний аспект ПО, називають *підсхемою*.

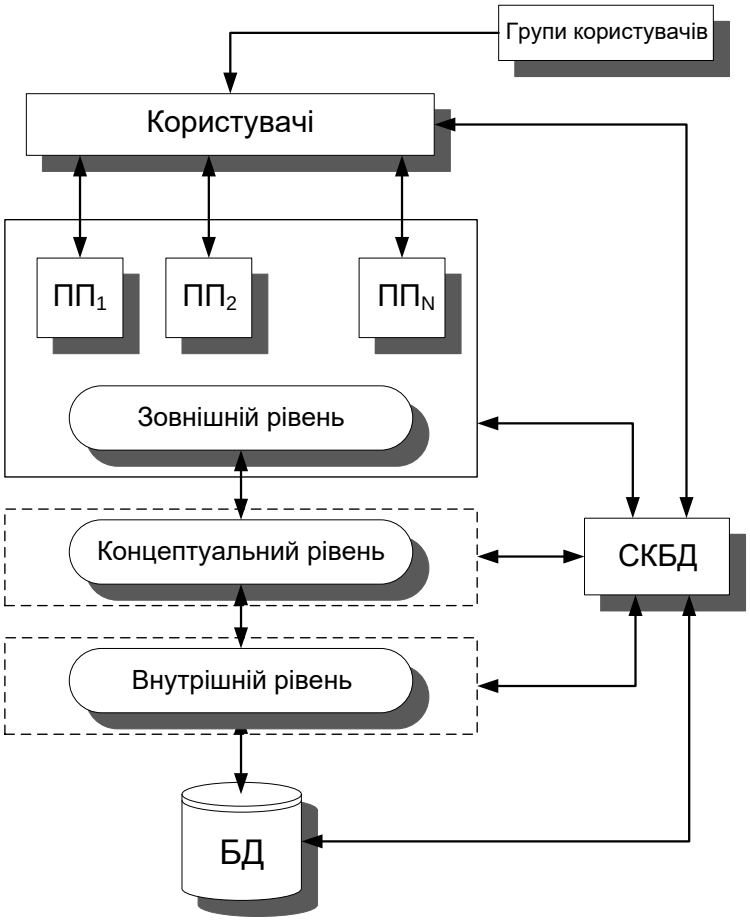


Рис. 2.2. Структура моделі *ANSI — SPARC*

Концептуальний рівень є проміжним і забезпечує подання *всієї інформації бази даних* в абстрактній формі. Опис БД на цьому рівні є результатом концептуального проєктування, яке передбачає аналіз інформаційних потреб усіх користувачів. Таким чином, *концептуальна схема* – це узагальнений логічний опис всіх даних і відношень між ними, інакше – логічна структура всієї бази даних. Для кожної бази даних є тільки одна концептуальна схема, яка визначає наступне: сутності та атрибути; зв'язки між сутностями; обмеження, що накладаються на дані; семантична інформація про дані, вона повинна також забезпечувати безпеку і підтримку цілісності даних.

Внутрішній рівень абстракції відображає внутрішнє розуміння даних. Внутрішня схема передбачає опис фізичної реалізації бази даних і призначений для досягнення оптимальної продуктивності і забезпечення економного використання дискового простору. На внутрішньому рівні зберігається інформація про розподіл дискового простору для зберігання даних і індексів, подробиці збереження записів (з вказівкою реальних розмірів елементів даних, що зберігаються); відомості про розміщення записів, стискання даних, вибрані методи їх шифрування. Для кожної бази даних існує тільки одна внутрішня схема.

Внутрішня схема СКБД встановлює відповідність між трьома типами схем різних рівнів.

Нижче за внутрішній рівень знаходиться *фізичний рівень*, який контролюється операційною системою, але під керівництвом СКБД. Він має індивідуальні риси для кожної системи.

Проектувальники фізичної бази даних працюють тільки з відомими операційній системі елементами. Проте функції СКБД і операційної системи на фізичному рівні не цілком чітко розділені і можуть варіюватися від системи до системи.

Фізична модель (*physical Model*) даних оперує категоріями, що стосуються організації зовнішньої пам'яті і структур зберігання у даному операційному середовищі. Сучасні фізичні моделі використовують різні методи розташування даних, які базуються на файлових структурах: це файли прямого й послідовного доступу, індексні, інвертовані та взаємозалежні файли, файли, що використовують різні методи хешування тощо. Сучасні СКБД використовують також сторінкову організацію даних. Фізичні моделі даних, засновані на сторінковій організації, є найбільш перспективними.

Основним призначенням трирівневої моделі (інакше кажуть - архітектури) є забезпечення незалежності від даних. Сутність цієї незалежності полягає в тому, що зміни на нижніх рівнях ніяк не впливають на верхні рівні. Розрізняють два типи незалежності від даних: логічну та фізичну.

Логічна незалежність від даних означає повну захищеність зовнішніх схем від змін, що вносяться до концептуальної схеми. Такі зміни концептуальної схеми, як додавання або видалення нової сутності, атрибутів або зв'язків, повинні здійснюватися без необхідності внесення змін у вже існуючі зовнішні схеми для інших груп користувачів.

Фізична незалежність від даних означає захищеність концептуальної схеми від змін, що вносяться до внутрішньої схеми. Такі зміни внутрішньої схеми, як використання різних файлових систем або структур зберігання, різних пристроїв зберігання, модифікація індексів або хешування, повинні здійснюватися без необхідності внесення змін в концептуальну або зовнішню схеми.

СКБД підтримує певну модель даних ПО і відображає її у відповідні структури фізичної бази даних.

Модель даних, яку підтримує СКБД на логічному рівні, визначається трьома компонентами:

- припустимою структурою даних, різноманітністю і кількістю типів об'єктів, які можна описати за допомогою моделі;
- множиною припустимих операцій над даними;
- обмеженнями для контролю цілісності.

Таким чином здійснюється інформаційне моделювання ПО в пам'яті ЕОМ. Мова програмування СКБД містить засоби опису як внутрішньої схеми, так і створення прикладних програм.

Треба відрізняти опис БД (її схему) від власне БД. Схему створюють на етапі проєктування БД і змінюють її дуже рідко. Власне БД містить інформацію, яка у процесі роботи весь час змінюється. Отже, одній і тій самі схемі можуть відповідати різні стани БД.

Узгодити дві описані класифікації можна, якщо розглядати концептуальну модель першої класифікації як зовнішній рівень абстракції у класифікації *ANSI*.

Оскільки моделі даних досить численні, то краще навести одну з їх найбільш детальних та повних класифікацій у вигляді схеми (рис. 2.3) [8].

Якщо попередні категорії моделей порівняти з класифікацією, поданою на рис. 2.3, то стає зрозумілим, що моделі реалізації охоплюють даталогічні та фізичні моделі.

Концептуальна (інфологічна) модель не залежить ні від програмного, ні від апаратного забезпечення автоматизованої інформаційної системи. Зовнішня та внутрішня моделі не залежать від апаратного забезпечення. Фізична модель залежить як від програмного, так і від апаратного забезпечення.

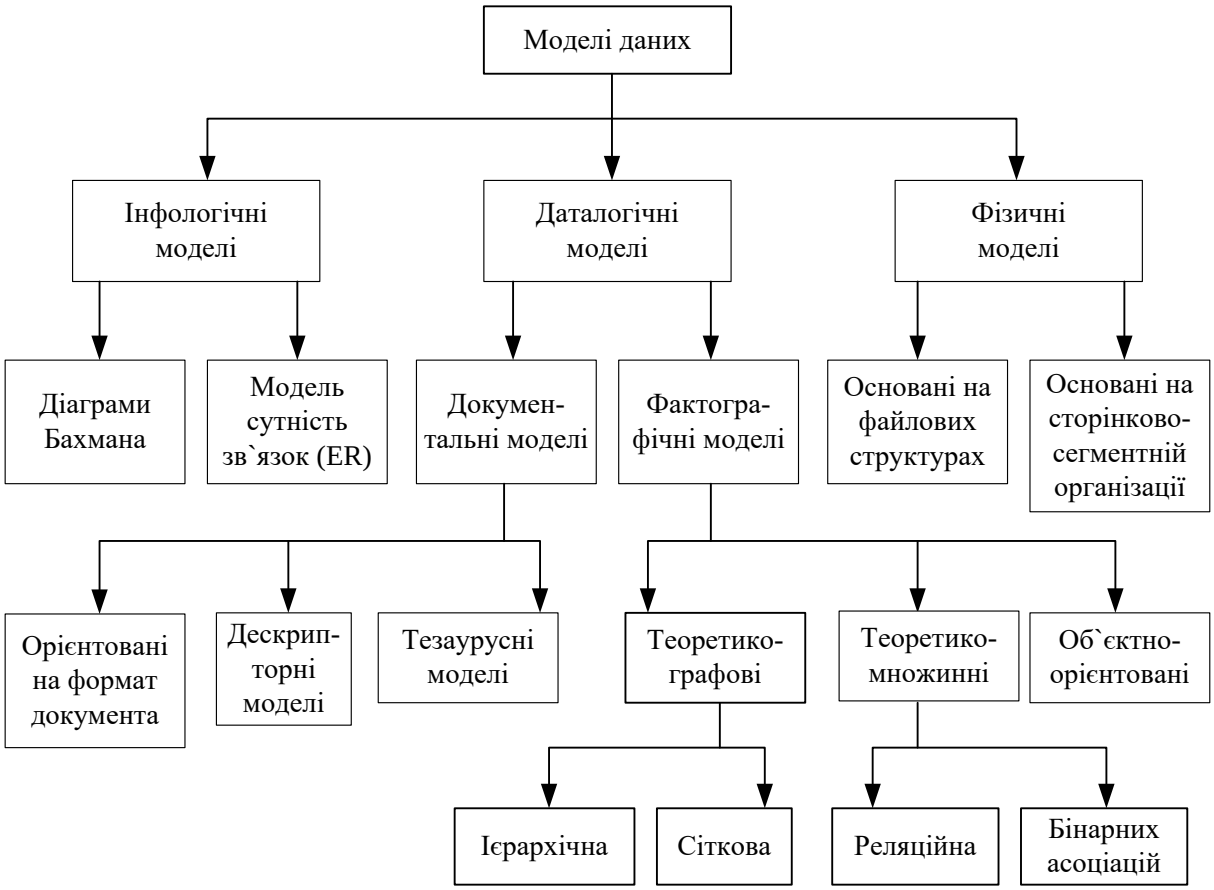


Рис. 2.3. Схема класифікації моделей даних

2.3. Інфологічна модель предметної області

Проектувальники інфологічної моделі розглядають модель предметної області у вигляді „***сутність – зв'язок”*** (*Entity – Relationship Model, ER – model*), запропоновану Пітером Пін-Шен Ченом (*Peter Pin-Shan Chen*) у 1976 р. Цей вид моделі не орієнтований на конкретні програмні та технічні засоби і призначений для логічного подання даних. При її створенні звичайно розглядають 3 основні конструктивні одиниці, а саме

- сутність;
- атрибут;
- зв'язок.

Предметну область розглядають як *множину сутностей*, між якими існує деяка *множина зв'язків*.

Сутність (*Entity*) – це об'єкт будь-якої природи (реальний або уявний), що може бути ідентифікований певним способом, який вирізняє його від інших об’єктів. Розрізняють сутності, як правило, за їхніми властивостями.

Існують такі поняття, як *тип* (*клас*) *сутності* й *екземпляр сутності*. ***Тип сутності*** (*Entity Set*) - множина однорідних особистостей, предметів, подій тощо, які мають однаковий перелік властивостей та виступають як дещо ціле.

Екземпляр сутності (*Entity Instance, Entity Occurrence*) – це окрема особистість, предмет, подія, тощо.

Наприклад, *сутність* (*тип сутності*) ПЕРВИННИЙ ПЕРЕТВОРЮВАЧ СИГНАЛІВ – це множина різноманітних первинних перетворювачів, е*кземпляр сутності* – певна термопара, певний термометр опору, певна трубка Бурдона тощо.

Сутність може бути *залежною від існування* іншої сутності (*Existence dependent*).

Наприклад, сутність РЕГУЛЯТОР залежить від існування сутності АВТОМАТИЧНА СИСТЕМА КЕРУВАННЯ. Сутність СХЕМА АВТОМАТИЗАЦІЇ залежить від сутності ПРОЕКТ АВТОМАТИЗАЦІЇ ВИРОБНИЦТВА.

Сутність може бути *незалежною від існування* іншої сутності (*Existence independent*). Наприклад, сутність РЕМОНТ КВП і А (контрольно-вимірювальних приладів і автоматики) може не залежати від сутності ФІРМА ПО РЕМОНТУ КВП і А, якщо ці прилади лагодитимуть фахівці того виробництва, на якому їх використовують.

Одна сутність може бути *обов'язковою* або *необов'язковою* для іншої (іноді в цьому контексті кажуть про *клас належності* сутності). Участь сутності у зв'язку з іншою сутністю необов'язкова тоді, коли один екземпляр однієї сутності не вимагає наявності відповідного екземпляра іншої сутності у кожному окремому зв'язку.

Наприклад, сутність ГРУПА обов'язкова для сутності СТУДЕНТ, оскільки кожний студент входить у склад якоїсь груп. Сутність ВИКОНАВЧИЙ МЕХАНІЗМ обов'язкова для сутності АВТОМАТИЧНА СИСТЕМА КЕРУВАННЯ. Сутність ВИКЛАДАЧ обов'язкова для сутності ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД, оскільки викладач навчає студентів, оцінює результати їх самостійної роботи, проводить заліки та екзамени, розробляє методичні посібники тощо.

Сутність може бути *необов'язковою* по відношенню до іншої. Наприклад, сутність СТУДЕНТ необов'язкова для сутності СПІВРОБІТНИК КАФЕДРИ, оскільки не кожний співробітник працює зі студентами.

Сутність називають *слабкою* (*залежною*), якщо вона не може існувати без якоїсь пов’язаної з нею сутністю. Наприклад, сутність ЦЕХ КВП і А не може існувати без сутності КВП і А. КАФЕДРА не може існувати без сутності ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД.

Атрибут (*Attribute*) – це поіменована логічно неподільна властивість (характеристика) сутності. Для сутності ПЕРВИННИЙ ПЕРЕТВОРЮВАЧ СИГНАЛІВ атрибутами можуть бути *Назва вимірюваної величини, Одиниця вимірювання, Нижнє значення вимірюваної величини, Верхнє значення вимірюваної величини, Нижнє значення вихідного сигналу, Верхнє значення вихідного сигналу, Завод - виготовник, Ціна* і т. ін.

Для кожного атрибуту є безліч значень, наприклад, для атрибуту *Прізвище* значеннями є “Гейтс”, “Шевченко”, “Паскаль”. Атрибути називають ще ***інформаційними елементами***. Припустиму потенціальну множину значень одного атрибута (тип даних та можливий діапазон значень) називають ***доменом***.

Сутність фактично є множиною атрибутів, що описують властивості всіх елементів даної сутності.

Сутність може володіти тільки одним значенням якоїсь властивості. Наприклад, кожна людина може мати тільки одну дату народження. Такі властивості називають ***однозначними*** (*single - valued Attribute*). Інші властивості можуть мати одночасно декілька значень для одного об'єкта, їх називають ***багатозначними*** (*multi - valued Attribute*).

Наприклад, при описі сутності ФАХІВЕЦЬ фіксують такі властивості, як вищі навчальні заклади, які він закінчив, та володіння іноземними мовами. Оскільки фахівець може закінчити декілька навчальних закладів та володіти декількома іноземними мовами, то визначимо ці властивості як ***багатозначні***.

Деякі властивості є сталими, їхні значення не змінюються з часом. Такі властивості називають ***статичними***, а ті властивості, значення яких можуть змінюватися з часом – ***динамічними***. До статичних належать одиниці вимірювання приладу, дата виготовлення пристрою, день, місяць та рік народження, до динамічних – адреса, прізвище, місце роботи, дата повірки пристрою, місце установки пристрою і т. ін.

Іншою характеристикою властивості є ознака того, чи притаманна ця властивість усім елементам сутності даного класу або тільки деяким. Наприклад, одні фахівці мають атрибут *Вчений ступінь*, а інші його не мають. Те саме стосується такої властивості, як наявність пільги для поселення студента у гуртожиток, або властивість пристрою мати вихідний сигнал. Такі властивості називають ***умовними***.

При інфологічному моделюванні атрибути поділяють також на ***прості*** (*simple Attribute*) та ***складені*** (*composite Attribute*).

Прикладами складених атрибутів можуть бути *Адреса*, що складається з назв *Міста*, *Вулиці*, номерів *Будинку* і *Квартири*, і *Дата народження*, що складається з *Числа*, *Місяця* і *Року*.

При даталогічному моделюванні даних атрибути класифікують також за належністю до одного з трьох типів: *описові*, *вказівні*, *допоміжні*. ***Описові атрибути*** – це факти, внутрішньо притаманні кожному екземпляру сутності. Атрибути, що вказують (***вказівні атрибути***), використовують для надання імені чи позначення унікальності екземплярів сутності. ***Допоміжні атрибути*** використовуються для зв'язку екземпляра однієї сутності з екземпляром іншої.

Нарешті, можуть існувати ***похідні атрибути*** (*derived Attribute*), тобто атрибути, які не треба зберігати у БД, а можна отримати за допомогою певного алгоритму.

Наприклад, вік співробітника *Співробітник_Вік* можна отримати як різницю між поточною датою та
датою народження.

Інший приклад - час наступної повірки технічного засобу автоматизації можна визначити схожим чином: до дати уведення засобу в експлуатацію треба додати тривалість роботи між профілактичними заходами.

Зв'язок (*Relationship*)– це спосіб, за допомогою якого відображають відношення між сутністю та атрибутами, а також між декількома сутностями.

Зв’язки характеризують *типом, направленістю, степенем, потужністю, обов’язковістю*. Їм надають імена.

Для позначення ***типу*** зв’язку використовують також термін ***зв’язнісь*** (*Connectivity*). ***Зв’язнісь*** відображає одноразовість чи багаторазовість участі екземпляра сутності у зв’язку. Типи бувають одно - та багатозначні.

За ***направленістю*** зв’язки поділяють на одно- та двонаправлені.

Розглянемо визначення зв’язків за направленістю та приклади з урахуванням зв’язності.

Однонаправлений зв'язок має наступні типи:

- *однозначний*, коли одному екземпляру сутності *A* відповідає точно один екземпляр сутності *B*, а зворотний зв'язок не визначений ($A \rightarrow B$), наприклад, КОМП’ЮТЕР → БЛОК ЖИВЛЕННЯ, СИСТЕМА ПІДГОТОВКИ СТИСЛОГО ПОВІТРЯ → ФІЛЬТР СТИСЛОГО ПОВІТРЯ, СТІЛЬНИКОВИЙ ТЕЛЕФОН → ДИСПЛЕЙ, СТУДЕНТ → СТИПЕНДІЯ[1],

[1] Технічний прогрес або зміни у суспільстві можуть змінити і тип зв’язку

КЕРІВНИК → СПІВРОБІТНИК, ДИПЛОМНИЙ ПРОЕКТ ЗІ СПЕЦІАЛЬНОСТІ “АВТОМАТИЧНЕ УПРАВЛІННЯ ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ” → СХЕМА АВТОМАТИЗАЦІЇ.

- ***багатозначний***, коли одному екземпляру сутності *A* відповідає декілька екземплярів сутності *B*, зворотний зв'язок теж не визначений, (*A* →→ *B*) наприклад, КОМП’ЮТЕР →→ ІНТЕГРАЛЬНА СХЕМА, ПРОЕКТ СИСТЕМИ АВТОМАТИЗАЦІЇ →→ КРЕСЛЕНИК, ТЕХНОЛОГІЧНИЙ ОБ’ЄКТ КЕРУВАННЯ →→ МОДЕЛЬ, КРЕДИТ →→ ПЛАТІЖ, СИРОВИНА →→ ПРОДУКЦІЯ, СТІЛЬНИКОВИЙ ТЕЛЕФОН→→ДИСПЛЕЙ,
СТУДЕНТ →→СТИПЕНДІЯ.

Останні два приклади виділені жирним, оскільки вони вже наводилися до зв’язку *A* → *B*. Пояснимо причину цього:

1) з розвитком технології виготовлення стільникових телефонів ці пристрої стають все складнішими, починають з’являтися і такі, що мають не один дисплей;

2) стипендії, які можуть отримати студенти урізноманітнюються, іноді статус однієї стипендії передбачає, що її можуть надати навіть і тоді, коли студент вже отримує якусь іншу стипендію.

Тобто, конкретна ситуація визначає, до якого типу може належати зв’язок між сутностями.

Двонаправлений зв'язок передбачає такі типи:

- ***однозначний*** зв'язок “1:1” („***один до одного***”) – одному екземпляру сутності *A* відповідає тільки один екземпляр сутності *B* і навпаки (*A* ↔ *B*), наприклад, ДЕКАН↔ ФАКУЛЬТЕТ, КОМП’ЮТЕР ↔ ВІДЕОПРОЦЕСОР, МІКРОПРОЦЕСОР ↔ ОПЕРАТИВНА ПАМ’ЯТЬ,

КОМП’ЮТЕР ↔ МАТЕРИНСЬКА ПЛАТА, ОДНОКОНТУРНА СИСТЕМА КЕРУВАННЯ ↔ РЕГУЛЯТОР, РЕГУЛЯТОР ↔ ЗАКОН КЕРУВАННЯ.

- ***багатозначні*** зв'язки:

а) “1:*M*” (“***один до багатьох”***) – одному екземпляру сутності *A* відповідає *M* екземплярів сутності *B*, а *M* екземплярам сутності *B* відповідає тільки один екземпляр сутності *A* (*A*⇔*B*), наприклад, ФАКУЛЬТЕТ⇔КАФЕДРА, СИСТЕМА АВТОМАТИЗАЦІЇ ⇔ АВТОМАТИЧНА СИСТЕМА КЕРУВАННЯ, ТЕХНОЛОГІЧНИЙ ОБ’ЄКТ КЕРУВАННЯ ⇔ РЕЖИМНИЙ ПАРАМЕТР (якщо властивості сировини на його вході сталі), СИРОВИНА ⇔ РЕЖИМНИЙ ПАРАМЕТР, СТУДРАДА ФАКУЛЬТЕТУ ⇔ СТУДЕНТ.

б) “*N*:*M*” (“***багато до багатьох”***), тобто *N* екземплярам сутності *A* відповідає *M* екземплярів сутності *B* і навпаки (*A*⇔*B*), наприклад, ВИКЛАДАЧ ⇔ СТУДЕНТ, СТУДЕНТ ⇔ ПРЕДМЕТ, СПІВРОБІТНИК ⇔ РОБОЧА ГРУПА, ЗАВОД – ВИГОТОВНИК ⇔ ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ, УСТАТКУВАННЯ ⇔ СИРОВИНА, УСТАТКУВАННЯ ⇔ ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ.

Зв'язки типу “1:1” зустрічаються частіше всього при розділенні доступу користувачів до певної інформації в базі даних і для створення різних зовнішніх уявлень для додатків БД. Наприклад, у бухгалтерії є інформація про зарплату співробітника та його табельний номер, а у відділі кадрів – тільки його табельний номер і посада. Зв'язок типу ”1:*M*” використовують для відображення ієрархічних зв'язків між різними сутностями.

Степінь (вимірність, порядок, арність) зв’язку (*Relationship Degree*) вказує на кількість пов’язаних сутностей.

Унарний зв’язок відповідає випадку, коли є рекурсія сутності саму на себе. При *бінарному* зв’язку задіяні дві сутності, при *тернарному* – три і т.д.

Потужність (кардинальність) зв’язку (*Cardinality*) вказує на певну кількість екземплярів однієї сутності, пов’язаних з одним екземпляром іншої сутності. Зазвичай вказують найменше і найбільше значення потужності , які називають *показниками кардинальності*.

Обов’язковість зв’язку випливає з такої властивості сутності, як *клас належності* (сутність *обов’язкова* або *необов’язкова*). Тоді і зв’язок може бути *обов’язковим* або *необов’язковим*.

2.4. Подання інфологічних моделей у вигляді *ER* – діаграм

Дуже важливою властивістю моделі “сутність – -зв'язок” є те, що вона може бути подана у вигляді графічної схеми (*ER* – діаграми, *ERD*), зокрема її часто реалізують у *CASE* – системах (*Computer – Aided Systems Engineering*). Це значно полегшує аналіз предметної області і проєктування БД.

Існує декілька стандартизованих *нотацій* – способів позначення елементів на діаграмі “сутність – зв'язок”, кожна з яких має свої позитивні риси.

Розглянемо дві нотації – Пітера Чена (*Peter Pin-Shan Chen*) та ”пташина лапка” (*Crow’s Foot Model*), яка була запропонована К.В. Бахманом (*C.W. Bachman*) [11]. Позначення, які використовують при їх застосуванні, наведено на рис. 2.4.

В обох нотаціях сутність зображують на схемі прямокутником, у полі якого вказують її ім’я (в однині, в основному великими літерами).

Для більшої конкретики всередині можуть вказувати і приклад екземпляра сутності.

Зв’язки між сутностями позначають лініями і надають їм імена. Для назв зв’язків використовують дієслова. У моделі Чена для назви зв’язку в середину лінії зв’язку між сутностями уводять графічний елемент (ромб) з текстом, що описує тип зв'язку. В нотації ”пташина лапка” назву вказують в розриві чи біля лінії.

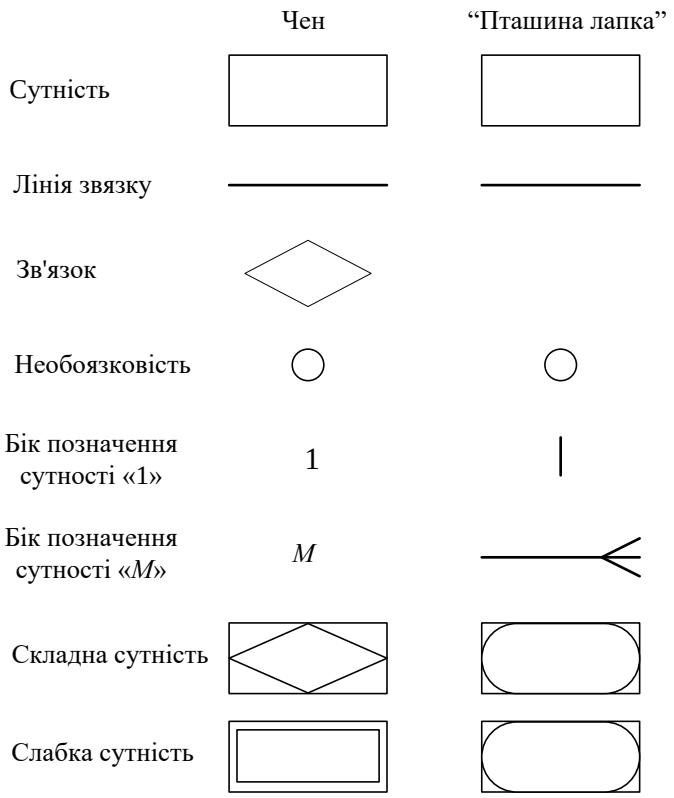


Рис. 2.4. Позначення елементів на *ER* – діаграмах Чена і ”пташина лапка”

У моделі Чена при зображенні типу зв’язку біля сутності, що відповідає “1” пишуть цю цифру, а біля сутності, що відповідає ”*M*” – цю літеру.

У нотації ”пташина лапка” для “1” використовують риску, що вертикально перекреслює лінію зв’язку, для ”*M*” – зображення трипальцевої пташиної лапки (звідки і походить назва нотації). Для зв’язку *N*:*M* у цій нотації ”лапки” стоять на обох кінцях лінії зв’язку.

На рис. 2.5 подано *ER* – діаграму для опису зв’язку між сутностями СИСТЕМА АВТОМАТИЗАЦІЇ ⇔ АВТОМАТИЧНА СИСТЕМА КЕРУВАННЯ (1:*M*) в обох нотаціях. Зв’язок матиме смисл ***містить***, тобто СИСТЕМА АВТОМАТИЗАЦІЇ ***містить*** АВТОМАТИЧНУ СИСТЕМУ КЕРУВАННЯ.

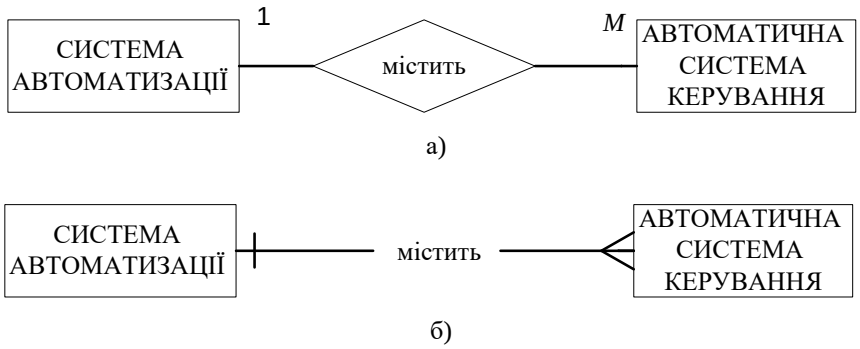


Рис. 2.5. *ER* – моделі для зв’язків між сутностями СИСТЕМА АВТОМАТИЗАЦІЇ та АВТОМАТИЧНА СИСТЕМА КЕРУВАННЯ в нотаціях: а) Чена, б) ”пташина лапка”

На рис. 2.6 зображено *ER* – діаграму для опису зв’язку між сутностями ЗАВОД – ВИГОТОВНИК ⇔ ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ (*N*:*M*) в обох нотаціях. Зв’язок матиме смисл ***виготовляє***, тобто ЗАВОД – ВИГОТОВНИК ***виготовляє*** ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ.

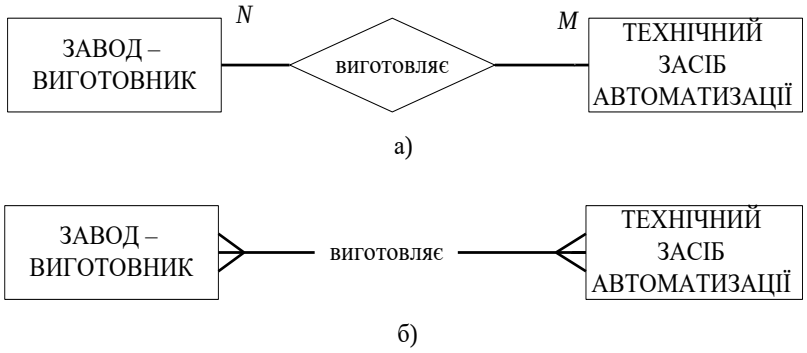


Рис. 2.6. *ER* – моделі для зв’язків між сутностями ЗАВОД – ВИГОТОВНИК та ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ в нотаціях:
а) Чена, б) ”пташина лапка”

Зупинимось на зображенні потужностей зв’язку. *Потужність* зв’язку показує інтервал кількості можливих входжень сутності в зв'язок, тому кожний зв'язок має 2 показники цієї властивості. Їх розташовують біля позначення сутності.

У моделі Чена потужність позначають парою чисел в форматі (x, y), де x – мінімальне, а y – максимальне значення потужності.

Для обов’язкової сутності мінімальна потужність зв’язку дорівнює 1.

На рис. 2.7 и 2.8 зображено *ER* – діаграми у відповідних нотаціях для сутностей ПРОФЕСОР та ГРУПА і назви зв’язку “*навчає*”. На діаграмах показано способи зображення зв’язностей і потужностей зв’язків у цих нотаціях. Наприклад, потужність (1,4), яка розташована біля ПРОФЕСОР у зв’язку ”ПРОФЕСОР *навчає* ГРУПУ” означає, що професор веде заняття хоча б в одній групі, але не більше ніж у чотирьох групах. Якби потужність описувалась (1, *M*), то це означало б, що немає обмежень на кількість груп, з якими може працювати один професор.

Потужність (1,1), розташована біля сутності ГРУПА, вказує на те, що в кожній групі вести заняття може тільки один професор (не більше і не менше).

У стандартній моделі ”пташина лапка“ числовий діапазон значень потужностей не відображають. Максимальне значення 4 у діапазоні (1,4) відображено у цій нотації як “багато”.

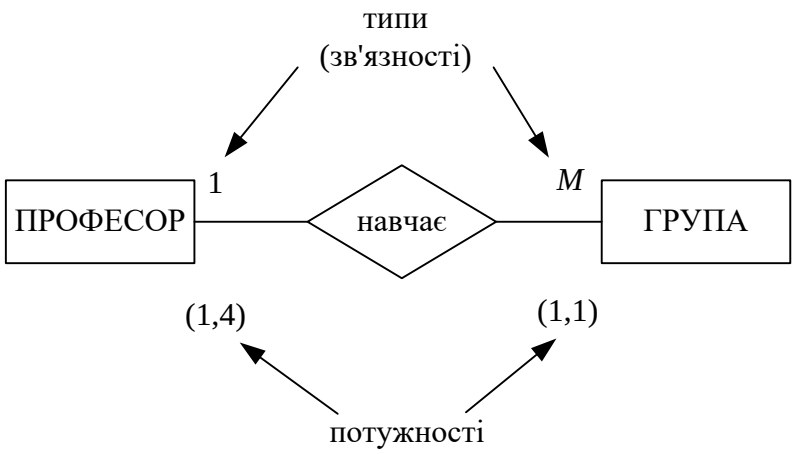


Рис. 2.7. Діаграма Чена з прикладами зображень зв’язностей і потужностей

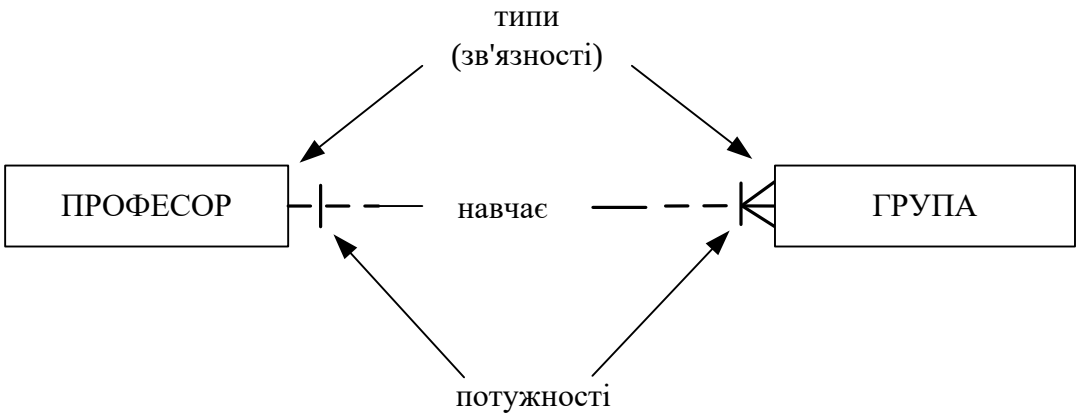


Рис. 2.8. Діаграма ”пташина лапка“ з прикладами зображень зв’язностей і потужностей

Іншою важливою характеристикою зв'язку є *обов'язковість*. Ця характеристика пов’язана з класом належності сутності, тобто тим, що ця сутність може бути *необов’язковою* по відношенню до іншої. Розглянемо приклад.

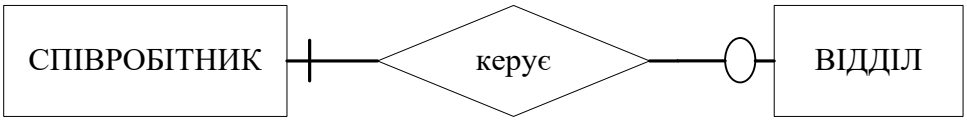
Нехай є дві сутності СПІВРОБІТНИК та ВІДДІЛ. Перша пов’язана з другою зв’язком з іменем “*керує*”. У кожному відділі обов'язково повинен бути керівник, тому кожній сутності ВІДДІЛ при такому зв’язку неодмінно повинна відповідати сутність СПІВРОБІТНИК.

Однак, не кожен співробітник є керівником відділу, отже для даного зв'язку не кожна сутність СПІВРОБІТНИК має асоційовану з нею сутність ВІДДІЛ. Отже, сутність СПІВРОБІТНИК має *обов'язковий клас належності*.

У діаграмах Чена і ”пташина лапка“ *не обов’зковий* зв’язок зображують невеликим колом з боку необов’язкової сутності. Для неї мінімальне значення потужності зв’язку дорівнює 0.

У нашому прикладі біля сутності СПІВРОБІТНИК на лінії зв’язку треба зобразити риску, а біля сутності ВІДДІЛ зобразити коло (рис. 2.9а).

Інший приклад – сутності СПІВРОБІТНИК ЦЕХУ КВП і А та ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ із зв’язком ”*ремонтує*“. Необов’язковою сутністю є ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ оскільки не кожний співробітник цеху КВП і А має за обов’язок виконувати саме ремонт (див. рис. 2.9б).



а)

б)

Рис. 2.9. *ER* - діаграми з необов’язковим типом зв’язку:
а) сутності СПІВРОБІТНИК і ВІДДІЛ;
б) сутності СПІВРОБІТНИК ЦЕХУ КВП і А та ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ

На рис. 2.10. зображені способи зображення типів зв’язків і потужностей у нотації “пташина лапка”.

Рис. 2.10. Типи і потужності зв’язків у нотації “пташина лапка”

Повернемось до сутностей СПІВРОБІТНИК та ВІДДІЛ і розглянемо потужність зв’язків. Співробітник може зовсім не керувати відділом, тоді мінімальне значення потужності зв’язку для нього 0, або може бути керівником тільки одного відділу, тоді максимальне значення потужності дорівнює 1.

Такий зв'язок можна описати як 0,1:1,1, і саме цей опис задаватиме *кардинальності зв'язку,* тобто *кількість екземплярів зв'язаної сутності, з якою зв'язаний кожний екземпляр головної сутності*.

Розглянемо ще один зв’язок між сутностями СПІВРОБІТНИК та ВІДДІЛ, а саме, ”*працює* “. У кожному відділі може працювати довільна кількість співробітників, але співробітник може працювати тільки в одному відділі. Для СПІВРОБІТНИК при такому смислі зв’язку потужність запишемо (1,1), а для ВІДДІЛ при необов'язковому класі належності вона (0,1).

Загальна *ER* – діаграма для обох аспектів зв’язку між сутностями СПІВРОБІТНИК та ВІДДІЛ наведені на рис. 2.11.

Позначення зв'язку типу 1:*N* (СПІВРОБІТНИК - *працює* у - ВІДДІЛ), 1,1:0,*N* (кожний співробітник працює лише у одному відділі, 1:1, кожний відділ може нараховувати скільки завгодно співробітників, у тому числі жодного, 0:*N*).

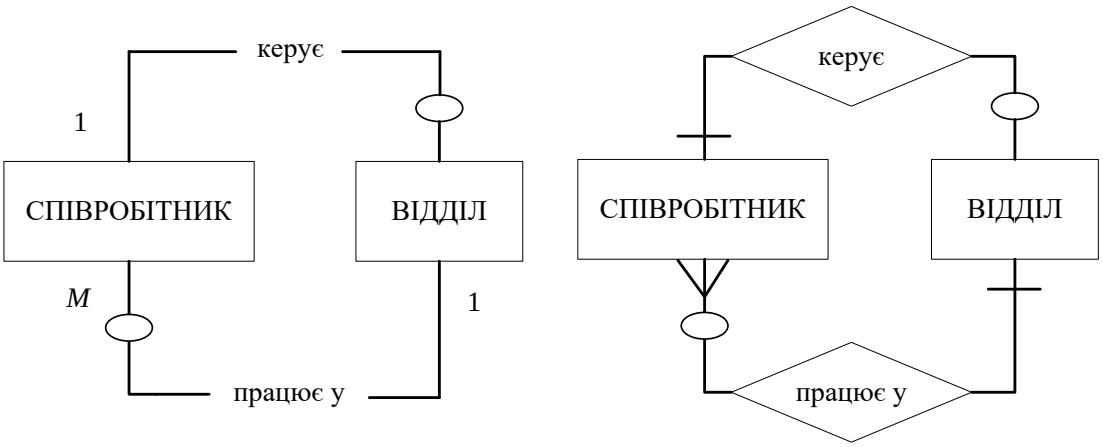


Рис. 2.11. *ER* – діаграми для зв’язків між сутностями СПІВРОБІТНИК та ВІДДІЛ
а) нотація Чена, б) нотація “пташина лапка”

Необхідно також враховувати клас приналежності сутностей. Кожен співробітник повинен працювати в якомусь відділі, але не кожен відділ (наприклад, новий) повинен мати хоча б одного співробітника. Тому сутність ВІДДІЛ має обов'язковий, а сутність СПІВРОБІТНИК необов'язковий класи належності.

Останньою розглянемо зв’язність *N:M*. Кожна з асоційованих сутностей може бути представлена будь-якою кількістю екземплярів.

Нехай для розробки проекту модернізації системи автоматизації та підвищення якості продукції створено робочі групи, у кожну з яких входять співробітники різних відділів.

Кожен співробітник може входити у декілька, *M* (у тому числі й у жодну) робочих груп, а кожна група повинна мати не менше одного співробітника, тобто *N*. *ER* – діаграма у нотації “пташина лапка” наведена на рис. 2.12а.

Розглянемо ще приклад. Певні співробітники цеху КВП і А можуть обслуговувати по декілька ТЗА, у той же час, начальник цеху та деякі інші співробітники обслуговуванням не займаються. На рис. 2.12б наведено відповідні *ER* – діаграми.

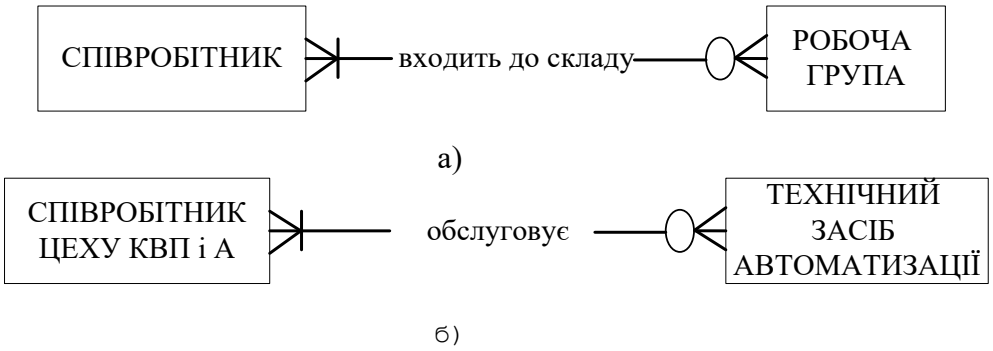


Рис. 2.12. *ER* – діаграма у нотації “пташина лапка” для зв’язності *N:M*
а) між сутностями СПІВРОБІТНИК і РОБОЧА ГРУПА;
б) між сутностями СПІВРОБІТНИК ЦЕХУ КВП і А та ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ

Нехай існування сутності *A* залежить від існування сутності *B* (кажуть, що сутність *A* “слабка”, а сутність *B* – “сильна”). Так, розглянемо зв'язок між сутностями РОБОЧА_ГРУПА та КОНТРАКТ. Робочу групу створюють тільки після того, як буде підписаний контракт із замовником, і вона припиняє своє існування після виконання контракту. Таким чином, сутність РОБОЧА_ГРУПА є слабкою (*залежною*) від сутності КОНТРАКТ.

На рис. 2.13 а,б наведемо приклади *ER* – діаграм з залежною сутністю для сутностей РОБОЧА_ГРУПА та КОНТРАКТ.

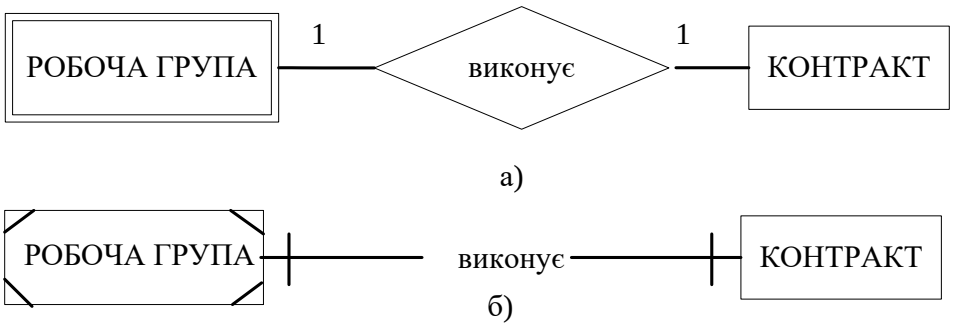


Рис. 2.13. Приклади *ER* – діаграм з залежною сутністю для сутностей РОБОЧА ГРУПА та КОНТРАКТ:
а) нотація Чена, б) нотація “пташина лапка”

Іншим прикладом є взаємодія сутностей ПРОЕКТ СИСТЕМИ АВТОМАТИЗАЦІЇ та ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ. Без розробленого проекту неможна з’ясувати, які саме ТЗА будуть використані у системі автоматизації. Тільки по завершенні робіт над проектом, зокрема після складання замовної специфікації на ТЗА можна закуповувати пристрої автоматизації. Відповідна *ER* – діаграма наведена на рис. 2.14 у нотації Чена.

Рис. 2.14. Приклади *ER* – діаграм з залежною сутністю для сутностей ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ та ПРОЕКТ СИСТЕМИ АВТОМАТИЗАЦІЇ

Кардинальність зв'язку з боку сильної сутності завжди буде (1,1). Клас належності і степінь зв'язку для залежної сутності можуть бути будь-якими. Для прикладу припустимо, що хімічне підприємство використовує декілька банківських кредитів. Виділимо сутність КРЕДИТ, запишемо біля неї в дужках позначення можливих атрибутів

КРЕДИТ (*Номер_Договору*, *Сума*, *Термін_Погашення*, *Банк*).

По кожному кредиту повинні здійснюватися платежі в рахунок його погашення. Тому виділимо сутність ПЛАТІЖ і запишемо її атрибути

ПЛАТІЖ (*Дата*, *Сума*).

Розглянемо зв'язки між КРЕДИТ і ПЛАТІЖ. У тому випадку, коли одержання запланованого кредиту скасовується, інформація про нього повинна бути вилучена з бази даних. Відповідно, повинні бути вилучені і всі відомості про планові платежі по цьому кредиту. Таким чином, сутність ПЛАТІЖ залежить від сутності КРЕДИТ. Діаграма для сутностей ПЛАТІЖ і КРЕДИТ наведено на рис.2.15.

Рис. 2.15. *ER* – діаграма для зв’язку між сутностями КРЕДИТ та ПЛАТІЖ

Розглянемо тепер позначення атрибутів на *ER* – діаграмах.

Нотація Чена подає атрибути у вигляді овалів, з’єднаних з прямокутниками сутностей, прямими лініями. В овал вписують імена атрибутів (рис.2.16). Багатозначний атрибут з’єднують з сутністю подвійною лінією, а похідний – пунктирною.

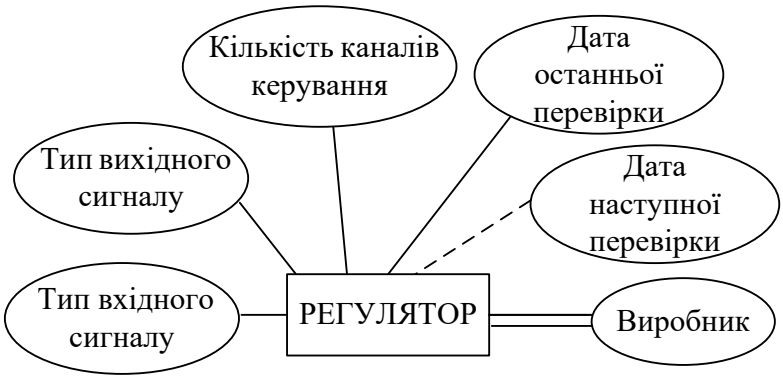


Рис. 2.16. Зображення атрибутів у нотації Чена

У нотації “пташина лапка” імена атрибутів вписують у прямокутники, приєднані знизу до прямокутників відповідних сутностей.

Багатозначні та похідні атрибути у цій нотації спеціальним чином не виділяють. На рис. 2.17 наведено зображення атрибутів у нотації “пташина лапка”.

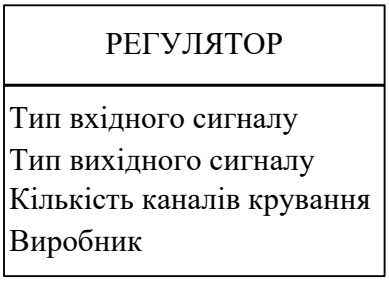


Рис. 2.17. Зображення атрибутів у нотації “пташина лапка”

При інфологічному моделюванні, як було сказано раніше, використовують й інші нотації, з доволі різноманітними способами позначення атрибутів. Оскільки етап визначення властивостей сутностей дуже відповідальний розглянемо ці способи.

Типи атрибутів відображають різними типами ліній зі стрілками та літерами. Для зображення однозначних властивостей використовують одинарну стрілку, а для багатозначних властивостей – подвійну.

Для позначення динамічних і статичних властивостей пишуть літери *D* і *S* над відповідною лінією зв’язку.

Для позначення складеної властивості зображують прямокутник, з якого виходять лінії, що сполучають його з позначеннями окремих елементів, які формують цю властивість.

Умовні атрибути сполучають з сутністю пунктирною лінією.

На рис. 2.18 умовно зображено сутність ФАХІВЕЦЬ з позначенням властивостей (див. п. 2.3).

Узагальнюючи викладений матеріал для формалізації інфологічного моделювання виділимо такі етапи:

1) визначення задач, які будуть розв’язуватися автоматизованою БД;

2) виокремлення сутностей, що пов’язані з цими задачами;

3) формування груп взаємопов’язаних сутностей;

4) визначення властивостей зв’язків: зв'язність, кардинальність і обов'язковість;

5) визначення атрибутів і можливих їхніх значень;

6) організація даних у вигляді моделі “сутність-зв'язок”;

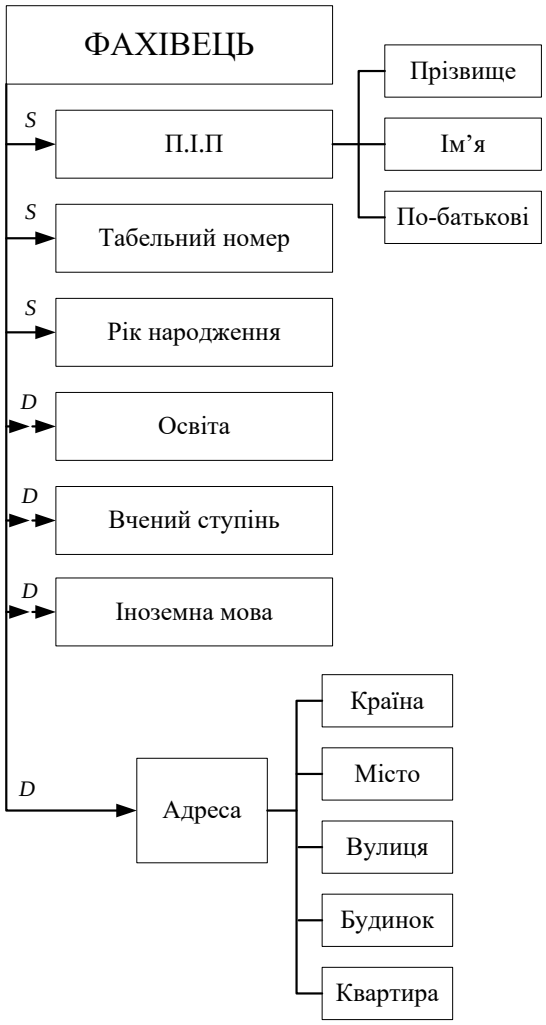


Рис.2.18. Приклад подання атрибутів для сутності ФАХІВЕЦЬ

7) формулювання зв'язків з точки зору кожної сутності, що бере в них участь;

8) виділення багатозначних атрибутів та вибір способу заміни їх однозначними (уведення додаткових сутностей, якщо це потрібно);

9) вибір нотації для представлення моделі;

10) нанесення на *ER* – діаграму сутностей, під'єднання до них атрибутів з урахуванням типів;

11) позначення зв'язків та їх параметрів;

12) перевірка відповідності *ER* – моделі задачам, поставленим при виконанні п.1.

2.5. Види даталогічних моделей

Інфологічні моделі даних використовують на початкових стадіях проєктування для опису структур даних в процесі розробки додатку користувача, а ***даталогічні моделі даних*** (див. рис.2.3) підтримуються конкретною СКБД. Будемо розглядати далі фактографічний тип даталогічних моделей, а з нього *теоретико – графові* та *теоретико – множинні моделі*. Ці моделі дають можливість відображати сукупність об'єктів реального світу у вигляді графа взаємопов'язаних інформаційних об'єктів.

Найбільш поширеними зараз є такі види даталогічних моделей [1]:

а) ієрархічні;

б) сіткові;

в) реляційні.

Ієрархічну модель БД відображають зв'язаним графом, що має вигляд перевернутого дерева, вершини якого розташовані на різних ієрархічних рівнях. При цьому одна з вершин, яка знаходиться на найвищому рівні, називається ***коренем***. Вважають, що корінь розташований на нульовому рівні. Він не підпорядковується жодній вершині. Всі інші вершини пов'язані тільки з однією вершиною, яка розташована на рівень вище, і безліччю інших вершин, розташованих на рівень нижче. Структура ієрархічної моделі наведена на рис. 2.19. За такою моделлю побудовано файлову систему.

Основними поняттями такої структури є ***рівень***, ***елемент (вузол)***, ***зв'язок***.

Дерево має *n* вершин і (*n*-1) дуг, не має циклів, кожна пара вершин з'єднується тільки однією простою дугою. ***Рівень вершини*** - це відстань від неї до кореня.

Ієрархічна модель характеризується жорсткою заданістю зв'язків між даними. Змінити зв'язки складно. Одержати інформацію можна тільки після послідовного проходження вищих рівнів ієрархії.

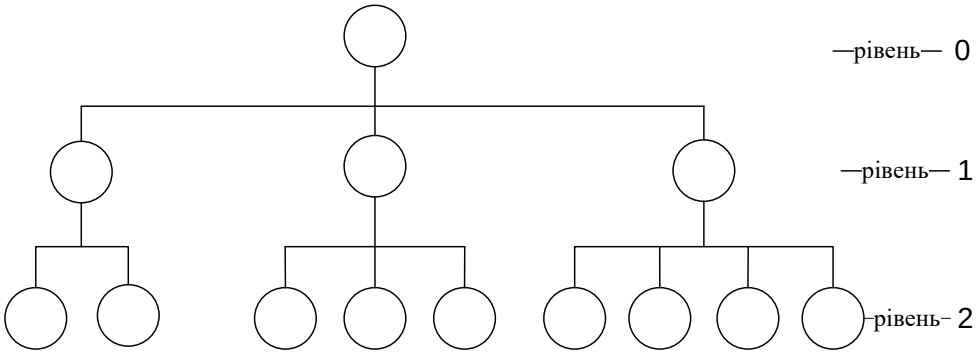


Рис.2.19. Структура ієрархічної моделі

Прикладом ієрархічної організації даних є поштова адреса:

країна – місто – вулиця – будинок – адресат.

Ієрархічну модель можна застосовувати як для логічної структури, так і для фізичної. Розглянемо приклад ієрархічної моделі БД для предметної області **ФАКУЛЬТЕТ**.

Вона може включати п'ять сегментів:

КАФЕДРА: (найменування кафедри, кількість викладачів, кількість студентів);

ВИКЛАДАЧ: (табельний номер, прізвище, посада);

ГРУПА: (номер групи, кількість студентів, староста);

РОЗКЛАД: (найменування предмету, прізвище викладача, день тижня, час, аудиторія);

СТУДЕНТ: (номер залікової книжки, прізвище, курс).

Дерево для цієї моделі може мати вигляд, наведений на рис. 2.20.

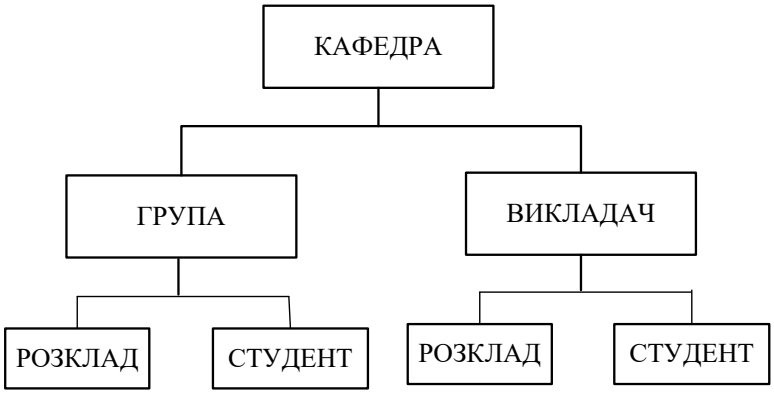


Рис.2.20. Граф ієрархічної моделі для ПО **ФАКУЛЬТЕТ**

Операції маніпулювання даними в ієрархічних системах орієнтовані, перш за все, на пошук інформації „зверху вниз”, тобто за заданим екземпляром сегменту – ”батька” можна знайти всі екземпляри сегментів – “синів”.

Зворотний же пошук (знайти екземпляр батька за заданим екземпляром сина) ускладнений, а часто і неможливий.

Вказана особливість мови маніпулювання даними призводить до того, що симетричні запити реалізуються по-різному і мають різний час виконання.

Наприклад, в схемі БД запит “*які викладачі проводять заняття з певною групою*?” реалізується простіше і швидше, ніж запит “*з якими групами веде заняття певний викладач*?”.

Ієрархічній моделі властиві і інші недоліки, наприклад, дублювання даних на логічному рівні.

Так, якщо один викладач проводить заняття одночасно з декількома групами, то відповідний екземпляр сегменту РОЗКЛАД повинен повторюватися декілька разів.

Сіткова модель має такі ж основні поняття, що й ієрархічна, але гнучкіша в порівнянні з нею. На відміну від ієрархічної моделі між її вузлами можна встановлювати множинні зв'язки. Приклад сіткової модель зображений на рис.2.21:

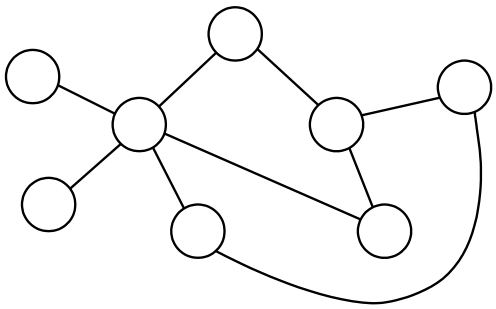


Рис. 2.21. Приклад сіткової моделі даних

Будь-який один елемент в такій моделі може бути пов'язаний з будь-яким іншим елементом. Сіткова модель дозволяє навіть цикли, в яких попередня вершина є в той же час і наступною. Такий зв'язок називають ***петлею***.

Приклад сіткової БД, що складається з п’яти сутностей (ВИКЛАДАЧ, ГРУПА, КАФЕДРА, РОЗКЛАД і СТУДЕНТ) та шести зв'язків, наведений на рис. 2.22.

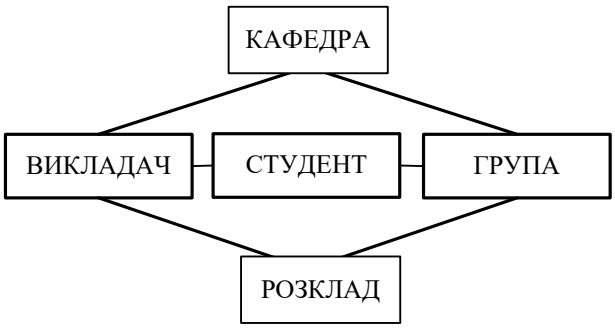


Рис. 2.22. Приклад сіткової структури для ПО **ФАКУЛЬТЕТ**

Сіткову (мережну) структуру можна перетворити в одне або декілька дерев, вводячи додаткові вершини. При цьому, якщо основою мережі є відношення не типу M:N, то не потрібно вводити нові рівні дерева.

Розглянемо мережу, що включає записи ВИКЛАДАЧ і ГРУПА, із зв'язками типу *M*:*N*. Цю мережу можна перетворити у дві ієрархічні мережі, записи яких зв'язані як 1:*N* і 1:*M* (див. рис.2.23).

Сіткові СКБД можуть підтримувати різноманітні моделі даних.

Модель БД є графом з вершинами двох типів. Цим двом типам вершин відповідають основні і залежні записи. На фізичному рівні говорять про основні і залежні файли.

Зв'язки можуть встановлюватися тільки між основними і залежними записами, причому залежний запис не може існувати самостійно, без зв'язків з основними файлами.

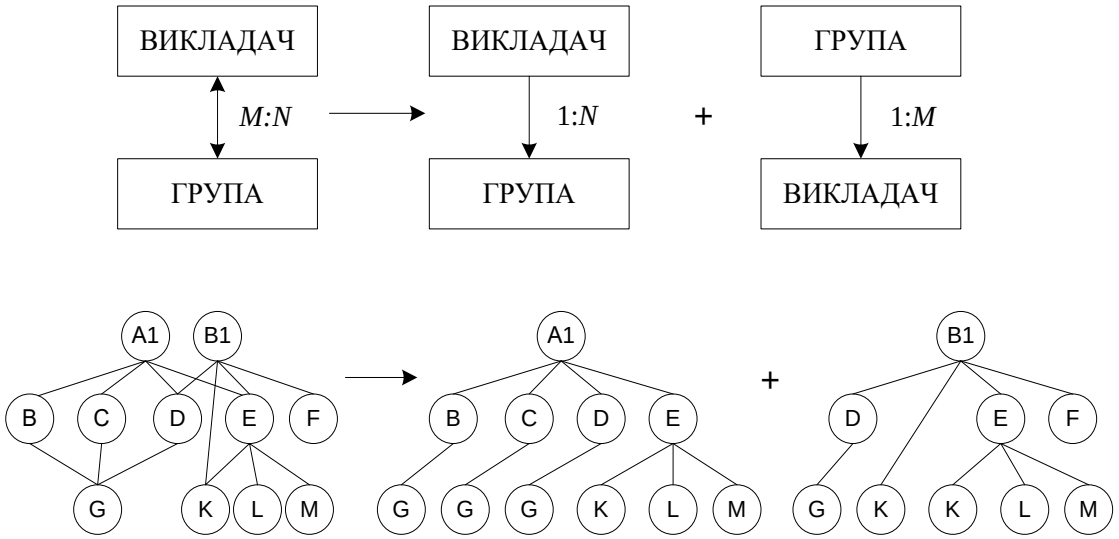


Рис. 2.23. Схема перетворення сіткової структури в ієрархічну

Перерахуємо основні операції пошуку даних у такій системі:

- послідовний перегляд записів основного файлу;

- перегляд всіх записів залежного файлу, пов'язаних з конкретним записом основного файлу;

- прямий пошук запису основного файлу по її ключу.

Крім того, в залежному файлі зберігаються ключі всіх основних записів, пов'язаних з цим файлом.

Отже, без особливих труднощів можна знайти запис основного файлу, зв'язаний певним чином з конкретним записом залежного файлу.

Сіткові моделі володіють перевагами в порівнянні з ієрархічними.

Зокрема, зменшується дублювання інформації, симетричні запити реалізуються за допомогою схожих алгоритмів.

Реляційну модель (***relation*** (англ.) - відношення) запропонував в 1970 р. Э.Ф. Кодд (*Edgar Frank Codd* – математик, інформатик, засновник теорії реляційних баз даних; народився в 1923 р. в Англії, працював у США та Канаді, відзначений премією Тюрінга.). Основною структурою даних в цій моделі є ***відношення***. Граф реляційної моделі зображений на рис. 2.24.

Реляційна база даних (РБД) має вигляд двовимірної таблиці. Таблиця зрозуміла людині і цілком осяжна для неї.

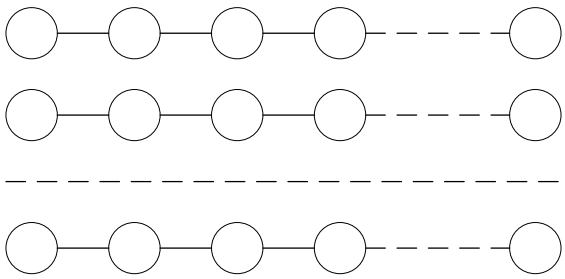


Рис. 2.24. Граф реляційної моделі даних

Далі зупинимось на реляційній моделі детальніше, оскільки вона є найпоширенішою у сучасних СКБД.

Запитання та завдання для самостійної роботи

1. Дайте визначення поняттю “Модель даних”.
2. Розкрийте поняття концептуальної моделі даних та моделі реалізації.
3. Поясніть класифікацію моделей даних, яку запропоновано *ANSI — SPARC*.
4. Розкрийте поняття інфологічної моделі даних. Які складові містить інфологічна модель?
5. Як співвідносяться поняття *сутність, тип сутності*, *екземпляр сутності*?
6. Наведіть приклади цих понять з Вашої спеціальності та навчального процесу.
7. Розкрийте поняття *атрибут*. Наведіть синоніми цього слова. Які є типи атрибутів? Наведіть приклади.
8. Розкажіть про типи зв’язків між сутностями. Наведіть приклади з Вашої спеціальності та навчального процесу.
9. Розкрийте поняття *ER* - моделі.
10. Як називають графічний спосіб подання *ER* – моделі?
11. Порівняйте зображення сутностей у нотаціях Чена і “пташина лапка”.
12. Як відображають смисл зв’зку між сутностями у нотаціях Чена і “пташина лапка”?
13. Як зображують властивості зв’язків у нотаціях Чена і “пташина лапка”?

14. Як зображують атрибути у нотаціях Чена і “пташина лапка”?

15. Опишіть способи зображення атрибутів у інших нотаціях.

16. Опишіть типи та зобразіть графи даталогічних моделей.

17. На рис. 2.25, 2.26 наведено *ER* – діаграми [2]. З кожною з них виконати такі завдання:

а) назвати сутності, які зображено на діаграмі, та дати характеристику кожної сутності по відношенню до іншої;

б) назвати атрибути сутностей, визначити їхні типи;

в) описати зв’язки між сутностями;

г) назвати задачі, які можуть бути розв’язані за допомогою бази даних, створеної на основі діаграми;

д) зобразити діаграму у нотації Чена (якщо вона не в цій нотації);

е) зобразити діаграму у нотації “пташина лапка” (якщо вона не в цій нотації).

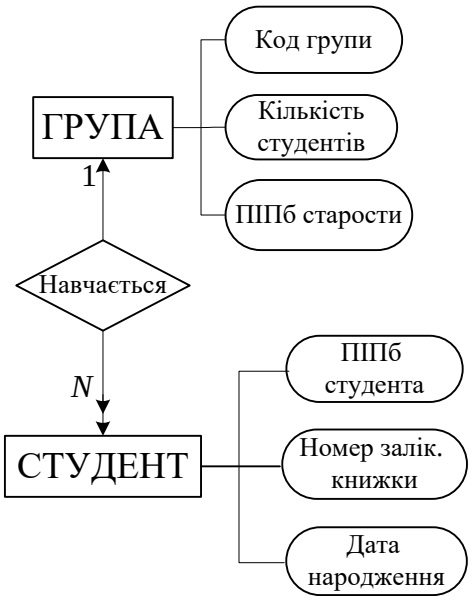


Рис. 2.25. Діаграма для сутностей СТУДЕНТ та ГРУПА

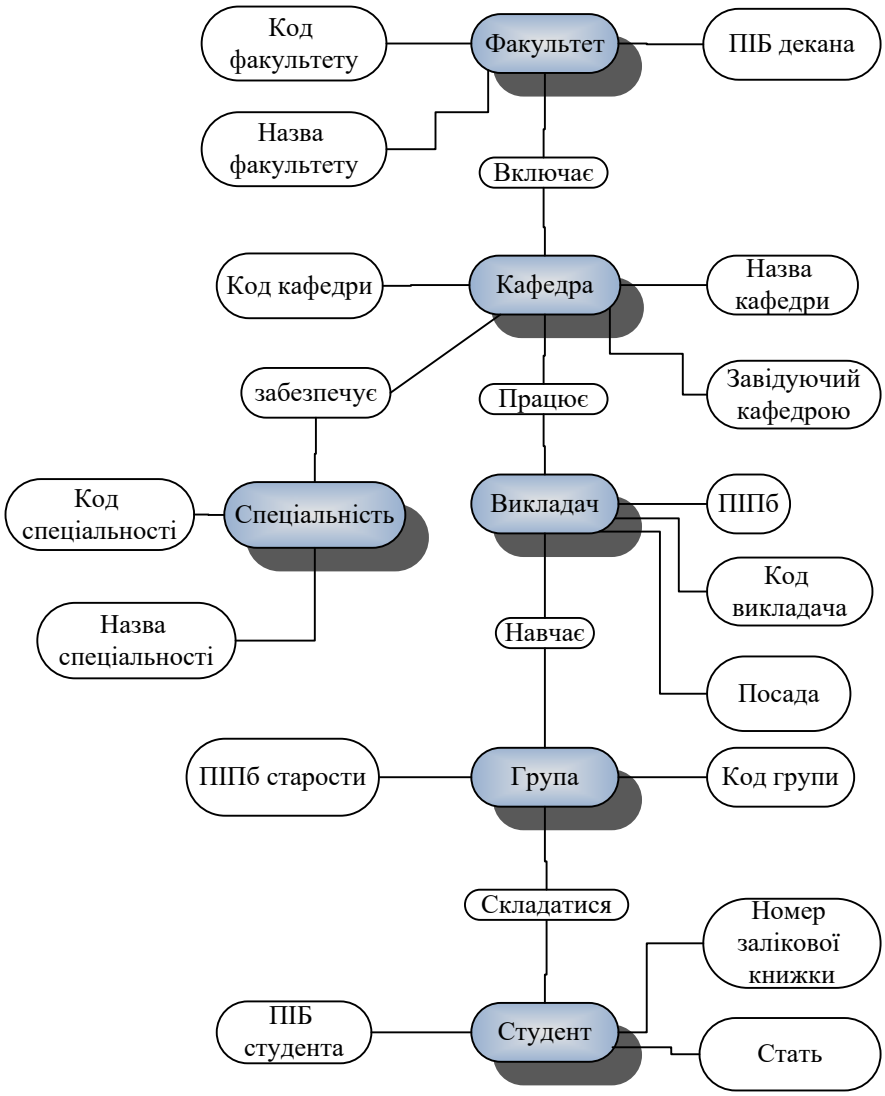


Рис. 2.26. Діаграма для сутностей предметної області ДЕКАНАТ

18. Зобразити сутність та її атрибути, наведені на рис. 2.18, у відповідності до нотацій Чена та “пташина лапка”.

3. ПРОЄКТУВАННЯ РЕЛЯЦІЙНИХ БАЗ ДАНИХ

3.1. Поняття реляційної бази даних. Структурні характеристики відношень

Певні поняття змінюють свої назви в залежності від етапу проєктування БД. Тому важливо правильно орієнтуватися у множині термінів. Реляційна модель передбачає використання специфічної термінології [12 – 14].

Реляційна база даних – це набір взаємопов'язаних відношень. Відношення реляційних БД бувають двох класів: об’єктні та зв’язані.

Поняття ***об’єктного відношення*** використовують в його загальноприйнятому математичному сенсі. Для заданих множин $S_1, S_2, ..., S_n$ (не обов'язково різних) R є відношенням на цих n множинах, якщо є набором кортежів степеня n, у кожного з яких перший елемент узятий з множини S_1, другий – з множини S_2 і т. д.

На рис. 3.1 зображено структуру реляційного об’єктного відношення R з позначенням множин $S_1 - S_n$.

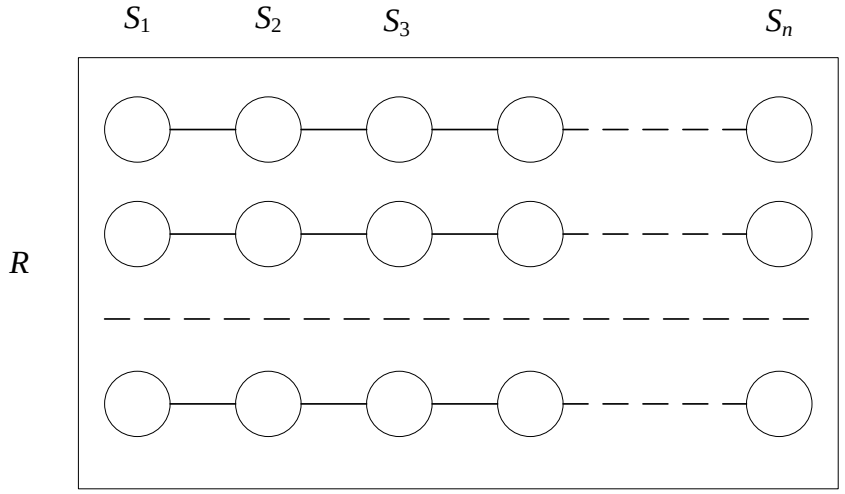


Рис. 3.1. Структура реляційного об’єктного відношення R з позначенням множин $S_1 - S_n$

Степінь відношення – це кількість його атрибутів. Відношення степеня 1 часто називають *унарним*, степеня 2 – *бінарним*, степеня 3 – *тернарними* і степеня n – n - *арним*. Множини $S_1, S_2, ..., S_n$ називають ***доменами***, отже ***домен*** – це певна множина значень окремого атрибуту (поняття домену наведено вперше у п.2.3). Множина R є в цьому випадку ***декартовим добутком множин*** $S_1, S_2, ..., S_n$, тобто набором всіляких поєднань з n елементів, де кожен елемент беруть з свого домену.

Відношення зручно подати як таблицю, де кожен рядок – це кортеж, а кожен стовпець відповідає одному компоненту декартового добутку. Наприклад, для опису сутності (інфологічне моделювання) СТУДЕНТ використовують об'єктне відношення (даталогічне моделювання) СТУДЕНТ, у якому властивості сутності розташовуються у стовпцях таблиці (табл.3.1).

Список імен атрибутів відношення називається ***схемою відношення***.

Таблиця 3.1. Таблиця відношення СТУДЕНТ

Прізвище, ініціали.	Дата_народження	Курс	Спеціальність
Іванов І.С.	12.01.2004	2	Автоматизація
Ткаченко В.Т.	15.05.2002	4	Машини і апарати
Федорчук Г.Т.	31.08.2003	3	Промислова екологія

Схема відношення СТУДЕНТ така:

СТУДЕНТ (*Прізвище_ініц***,** *Дата_народження***,** *Курс***,** *Спеціальність***).**

Кожна БД подається в ЕОМ у вигляді файлу. При роботі з файлами виникають поняття поля та запису.

Поле – це одиниця логічної організації даних, яка відповідає атрибуту. Поля мають імена, типи та інші характеристики.

Запис – це сукупність логічно пов’язаних полів.

Потужність відношення (***кардинальне число***) – кількість кортежів у відношенні.

У табл.3.2 вказано відповідності між поняттями, які відповідають різним етапам проєктування реляційної БД:

Таблиця 3.2. Відповідності між поняттями етапів проєктування БД

Інфологічне моделювання	Даталогічне моделювання	Даталогічне моделювання і реалізація на ЕОМ	Реалізація на ЕОМ
сутність	відношення	таблиця	файл
екземпляр сутності	кортеж	рядок	запис
атрибут	домен	стовпець	поле

Зв’язане відношення складається з декількох взаємопов’язаних об’єктних відношень. Приклади цих відношень можна знайти у п. 2.3 при описі зв’язків між сутностями, зокрема,

ТЕХНОЛОГІЧНИЙ ОБ’ЄКТ КЕРУВАННЯ $\twoheadrightarrow$ МОДЕЛЬ.

Е.Ф. Кодд показав, що набір відношень (таблиць) може бути використаний для зберігання даних про об'єкти реального світу і моделювання зв'язків між ними.

Таблиці можна розглядати як відношення за таких умов:

1. Усі рядки повинні мати одну і ту саму структуру, тобто одну і ту саму кількість атрибутів з однаковими іменами.

2. Імена стовпців повинні бути різні, а значення в кожному стовпці повинні бути однотипні (однорідні).

3. Значення атрибутів повинні бути елементарними, тобто не включати інші відношення.

4. У таблиці не може бути однакових рядків (записів).

5. Порядок розташування рядків у таблиці не має значення, оскільки впливає лише на швидкість доступу до рядка.

6. Повинна дотримуватися посилальна цілісність для зовнішніх ключів.

Таблиці зручні, але не враховують необхідності узгодження трьох способів маніпулювання даними:

- впорядкування;
- угрупування по значенню індексів;
- доступ по дереву параметрів.

У таблицях всі три способи маніпулювання жорстко пов'язані між собою – упорядковані по одному параметру дані не впорядковані по іншому. Таким чином, полегшується доступ до потрібних даних по одному ланцюжку, а по іншому пошук ускладнюється.

Оригінальність підходу Е.Ф. Кодда полягала в тому, що він запропонував застосовувати до відношень (таблиць) таку систему операцій, яка дозволяє одержувати (обчислювати подібно до арифметичних операцій) одні відношення з інших. Це дало можливість ділити інформацію на таку, що зберігають, і таку, що розраховують. Такий підхід економить пам'ять ЕОМ.

3.2. Типи даних

Тип даних (*Data Type*) визначає допустиму множину значень, які може приймати змінна величина. При виборі типу враховують і операції, які виконуватимуть з цими значеннями. Конкретні значення полів випливають зі змісту конкретної задачі. У кожній СКБД передбачено використання декількох типів даних. Розглянемо основні типи, передбачені у СКБД *MS Access*.

Текстовий *(Text)* — тип даних за умовчанням, його можна надавати значенням (текст або цифри), які не беруть участі в розрахунках.

Наприклад, назва приладу, номер телефону тощо. Кількість символів у полі не повинно перевищувати 255. Користувач може задати необхідну для його задачі кількість символів.

Поле ***МЕМО*** *(Memo).* Його надають текстовим даним завдовжки до 64 000 символів. Наприклад, інструкція по експлуатації приладу.

Числовий *(Number)* тип застосовують для числових даних, які передбачають використовувати у математичних обчисленнях. Конкретні варіанти числового типу (*Байт*, *Ціле*, *Довге ціле* тощо) користувач має можливість вибрати при роботі з СКБД.

Для проведення грошових розрахунків визначений інший тип даних —***Грошовий*** (*Currency*). Цей тип надає можливість зберігати точні значення великих чисел – 15 знаків до коми і 4 знаки після неї.

Дата/час *(Date/Time).* Цей тип передбачає різні форми подання дати та часу.

Лічильник *(AutoNumber).* У поле такого типу для кожного нового запису автоматично вводяться унікальні цілі, що послідовно зростають

на 1 або випадкові числа. Значення цього поля не можна змінити або видалити, довжина поля складає 4 байти. За умовчанням в поле вводяться послідовні значення.

Логічний *(Yes/No*). До такого типу належать дані, які можуть мати одне з двох можливих значень *Так/Ні* (*Істина/Неправда, Увімкнено/Вимкнено*). Довжина поля 1 біт.

Поле об'єкту ***OLE*** *(OLE Object).* Об'єкт (наприклад, електронна таблиця *Microsoft Excel,* документ *Microsoft Word,* малюнок, звукозапис або інші дані в двійковому форматі), зв'язаний або упроваджений в таблицю *MS Access.* Для полів типів *OLE* та *MEMO* не допускається сортування.

Поля з типом даних ***Гіперпосилання*** (*Hyperlink*) використовують для переходу до документів, створених у різноманітних додатках *Microsoft Office*, що зберігають у файлах на локальних чи сітьових дисках, зокрема, до інших баз даних, до сторінок *Web*-серверів в Інтернет чи Інтранет.

У полях *MS Access* гіперпосилання записують як *URL*- чи *UNC*-адреси. Відображають у полі у вигляді підкресленого тексту, заданого при визначенні гіперпосилання. При клацанні мишкою на гірперпосиланні відбувається перехід до документу, який відкривається при цьому створившим його додатком.

Поле ***Вкладення*** *MS Access* (*Attachment*) можна використовувати для зберігання декількох файлів в одному полі, причому в цьому полі можна зберігати файли різних типів. Наприклад, у таблиці ТЗА у полі цього типу можна подати декілька файлів з відомостями про пристрій: сертифікат походження, сертифікат якості, інструкцію з експлуатації, рекламації, фотографії, схеми і так далі. Вкладати файли можна тільки в бази даних, створені в додатку *MS Access* 2007/2010 в новому форматі файлу *accdb*.

Таке поле пропонують для зберігання зображень і документів замість типу даних OLE (OLE Object). Вкладенням потрібно значно менший обсяг пам’яті для зберігання, вони не вимагають застосування програм, званих OLE-серверами.

Тип ***Обчислюваний*** (*Calculated*) призначений для створення обчислюваних полів: числових, текстових, грошових, дата/час, логічних. Значення обчислюваного поля визначається виразом, записаним у полі.

Цей вираз використовує інші поля поточного запису, деякі вбудовані функції і константи, зв'язані арифметичними, логічними або рядковими операторами.

Майстер підстановок*.(Lookup Wizard.).* Вибір цього типу даних запускає інструмент СКБД, який називають *Майстром підстановок. Майстер* дозволяє формувати перелік можливих значень однієї таблиці або зі значень іншої або з деякого списку. Наприклад, у список можна увести стандартні електричні чи пневматичні сигнали на виході перетворювачів, певні відомості про фірми – виготовник ТЗА, класи точності пристроїв тощо.

Наприклад, якщо заповнюють таблицю для відношення АВТОМАТИЧНА СИСТЕМА КЕРУВАННЯ і планують вказувати типи первинних перетворювачів тиску, то можна заздалегідь створити список можливих типів і далі тільки вибирати необхідне зі створеного списку, а не вписувати кожний раз з клавіатури.

Відповідно, тип даних поля створюваної таблиці визначатиметься типом даних поля списку. Крім того, ***Майстер підстановок*** дозволяє визначити зв'язок таблиць і включити перевірку зв'язної цілісності даних.

Нові версії СКБД можуть дещо модернізувати перелік типів даних.

3.3. Ключі відношень. Встановлення зв'язків між відношеннями

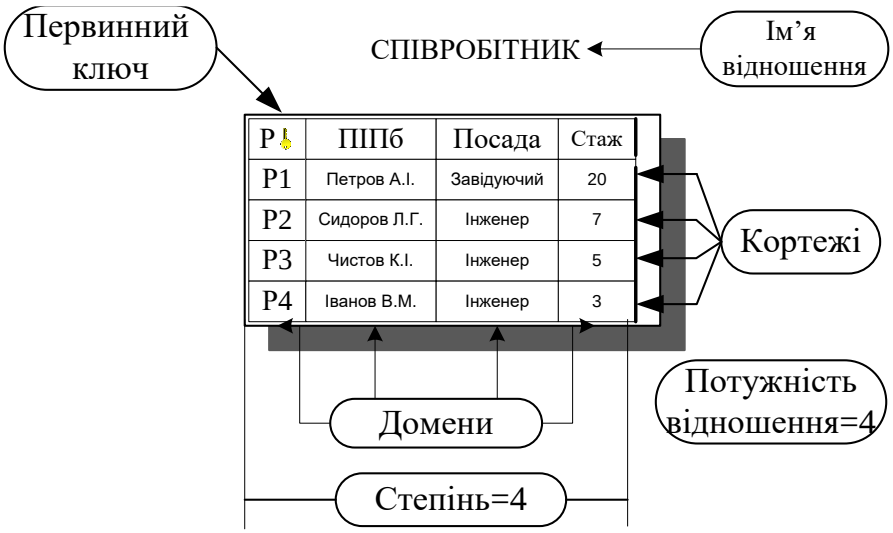
Один домен (або комбінація доменів) відношення повинен містити значення, що дозволяють однозначно ідентифікувати кожен кортеж цього відношення. Такий домен (або комбінацію) називають ***первинним ключем*** (*Primary Key*). Первинний ключ є *не надмірним*, якщо він є або простим доменом (не комбінацією), або такою комбінацією, при якій жоден з вхідних в неї доменів не зайвий.

При створенні БД можливі ситуації, коли декілька доменів (або груп доменів) задовольняють вимогам до первинного ключа. Зазвичай їх називають *кандидатами* і вибирають один з них.

За первинний ключ рекомендують вибирати числове поле, оскільки обробка такого типу полів відбувається набагато ефективніше, ніж текстових.

Вибір складеного первинного ключа з декількох текстових полів може серйозно погіршити продуктивність СКБД.

У випадку, якщо первинний ключ підібрати складно, або він досить громіздкий, рекомендують створити штучний ідентифікатор у вигляді поля типу "довге ціле" і заповнити його унікальними значеннями, що кодують записи початкової таблиці. У СКБД *MS Access* для цього передбачений спеціальний тип даних "Лічильник". Це поле нагадує поле типу "довге ціле". Його особливістю є те, що СКБД автоматично надає номер кожному новому запису і відслідковує унікальність нумерації. На рис. 3.2 зображено об'єктне відношення СПІВРОБІТНИК з позначеннями його характеристик [2].



Р	ПІПб	Посада	Стаж
Р1	Петров А.І.	Завідуючий	20
Р2	Сидоров Л.Г.	Інженер	7
Р3	Чистов К.І.	Інженер	5
Р4	Іванов В.М.	Інженер	3

Рис. 3.2. Зображення відношення СПІВРОБІТНИК з позначеннями його характеристик

Усі атрибути відношення, які не входять у первинний ключ, називають ***описовими*** (див. п.2.3).

При створенні БД часто використовують поняття "інформаційний об'єкт".

Інформаційний об'єкт – це опис певної сутності у вигляді сукупності логічно пов'язаних атрибутів. Він має таке ж ім'я, що і відповідна сутність, наприклад, СТУДЕНТ, ФАХІВЕЦЬ тощо. Цей об'єкт має сукупність реалізацій і ідентифікується значенням ключа.

Зв'язки між таблицями дають можливість сумісно використовувати дані з різних таблиць. У п. 2.3 були перелічені види зв'язку між двома сутностями, які треба враховувати при їх зв'язуванні.

Одну з пов'язаних таблиць назвемо *головною*, іншу – *зв'язуваною*. Головною є таблиця з одноразовою зв'язністю (1), а зв'язуваною – таблиця з багатозначною зв'язністю (М). Обидві вони мають власні первинні ключі.

Для забезпечення зв’язку між цими таблицями у зв’язуваній таблиці потрібно передбачити зовнішній ключ (***ключ зв’язку***, *Foreign Key*), так званий *допоміжний атрибут* (див. п. 2.3). Отже, зовнішній, як і первинний ключ, може бути як простим (одне поле), так і складеним (декілька полів). Нагадаємо, що поля зовнішнього ключа мають той же склад (смисл і тип полів), що і первинний ключ головної таблиці, а назви відповідних полів можуть відрізнятися.

Значення первинного ключа у головній таблиці повинні бути унікальні для різних записів, тоді як значення зовнішнього ключа у зв’язаній таблиці можуть повторюватися багато разів.

Сукупність двох відношень, поєднаних за допомогою ключа зв’язку, називають ***зв’язаним відношенням***. Наведемо приклади їх створення.

Встановлення зв’язку “*1:1*” (“*один до одного*”) *між таблицями А і В*. У цьому випадку зв’язуваності обох таблиць дорівнюють 1. В одній з таблиць (наприклад, *В*) треба створити зовнішній ключ, який посилатиметься на первинний ключ іншої таблиці (*А*). Цей зовнішній ключ водночас є також первинним ключем *В* або її індексованим полем (у ньому повторення значень не припустимі). На рис. 3.3 наведена схема даних для таблиць СИРОВИНА (*А*) та ВЛАСТИВОСТІ СИРОВИНИ (*В*), на рис. 3.4 показана робота з цими таблицями – з таблиці СИРОВИНА можна побачити властивості кожної сировини.

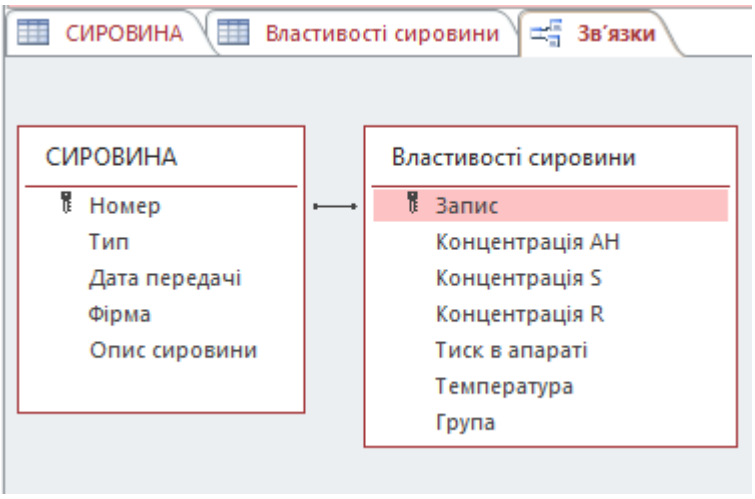


Рис. 3.3. Схема даних для зв'язку типу "1:1"

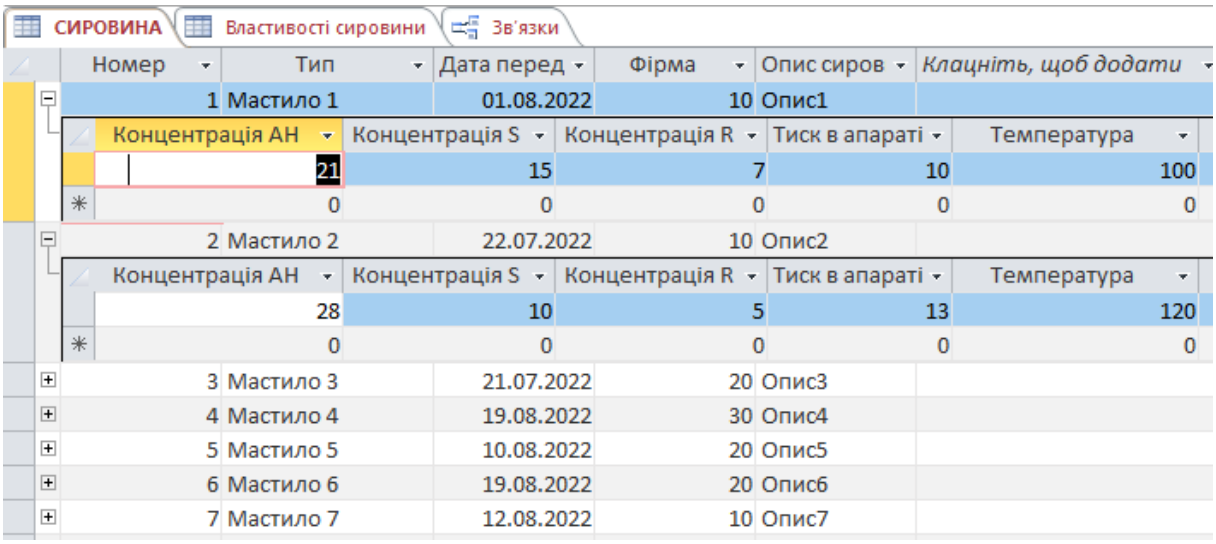


Рис. 3.4. Зразок використання таблиць при зв'язку типу "1:1"

Встановлення зв'язку "1:M" ("один до багатьох") між таблицями A і B. На рис. 3.5 наведена схема даних для таблиць ПОСТАЧАЛЬНИК (*A*) та СИРОВИНА (*B*), на рис. 3.6 показана робота з цими таблицями – з таблиці ПОСТАЧАЛЬНИК можна побачити різні види сировини, які можна придбати у постачальників.

У цьому прикладі розглядаємо ситуацію на ринку сировини, коли кожний постачальник реалізує власну сертифіковану сировину, якої не має інший постачальник.

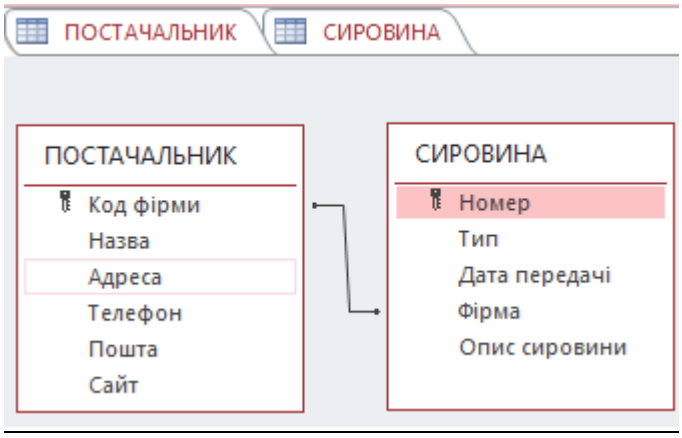


Рис. 3.5. Схема даних для зв’язку типу “1:*M*”

Код фірми	Назва	Адреса	Телефон	Пошта
110	Ф1	Адреса1	Телефон1	ааа@111

Номер	Тип	Дата перед	Опис сиров	Клас
1	Мастило 1	01.08.2022	Опис1	
2	Мастило 2	22.07.2022	Опис2	
7	Мастило 7	12.08.2022	Опис7	
(Новий)				

120	Ф2	Адреса2	Телефон2	bbb@111

Номер	Тип	Дата перед	Опис сиров	Клас
3	Мастило 3	21.07.2022	Опис3	
5	Мастило 5	10.08.2022	Опис5	
6	Мастило 6	19.08.2022	Опис6	
(Новий)				

130	Ф3	Адреса3	Телефон3	ccc@111

Номер	Тип	Дата перед	Опис сиров	Клас
4	Мастило 4	19.08.2022	Опис4	

Рис. 3.6. Зразок використання таблиць при зв’язку типу “1:*M*”

Встановлення зв’язку “N:M” (“багато до багатьох”).між таблицями A і B. Реляційна модель даних не підтримує цей зв’язок, тому його реалізують через два зв’язки типу “1:*M*”. Отже, в цьому випадку необхідно створити допоміжну таблицю *C* і зв’язками “один до багатьох” приєднати до неї обидві таблиці *A* і *B*.

У цьому прикладі розглядаємо сутності попереднього прикладу, але іншу ситуацію на ринку сировини, а саме, коли постачальники можуть реалізувати однакові сировини. Створимо додаткову таблицю *C* під назвою “ПОСТАЧАННЯ”. Обов'язковими її полями повинні бути тільки поля зв’язку, інші поля можна вводити в її склад за необхідністю.

На рис. 3.7 наведена схема даних для таблиць ПОСТАЧАЛЬНИК (*A*), СИРОВИНА (*B*) та “ПОСТАЧАННЯ”(*C*), на рис. 3.8 показано таблицю “ПОСТАЧАННЯ”. На рис. 3.9 та 3.10 показана робота з БД з боку таблиць як *A*, так і *B*.

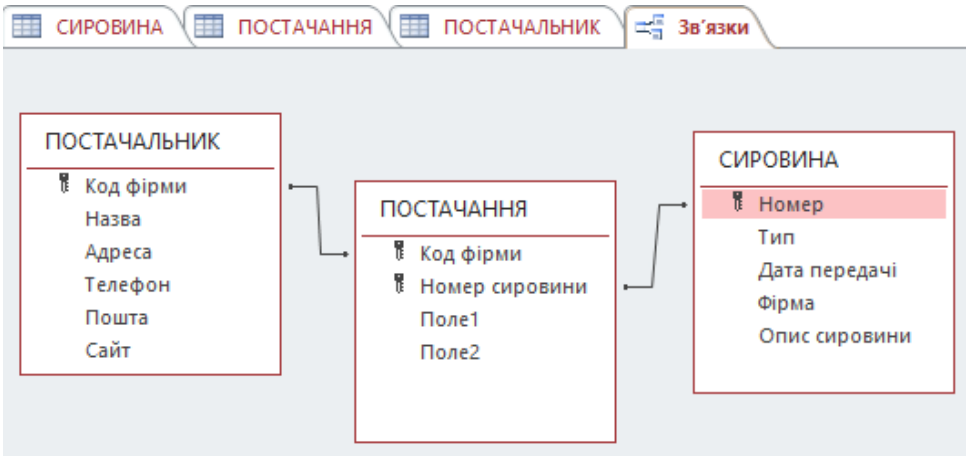


Рис. 3.7. Схема даних для реалізації зв’язку типу “*N*:*M*”

СИРОВИНА | ПОСТАЧАННЯ | ПОСТАЧАЛЬНИК | Зв'язки

Код фірми	Номер сирс	Поле1	Поле2	Кла
110	1	Ф1	Мастило 1	
110	2	Ф1	Мастило 2	
110	7	Ф1	Мастило 7	
120	3	Ф2	Мастило 3	
120	5	Ф2	Мастило 5	
120	6	Ф2	Мастило 6	
130	4	Ф3	Мастило 4	

Рис. 3.8. Вид додаткової таблиці “ПОСТАЧАННЯ”

СИРОВИНА | ПОСТАЧАННЯ | ПОСТАЧАЛЬНИК | Зв'язки

Код фірми	Назва	Адреса	Телефон	Пошта
110	Ф1	Адреса1	Телефон1	aaa@111
120	Ф2	Адреса2	Телефон2	bbb@111
130	Ф3	Адреса3	Телефон3	ccc@111

Номер сирс	Поле1	Поле2	Клацніть, ι
1	Ф1	Мастило 1	
2	Ф1	Мастило 2	
7	Ф1	Мастило 7	
0			

Номер сирс	Поле1	Поле2	Клацніть, ι
3	Ф2	Мастило 3	
5	Ф2	Мастило 5	
6	Ф2	Мастило 6	
0			

Рис. 3.9. Зразок використання таблиць при зв'язку типу "*N*:*M*" з таблиці "ПОСТАЧАЛЬНИК"

СИРОВИНА | ПОСТАЧАННЯ | ПОСТАЧАЛЬНИК | Зв'язки

Номер	Тип	Дата перед	Фірма	Опис сиров
1	Мастило 1	01.08.2022	110	Опис1
2	Мастило 2	22.07.2022	110	Опис2
3	Мастило 3	21.07.2022	120	Опис3
4	Мастило 4	19.08.2022	130	Опис4

Код фірми	Поле1	Поле2	Клацніть, ι
110	Ф1	Мастило 1	
0			

Код фірми	Поле1	Поле2	Клацніть, ι
110	Ф1	Мастило 2	
0			

Код фірми	Поле1	Поле2	Клацніть, ι
120	Ф2	Мастило 3	
0			

Рис. 3.10. Зразок використання таблиць при зв'язку типу "*N*:*M*" з таблиці "СИРОВИНА"

Значення первинного ключа у головній таблиці повинні бути унікальні для різних записів, тоді як значення зовнішнього ключа у зв'язаній таблиці можуть повторюватися багато разів.

3.4. Цілісність та достовірність даних

Цілісність даних означає, що в базі даних встановлено і коректно підтримуються взаємозв'язки між записами різних таблиць при завантаженні, додаванні і видаленні записів у зв'язаних таблицях, а також при зміні значень ключових полів.

Цілісність бази даних (*Database Integrity*) — це відповідність інформації, що знаходиться в базі даних, її внутрішній логіці, структурі і всім явно заданим правилам. Кожне правило, що накладає деяке обмеження на можливий стан бази даних, називається ***обмеженням цілісності*** (*Integrity Constraint*). Приклади обмежувальних правил наступні: вага деталі має бути додатною; кількість знаків у телефонному номері не повинна перевищувати 10; вік батьків не може бути менше віку їхньої дитини і так далі. Перелік обмежень визначає розробник БД.

Для забезпечення цілісності СКБД повинна перевіряти наступну умову: «*зовнішній ключ зв'язаної таблиці не може містити значень, яких не містить ключ головної таблиці»*. Активування цієї опції при налаштуванні СКБД виконують встановленням прапорця "Забезпечення цілісності даних" при створенні зв'язку між таблицями у діалоговому вікні "Редагування (зміна) зв'язків" [14]. При такому налаштуванні СКБД буде виникати повідомлення про помилку при порушенні зазначеної вище умови.

Передбачено, що зв'язки, сформовані з підтримкою цілісності, будуть позначені символами множинності зв'язків (1 або $\propto$) на лініях схеми даних.

Зазначимо, що у діалоговому вікні "Редагування зв'язків", про яке йшлося раніше, є також прапорці "Каскадне оновлення пов'язаних полів" та "Каскадне видалення пов'язаних полів".

Ця опція передбачає, що видалення певного запису з головної таблиці автоматично викликає видалення пов’язаних з ним записів з підпорядкованих таблиць. Якщо “Каскадне видалення пов’язаних полів” не активовано, то автоматичне видалення не відбувається.

Цілісність БД не гарантує достовірності інформації, що міститься в ній, але забезпечує принаймні правдоподібність цієї інформації, відкидаючи вочевидь неймовірні, неможливі значення.

Достовірність БД (або істинність) є відповідність фактів, що зберігаються в базі даних, реальному світу.

Очевидно, що для визначення достовірності БД потрібне володіння повними знаннями як про вміст БД, так і про реальний світ. СКБД може і повинна контролювати цілісність БД, але принципово не в змозі контролювати достовірність БД. Контроль достовірності БД може бути в обмеженому обсязі покладений тільки на людину.

Отже, БД може бути цілісною, але не достовірною. Можливо і зворотне: БД може бути достовірною, але не цілісною. Останнє має місце, якщо правила (обмеження цілісності) задані неправильно.

3.5. Операції з відношеннями

Основними операціями з відношеннями у реляційній базі даних є такі:

- традиційні операції над множинами: об'єднання; перетин; віднімання (різниця); декартовий добуток; ділення;
- спеціальні реляційні операції: проекція; з'єднання; вибір (селекція).

Традиційні операції виконують над двома відношеннями.

Результатом операції ***об'єднання*** є нове відношення, що вміщує всі записи першого відношення і ті записи другого, які були відсутні у першому.

Результатом операції ***перетину*** є нове відношення, що вміщує тільки ті записи, які присутні одночасно в обох відношеннях.

Результатом операції ***віднімання*** є нове відношення, що вміщує ті записи першого відношення, яких немає у другому відношенні.

Операція ***декартового добутку*** передбачає, що степінь кінцевого відношення дорівнює сумі степенів первинних відношень, а потужність – добутку їх потужностей. Результат містить всі можливі комбінації з записів первинних відношень (див. табл. 3.3 – 3.5).

Таблиця 3.3. Відношення 1

Прізвище та ініціали студента
Тесленко Л.Д.
Франчук Я.М
Ковальова Т.М.

Таблиця 3.4. Відношення 2

Предмет	Дата екзамену
Математика	10.01.2019
Фізика	15.01.2019

Таблиця 3.5. Відношення – результат (декартовий добуток)

Прізвище та ініціали студента	Предмет	Дата екзамену
Тесленко Л.Д.	Математика	10.01.2019
Тесленко Л.Д.	Фізика	15.01.2019
Франчук Я.М	Математика	10.01.2019
Франчук Я.М	Фізика	15.01.2019
Ковальова Т.М.	Математика	10.01.2019
Ковальова Т.М.	Фізика	15.01.2019

При операції ***ділення*** відношення дільника повинно містити підмножину атрибутів відношення діленого. Кінцеве відношення містить тільки ті записи декартового добутку кінцевого відношення з дільником, які знаходяться в діленому.

Крім того, кінцеве відношення буде містити тільки ті атрибути діленого, яких немає в дільнику.

Так, якщо розділити *Відношення – результат* (табл.3.5) попереднього прикладу на *Відношення 2* (табл.3.4), то отримаємо *Відношення 1* (табл.3.3).

Розглянемо спеціальні реляційні операції. Операцію ***проекції*** виконують над одним відношенням. Результат – це відношення, що включає частину атрибутів початкового відношення, на які виконують проекцію.

Нехай є відношення з переліком прізвищ службовців, їхніми посадами та номерами відділів, у яких вони працюють (табл.3.6). Треба визначити перелік посад кожного відділу (результат – табл.3.7).

Таблиця 3.6. Початкове відношення

Прізвище та ініціали	Номер відділу	Посада
Коваленко Л.Д.	2	Технік
Братюк А.П.	1	Економіст
Жирнов В,В,	1	Інженер
Сайчук В.М	2	Технік

Таблиця 3.7. Відношення – результат (проекція)

Номер відділу	Посада
1	Економіст
1	Інженер
2	Технік

Операцію ***з'єднання*** виконують над двома відношеннями (табл.3.8,3.9). Вони мають містити хоча б один спільний атрибут (тут – "Код студента"). Відношення – результат (табл.3.10) матиме усі атрибути обох відношень.

Таблиця 3.8. Відношення 1

Спеціальність	Код студента
Автоматика	2
Менеджмент	3
Право	5

Таблиця 3.9. Відношення 2

Прізвище та ініціали студента	Код студента	Курс
Ковальчук Я.Д.	2	2
Бражко М.П.	3	4
Васяник В.В.	5	1
Салов В.М	8	5

Таблиця 3.10. Відношення – результат (з'єднання)

Спеціальність	Код студента	Прізвище та ініціали студента	Курс
Автоматика	2	Ковальчук Я.Д.	2
Менеджмент	3	Бражко М.П.	4
Право	5	Васяник В.В.	1

Операцію ***вибору*** виконують над одним відношенням (початкове відношення у табл.3.11).

Кінцеве відношення містить тільки ті записи, які відповідають певній умові з вибраного атрибуту. Нехай треба вибрати тільки ті записи БД, у яких середній бал успішності менше ніж 3,5. Відношення – результат наведено у табл.3.12.

Таблиця 3.11. Початкове відношення

Прізвище та ініціали студента	Середній бал
Коваленко Л.Д.	3,1
Братюк А.П.	4,5
Жирнов В,В,	3,4
Сайчук В.М	3,8

Таблиця 3.12. Відношення – результат (вибір)

Прізвище та ініціали студента	Середній бал
Коваленко Л.Д.	3,1
Жирнов В,В,	3,4

3.6. Нормалізація відношень

При проєктуванні бази даних потрібно зробити так, щоб структура моделі даних не суперечила семантиці (сенсу) предметної області. Модель повинна бути ефективною, тобто мати мінімальне дублювання даних. Це буде зручним при їх обробці і оновленні. Мінімальне дублювання здійснюють за допомогою методу нормалізації відношень.

Нормалізація відношень - покроковий процес розділення (декомпозиції) початкових відношень БД на простіші. Кроки цього процесу переводять схему відношення БД в послідовні ***нормальні форми***. Кожна наступна форма володіє кращими властивостями ніж попередня.

У теорії реляційних баз даних прийнято виділяти таку послідовність нормальних форм:

- перша нормальна форма, 1*NF* (*First Normal Form*);
- друга нормальна форма, 2*NF* (*Second Normal Form*);
- третя нормальна форма, 3*NF* (*Third Normal Form*);
- нормальна форма Бойса – Кодда, *BCNF* (*Brice – Codd Normal Form*);
- четверта нормальна форма, 4*NF* (*Fourth Normal Form*);
- п'ята нормальна форма, 5*NF* (*Fifth Normal Form*).

Кожній нормальній формі відповідає певний набір обмежень. При переведенні структури відношення у форми вищого порядку досягають видалення з таблиць надмірної описової інформації.

Атрибут *B* ***функціонально залежить*** від атрибуту *A* (позначають $A \rightarrow B$), якщо у будь-який момент часу кожному значенню атрибуту *A* відповідає не більше одного значення атрибуту *B*.

Якщо описовий атрибут залежить від усього складеного ключа і не залежить від його частин, то мова йде про ***повну функціональну залежність*** атрибуту від складеного ключа.

Якщо атрибут *A* залежить від атрибуту *B*, а *B* залежить від атрибуту *C*, але зворотна залежність відсутня, то кажуть, що атрибут *A* залежить від *C* ***транзитивно***. Наприклад, відношення має два описові атрибути, причому один залежить від ключа, а другий – від першого описового атрибуту.

Перша нормальна форма. Відношення відповідає 1*NF* тоді, коли на перетині кожного стовпця і кожного рядка знаходяться тільки елементарні (неподільні) значення атрибутів і не містяться групи, що повторюються.

Розглянемо відношення ТЕЛЕФОННА КНИГА (див табл.3.13).

Таблиця 3.13. Таблиця до відношення ТЕЛЕФОННА КНИГА

Прізвище	Адреса	Телефон
Корнійчук	вул. Зелена, буд. 2, кв. 36.	123-45-67 - домашній 765-43-21 - робочий
Савченко	вул. Хрещатик, буд.15, кв. 196.	063 - 322-22-33 - мобільний

Текстове поле *Адреса* містить не елементарну інформацію. Його потрібно розбити на декілька окремих неподільних полів. Такими полями будуть *Вулиця* (текстове), *Будинок* (числове) і *Квартира* (числове).

Крім того, на перетині першого рядка і третього стовпця таблиці знаходиться цілий набір елементарних значень номерів телефонів – телефони різних типів – робочий, домашній та мобільний.

Для виключення групи, що повторюється у полі *Телефон*, у таблицю потрібно додати ще поле для зберігання інформації про тип телефону (див. табл. 3.14).

Таблиця 3.14. Форма 1*NF* відношення ТЕЛЕФОННА КНИГА

Прізвище	Вулиця	Будинок	Квартира	Телефон	Тип телефону
Корнійчук	Зелена	2	36	123-45-67	домашній
Корнійчук	Зелена	2	36	765-43-21	робочий
Савченко	Хрещатик	15	196	063-322-22-33	мобільний

Друга нормальна форма. Відношення знаходиться в 2*NF*, якщо виконуються обмеження 1*NF* і кожен описовий атрибут функціонально повно залежить від первинного ключа (у тому числі і складеного).

Якщо первинний ключ таблиці складається з одного атрибуту, то ця умова виконується автоматично. Якщо первинний ключ складається з декількох полів, то жоден з описових атрибутів не повинен функціонально залежати тільки від частини складеного первинного ключа.

Розглянемо відношення СЕСІЯ, яке моделює результати сесії (див.табл. 3.15).

Таблиця 3.15. Таблиця до відношення СЕСІЯ

Прізвище, ініціали	Спеціальність	Номер групи	Номер у списку	Дисципліна	Оцінка
Котляр І.І.	ЛА	61	15	Інформатика	відмінно
Котляр І.І.	ЛА	61	15	Вища математика	добре
Котляр І.І.	ЛА	61	15	Фізика	добре
Бовкун П.П.	ЛЕ	71	3	Інформатика	добре

Оскільки кожен студент здає у ході сесії декілька дисциплін, то первинним ключем відношення може бути складений: *Спеціальність*, *Номер групи*, *Номер у списку* (цей набір атрибутів формує код залікової книжки) та *Дисципліна*. Він однозначно визначає кожен рядок відношення. З іншого боку, описовий атрибут *Прізвище, ініціали* залежить тільки від частини первинного ключа – від значень атрибутів *Спеціальність*, *Номер групи*, *Номер у списку*.

Отже, у даному відношенні є неповні функціональні залежності. Для приведення відношення СЕСІЯ до другої нормальної форми потрібно розділити його на два зв'язані відношення (дві таблиці).

До першої таблиці (табл.3.16) увійдуть *Спеціальність*, *Номер групи*, *Номер у списку* і описовий атрибут *Прізвище, ініціали*, який функціонально залежний від ключа у цілому. В другій таблиці (табл.3.17) залишаться *Спеціальність*, *Номер групи*, *Номер у списку* як частина первинного ключа з одного боку, а з іншого боку – як зовнішній ключ для зв'язку першої таблиці з другою. Окрім них там залишаться *Дисципліна* та *Оцінка*. У табл. 3.17 первинним ключем є складений ключ, той, що був у початковій таблиці. Атрибут *Оцінка* є описовим, який функціонально залежний від ключа у цілому.

Табл. 3.16. Перша таблиця відношення СЕСІЯ для переходу до 2*NF*

Прізвище, ініціали	Спеціальність	Номер групи	Номер у списку
Котляр І.І.	ЛА	61	15
Котляр І.І.	ЛА	61	15
Котляр І.І.	ЛА	61	15
Бовкун П.П.	ЛЕ	71	3

Табл. 3.17. Друга таблиця відношення СЕСІЯ для переходу до 2*NF*

Спеціальність	Номер групи	Номер у списку	Дисципліна	Оцінка
ЛА	61	15	Інформатика	відмінно
ЛА	61	15	Вища математика	добре
ЛА	61	15	Фізика	добре
ЛЕ	71	3	Інформатика	добре

Цей набір відношень (таблиць) не містить неповних функціональних залежностей, тому можна сказати, що кожне відношення знаходиться в другій нормальній формі.

Третя нормальна форма. Відношення знаходиться у 3*NF*, якщо виконуються обмеження 2*NF* і всі описові атрибути відношення взаємно незалежні і повністю залежать від первинного ключа, тобто кожний описовий атрибут не транзитивно залежить від ключа.

У більшості випадків третя нормальна форма є компромісом між повною нормалізацією і функціональністю в сукупності з легкістю реалізації. Існують нормальні форми вище 3*NF*, але на практиці вони ускладнюють розробку структур даних і знижують їх функціональність.

Наведемо складніший приклад нормалізації відношень, для чого розглянемо предметну область **ЛІТЕРАТУРНІ ДЖЕРЕЛА З БАЗ ДАНИХ**.

Один автор може написати декілька книг, що вийшли в різних видавництвах. Кожна книга, що вийшла у видавництві, має певного автора.

У БД повинні бути наступні дані: прізвище та ініціали автора книги, назва книги, шифр УДК, видавництво, що опублікувало книгу, рік видання і кількість сторінок.

У табл. 3.18 та 3.19 наведено дані для створення БД у початковому виді, коли до розгляду запропоновано інформаційні об’єкти АВТОР і ВИДАВНИЦТВО.

Таблиця 3.18. Інформація про авторів видань у початковому виді

АВТОР			
ПІП/б	**Назва книги**	**Кількість сторінок**	**Рік видання**
Хомоненко А.Д.	Базы данных	416	2000
	Базы данных	672	2002
	Основы современных компьютерных технологий	730	2005
	Работа с базами данных в Delphi	656	2001
	Работа с базами данных в Delphi	640	2005
Фуфаев Э.В.	Базы данных	320	2005
Заверач М.М.	Бази даних. Інформаційні системи	303	2007
Карпова Т.С.	Базы данных: Модели, разработка, реализация	304	2002

Таблиця 3.19. Інформація про видавництва у початковому виді

ВИДАВНИЦТВО						
Назва	**Адреса**	**Назва книги**	**Автор книги**	**Рік вид.**	**Кільк стор.**	**Шифр УДК**
Корона принт	Санкт-Петербург, Мойка., д.2	Базы данных Базы данных	Хомоненко А.Д. Хомоненко А.Д.	2000 2002	416 672	681.5 681.5
БХВ-Петербург	Санкт-Петербург, Морской просп., д.4	Работа с базами данных в Delphi Работа с базами данных в Delphi	Хомоненко А.Д. Хомоненко А.Д.	2001 2005	656 640	678.9 678.9
Академия	Москва, ул. Правды, д.24	Базы данных	Фуфаев Э.В.	2005	320	681.5
Вид-во Хмельни-цького нац. ун-ту (ХНУ)	Хмельницький, ул. М.Кривоноса, б.7	Бази даних. Інформаційні системи	Заверач М.М.	2007	303	685.2
Питер	Санкт-Петербург, ул. Благодатная, д.67	Базы данных: Модели, разрабо-тка, реализация	Карпова Т.С.	2002	304	681.3

Як видно з таблиць, подання даних у них не задовольняє вимог навіть першої нормальній форми. Приведемо їх до (табл.3.20, 3.21).

Таблиця 3.20. Відношення АВТОР у 1*NF*

АВТОР		
Прізвище, ініціали	**Назва книги**	**Кільк. стор.**
Хомоненко А.Д.	Базы данных	416
Хомоненко А.Д.	Базы данных	672
Хомоненко А.Д..	Работа с базами данных в Delphi	656
Хомоненко А.Д..	Работа с базами данных в Delphi	640
Фуфаев Э.В.	Базы данных	320
Заверач М.М.	Бази даних. Інформа-ційні системи	303
Карпова Т.С.	Базы данных: Модели, разработка, реализация	304

Таблиця 3.21. Відношення ВИДАВНИЦТВО у 1*NF*

ВИДАВНИЦТВО								
Назва	**Місто**	**Вулиця**	**Буд**	**Назва книги**	**Автор книги**	**Рік вид.**	**Кільк. стор.**	**Шифр УДК**
Корона принт	Санкт-Петербург	Мойка	2	Базы данных	Хомоненко А.Д.	2000	416	681.5
Корона принт	Санкт-Петербург	Мойка	2	Базы данных	Хомоненко А.Д.	2002	672	681.5
БХВ-Петербург	Санкт-Петербург	Морской просп	4	Работа с базами данных в Delphi	Хомоненко А.Д.	2001	656	678.9
БХВ-Петербург	Санкт-Петербург	Морской просп	4	Работа с базами данных в Delphi	Хомоненко А.Д.	2005	640	678.9
Академия	Москва	Правды	24	Базы данных.	Фуфаев Э.В.	2005	320	681.5
Вид-во Хмельницького нац. ун-ту (ХНУ)	Хмельницький	М.Криво-носа	7	Бази даних. Інформаційні системи	Заверач М.М.	2007	303	685.2
Питер	Санкт-Петербург	Благодат-ная	67	Базы данных: Модели, разработка, реализация	Карпова Т.С.	2002	304	681.3

Оскільки в обох відношеннях первинний ключ не складений, то від 1*NF* переходимо відразу до 3*NF*.

У відношенні АВТОР від первинного ключа функціонально залежить тільки атрибут *Прізвище, ініціали*. Запишемо 3*NF* для АВТОР (табл.3.22).

Таблиця 3.22. Відношення АВТОР у 3*NF*

АВТОР	
Код автора	**Прізвище, ініціали**
1	Хомоненко А.Д.
2	Фуфаев Э.В.
3	Заверач М.М.
4	Карпова Т.С.

У відношенні ВИДАВНИЦТВО від *Коду видавництва* залежать атрибути *Назва, Місто, Вулиця, Будинок*. Винесемо їх в окреме відношення у 3*NF* (табл.3.23).

Таблиця 3.23. Відношення ВИДАВНИЦТВО у 3*NF*

ВИДАВНИЦТВО				
Код видавництва	**Назва**	**Місто**	**Вулиця**	**Будинок**
1	Корона принт	Москва	Мойка	2
2	БХВ-Петербург	Москва	Морской просп	4
3	Академия	Москва	Правды	24
4	Вид-во Хмельницького нац. ун-ту (ХНУ)	Хмельницький	М.Кривоноса	7
5	Питер	Санкт-Петербург	Благодатная	67

З першого і другого відношень залишилися не використаними наступні атрибути, що характеризують літературне джерело: *Назва книги; Рік видання; Кількість сторінок; Шифр УДК*. Для того, щоб врахувати ці атрибути, створимо додатково інформаційний об’єкт КНИГА, за первинний ключ для нього уведемо атрибут *Код книги*. Для зв'язку КНИГА з відношеннями АВТОР і ВИДАВНИЦТВО уведемо в нього зовнішні ключі *Код автора* і *Код видавництва*. Одержане відношення у 3*NF* має вигляд (табл.3.24):

Таблиця 3.24. Відношення КНИГА у 3*NF*

КНИГА						
Код книги	**Назва книги**	**Рік видання**	**Кількість сторінок**	**Шифр УДК**	**Код автора**	**Код вид-ва**
1	Базы данных	2000	416	681.5	1	1
2	Базы данных	2002	672	681.5	1	1
3	Базы данных	2005	320	681.5	2	3
4	Работа с базами данных в Delphi	2001	656	678.9	1	2
5	Работа с базами данных в Delphi	2005	640	678.9	1	2
6	Бази даних. Інформа-ційні системи	2007	303	685.2	3	4
7	Базы данных: Модели, разработка, реализация	2002	304	681.3	4	5

Розглянутий приклад показує, як може змінитися попередня уява про структуру БД при застосуванні процедури нормалізації. Тепер зрозуміло, що нормалізація відношень - не марне витрачання часу. Багаторазове дублювання інформації в базі даних, крім різних аномалій редагування (аномалія вставки, аномалія редагування, аномалія видалення) призводить до зниження продуктивності СКБД. Кожен елемент даних повинен зберігатися в базі в одному і лише одному екземплярі, що і досягається нормалізацією.

3.7. Створення схеми даних

При визначенні логічної структури РБД на основі реляційної моделі даних кожний інформаційний об’єкт (відношення) відображають таблицею, а зв’язки між таблицями відповідають зв’язкам між об’єктами.

Схемою даних називають умовне графічне зображення об’єктних відношень РБД та зв’язків між ними. Створення схеми даних є одним з етапів створення РБД.

Вона дозволяє не тільки отримати уяву про структуру БД, але і забезпечує об’єднання відношень при обробці даних і цілісність БД.

Розташування інформації про об’єкт в окремій таблиці і подальше зв’язування цих таблиць запобігає повторенню даних у таблицях і спрощує пошук у БД, забезпечує цілісність взаємопов’язаних даних при їхньому коригуванні. Створення схеми даних спрощує створення багатотабличних об’єктів РБД, таких як запити, форми та звіти.

На схемі об’єкт позначають прямокутником, у верхній частині якого вказують його ім’я, нижче – атрибути, причому першим (першими при складеному ключі) подають ключовий атрибут. Прямокутники з’єднують лініями, які відповідають зв’язкам між сутностями. Лінію починають від ключа головного об’єкта і закінчують на відповідному атрибуті (атрибутах) підпорядкованого.

При виконанні схеми даних засобами СКБД мова йде про таблиці.

Приклад схеми даних РБД у середовищі *MS Access* зображено на рис. 3.11.

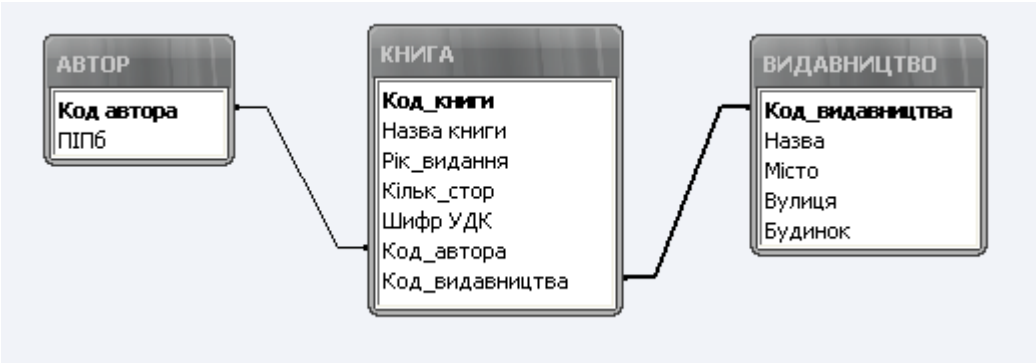


Рис. 3.11. Приклад схеми даних РБД для предметної області
ЛІТЕРАТУРНІ ДЖЕРЕЛА З БАЗ ДАНИХ

3.8. Узагальнений алгоритм проєктування реляційної бази даних

Перед створенням бази даних необхідно мати детальний опис предметної області, з якого буде зрозуміло, які в ній діють реальні об'єкти і процеси, і які задачі користувачів треба задовольняти інформацією з цієї бази даних.

Проєктування БД передбачає виконання цілої низки кроків, які дозволять перейти від загальної постановки задачі до створення необхідної структури даних. Структуру відображають спочатку інфологічною та даталогічною моделями, а потім тими об’єктами, які передбачені конкретною СКБД, у реляційних – таблицями.

Існують два підходи для дослідження властивостей предметної області. Перший підхід передбачає першочерговим – визначення основних *завдань*, для яких власне і створюють БД, а другий – визначення *об'єктів* предметної області. Гарні результати дає сполучення зазначених підходів. Це пояснюється тим, що на початку проєктування ще відсутній вичерпний перелік необхідних задач.

Використання такої технології можливе завдяки гнучким засобам сучасних реляційних СКБД, зокрема *MS Access*, а це дозволяє на будь-якому етапі проєктування внести зміни в БД і модифікувати її структуру.

Реляційна база даних складається з взаємопов’язаних таблиць. Структура таблиці визначається сукупністю стовпців (полів), типом і розміром даних кожного стовпця, а також первинним ключем таблиці. Рядки таблиці однозначно ідентифікуються значенням ключа. Логічні зв'язки між таблицями в реляційній базі даних реалізуються за рахунок однакових полів у таблицях, які зв'язують.

Отже, у результаті проєктування має бути визначена логічна структура бази даних, тобто склад реляційних таблиць, їх структура і міжтабличні зв'язки.

Створеною вважають БД, у таблиці якої внесені необхідні дані, з якими працює користувач БД.

Наведемо спрощений алгоритм проєктування РБД:

1. визначити мету створення бази даних;
2. визначити інформаційні об’єкти, які містяться в базі даних;
3. визначити зв’язки між об’єктами;
4. визначити основні властивості об’єктів;
5. визначити зв’язки між властивостями об’єктів;
6. визначити відношення між таблицями баз даних, базуючись на зв’язках між об’єктами даних, що містяться в таблиці, і включити цю інформацію до словника даних;
7. передбачити операції, що треба виконати при створенні та зміні інформації таблиць, включаючи забезпечення цілісності даних;
8. визначити користувачів, яким дозволений доступ до даних, їх редагування, а також зміна при необхідності структури таблиць;

Добре ілюструє етапи проєктування схема, подана на рис.3.12 [15].

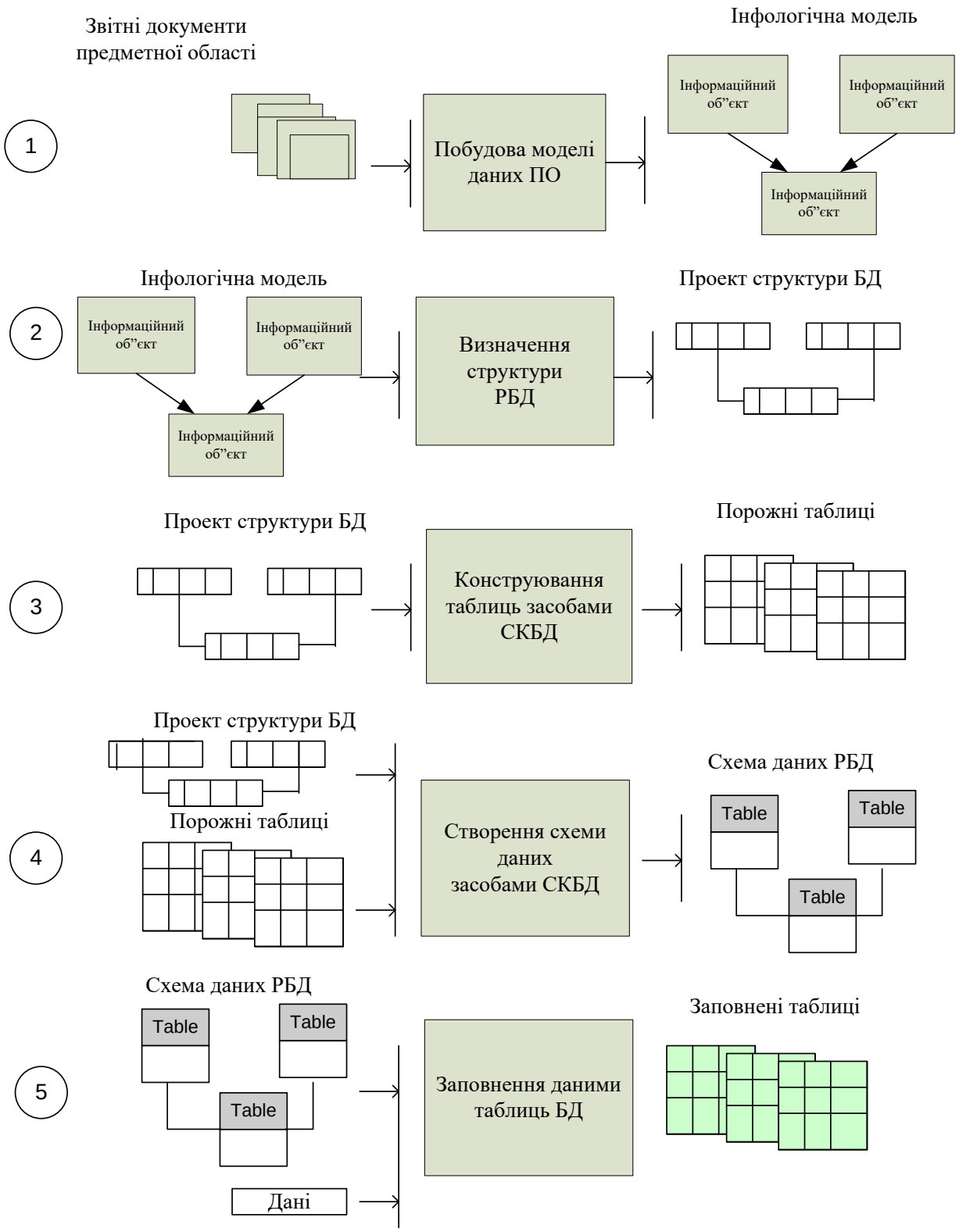


Рис. 3.12. Схема етапів проєктування РБД

Розглянемо проєктування БД на прикладі предметної області **НАВЧАЛЬНИЙ ПРОЦЕС** [15].

Нехай на поточний момент навчальний процес супроводжують наступні звітні документи:

1) екзаменаційні і залікові відомості для кожної студентської групи з певної навчальної дисципліни, у відомостях треба вказувати, який викладач приймає ці види семестрової звітності, у табл. 3.25 наведемо приклад відомості.

Таблиця 3.25. Екзаменаційна відомість

Назва групи	**ЛА-11**		
Вид семестрової звітності	**Екзамен**		
Назва предмету	**Комп'терні технології**		
Прізвище, ім'я, по-батькові викладача	**Акінфієв Павло Миколойович**		
№ п/п	Прізвище, ініціали студента	Оцінка	Підпис викладача
1	Антощук І.А.	Відмінно	
2	Андрійко В.В.	Добре	
…	…	…	…

2) списки студентських груп, у табл. 3.26 наведемо приклад списку для групи ЛА-01

Таблиця 3.26. Список студентської групи

Гр. ЛА-01						
Номер за списком	Відмітка для старости	Прізвище, ім'я, по - батькові	Дата народження	Адреса	Телефони	Бюджет / контракт
1		Аненков Петро Гнатович	12.09.1997	м.Київ, в. Гетьмана, б.10, кв.6	454- 67- 90, 095-675-65-89	Б
2	С	Борчук Наталія Семенівна	03.10.1998	м. Чернівці, в. Федорова, б. 5, кв. 8	254- 17- 94, 093-135-63-83	К
…		…	…	…	…	…

3) списки викладачів кафедр, у табл. 3.27 наведемо приклад списку.

Таблиця 3.27. Список викладачів кафедри АСК

Назва кафедри **АСК**					
Код кафедри	**306**	Завідувач каф.	**Жуков А.І.**	Телефон каф.	**241-56-76**
Таб. номер	Прізвище, ім'я, по – батькові		Посада	Вч. звання	Вч. ступінь
642	Акінфієв Павло Миколойович		професор	професор	д.т.н.
643	Барклай Інна Федорівна		професор	доцент	к.т.н.
642	Жадан Марта Павлівна		ст. викладач	немає	немає
…	…		…	…	…

4) інформація про педагогічне навантаження викладачів, у табл. 3.28 наведемо приклад

Таблиця 3.28. Педагогічне завантаження викладачів каф. АСК

Прізвище, ім'я, по – батькові	Назва предмету	Вид занять	Кількість годин	Шифр групи	Семестрова звітність
Акінфієв Павло Миколайович	Комп'ютерні технології	Лекції	36	ЛА-11	екзамен
" – "	Комп'ютерні технології	Лабор. роботи	18	ЛА-11	-
" – "	Числові методи	Практичні заняття	18	ЛА-12	-
Барклай Інна Федорівна	Числові методи	Лекції	36	ЛА-12	залік
" – "	Метрологія	Лекції	36	ЛА-01	екзамен
" – "	Метрологія	Практичні заняття	36	ЛА-01	-
	Метрологія	Практичні заняття	36	ЛА-02	-
…	…	…	…	…	…

Наведені документи ПО слід розглядати як основу для створення таких об'єктів БД, як форми і звіти.

На основі цих документів звітності виділимо інформаційні об'єкти ПО.

Звернемось до списку студентської групи (табл.3.26).

Можна побачити, що кожний студент згідно з задачами деканату має такі властивості: прізвище, ім'я, по – батькові, дату народження, адресу, телефони, джерело фінансування (бюджет або контракт). Спільними властивостями для усіх студентів є шифр групи (надають в залежності від спеціальності), її номер (визначають роком вступу), а також прізвище, ім'я, по-батькові (ПІПб) старости. Деканат вирізняє старост оскільки через них відбувається зв'язок з групою по загальних питаннях. Відзначимо, що усі зазначені властивості можуть співпасти для декількох студентів групи. Неповторним для студента є його порядковий номер у групі. Враховуючи наявність однакових порядкових номерів у різних групах, за ключ відношення візьмемо декілька властивостей: *Шифр_групи*, *Номер_групи*, *Номер_у групі*. Решту зазначених атрибутів розглядаємо як описові. Аналізуючи цю інформацію, визначимо відношення (об'єкт) СТУДЕНТ і запишемо його схему:

СТУДЕНТ (*Шифр_групи, Номер_групи, Номер_у групі*, *Прізвище_ім'я_по – батькові, Дата народження, Адреса, Телефони, Джерело фінансування*).

Тепер запишемо схему з такими назвами полів, які будуть у *MS Access*:

СТУДЕНТ (*Шифр_групи, Номер_групи, Номер_у групі*, *ПІПб, Дата_нар, Адреса, Телефони, Фінанс*).

Зі списків студентських груп стає зрозумілим, що можна виокремити ще один об'єкт – СТУДЕНТСЬКА ГРУПА або просто ГРУПА. Кожна група володіє наступними властивостями: шифр, номер, кількість студентів, ПІПб старости, ПІПб куратора та іншими, які визначає деканат. За ключ об'єкта візьмемо дві з них *Шифр_групи* та *Номер_групи*.

Решта зазначених атрибутів є описовими. Запишемо можливу схему відношення ГРУПА**:**

ГРУПА (***Шифр_групи, Номер_групи***, *ПІПб_старости, ПІПб_куратора, Рік_вступу*).

Тепер запишемо схему з такими назвами полів, які будуть у *MS Access*:

ГРУПА (***Шифр_групи, Номер_групи***, *Староста, Куратор, Рік_вступу*).

Між об'єктами ГРУПА та СТУДЕНТ існує зв'язок 1:*N*, оскільки група може складатися з декількох студентів, а студент навчається в одній групі[2].

ГРУПА – головна, СТУДЕНТ – підпорядкована сутності. Для їхнього зв'язування у СТУДЕНТ вже існують атрибути *Шифр_групи*, *Номер_групи*, що є складеним первинним ключем сутності ГРУПА.

Студенти – суб'єкти навчального процесу з одного боку. З іншого суб'єктами є викладачі. З ними пов'язана інформація, наведена у табл.3.25, 3.27, 3.28.

Розглянемо дані, пов'язані з сутністю **ВИКЛАДАЧ**. Властивостями кожного викладача є табельний номер, прізвище, ім'я та по – батькові, назва кафедри, на якій вони працюють, посада, вчений ступінь, предмети, які вони викладають, типи занять, відповідні їм назви груп, час та аудиторії занять тощо. Цей великий перелік властивостей підказує, що вони можуть характеризувати декілька об'єктів.

Зокрема, з табл. 3.27 видно, що усім викладачам однієї кафедри притаманні спільні риси: назва кафедри, ПІПб завідувача кафедри, телефон кафедри, розташування або кабінет завідувача, телефон кафедри тощо. Тому виділимо кафедру як окремий об'єкт КАФЕДРА.

[2] Випадок, коли студент набуває другу освіту у тому самому закладі, не розглядаємо.

Нехай він володіє такими властивостями: код кафедри, назва, ПІПб завідувача, телефон, кабінет завідувача. За ключ доцільно взяти код кафедри. Інші властивості розглядатимемо як описові. Запишемо схему відповідного відношення:

КАФЕДРА (***Код_кафедри,*** *Назва_кафедри, ПІПб_завідувача, Телефон, Кабінет_завідувача*).

Запишемо схему з такими назвами полів, які будуть у *MS Access*:

КАФЕДРА (***Код_кафедри,*** *Назва, Завідувач, Телефон, Каб_зав*).

З таблиць 3.25, 3.27, 3.28 виокремимо тепер інформаційний об'єкт ВИКЛАДАЧ. Він володітиме в нашому випадку такими властивостями: табельний номер, прізвище, ім'я та по – батькові, назва кафедри, на якій викладач працює, посада, вчений ступінь. За ключ відповідного відношення візьмемо табельний номер, інші – описові властивості. Запишемо схему відношення з такими назвами полів, які будуть у *MS Access*:

ВИКЛАДАЧ (***Таб_номер****, ПІПб, Назва_каф, Посада, Вч_ступінь*).

Між сутностями КАФЕДРА та ВИКЛАДАЧ існує зв'язок типу 1:*N*, перша – головна, а друга – підпорядкована. Для того, щоб їх зв'язати уведемо у підпорядковану сутність ВИКЛАДАЧ атрибут *Код_кафедри*. Остаточно схема відношення ВИКЛАДАЧ з такими назвами полів, які будуть у *MS Access*, набуде такого вигляду

ВИКЛАДАЧ (***Таб_номер****, Код_кафедри, ПІПб, Назва_каф, Посада, Вч_ступінь*).

Розглянемо табл. 3.28 з даними про педагогічне навантаження викладачів. У ній є інформація про навчальні *дисципліни* (*предмети*), які

повинні вивчати студенти різних *груп* (за спеціальностями), та про *викладачів*. Ця таблиця досить складна у порівнянні з іншими з точки зору виділення інформаційних об'єктів. Оскільки об'єкти ГРУПА та ВИКЛАДАЧ вже створені, то виокремимо об'єкт ПРЕДМЕТ. Судячи з табл. 3.28, він повинен мати наступні властивості: назва предмету, загальний обсяг годин на його вивчення (обмежимося аудиторними заняттями), види занять (лекції, практичні та лабораторні заняття) та їхні обсяги, вид семестрової звітності (екзамен чи залік), програма предмету (перелік розділів та тем), назва групи, яка вивчає цей предмет. За ключ відповідного об'єкту доцільно взяти штучний атрибут – код предмету, який іноді вказують навіть у навчальних планах.

Запишемо схему відношення ПРЕДМЕТ:

ПРЕДМЕТ (***Код_предм***, *Назва_предм*, *Загальн_обсяг, Лекц, Практ, Лаб, Звітність, Програма, Шифр_групи, Номер_групи*)

Зробимо попередній висновок з наведених таблиць звітної документації – нами виділено такі об'єкти: СТУДЕНТ, ГРУПА, КАФЕДРА, ВИКЛАДАЧ та ПРЕДМЕТ. Ці сутності добре зрозумілі і користувачам і проектувальникам.

Зауважимо, що ВИКЛАДАЧ теж може містити інформацію про види занять, які проводить кожний викладач. Отже, у об'єктів ПРЕДМЕТ і ВИКЛАДАЧ могли б бути спільні властивості. Треба тепер з'ясувати, чи можна обмежитися цими сутностями для задоволення вимог реляційної БД. Для цього вивчимо властивості зв'язків між сутностями: один викладач може проводити заняття з декількох предметів, а один предмет може викладатися декількома викладачами, тобто зв'язок типу N:M.

Розглянемо наступну пару об'єктів – ГРУПА та ВИКЛАДАЧ. З однією групою працює декілька викладачів, а один викладач може

викладати у декількох групах, отже зв'язок між сутностями також типу *N*:*M*.

Розглянемо дві сутності – ГРУПА та ПРЕДМЕТ. Студентська група вивчає декілька предметів, а один предмет може викладатися у декількох групах, отже зв'язок між сутностями також типу *N*:*M*.

На рис. 3.13 зобразимо *ER* – діаграму для зазначених сутностей у нотації Чена.

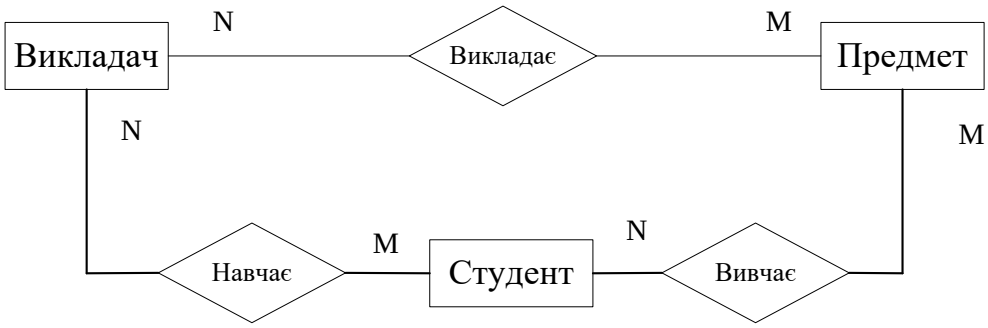


Рис. 3.13. *ER* – діаграма для сутностей ВИКЛАДАЧ, ПРЕДМЕТ та СТУДЕНТ

Реляційна БД не зможе реалізувати таку модель даних, оскільки не підтримує зв'язки типу *N*:*M* ("багато – до багатьох").

Отже, наступним кроком при проєктуванні буде уведення штучних сутностей, які дозволять перейти від одного зв'язку типу *N*:*M* до декількох зв'язків типу 1:*N*. Ці сутності будуть не такими очевидними для сприйняття, як визначені раніше.

Згадаємо п.3.3, у якому мова йшла про правило створення ключів зв'язку між двома об'єктами. Первинний ключ головного об'єкту повинен стати атрибутом (або декількома атрибутами у випадку складеного первинного ключа) підпорядкованого об'єкту.

Знову повернемось до об'єктів ПРЕДМЕТ і ВИКЛАДАЧ. Уведемо штучний об'єкт ВИКЛАДАННЯ, який міститиме атрибути, що відіграють ролі первинних ключів обох основних об'єктів, а саме код предмета та

табельний номер викладача. Описовими властивостями ВИКЛАДАННЯ стануть вид занять з предмету, обсяги різних видів занять, номер студентської групи. Цей новий об'єкт пов'яже різні види занять з конкретними групами і викладачами.

Оскільки ПРЕДМЕТ, ГРУПА та ВИКЛАДАЧ взаємопов'язані, то спробуємо використати об'єкт ВИКЛАДАННЯ і для зв'язку з сутністю ГРУПА. Для цього уведемо до складу атрибутів ВИКЛАДАННЯ первинний складений ключ об'єкту ГРУПА – шифр та номер групи.

Запишемо схему відношення ВИКЛАДАННЯ:

ВИКЛАДАННЯ (***Таб_номер***, ***Код_предм***, ***Шифр_групи***, ***Номер_групи***, *Вид_заняття*, *Обсяг_заняття*).

На рис.3.14. зобразимо *ER* – діаграму ВИКЛАДАННЯ у нотації Чена.

Рис.3.14. *ER* – діаграма для об'єкту ВИКЛАДАННЯ

На рис.3.15 зображено спрощену *ER* – діаграму, на якій видно зв'язки між об'єктами ВИКЛАДАЧ, ПРЕДМЕТ та ГРУПА при уведенні сутності ВИКЛАДАННЯ.

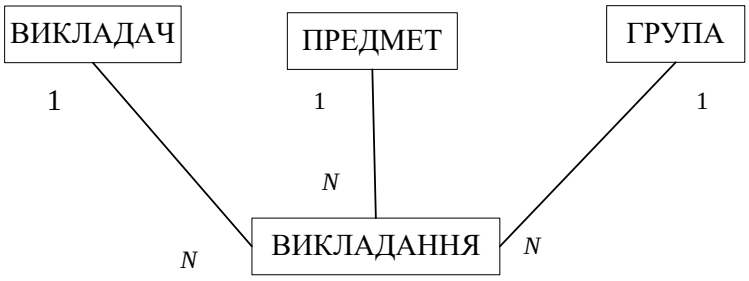


Рис.3.15. *ER* – діаграма для об'єкту ВИКЛАДАННЯ

На рис. 3.16 показано схему даних, збудовану засобами *MS Access*. На рис. 3.17 показано таблиці бази даних *MS Access*, які відповідають поточному етапу її проєктування.

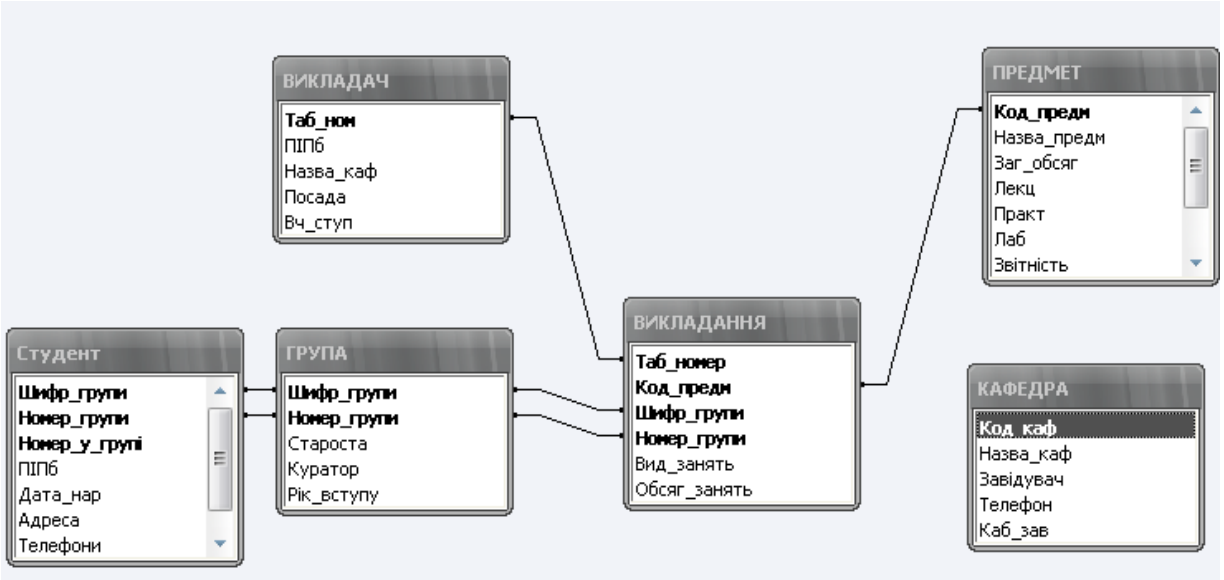


Рис. 3.16. Схема даних *MS Access* для сутностей СТУДЕНТ, ВИКЛАДАЧ, ГРУПА, ПРЕДМЕТ, КАФЕДРА, ВИКЛАДАННЯ

Відзначимо, що визначені на цьому етапі проєктування об’єкти не містять усієї звітної інформації, зокрема екзаменаційних оцінок. Щоб врахувати і їх уведемо додатковий (штучний) об’єкт і назвемо його УСПІШНІСТЬ. Зрозуміло, що з ним будуть пов’язані об’єкти СТУДЕНТ, ВИКЛАДАЧ, ПРЕДМЕТ.

ПРЕДМЕТ : таблиця

Код_предм	Назва_предм	Заг_обсяг	Лекц	Практ	Лаб	Звітність	Програма	Шифр_гр	Номер_гр
НП-01	Механіка	108	72	18	0	Екзамен		ЛА	1

Таб_номер	ПІПБ_Викл	Назва_предм	Шифр_групи	Номер_групи	Вид_занять	Обсяг_занять
121	Проклов С.С.	Механіка	ЛА	1	лекц	72

Запись: 1 из 2

КАФЕДРА : таблиця

Код_каф	Назва_каф	Завідувач	Телефон	Каб_зав
13	АХВ	Жуков А.І.	454-90-78	306-19
14	ХПСМ	Паров Є.М.	454-89-90	305-19
15	МАХНВ	Корнійчук Я.М	241-80-67	202-19
16	ЕТРП	Горєлов М.Д.	454-56-43	102-4

ГРУПА : таблиця

Шифр_групи	Номер_групи	Староста	Куратор	Рік_вступу
ЛА	1	Кулик Т.П.	Котов А.А.	01.09.2010

Таб_номер	ПІПБ_Викл	Код_предм	Назва_предм	Вид_занять	Обсяг_занять
121	Проклов С.С.	НП-01	Механіка	лекц	72
122	Сутко А.А.	НП-01	Механіка	Практ	18

Запись: 1 из 2

ВИКЛАДАННЯ : таблиця

Таб_номер	ПІПБ_Викл	Код_предм	Назва_предм	Шифр_групи	Номер_групи	Вид_занять	Обсяг_занять
121	Проклов С.С.	НП-01	Механіка	ЛА	1	лекц	72
122	Сутко А.А.	НП-01	Механіка	ЛА	1	Практ	18

Запись: 1 из 2

ВИКЛАДАЧ : таблиця

Таб_ном	ПІПб	Назва_каф	Посада	Вч_ступ
121	Проклов С.С.	АХВ	доцен	к.т.н.
122	Судко А.А.	АХВ	професор	д.т.н.
221	Ваткін М.М.	ХПСМ	ст_виклад	б/ст
222	Колесник Р.Р.	ХПСМ	доцент	к.т.н.

ПІПБ_Викл	Код_предм	Назва_предм	Шифр_групи	Номер_групи	Вид_занять	Обсяг_занять
Проклов С.С.	НП-01	Механіка	ЛА	1	лекц	72

Запись: 1 из 1

студент : таблиця

Шифр_групи	Номер_групи	Номер_у_групі	ПІПб	Дата_нар	Адреса	Телефони	Джерело_фінан
ЛА	1	1	Зощенко Л.Д.	01.09.1998	Полтава	342-78-5	бюджет
ЛА	1	2	Зубов Т.Р.	05.10.1997	Вінниця	543-78-8	контракт

Рис. 3.17. Таблиці бази даних *MS Access*, які відповідають рис. 3.8

Виходячи з табл. 3.25 об’єкт УСПІШНІСТЬ повинен мати такі властивості: назви груп, назви предметів, ПІПб викладачів та студентів, оцінки. Запишемо зрозумілу для цього етапу проєктування схему цього відношення:

УСПІШНІСТЬ (***Шифр_групи***, ***Номер_групи***, ***Номер_у_групі***, ***Таб_номер***, ***Код_предм***, *Назва_предм*, *Оцінка*)

Розглянемо детально впровадження цього об’єкту у загальну схему даних.

На екзамені викладач ставить завдання студенту з певного предмета, студент виконує ці завдання і отримує оцінку. Студент отримує оцінки з декількох предметів.

На рис.3.18 зобразимо *ER* – діаграму для об’єктів ВИКЛАДАЧ та СТУДЕНТ у нотації Чена.

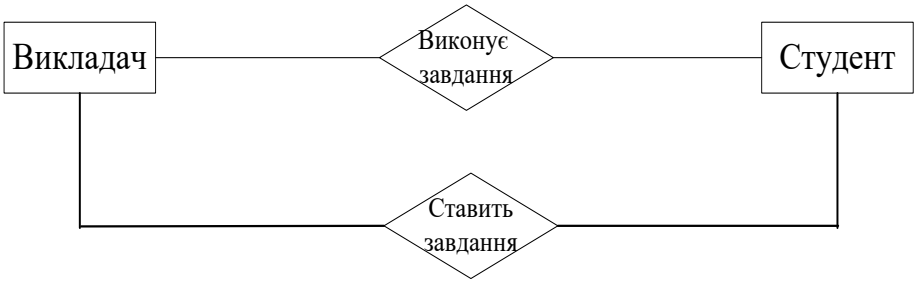


Рис. 3.18. *ER* – діаграма для об’єктів ВИКЛАДАЧ та СТУДЕНТ

Зрозуміло, що така діаграма повторюється для декількох предметів. По одному і тому самому предмету отримують оцінки декілька студентів. Відношення УСПІШНІСТЬ повинно перетворити зв’язок *N*:*M* між сутностями СТУДЕНТ, ВИКЛАДАЧ і ПРЕДМЕТ на три зв’язки типу 1:*N*.

Оскільки об’єкт ВИКЛАДАННЯ раніше був застосований з такою ж метою але без врахування об’єкту СТУДЕНТ, то зв’яжемо тепер об’єктом УСПІШНІСТЬ два об’єкти – СТУДЕН та ВИКЛАДАННЯ.

На рис.3.19 зображено кінцеву *ER* – діаграму, на якій видно зв'язки між усіма виділеними інформаційними об'єктами – ВИКЛАДАЧ, ПРЕДМЕТ, ГРУПА, ВИКЛАДАННЯ, СТУДЕНТ та УСПІШНІСТЬ.

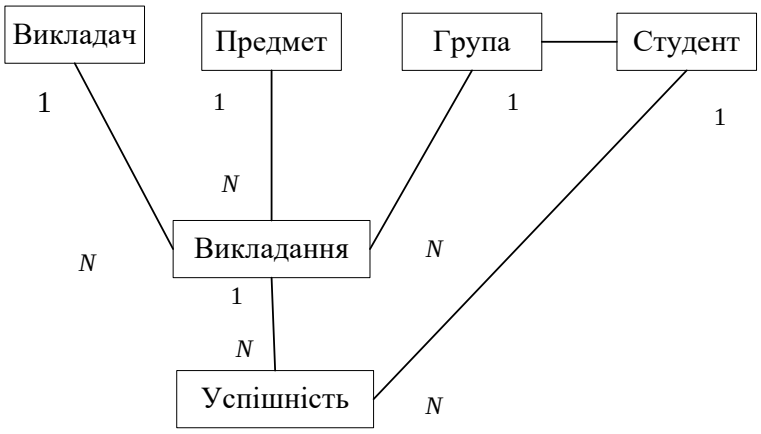


Рис.3.19. Кінцевий вид *ER* – діаграми для об'єктів ПО **НАВЧАЛЬНИЙ ПРОЦЕС**

На рис. 3.20 показано схему даних, збудовану засобами *MS Access*. На рис. 3.21 наведено кінцевий склад таблиць бази даних *MS Access*.

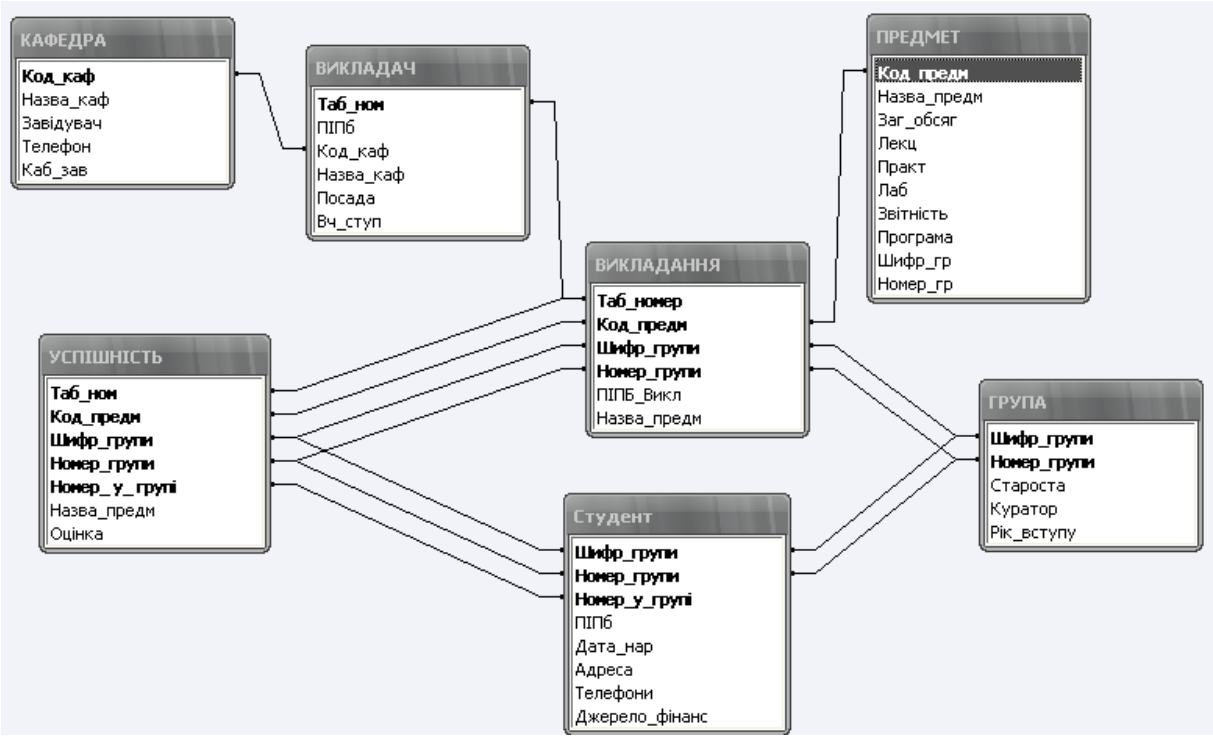


Рис. 3.20. Кінцева схема даних *MS Access* для предметної області **НАВЧАЛЬНИЙ ПРОЦЕС**

ВИКЛАДАЧ : таблиця

Таб_ном	ПІПб	Код_каф	Назва_каф	Посада	Вч_ступ
121	Проклов С.С.	13	АХВ	доцен	к.т.н.
122	Судко А.А.	13	АХВ	професор	д.т.н.
221	Ваткін М.М.	14	ХПСМ	ст_виклад	б/ст
222	Колесник Р.Р.	14	ХПСМ	доцент	к.т.н.

Код_предм	Шифр_групи	Номер_групи	ПІПБ_Викл	Назва_предм
НП-01	ЛА	1		

ГРУПА : таблиця

Шифр_групи	Номер_групи	Староста	Куратор	Рік_вступу
ЛА	1	Кулик Т.П.	Котов А.А.	01.09.2010
ЛА	91	Торопов М.М.	Зібров Є.М.	01.09.2009
ЛЕ	1	Саблін П.Н.	Дубов С.С.	01.09.2010

Таб_номер	Код_предм	ПІПБ_Викл	Назва_предм
121	НП-01		
122	НП-01		

ПРЕДМЕТ : таблиця

Код_предм	Назва_предм	Заг_обсяг	Лекц	Практ	Лаб	Звітність	Програма	Шифр_гр	Номер_гр
НП-01	Механіка	108	72	18	0	Екзамен		ЛА	1
НП-02	Гідравліка	72	18	0	18	Залік			

Таб_номер	Шифр_групи	Номер_групи	ПІПБ_Викл	Назва_предм
121	ЛА	1		
122	ЛА	1		

Запись: 1 из 2

Таб_номер	Код_предм	Шифр_групи	Номер_групи	ПІПБ_Викл	Назва_предм
121	НП-01	ЛА	1		
122	НП-01	ЛА	1		

Номер_у_групі	Назва_предм	Оцінка
1	Механіка	95
2	Механіка	78
0		0

Запись: 1 из 2

Студент : таблиця

Шифр_групи	Номер_групи	Номер_у_групі	ПІПб	Дата_нар	Адреса	Телефони	Джерело_фіна
ЛА	1	1	Зощенко Л.Д.	01.09.1998	Полтава	342-78-5	бюджет
ЛА	1	2	Зубов Т.Р.	05.10.1997	Вінниця	543-78-8	контракт

Таб_ном	Код_предм	Назва_предм	Оцінка
121	НП-01	Механіка	95
0			0

Рис. 3.21.Кінцевий склад таблиць БД для предметної області **НАВЧАЛЬНИЙ ПРОЦЕС**

Запитання та завдання для самостійної роботи

1. Розкрийте поняття відношення у реляційній базі даних (РБД).

2. Що таке домен? Наведіть приклад.

3. Що таке кортеж? Наведіть приклад.

3. Що таке схема відношення? Наведіть приклад.

5. Назвіть параметри відношення.

6. Яким вимогам повинна відповідати таблиця для того, щоб її можна було розглядати як РБД?

7. Що таке інформаційний об’єкт?

8. Які основні типи даних використовують при створенні БД?

9. Що таке первинний ключ? Які є типи первинних ключів? Наведіть приклади.

10. Що таке зовнішній ключ? Як його створити? Наведіть приклади.

11. На основі рис. 2.16 запишіть схему відношення РЕГУЛЯТОР.

12. На основі рис. 2.25 визначте ключ зв’язку між сутностями СТУДЕНТ та ГРУПА.

13. На основі рис. 2.26 запишіть схеми усіх відношень, визначте їхні первинні ключі.

13. На основі рис. 2.26 визначте ключі зв’язку між сутностями

а) ФАКУЛЬТЕТ та КАФЕДРА;

б) ВИКЛАДАЧ та СПЕЦІАЛЬНІСТЬ.

15. Що означають цілісність даних та цілісність БД?

16. Назвіть найбільш поширені операції над відношеннями. Наведіть приклади.

17. Для чого використовують нормалізацію відношень?

18. Назвіть існуючі нормальні форми і поясніть їх вимоги.

19. Наведіть приклади нормалізації відношень до 1NF, 2NF та 3NF.

20. Поясніть призначення схеми даних.

21. Як зобразити схему даних?

22. Які вимоги висувають до структури інформаційних об’єктів для створення схеми бази даних у реляційній СКБД?

23. Розробіть схеми даних для вибраних відношень, які зображені на рис.2.25 та 2.26, проаналізуйте необхідність їх нормалізації відношень.

4. ПРИКЛАДИ ВИКОРИСТАННЯ БАЗ ДАНИХ У ЗАДАЧАХ АВТОМАТИЗАЦІЇ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

4.1. Створення БД для визначення режимних параметрів екструзії полімерів при зміні сировини

Нові для даного обладнання властивості сировини викликають необхідність визначення тих режимних параметрів, які забезпечать вимоги до якості продукції і економічну ефективність у цілому. Тому час досягнення оптимального технологічного режиму для нової сировини, як правило, тривалий, що призводить до значних енерговитрат.

Для прикладу розглянемо технологію виготовлення полімерних виробів в одношнековому екструдері. На рис. 4.1 наведено його принципову схему.

Задачею дослідження є створення такої інформаційної системи, яка дозволить автоматично підбирати режимні параметри роботи екструдера для нової сировини на основі інформації про оптимальні параметри за минулий час роботи обладнання.

Згідно з загальною схемою проєктування РБД (рис. 3.4.) вивчимо особливості технологічного об’єкту та документацію, яку можна розглядати як «Звітні документи».

4.1.1. Дослідження технологічної системи та розробка моделі даних

Термопластичний матеріал з бункера 4 надходить у зону завантаження екструдера 5. Обертовий шнек 6 захоплює і просуває матеріал вздовж циліндра.

За рахунок тертя матеріалу об стінки циліндра і поверхні шнека, а також за допомогою нагрівачів 7 матеріал розігрівається
до температури плавлення полімеру (~ 150-200^{0}С), перемішується і ущільнюється.

Високий тиск, що виникає в ущільненій зоні, дозволяє продавити в'язкий компаунд через філь'єру 8. Видавлений матеріал (екструдат) 9 виходить з філь'єри і потім охолоджується.

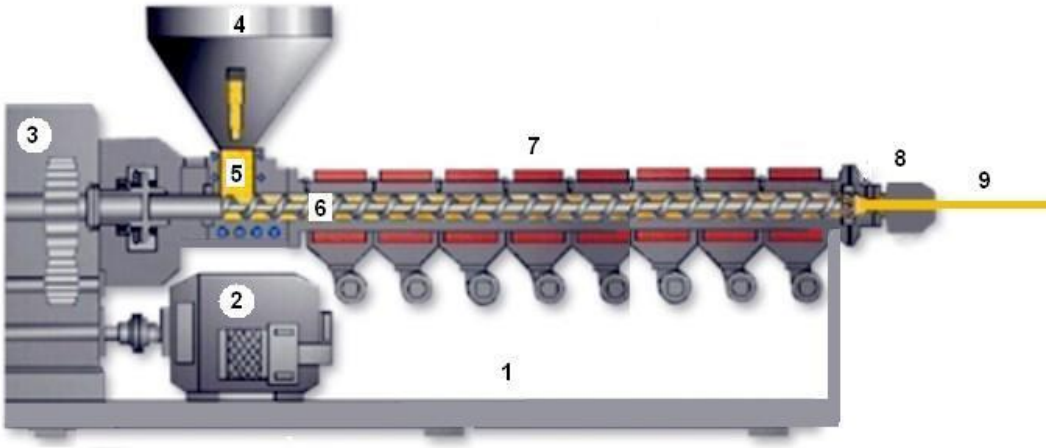


Рис. 4.1. Принципова схема одношнекового екструдера
1 - станина; 2 - електродвигун; 3 - трансмісія; 4 - бункер; 5 - зона завантаження;
6 - обертовий шнек; 7 - нагрівачі; 8 - філь'єра; 9 - екструдат

Основні вузли екструдера монтують на станині 1. Обертання шнека здійснюється за допомогою електродвигуна 2 і трансмісії 3.

Для підтримання необхідної температури у зонах екструдера та черв'яку застосовують ***пристрої нагрівання*** та ***охолодження***. Пристроями нагрівання є електронагрівачі, що характеризуються значеннями потужності (напруги, струму або частоти).

Охолодження здійснюється шляхом примусової подачі холодної води або повітря.

Повний цикл роботи екструдера складається з декількох етапів:

1) розігрів екструдера до заданого технологічними умовами температурного режиму;

2) пуск – перехід від стану, коли продукція на виході екструдера відсутня, до стану, коли вихідна продукція екструдера відповідає заданим характеристикам;

3) нормальне функціонування згідно до вимог технологічного регламенту;

4) зупинення.

У навчальному посібнику обмежимося розглядом етапу нормальної експлуатації, оскільки решта етапів вимагають більш складних способів керування.

Головною задачею керування процесом екструзії є забезпечення однорідності та певної текучості екструдату. Цієї мети можна досягти дотриманням певного температурного поля по довжині екструдера та узгодженістю швидкостей усіх механізмів черв’ячної машини (приймальних, власне черв’яка та тягових пристроїв). Температурне поле та швидкості визначають за властивостями сировини.

Основним документом для створення БД з такою постановкою задачі є Технологічний регламент. У ньому визначені вимоги до якості готової продукції та наведено припустимі діапазони значень режимних параметрів.

Властивості сировини розділяють на такі групи:

- механічні (форма і розмір гранул, сипкість, міцність, еластичність, коефіцієнт тертя);

- фізичні (щільність, теплопровідність, питома теплоємність, температура плавлення, в’язкість розплаву);

- хімічні (розподіл молекулярної маси, вміст газу, термостійкість і т.ін.).

Режимними параметрами, найчастіше визначають температури у зонах обігріву (для прикладу розглянуто 4 зони), швидкість обертання шнека (або частота обертання) та тиск у головці шнека (перед філь'єрою) перед видавлюванням матеріалу. Тиск, одначе, залежить від керувальних змінних.

Отже, пропонується створити автоматизовану інформаційну систему на основі реляційної бази даних ПОЧАТКИВИЙ РЕЖИМ, у якій передбачені різноманітні запити, що дозволяють виконувати пошук тих режимних параметрів процесу екструзії (таких як температури в зонах, швидкість обертання шнека тощо). Вони які найкраще відповідають властивостям нової сировині.

Поставлену задачу можна реалізувати різними способами.

Розглянемо перший спосіб. Для інфологічної моделі виділимо сутності СИРОВИНА та РЕЖИМНИЙ ПАРАМЕТР.

Зупинимось на їх атрибутах, що базуються на Технологічному регламенті. Сутність СИРОВИНА БД буде призначена для пошуку сировини, схожої за властивостями на нову.

З практики відомо наступне:

- властивості задають у певному діапазоні, номінальне значення знаходиться посередині цього діапазону;
- незначні відхилення властивостей не призводять до суттєвих змін режимних параметрів.

Виходячи з цього реляційна даталогічна модель міститиме таблицю СИРОВИНА, у якій фіксуватимуть назву властивості (змінної), одиниці її вимірювання, середнє значення для партії, припустиме відхилення її значення від номінального, а також нижню (НГ) та верхню (ВГ) границі.

Така структура (табл. 4.1) має певну надмірність, оскільки нижню і верхню границі розраховують з припустимого відхилення та числового значення. Ця надмірність доцільна, оскільки процес визначення керувальних впливів досить відповідальний. Особа, що приймає рішення (ОПР) може співставляти як відсоток відхилення, так і конкретні значення керувальних змінних (тобто, така надмірність підвищує надійність системи прийняття рішень). Після аналізу режимних параметрів визнано за доцільне описати їх за тими ж значеннями, що і властивості сировини. Наведемо табл. 4.1, яка це ілюструє.

4.1.2. Опис алгоритму визначення режимних параметрів

При надходженні нової партії сировини дані про її властивості треба внести у таблицю 4.1 (числові значення основних властивостей сировини). Після заповнення нового запису у таблиці СИРОВИНА, відбувається пошук схожих записів у цій же таблиці для кожного з атрибутів. Якщо атрибути нового запису входять у діапазони відповідних атрибутів інших записів (НГ і ВГ), то в систему керування можна подати як завдання регуляторам оптимальні режимні параметри таблиці одного з попередніх записів.

У тому випадку, коли атрибути нового запису виходять за припустимі межі атрибутів існуючих записів, то користувач зупиняється на такому записі, дані якого найближчі до даних нового запису і використовує їх для системи керування. У процесі роботи екструдера ці режимні параметри уточнюють і, підібравши оптимальні, уводять в останній запис для нової сировини.

Таблиця 4.1. **Таблиця початкових режимів**

	Назва властивості чи параметру	**Числове значення**	**Одиниці вимірювання**	**Припустиме відхилення, %**	**НГ**	**ВГ**
Властивості сировини	Щільність					
	Показник текучості розплаву (ПТР)					
	Коеф. еластичності					
	Міцність					
	Температура плавлення					
Технологічні режимні параметри	Температура в зоні I					
	Температура в зоні II					
	Температура в зоні III					
	Температура в зоні IV					
	Тиск у головці					
	Швидкість шнека					

Запропонована база даних спростить визначення початкових значень керувальних впливів для відповідної сировини шляхом швидкого пошуку серед існуючих даних. Під час пошуку передбачено гнучкий алгоритм врахування ступеня близькості поточних значень властивостей до нижньої або верхньої межі припустимих діапазонів.

Опишемо схему цього алгоритму, подану на рис 4.1.

Блок 1 «Результати аналізу сировини» передбачає внесення номінальних значень та припустимих діапазонів властивостей сировини, які отримують з її паспорта або від виробника.

Блок 2 започатковує циклічні дії по перевірці наявності властивостей нової сировини (усього їх *N*) у БД.

Це відбувається наступним чином: спочатку оператор формує запит №1 (блок 3), куди вносить певні діапазони значень властивостей сировини, далі автоматично здійснюється пошук записів у таблиці, які відповідають заявленим нижній та верхній границям діапазонів (перевірка відповідності у – блоці 4).

Якщо у таблиці знайдено необхідні дані, наступним є блок 5 «Вибір режиму із списку». На цьому етапі оператор має обрати один із режимів, представлених у списку результатів пошуку.

Після цього у систему керування передають значення режимних параметрів згідно з результатами пошуку таблиці РЕЖИМНИЙ ПАРАМЕТР і починають переробку нової сировини (**блок 6 –** «Запуск початкового режиму»). Після певного часу експлуатації екструдера, визначають дійсно оптимальні режимні параметри для нової сировини (**блок 7** – «Обраний оптимальний режим»).

Блок 8 «Створити новий запис у БД?». За результатами початкового режиму оператору надається можливість додати новий запис таблиці РЕЖИМНИЙ ПАРАМЕТР БД, або оновити вже існуючий запис (для сировини, яка вже є у базі) з уточненими оптимальними параметрами процесу. Якщо оператор обирає створення нового запису, наступним буде блок 10. Якщо необхідно оновити існуючий запис – блок 9.

Якщо у таблиці властивості нової сировини відсутні, то наступним буде **блок 11** «Задати діапазон властивостей сировини». Оператор у цьому випадку формує запит про попадання властивості нової сировини у діапазон між нижньою та верхньою границями сировини.

Блок 12 «Входить в діапазон по НГР ?». На цьому етапі оператор має визначити чи необхідні дані входять тільки по нижній границі діапазону. Якщо так, то наступним буде **блок 15** «Входить в діапазон по ВГР ?». На цьому етапі оператор має визначити чи необхідні дані входять тільки по верхній границі. Наступним є **блок 13.**

Якщо ж діапазон властивостей сировини не входить по НГР, то переходимо до **блоку 13** « Збільшення діапазону». У цьому блоці оператор має можливість збільшити діапазон значень властивостей сировини. Проте, такий пошук розпочнеться після перевірки **блоку 14,** в якому % відхилення має бути менше % критичного.

Далі цикл повторюють з блоку 3. До блоку 6 переходять тоді, коли переглянуто усі N заявлених властивостей.

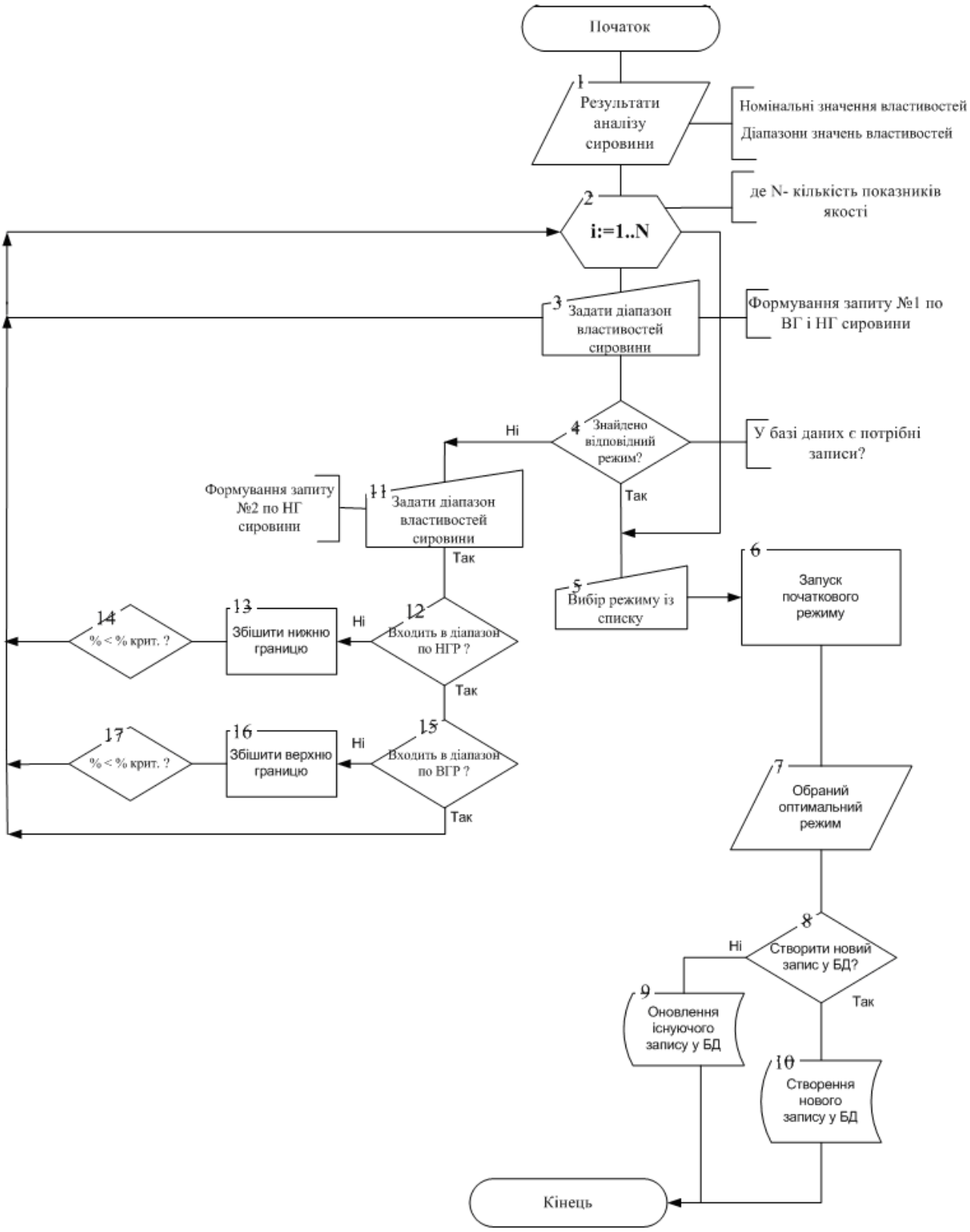


Рис. 4.2. Схема алгоритму роботи бази даних
ПОЧАТКОВИЙ РЕЖИМ

Розглянемо другий спосіб. Він передбачає збереження даних про вказані властивості сировини і режимні параметри у одній сутності РЕЖИМНИЙ ПАРАМЕТР. Інша сутність ТИП СИРОВИНИ міститиме дані про види її дисперсності – гранули різного розміру чи порошок. Це для деяких виробництв треба враховувати.

Алгоритм визначення оптимальних режимних параметрів для цієї даталогічної моделі той самий, що для попередньої.

Схема бази даних ПОЧАТКОВИЙ РЕЖИМ зображено на рис.4.3.

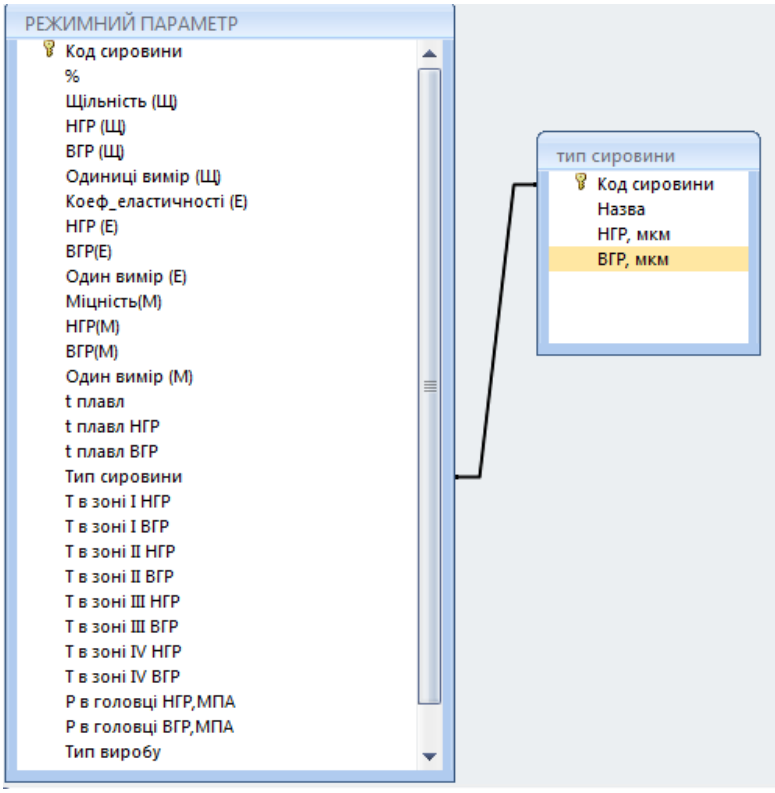


Рис.4.3. Схема бази даних ПОЧАТКОВИЙ РЕЖИМ

Приклад таблиці ТИП СИРОВИНИ зображено на рис. 4.4.

Код сирови	Назва	НГР, мкм	ВГР, мкм
1	Гранули	100	1000
2	Дрібні гранули	1000	5000
3	Порошок тонкий	10	100
4	Порошок зверхтонкий	1	10
5	Порошок ультратонкий	0	1

Рис. 4.4. Документ *MS Access* – таблиця ТИП СИРОВИНИ

Приклад заповнення таблиці РЕЖИМНИЙ ПАРАМЕТР зображено на рис.4.5.

Початковий режим

Код сирови	%	Питома ваг	НГР (П)	ВГР (П)	Одиниці ви	Коеф_еласт	НГР (Е)	ВГР(Е)	Один вимір
1	5	1,14	1,08	1,2	г/см3	1500	1425	1575	МН/м2
2	5	0,95	0,9	1	г/см3	1000	950	1050	МН/м2
3	5	1,41	1,34	1,48	г/см3	3000	2850	3150	МН/м2
4	5	0,92	0,87	0,97	г/см3	1300	1235	1365	МН/м2
5	5	1,18	1,12	1,24	г/см3	1600	1520	1680	МН/м2
6	5	1,36	1,29	1,43	г/см3	2900	2850	3000	МН/м2
7	5	1,05	1	1,1	г/см3	1400	1425	1570	МН/м2
8	5	2,12	2,01	2,23	г/см3	3500	3325	3675	МН/м2
9	5	1,2	1,14	1,26	г/см3	1550	1425	1575	МН/м2
10	5	1,13	1,07	1,19	г/см3	1450	1380	1520	МН/м2

Початковий режим

Міцність(М	НГР(М)	ВГР(М)	Один вимір	t плавл	t плавл НГР	t плавл ВГР	Тип сировин	Т в зоні I НГ	Т в зоні I ВГ
17	16,2	17,9	КДж/м2	240	225	265	Гранули	120	130
3	2,9	3,2	КДж/м2	111	108	115	Гранули	125	130
30	28,5	31,5	КДж/м2	200	180	220	Гранули	135	140
6	5,7	6,3	КДж/м2	163	160	170	Гранули	180	185
16,5	16,2	17,9	Кдж/м2	228	225	265	Гранули	120	130
29	28,6	30	Кдж/м2	245	225	250	Гранули	130	140
17,2	16	17,5	Кдж/м2	255	220	260	Гранули	115	125
15	14	34,5	Кдж/м2	327	311	343	Дрібні гранул	170	180
62	60	70	Кдж/м2	225	220	240	Гранули	120	130
65	61,7	68,3	Кдж/м2	213	210	218	Гранули	130	140

Початковий режим

Т в зоні II ВГ	Т в зоні III Н	Т в зоні III В	Т в зоні IV Н	Т в зоні IV В	Р в головці	Р в головці	Тип виробу	V шнеку,об	Код АСУТП
140	130	140	140	150	10	15	плівка	45	1
150	155	160	165	175	12	17	плівка	36	4
160	170	175	175	185	13	19	плівка	35	2
235	230	245	230	250	15	20	плівка	45	2
140	130	140	140	150	10	15	труба	40	2
155	165	170	165	173	12	15	плівка	32	2
135	135	140	140	152	9	13	плівка	42	1
230	230	245	245	255	22	25	профіль	55	3
140	135	145	145	152	13	16	плівка	45	1
145	145	150	150	155	15	15	плівка	45	1

Рис.4.5. Документ *MS Access* – таблиці РЕЖИМНИЙ ПАРАМЕТР

4.2. Створення інформаційно – дослідницької бази даних моделей та систем керування

Основним завданням створення такої БД є скорочення часу пошуку інформації про існуючі автоматичні та автоматизовані системи контролю та керування в галузі переробки пластмас у черв'ячних екструдерах.

Різноманіття систем керування ускладнює задачу вибору конкретної системи керування. Тому розглянемо задачу класифікації систем керування та створення відповідної бази даних, в якій буде передбачено зберігати різноманітну інформацію.

Враховуючи складність процесу екструзії і тривалість використання його в промисловості різних країн, зрозумілою стає різноманітність створених для нього моделей та систем контролю і керування. У наш час пошук інформації і доступ до неї перетворюється в окрему проблему. Суттєву допомогу в отриманні та обробці інформації надають автоматизовані інформаційні системи (АІС).

Оскільки для процесу екструзії розроблено багато різних систем керування простих і складних за структурою та функціями, то варто створити таку базу даних, яка дозволить ознайомитися як зі структурою, так і з описом цих систем. На жаль, відкриті бази даних такого типу, відсутні в мережі Інтернет.

Така інформаційна система дозволить скоротити час ознайомлення користувачів, спрощення науково-технічної творчості студентів і фахівців з існуючими рішеннями в області автоматизації черв'ячних екструдерів, забезпечить швидкий пошук даних, які представлені в зручній формі і виконати моделювання обраної системи.

Така інформаційна система орієнтована в першу чергу на студентів та науковців, а також на працівників проектних організацій, які не мають значного обсягу знань у цій предметній області. Розвитком цього напрямку може бути створення баз даних для систем керування іншими технологічними процесами.

Дамо назву базі даних БАЗА ДАНИХ ПРОЦЕСУ ЕКСТРУЗІЇ.

Згідно до поставленої задачі вона буде містити два об'єктних відношення: АСУТП і МОДЕЛЬ. Запропоновано наступні схеми цих відношень:

- АСУТП (Керована змінна, Схема системи управління, Опис системи, Модель, Джерело інформації);
- МОДЕЛЬ (Тип системи, Імітаційна модель).

Відношення АСУТП дає можливість отримати інформацію про відомі системи керування і використовувати її для вибору аналогів при створенні нового способу або пристрою керування, підготовки аналітичних оглядів у навчальних, проектних і наукових працях.

Уведення в базу даних відношення МОДЕЛЬ дозволяє подати структурні (імітаційні) моделі одноконтурної, двоконтурною і каскадної систем у *MatLab+Simulink*. Користувач може зібрати з них модель обраної системи управління в цілому або якогось певного фрагмента для вибору оптимальних параметрів і порівняння з іншими системами.

Користувачу надається можливість, знаючи керувальну і керовану змінні, а також модель екструдера, підібрати схему автоматизації та імітаційну модель у *MatLab+Simulink*. Додатково можна буде отримати опис схеми системи, конструкцію екструдера та джерело інформації.

4.2.1. Схема даних

У розробленій базі даних використані три основні типи даних: текстовий, числовий та OLE (для впровадження інших об’єктів). Таблиці відношень АСУТП та МОДЕЛЬ у БАЗІ ДАНИХ ПРОЦЕСОМ ЕКСТРУЗІЇ мають поля, які унікально ідентифікують запис, в даному випадку таким первинним ключем є: **Код АСУТП.** Відношення АСУТП включає в себе атрибут Модель, у якому зазначені типи систем, що входять в загальну схему управління, наприклад, одноконтурна, каскадна і т. д.

Зобразимо зв'язок між сутностями у вигляді нотації «Пташина лапка» рис 4.6:

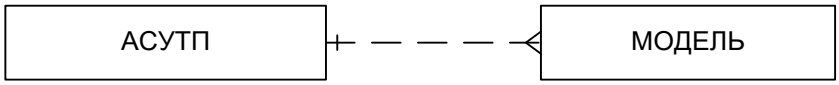


Рис. 4.6. Зображення відношень у нотації «Пташина лапка»

Структура двох відношень МОДЕЛЬ та АСУТП в *Microsoft Access* зображено на рис.4.7.

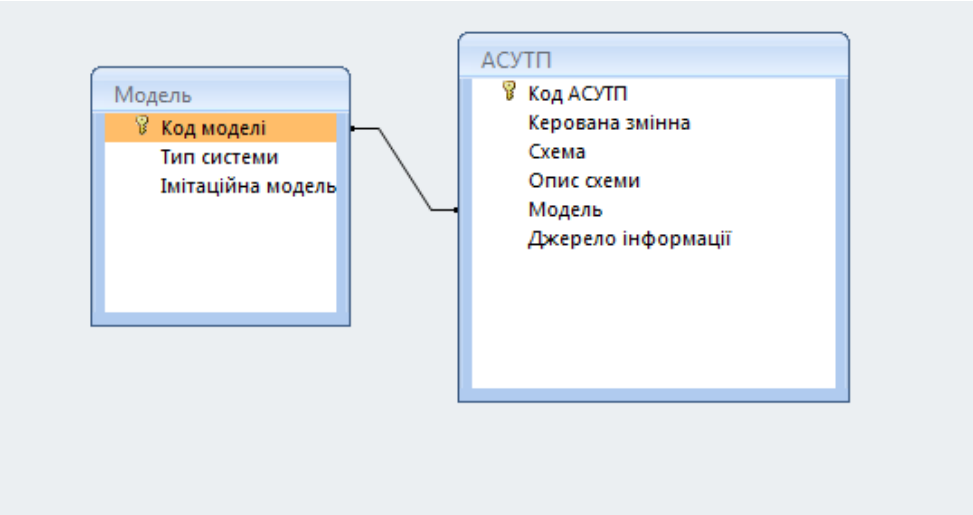


Рис. 4.7. Схема даних АСУТП

4.2.2. Приклад бази даних

Зображення відношення АСУТП у режим **Таблиця** зображено на рис. 4.8.

Код АСУТП	Керована змінна	Схема	Опис схеми	Модель	Джерело інформації
1	Температура	Bitmap Image	:кой макросов	двоконтурна	http://patentdb.su/3-83
2	Температура	Bitmap Image	:кой макросов	каскадна	http://msd.com.ua/obo
3	Тиск	Bitmap Image	:кой макросов	двоконтурна	http://patentdb.su/3-13
4	Тиск	Bitmap Image	:кой макросов	багатоконтурн	http://patentdb.su/3-89
5	Якість	Bitmap Image	:кой макросов	двоконтурна	http://patentdb.su/3-13
6	Якість	Bitmap Image	:кой макросов	багатоконтурн	file://localhost/C:/User
7	Температура, Тиск	Bitmap Image	:кой макросов	багатоконтурн	http://bankpatentov.ru

Рис. 4.8. Документ *Microsoft Access* – відношення АСУТП у режимі **Таблиця**

Зображення відношення АСУТП в режимі **Конструктор** зображено на рис. 4.9.

Имя поля	Тип данных
Код АСУТП	Числовой
Керована змінна	Текстовый
Схема	Поле объекта OLE
Опис схеми	Поле объекта OLE
Модель	Числовой
Джерело інформації	Гиперссылка

Рис. 4.9. Документ *Microsoft Access* – відношення АСУТП бази даних у режимі **Конструктор**

Зображення відношення МОДЕЛЬ у режим **Таблиця** зображено на рис. 4.10.

Модель

Код моделі	Тип системи	Імітаційна модель	*Добавить поле*
1	одноконтурна	Bitmap Image	
2	каскадна	Bitmap Image	
3	двоконтурна	Bitmap Image	
4	багатоконтурна	Bitmap Image	

Рис. 4.10. Документ *Microsoft Access* для відношення МОДЕЛЬ у режимі **Таблиця**

На рис. 4.11 наведено форму для створеної бази даних у *програмі Microsoft Access*, з якої можна перейти до кожного з об’єктних відношень при натисканні відповідної кнопки (заставка для форми запозичена з мережі Інтернет).

Рис. 4.11. Форма для бази даних

При натисканні на кнопку **АСУТП** відбувається перехід до форми, зображеної на рис. 4.12, а при натисканні на кнопку **Модель** – до форми, зображеної на рис.4.13.

Рис. 4.12. Документ *Microsoft Access* – форма АСУТП

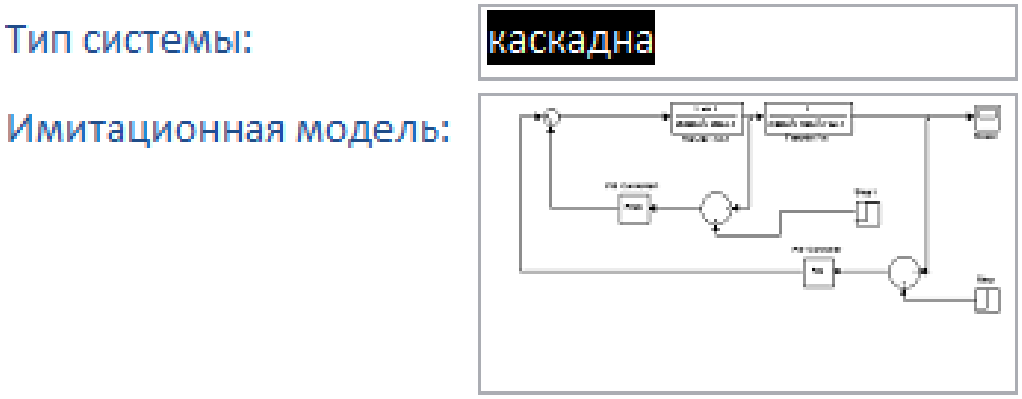


Рис. 4.13. Документ *Microsoft Access* – форма МОДЕЛЬ

4.3. Створення бази даних для вибору технічних засобів автоматизації

Розглянемо питання створення бази даних для технічних засобів автоматизації (ТЗА), які використовують при автоматизації компресорних станцій та інших близьких виробництв.

Кожну предметну область характеризує значна кількість відношень. Вдале використання найбільш інформативних з них та розкриття властивостей відношень через їх атрибути (характеристики) дозволяє створити вдалу модель досліджуваної предметної області.

Предметна область в нашому випадку включає відношення, які сформовані у такі групи:

- первинні перетворювачі;
- дистанційні нормуючі перетворювачі;
- регулятори;
- вторинні прилади.

Кожна група складається з підгруп пристроїв, оскільки первинні перетворювачі (датчики) мають різне призначення: датчики температур, датчики тиску, витрат тощо, та різні принципи дії і інші характеристики. Це стосується і інших груп пристроїв.

Однак для кожної групи ТЗА є і спільні риси – певні метрологічні характеристики, фірми виробники, граничні умови режимних параметрів (для тисків, температур тощо) та інше.

Зрозуміло, що структура бази даних може мати альтернативні варіанти.

Для визначення структури бази даних “ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ” дослідимо можливі інформаційні об’єкти всієї області.

4.3.1. Визначення інформаційних об’єктів предметної області

При визначенні інформаційних об’єктів слід сказати , що важливою задачею є зберігання та використання характеристик технологічних змінних (параметрів). Також можна зазначити, що велика різниця між характеристиками кожної моделі пристроїв не дає змогу створити одне відношення, тобто одну велику таблицю. Тому запропоновано створити декілька відношень – для кожної підгрупи пристроїв окреме відношення (таблицю).

Відношення ДАТЧИКИ ТИСКУ можна записати у вигляді такої реляційної моделі (рис. 4.14).

Код	Тип	Назва	Min Діап	Max Діап	Вихід сигн	Живлення	Похибка	Min темп	Max темп	Зах IP-65	Фірма
501	датчик тиску	SML-10,0	0	2000	4-20 мА	4-30 VDC	0,5	-40	100	☑	304
502	датчик тиску	SML-20,0	0	2000	0-10 V	12-30 VDC	0,5	-40	100	☑	304
503	датчик тиску	SML-30,0	0	2000	0,5-4,5 V	5,0 VDC	0,5	-40	100	☑	304
504	датчик тиску	SML-40,0	0	2000	0,5-4,5 V	8-32 VDC	0,5	-40	100	☑	304
505	датчик тиску	SMK	0	2000	4-20 мА	12-32 VDC	0,5	-40	180	☑	304
506	датчик тиску	SML-M01	0	2000	PWM	8-32 VDC	0,5	-40	110	☑	304
507	датчик тиску	SML-M02	0	4000	4-20 мА	0-10 V	0	-40	105	☑	304
508	датчик тиску	SML-CAN	1	2000	1 Mbit/sec	CAN - Protocol	0,5	-40	100	☑	304
509	датчик тиску	SML-PS1	0	25	0-10 V	12-32 VDC	0,25	-40	100	☑	304
510	датчик тиску	KS-1 T+P	0	2000	0,5-4,5 V	5,0 VDC	0,5	-50	200	☑	304
511	датчик тиску	DC-3	100	2000	Реле , 24V ,	12-32 VDC	1,5	-10	80	☑	304
512	датчик тиску	DC-4	0,6	4000	Реле , 24V ,	10-32 VDC	1,5	-40	125	☑	304
513	датчик тиску	DC-DG	5	2000	4-digit Displa	12-27 VDC	1,5	-40	80	☑	304
514	датчик тиску	SML-SME	0	250	0,5-4,5 V	8-32 VDC 5,0 V	0,5	-40	100	☑	304
515	напорометр	НС-П1	0	40	4-20 мА	12-32 VDC	1	-50	50	☐	306
0			0	0			0	0	0	☐	0

Рис. 4.14. Реляційна модель відношення ДАТЧИКИ ТИСКУ

Опишемо схему цього відношення.

ДАТЧИКИ_ТИСКУ:

- “Тип” – тут зазначено тип приладу (його направленість)
- “Назва” – Особиста назва приладу від виробника
- “Min Діап” – мінімальний діапазон приладу
- “Max Діап” – максимальний діапазон приладу
- “Вихід сигн” – вихідний сигнал приладу
- “Живлення” – сигнал живлення приладу
- “Похибка” – похибка вимірювань приладу

- “Min темп” – мінімальна температура дієздатності приладу
- “Max темп” – максимальна границя температури роботи пристрою
- “Зах IP 65” – наявність у приладі стандартного степеня захисту IP 65
- “Фірма” – код фірми виробника або фірми продавця.

Відношення МІКРОПРОЦЕСОРНІ ПРИЛАДИ подано у вигляді реляційної моделі на рис. 4.15.

Код	Назва	Тип	Ан Вход	Ан Вих	Дискр Вход	Дискр Вих	Заказ	Min Темп	Max Темп	Потужн	Період Вимір	Похибка %	Фірма
101	Мик-51	Контроллер	4	1	3	5	☑	0	50	15	0,1	0,2	301
102	Мик-21	Регулятор	1	0	2	4	☐	0	50	9	0,1	0,2	301
103	Мик-25	Регулятор	2	0	2	4	☐	0	50	9	0,1	0,2	301
104	Мик-22	Регулятор	0	0	2	4	☐	0	50	9	0,1	0,2	301
105	Мик-2	Регулятор	0	0	2	4	☐	0	50	9	0,1	0,2	301
106	МТР-8	терморегул	8	1	0	8	☑	0	50	8,5	0,2	0,2	301
107	МТР-11	терморегул	0	0	0	0	☐	0	0	0	0	0	0
0			0	0	0	0	☐	0	0	0	0	0	0

Рис. 4.15. Реляційна модель відношення МІКРОПРОЦЕСОРНІ ПРИЛАДИ

Опишемо схему цього відношення.

МІКРОПРОЦЕСОРНІ_ПРИЛАДИ**:**

- “Назва” – Назва приладу від виробника
- “Тип” – тут зазначено тип приладу(його направленість)
- “Ан вход” – кількість аналогових входів регулятора
- “Ан вих” – кількість аналогових виходів регулятора
- “Дискр вход” – кількість дискретних входів приладу
- “Дискр вих” – кількість дискретних виходів приладу
- “Заказ” – можливість заказу кількості виходів та входів замовником
- “Min темп” – мінімальна робоча температура навколишнього середовища регулятора

- “Мах темп” – максимальна робоча температура навколишнього середовища регулятора
- “Потужн” – споживча потужність приладу
- “Період вимір” – період вимірювання приладу в секундах
- “Похибка” – похибка вимірювань приладу %
- “Фірма” – код фірми виробника або фірми продавця

Реляційну модель відношення ДАТЧИКИ СИЛИ подамо на рис. 4.16.

Код	Тип	Назва	Модель	Інтерв	max Отк	Темп 0	Темп номін	min Темп	max Темп	Захист	Живл	Вихід	Фірма
401	Прості елемент	Megatron	FF100	4,22,44,22	0	0,36		-10	85	----------	5тип	змінний с	304
402	Прості елемент	Megatron	TF100	1,2,4,10,2	0,25	0,36	0,25	-5	50	----------	10тип,15	2 mV/V	304
403	Прості елемент	Megatron	K100	0,5 ,1,2,5,	0	0,03		-10	70	----------	10тип,15	змінний с	304
404	Прості елемент	Megatron	K250	9,81:19,62	0	0,03		-5	50	----------	5....2		304
405	Індустріал	Megatron	K200	10	0,15	0,8	0,8	0	40	IP 00	5тип, 8м	0,65 mV/V	304
406	Індустріал	Megatron	K200I	50	0,1	0,1	0,1	0	55	IP 00	10тип, 15	2 mV/V	304
407	Індустріал	Megatron	K300	60;100;15	0,1	0,5	0,5	-10	50	IP 65	10тип, 20	2 mV/V	304
408	Індустріал	Megatron	K500	50;200;50	0,15	0,2	0,2	-10	40	IP 00	10тип, 15	2 mV/V	304
409	Індустріал	Megatron	K700	200, 500,	0,25	0,1	0,15	0	55	IP 40	10тип, 15	20 mV/0-5	304
410	Індустріал	Megatron	K700I	200, 500,	0,4	0,2	0,2	0	55	IP 40	10тип, 15	20 mV/0-5	304
411	Індустріал	Megatron	K800		0	0	0	0	0				0
0					0	0	0	0	0				0

Рис. 4.16. Реляційна модель відношення ДАТЧИКИ СИЛИ

Опишемо схему цього відношення.

ДАТЧИКИ СИЛИ**:**

- “Тип” – тип приладу (його направленість)
- “Назва” – назва приладу від виробника
- “Модель” – модель приладу від виробника
- “Інтерв” – інтервал вимірювань від 0 до (Н)
- “Мах откл” – максимальне відхилення від номіналу (%F.S.)
- “Темп 0” – температурний коефіцієнт нульової точки (%F.S.)
- “Темп номін” – температурний коефіцієнт номіналу (%F.S./10K)
- “min темп” – мінімальна температура
- “max темп” – максимальна температура

- “захист” – вид захисту приладу за міжнародним стандартом
- “живл” – характеристика сигналу живлення
- “вихід” – вихідний сигнал приладу
- “Фірма” – код фірми виробника або фірми продавця.

Відношення ПЕРТВОРЮВАЧІ можна записати у вигляді реляційної моделі, яку показано на рис. 4.17.

	Код	Назва	перетвор	Обл вик	min сигн	max сигн	вихід сигн	Кіл-сть кан	Схема підкл	Спож потужн	Фірма
+	201	БПВІ-1	Взаємної індук	Системи вимір	0	10мі	☑	1	4-х провідна	6	301
+	202	БПО-32	термоопору	Вимір темпера	3	200 О	☑	2	3-х провідна	6	301
+	203	БПО-42	Термоопору	Вимір темпера	3	200 О	☑	2	4-ч провідна	6	301
✳	0				0		☐	0		0	0

Рис. 4.17. Реляційна модель відношення ПЕРТВОРЮВАЧІ

Опишемо схему цього відношення.

ПЕРЕТВОРЮВАЧІ**:**

- “Назва” – назва приладу від виробника
- “перетвор” – вид перетворювача, його напрямок
- “Обл вик” – область використання приладу
- “min сигн” – мінімальне значення вхідного сигналу
- “max сигн” – максимальне значення вхідного сигналу
- “вихід сигн” – стандартний вихідний сигнал
- “Кіл– сть кан” – кількість каналів перетворювача
- “Схема підкл” – схема підключення перетворювача
- “Спож потужн” – споживча потужність перетворювача
- “Фірма” – код фірми виробника або фірми продавця.

Оскільки база даних ТЕХНІЧНИ ЗАСОБИ АВТОМАТИЗАЦІЇ створюється для оснащення технічними засобами, то однією з характеристик, як ми бачимо, є фірми – виробники або продавці.

Таким чином, ми бачимо, що виникає потреба в уведені додаткового інформаційного об’єкту ФІРМИ_ВИРОБНИКИ. Це відношення можна записати у вигляді реляційної моделі, яку показано на рис. 4.18.

	Код	Назва Фірми	Мікроконтр	Мікрорегул	Перетвор	Датчики	П З	Країна	Індекс	Місто	Вул	Тел	E-MAIL	Web
+	301	"Мікрол"	☑	☑	☑	☐	☑	Україна	76036	Івано-Фра	Автолитм	8(0342) 5	microl@n	www.mic
+	302	ПНВП "Промп	☑	☑	☐	☐	☐	Україна	10001	Житомир	Гоголя ,5	8(0412)4		www.pro
+	303	АОЗТ "ТЄРА"	☑	☑	☑	☐	☑	Україна	14030	Чернігів	50 років	8(0462)9	info@ao-t	www.ao-
+	304	Мікро прилад	☐	☐	☐	☑	☐	Україна	3115	Київ	Котельні	8(044)45	info@mic	www/mic
*	0		☐	☐	☐	☐	☐		0					

Рис.4.18. Реляційна модель відношення ФІРМИ_ВИРОБНИКИ

Опишемо схему цього відношення.

ФІРМИ_ВИРОБНИКИ:

- “Назва фірми” – юридична назва фірми або виробника
- “Мікроконтр” – наявність мікроконтролерів в асортименті
- “Мікрорегул” – наявність мікрорегулятора в асортименті
- “Перетвор” – наявність перетворювачів в асортименті
- “Датчики” – наявність датчиків в асортименті виробника
- “ПЗ” – наявність програмного забеспечення в асортименті
- “Країна” – країна виробника або фірми продавця
- “Індекс” – поштовий індекс адреси виробника
- “Місто” – місто виробника або фірми продавця
- “Вул” – юридична адреса виробника або фірми продавця
- “Тел” – контактні телефони та факси виробника
- “E– MAIL” – електронна пошта фірми
- “WEB” – офіційна веб– сторінка виробника або фірми продавця.

Кожна таблиця складається із записів і полів. Кількість полів у запису і їхній тип визначаються в процесі конструювання таблиці. Наприклад, на етапі складання таблиці паспортів необхідно створити поля для назв змінних, одиниць, періодичності та способу їх вимірювання і т.п.

Тип даних у кожному полі залежить від того, яка інформація буде введена. Наприклад, у випадку введення назви приладу ми використовували текстову поле (реляційна модель фірми виробники; поле “назви фірми”), а коли треба ввести робочі граничні температури приладу, використали числові поля (реляційна модель мікропроцесорні прилади; поле ”Мах темп”). Якщо зустрічалась характеристика приладу, яка або присутня або ні, то використовували логічні поля (реляційна модель ДАТЧИКИ ТИСКУ; поле ”захист IP 65”).

4.3.2. Визначення первинних ключів та ключів зв’язку. Схема даних

У *MS Access* таблицею є сукупність даних, об'єднаних загальним смислом. Обов'язковою умовою нормальної роботи з базою даних є однозначна ідентифікація кожного запису. Поле (поля), яке дозволяє ідентифікувати запис у відношенні оголошують полем первинного ключа.

У *MS Access* існує автоматична нумерація записів (тип “Лічильник”), яка може слугувати за первинний ключ, але запропоновано спеціальне поле під назвою “Код”. Це поле оголошено полем первинного ключа та заповнюється вручну так, щоб номер кожного запису не повторювався. На рис. 4.19. зображено вікно **Конструктора таблиць** з вибраним первинним ключем “Код”.

Фірми виробники : таблица

Имя поля	Тип данных	Описание
Код	Числовой	Особистий код кожної Фірми виробника
Назва Фірми	Текстовый	Назва Фірми
Мікроконтр	Логический	Наявність мікроконтроллерів в асортименті
Мікрорегул	Логический	Наявність мікрорегуляторів в асортименті
Перетвор	Логический	Наявність блоків перетворювачів сигналів

Рис. 4.19. Визначення первинного ключа для відношення ФІРМИ_ВИРОБНИКИ

Створення схеми даних потрібно для встановлення зв’язків між таблицями та при подальшому створенню запитів, адже запити можна створювати для добору даних як з однієї, так і з декількох таблиць. При цьому слід враховувати наявність зв'язків між таблицями. Якщо розробляється запит до декількох таблиць, між якими немає зв'язку, то *MS Access* розглядає кожну таблицю як незалежний об'єкт і дозволяє скласти будь-яку можливу комбінацію з полів у вікні конструктора.

У цьому випадку модель реляційної бази даних працює правильно лише в тому випадку, якщо коректно установлені відношення між полями таблиць бази. Обов'язковою умовою установки зв'язку між таблицями є наявність двох однакових полів.

У нашому випадку головною є таблиця ФІРМИ_ВИРОБНИКИ, усі інші таблиці пов’язані з нею через її первинний ключ (Код). Схему відношень між розглянутими інформаційними об’єктами подано на рис 4.20.

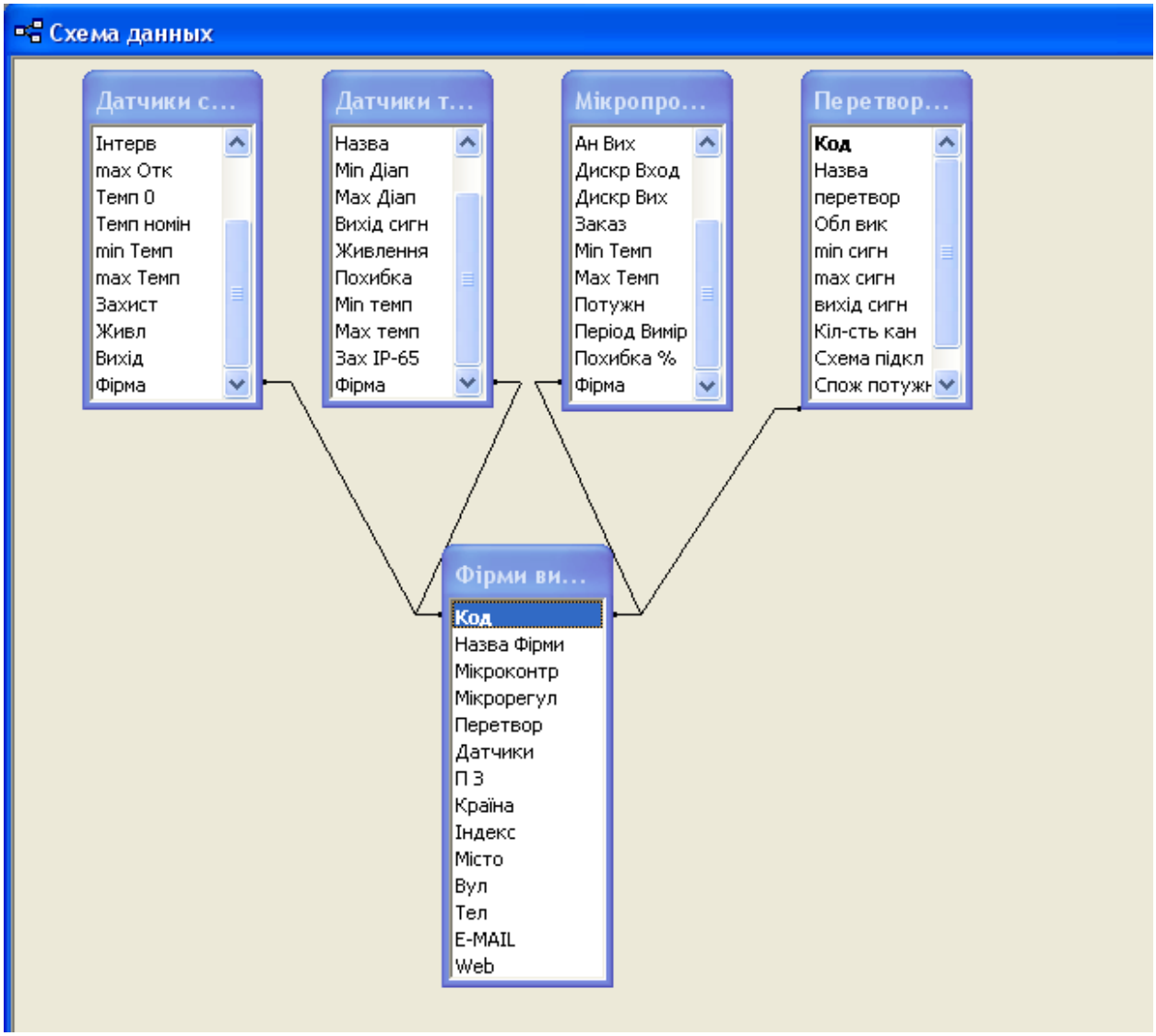


Рис.4.20. Схема даних об’єктів предметної області ТЕХНІЧНІ ЗАСОБИ АВТОМАТИЗАЦІЇ

Перелік можливих таблиць по ТЗА визначаємо за структурою автоматичних систем керування. Загальна її структура наведена на рис. 4.21.

Рис. 4.21. Загальна структура автоматичних систем керування

Структура БД ТЗА визначена моделлю предметної області ТЗА, а саме переліком інформаційних об'єктів та їхніми взаємозв'язками.

Кожний інформаційний об'єкт предметної області відображається таблицею. Вікно бази даних з переліком таблиць наведено на рис. 4.22.

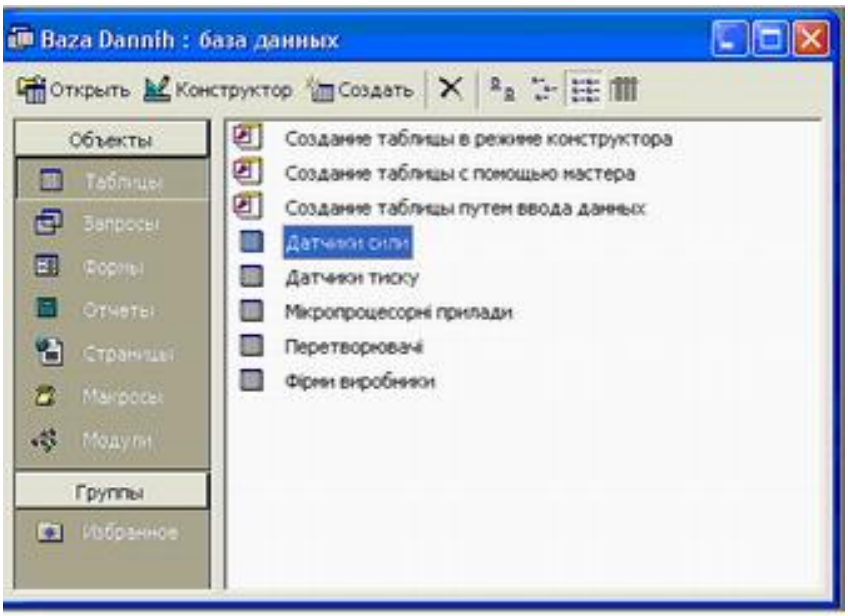


Рис. 4.22. Вікно об'єктів бази даних ТЗА

Для прикладу розроблені таблиці тих датчиків, які використовуються при автоматизації компресорних станцій.

Треба, однак, зауважити, що принцип побудови БД дозволяє нарощувати кількість інформаційних об’єктів без змін існуючих таблиць.

Запропоновано первинні перетворювачі класифікувати за призначенням, тобто, за технологічним параметром, який вимірюється.

Наприклад : ДАТЧИКИ ТИСКУ, ДАТЧИКИ СИЛИ, ДАТЧИКИ ТЕМПЕРАТУРИ і т.д.

При розширені БД може бути застосований і дещо інший спосіб визначення інформаційних об’єктів, а, отже, і таблиць – за галузями застосування. Наприклад, при автоматизації паперового виробництва використовують специфічні прилади, для яких недоцільно створювати окрему таблицю.

Таблиці пов’язують також з перетворювачами, регуляторами, виконавчими механізмами та ін.

Користувачі використовують бази даних для різних задач, які називають прикладними.

До прикладних задач користувача слід віднести, насамперед, вибір потрібного приладу із бази даних для використання у системі автоматизації. Для цього треба знати, які характеристики користувач буде враховувати при підборі того або іншого приладу.

При створені проекту з автоматизації технологічних процесів виникають, зокрема, наступні задачі :

- для певної технологічної змінної і визначеного діапазону її зміни вибрати датчик (при цьому, враховуючи його виробника, точність, ціну і т.д.);
- вибрати перетворювачі за заданими вхідними сигналами.

Ці задачі, у загальному випадку, можна розв’язати за допомогою створених таблиць, але це не зовсім раціонально, оскільки значна кількість інформації перевтомлює і відволікає користувача від головної мети.

Такі пошуки виконують за допомогою запитів, які дозволяють виводити тільки необхідну інформацію

Наведемо приклади можливих пошукових задач:

- пошук по галузі;
- пошук за типами приладів;
- пошук за параметрами, притаманними виключно одному типу приладів;
- пошук за точністю вимірювань приладу;
- пошук за границями температур навколишнього середовища, для нормальної роботи приладу
- пошук за фірмами – виробникам та фірмами – продавцями.

Створимо для прикладу запит з таблиці ДАТЧИКИ_ТЕМПЕРАТУРИ, який буде містить характеристики мінімальної, максимальної температури вимірювання, похибку та фірму виробника (рис.4.22).

Датчики температури Запрос : запрос на выборку

Назва	Min	Max	Похиб	Фірма
ТСП-5071	-200	600	1	306
ТСМ-5071	-50	150	0,5	306
ТСП-6097	-50	250	1	306
ТСП-8012	-50	100	1,5	306
ТСП-175	-50	500	1	306
ТСП-712	-50	400	1,5	306
	0	0	0	0

Запись: 3 из 6

Рис.4.22. Запит на вибірку для датчиків температури.

СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1 Зарицька О. Л. Бази даних та інформаційні системи: методичний посіб. – Житомир: Вид-во ЖДУ ім. І. Франка, 2010. – 132 с.

2 Бураков, П.В. Введение в системы баз данных [Текст]: учебн. пособ. / П.В. Бураков, В.Ю. Петров. – СПб.: СПбГУ ИТМО, 2010. – 128 с.

3 Глушков, С.В. Базы данных [Текст]: учебный курс / С.В. Глушков, Д.В. Ломотько. – Харьков: Фолио, 2000. – 504 с.

4 Дейт, К. Дж. Введение в системы баз данных [Текст] / К. Дж. Дейт. – – М. : Вильямс, 2000. – 848 с.

5 Диго, С. М. Базы данных: проектирование и использование [Текст]: учебник / С. М. Диго. – М. : Финансы и статистика, 2005. – 592 с.

6 Методологія інформаційних систем та баз даних: теоретичний і практичний підходи : навч. посібник / уклад. Ю.О. Ушенко, М.Л. Ковальчук, М.С. Гавриляк, А.Л. Негрич. – Чернівці : Чернівецький нац. ун-т ім. Ю. Федьковича, 2021. – 240 с.

7 Заверач, М.М. Бази даних. Інформаційні системи [Текст]: навч. посіб. / М. М. Заверач,, В. В Третько. – Хмельницький : ХНУ, 2007, – 303 с.

8 Карпова, Т. С. Базы данных: модели, разработка, реализация [Текст] : учебное пособие / Т. С. Карпова. – СПб. : Питер, 2001. – 304 с.

9 Ролланд, Ф. Д. Основные концепции баз данных [Текст]/ Ф. Д. Ролланд. – М. : Вильямс, 2002, –256 с.

10 Харрингтон, Д. Разработка баз данных [Текст]/ Д. Харрингтон. – М. : ДМК Пресс, 2005. – 272 с.

11 Роб, П. Системы баз данных: проектирование, реализация и управление [Текст] / П. Роб, К. Коронер – СПб.: БХВ – Петербург, 2004, – 1040 с.

12 Організація реляційних баз даних : навч. посіб. / Я.Г. Куваєв, О.А. Жукова, І.А. Сечкін – 2-ге вид., допов. та переробл. – Дніпро : НГУ, 2017. – 157 с.

13 Гайна Г.А. Основи проектування баз даних: Навчальний посібник. − К.: КНУБА, 2005. – 204 с.

14 Основи баз даних: навч. посіб. / І. О. Завадський. – К.: Видавець І. О. Завадський, 2011. – 192 с.

15 Бекаревич, В.Ю. СУБД *Access* для *Windows* 95 в примерах [Текст] / В.Ю. Бекаревич, Н.В. Пушкина. – СПб.: *BHW* – Санкт-Петербург, 1997. – 400 с.

ДОДАТКИ

Додаток 1.

БЛОК ТЕСТІВ ДО ТРЬОХ ТЕМ КУРСУ

1
Баллов: --/1
Поставте у відповідність поняття і їх смисл:

Степінь відношення	Кількість його кортежів/Список імен його атрибутів/ Кількість його атрибутів
Потужність відношення	Кількість його кортежів/Список імен його атрибутів/ Кількість його атрибутів
Схема відношення	Кількість його кортежів/Список імен його атрибутів/ Кількість його атрибутів

2
Баллов: --/1

Задайте первинні і зовнішній ключі двох інформаційних об'єктів ТОВАР і **ВИРОБНИК**.

Товар

Артикул	Назва товару	Колір Кількість

Виробник

Назва фірми	E-mail Країна	Адреса	Телефон

Первинний ключ для ТОВАР	Артикул/ Назва фірми
Первинний ключ для ВИРОБНИК	Артикул/ Назва фірми
Зовнішній ключ	Артикул Назва фірми

3
Баллов: --/1

Поставте у відповідність пункти класифікації СКБД і типи цих систем

за мовами	Документальні/Фактографічні/Клієнт-сервер/Файл-сервер/Замкнені/З базовою мовою централізовані/Розподілені сіткові
за призначенням	Документальні/Фактографічні/Клієнт-сервер/Файл-сервер/Замкнені/З базовою мовою централізовані/Розподілені сіткові
за способом доступу до даних	Документальні/Фактографічні/Клієнт-сервер/Файл-сервер/Замкнені/З базовою мовою централізовані/Розподілені сіткові
за локалізацією	Документальні/Фактографічні/Клієнт-сервер/Файл-сервер/Замкнені/З базовою мовою централізовані/Розподілені сіткові
за структурами даних	Документальні/Фактографічні/Клієнт-сервер/Файл-сервер/Замкнені/З базовою мовою централізовані/Розподілені сіткові

4
Баллов: --/1

Які пари термінів застосовують на різних етапах проєктування БД?

Кортеж	Поле/Стовпець/Атрибут
Рядок	Поле/Стовпець/Атрибут
Запис	Поле/Стовпець/Атрибут
Екземпляр сутності	Поле/Стовпець/Атрибут

5
Баллов: --/1

Що означає "інтегрованість даних" ?

Выберите один ответ.

- a. одні дані входять в інші як "матрьошка" (інтегровані)
- b. дані, які можна підсумовувати (інтегрувати)
- c. дані охоплюють широке коло питань предметної області

6

Баллов: --/1

Наведено інформаційний об’єкт СЕСІЯ. Вкажіть номер описового атрибуту, який залежить від повного складу первинного ключа.

ПІП/б	Спеціальність	Номер групи	Номер у списку	Дисципліна	Оцінка
Котляр І.І.	ЛА	61	15	Інформатика	відмінно
Котляр І.І.	ЛА	61	15	Вища математика	добре
Котляр І.І.	ЛА	61	15	Фізика	добре
Бовкун П.П.	ЛЕ	71	3	Інформатика	добре

Выберите один ответ.

- a. 4
- b. 5
- c. 2
- d. 3
- e. 6
- f. 1

7

Баллов: --/1

Скільки полів матиме відношення, наведене на рисунку, якщо його перевести у 1NF?

Прізвище	Адреса	Телефон
Корнійчук	вул. Зелена, буд. 2, кв. 36.	123-45-67 домашній 765-43-21 робочий
Савченко	вул. Хрещатик, буд.15, кв. 196.	8 (916) 322-22-33 мобільний

Выберите один ответ.

- a. 6
- b. 2
- c. 4
- d. 5

8
Баллов: --/1
На якому етапі проєктування БД створюють інфологічну модель?
Выберите один ответ.

- a. Коли заповнюють БД інформацією, яка логічно між собою пов"язана
- b. Коли розділяють предметну область на сутності та зв"язки між ними
- c. Вже створена БД і пояснюють логіку її будови замовнику

9
Баллов: --/1

Переведення відношення у першу форму передбачає:

Выберите один ответ.

- a. ключ зробити обов"язково числом
- b. вміст полів зробити елементарним
- c. перейти від складеного ключа до простого

10
Баллов: --/1
Що таке база даних?
Выберите по крайней мере один ответ:

- a. Набір будь-якої інформації, яка цікава та корисна людині з різних предметних областей
- b. пойменована структурована сукупність даних, що відображає стан об'єктів і їх відношення в даній предметній області
- c. сукупність взаємопов'язаних даних, які зберігаються разом, вона має мінімальну надмірність для використання у декількох задачах
- d. Пойменована структурована інформація, яка зберігається на базовому (основному) комп"ютері мережі
- e. сукупність інформації про предметну область без вимог до її структури

11
Баллов: --/1

Що може бути багатозначним?

Выберите по крайней мере один ответ:

- a. сутність
- b. зв"зок
- c. атрибут

12
Баллов: --/1

Яку модель створюють першою при проєктуванні БД?

Выберите по крайней мере один ответ:

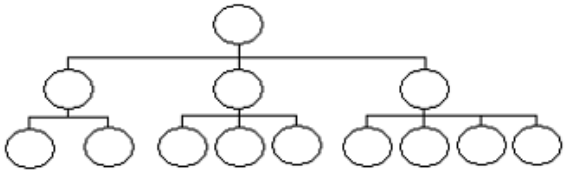
- a. Реляційну
- b. інфологічну
- c. Сутність-зв"язок
- d. Даталогічну

13
Баллов: --/1

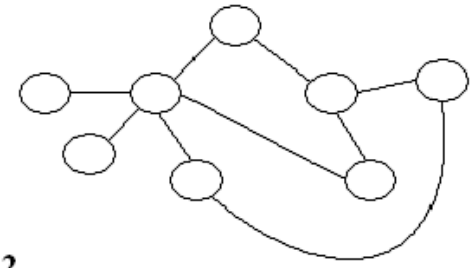
Як називають домен (чи декілька доменів), який однозначно ідентифікує кожний кортеж?

Ответ:

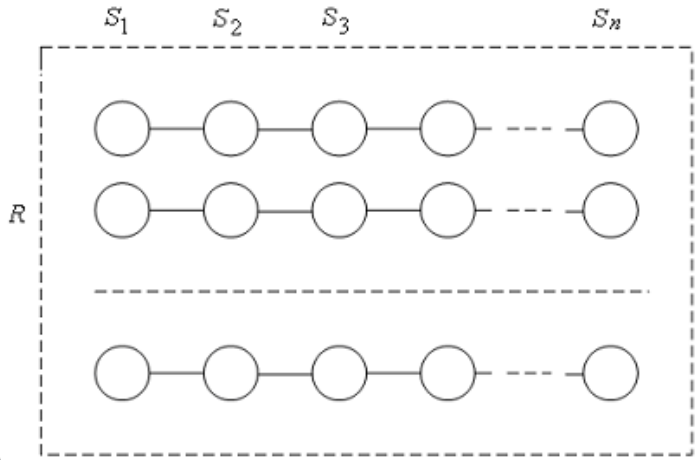
14
Баллов: --/1
Вкажіть номер графа, який відповідає сітковій моделі



1



2



3

Ответ:

15
Баллов: --/1

Таблиці можуть бути використані як реляційна база даних, якщо виконуються подані далі умови. Які вимоги лишні?

1. Всі рядки повинні мати одну і ту ж структуру, тобто одну і ту ж кількість атрибутів з однаковими іменами.

2. Імена стовпців повинні бути різні, а значення в кожному стовпці повинні бути однотипні (однорідні).

3. Значення атрибутів повинні бути елементарними, тобто не включати інші відношення.

3. Не може бути однакових рядків (записів) у таблиці.

5. Порядок розташування рядків у таблиці не має значення, оскільки впливає лише на швидкість доступу до рядка.

6. Повинна дотримуватися посилальна цілісність для зовнішніх ключів.

7. Таблиці повинні мати прості ключі.

8. Таблиці повинні бути між собою пов’язані.

9. Таблиці повинні мати однакову кількість записів.

Ответ:

16

Баллов: --/1

Існує _____ нормальних форм. (Уведіть відповідну цифру)

Ответ:

17

Баллов: --/1

Яку модель даних підтримує СКБД MS Access?

Ответ:

18

Баллов: --/1

Надано 2 інформаційні об’єкти: Товар і Виробник. Їх треба зв’язати за допомогою зовнішнього ключа. Назвіть об’єкт, у який треба додати поле зовнішнього ключа?

Товар

Артикул	Назва товару	Колір Кількість

Виробник

Назва фірми	E-mail Країна	Адреса	Телефон

Ответ:

19
Баллов: --/1

Потужність відношення - 6, а степінь відношення 5.

Скільки у відношенні може бути елементів?

Ответ:

20
Баллов: --/1

Інформаційний об’єкт Первинний перетворювач має такі атрибути:

Тип Довжина Ціна Клас точності

Виробник Серійний номер Місце установки

Який з них вибрати первинним ключем?

Выберите один ответ.

- a. Місце установки
- b. Серійний номер
- c. Ціна
- d. Довжина
- e. Виробник
- f. Клас точності
- g. Тип

Додаток 2.

ПРИКЛАДИ КОМПЛЕКСНИХ КОНТРОЛЬНИХ ЗАВДАНЬ ТА ЇХ ВИКОНАННЯ

Контрольне завдання № 1

Розглянути предметну область АВТОМАТИЗАЦІЯ з сутностями АВТОМАТИЧНА СИСТЕМА КЕРУВАННЯ (АСК) певного виробництва та РЕГУЛЯТОР.

1). Визначити для кожної сутності не менше ніж 3 атрибути.

2). Визначити тип зв’язку між сутностями.

3). Створити ER – діаграму у нотації Чена.

4). Записати схеми відношень.

5). Визначити первинні ключі та ключ зв’язку.

6). Визначити необхідні таблиці, поля і типи даних для створення БД у MS Access.

7). Побудувати схему даних.

8). Зобразити бланк запиту для визначення керувальних впливів і місць установки датчиків тих АСК, які містять дискретний ПІ – регулятор.

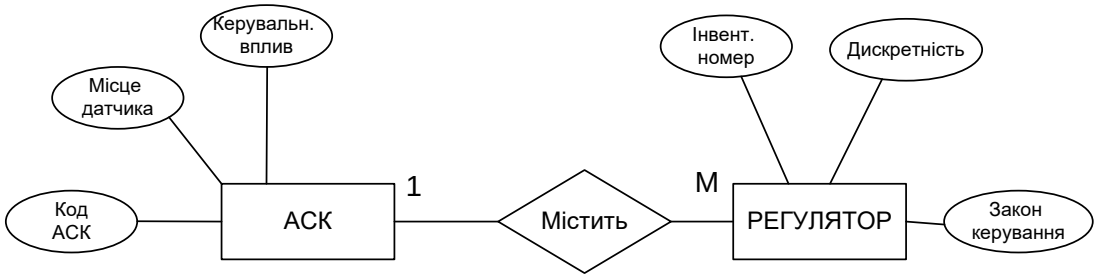
Еталон відповіді на контрольне завдання № 1

1. Атрибути відношень

АСК	Код_АСК	Місце_датчика	Керувальн. вплив
РЕГУЛЯТОР	Інвентар_номер	Закон_керування	Дискретність

2. АСК – РЕГУЛЯТОР – 1:N

3.



4. АСК (**Код АСК**, Місце_датчика, Керувальн_вплив)
РЕГУЛЯТОР (**Інвентар_номер**, Закон_керування, Дискретність)

5. Первинні ключі: АСК – Код АСК; РЕГУЛЯТОР – Інвентарний номер
Ключ зв'язку: Код АСК.

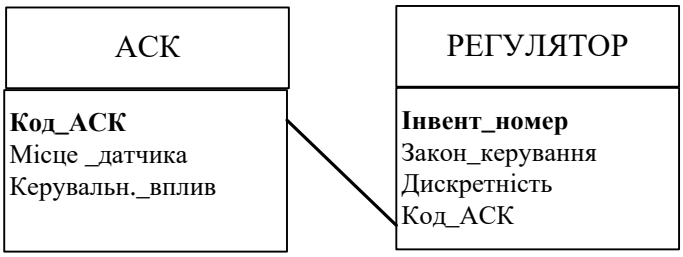
6. Таблиця АСК

Назва поля	Код_АСК	Місце_датчика	Керув_вплив
Тип поля	Числовий	Текстовий	Текстовий

Таблиця РЕГУЛЯТОР

Назва поля	Інвентар_номер	Закон_керування	Дискретність	Код_АСК
Тип поля	Числовий	Текстовий	Текстовий	Числовий

7.



8.

Таблиця	АСК	АСК	РЕГУЛЯТОР	РЕГУЛЯТОР
Поле	Місце датчика	Керувальн_вплив	Дискретність	Закон керування
Умова відбору			Дискретний	ПІ

Контрольне завдання №2

Розглянути предметну область ВИРОБНИЦТВО ТЕХНІЧНИХ ЗАСОБІВ АВТОМАТИЗАЦІЇ з сутностями ЗАВОД-ВИРОБНИК та ТЕХНІЧНИЙ ЗАСІБ АВТОМАТИЗАЦІЇ (ТЗА).

Умова! - Кожний тип пристрою може бути виготовлений тільки на одному заводі.

- Один завод може виготовляти різні типи пристроїв.
- Кожний ТЗА має власний заводський номер (код, шифр).

1). Визначити для кожної сутності не менше ніж 3 атрибути.

2). Визначити тип зв'язку між сутностями.

3). Створити ER – діаграму у нотації Чена.

4). Записати схеми відношень.

5). Визначити первинні ключі та ключ зв'язку.

6). Визначити необхідні таблиці, поля і типи даних для створення БД у MS Access.

7). Побудувати схему даних.

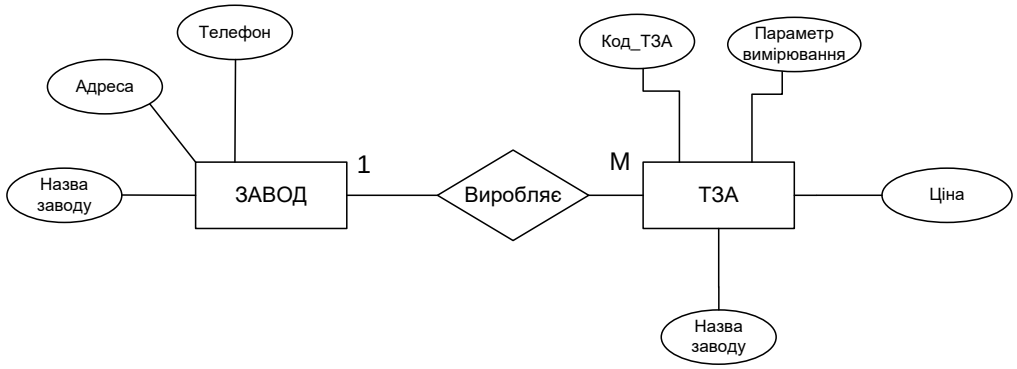
8). Зобразити бланк запиту для визначення адреси та телефону заводу, який виготовляє пристрої вимірювання рівня не дорожчі за 2500 грн.

Еталон відповіді на контрольне завдання № 2

1. Атрибути відношень

ЗАВОД-ВИРОБНИК	Назва заводу	Адреса	Телефон	
ТЗА	Код_ТЗА	Параметр вимірювання	Ціна	Назва заводу

2. ЗАВОД-ВИРОБНИК – ТЗА – 1:N
3.



4. ЗАВОД (**Назва заводу**, Адреса, Телефон);
ТЗА (**Код_ТЗА**, Параметр вимірювання, Ціна, Назва заводу)

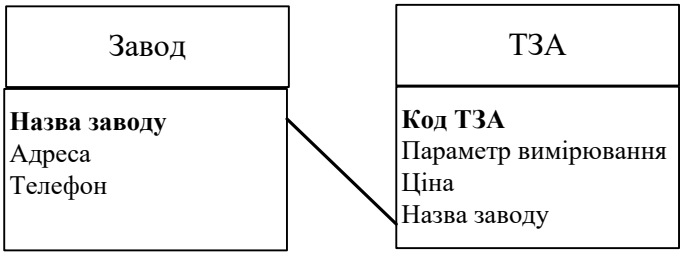
5. Первинні ключі: ЗАВОД– Назва заводу; ТЗА – Код_ТЗА
Ключ зв'язку: Назва заводу.
6. Таблиця ЗАВОД

Назва поля	Назва заводу	Адреса	Телефон	
Тип поля	Текстовий	Текстовий	Текстовий	

Таблиця ТЗА

Назва поля	Код_ТЗА	Параметр вимірювання	Ціна	Назва заводу
Тип поля	Числовий	Текстовий	Грошовий	Текстовий

7.



8.

Таблиця	ЗАВОД	ЗАВОД	ТЗА	ТЗА
Поле	Адреса	Телефон	Параметр вимірювання	Ціна
Умова відбору			Тиск	<1500

Контрольне завдання №3

Розглянути предметну область ДИПЛОМНЕ ПРОЄКТУВАННЯ з сутностями КЕРІВНИК ДИПЛОМНОГО ПРОЕКТУ та СТУДЕНТ.

Умова! Бухгалтерія інституту надає кожному викладачу окремий табельний номер.

1). Визначити для кожної сутності не менше ніж 3 атрибути.

2). Визначити тип зв'язку між сутностями.

3). Створити ER – діаграму у нотації Чена.

4). Записати схеми відношень.

5). Визначити первинні ключі та ключ зв'язку.

6). Визначити необхідні таблиці, поля і типи даних для створення БД у MS Access.

7). Побудувати схему даних.

8). Вказати умови створення запиту для того, щоб визначити, у яких студентів керівником проекту є Тимченко Т.А., його посаду, науковий ступінь і розклад консультацій. Вивести також теми дипломних проектів.

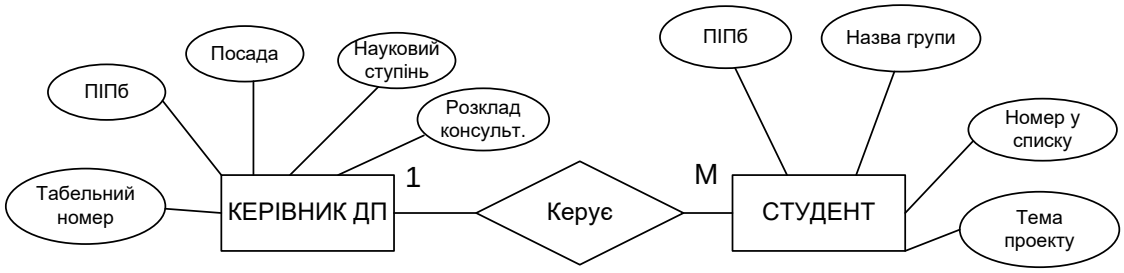
Еталон відповіді на контрольне завдання №3

1. Атрибути відношень

КЕРІВНИК ДИПЛОМНОГО ПРОЕКТУ	Табельн. номер	ПІПб	Науковий ступінь	Посада	Розклад
СТУДЕНТ	Назва групи	Номер за списком	ПІПб	Тема проекту	

2. КЕРІВНИК ДИПЛОМНОГО ПРОЕКТУ – СТУДЕНТ – 1:N

3.



4. КЕРІВНИК ДИПЛОМНОГО ПРОЕКТУ (Табельн.номер, ПІПб, Науковий ступінь, Посада, Розклад);
СТУДЕНТ (Назва групи, Номер за списком, ПІПб, Тема проекту)

5. Первинні ключі: КЕРІВНИК ДИПЛОМНОГО ПРОЕКТУ – Табельн. номер;
СТУДЕНТ – Назва групи, Номер за списком; Ключ зв'язку: Табельн. номер

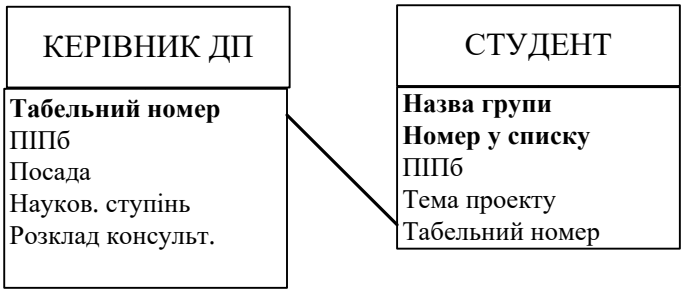
6. Таблиця КЕРІВНИК ДИПЛОМНОГО ПРОЕКТУ

Назва поля	Табельн. номер	ПІПб	Науковий ступінь	Посада	Розклад
Тип поля	Числовий	Текстовий	Текстовий	Текстовий	Мемо

Таблиця СТУДЕНТ

Назва поля	Назва групи	Номер за списком	ПІПб	Тема проекту	Табельн. номер
Тип поля	Текстовий	Числовий	Текстовий	Мемо	Числовий

7.



8.

Таблиця	СТУДЕНТ	СТУДЕНТ	КЕРІВНИК ДП	КЕРІВНИК ДП	КЕРІВНИК ДП	КЕРІВНИК ДП
Поле	ПІПб	Тема проекту	ПІПб	Посада	Наук. ступінь	Розклад
Умова відбору			Тимченко Т.А.			

Контрольне завдання №4

Розглянути предметну область АВТОМАТИЗАЦІЯ з сутностями АВТОМАТИЧНА СИСТЕМА КЕРУВАННЯ (АСК) та РЕГУЛЯТОР.

1). Визначити для кожної сутності не менше ніж 3 атрибути.

2). Визначити тип зв’язку між сутностями.

3). Створити ER – діаграму у нотації «Пташина лапка».

4). Записати схеми відношень.

5). Визначити первинні ключі та ключ зв’язку.

6). Визначити необхідні таблиці, поля і типи даних для створення БД у MS Access.

7). Побудувати схему даних.

8). Зобразити бланк запиту для визначення керувальних впливів і місць установки датчиків тих АСК, які містять аналоговий П – регулятор.

Еталон відповіді на контрольне завдання № 4

1. Атрибути відношень

АСК	Код_АСК	Місце_датчика	Керувальн. вплив
РЕГУЛЯТОР	Інвентар_номер	Закон_керування	Дискретність

2. АСК – РЕГУЛЯТОР – 1:N

3.

4. АСК (**Код**, Місце_датчика, Керув_вплив)
РЕГУЛЯТОР (**Інвентар_номер**, Закон_керування, Дискретність)

5. Первинні ключі: АСК – Код АСК; РЕГУЛЯТОР – Інвентарний номер
Ключ зв'язку: Код АСК.

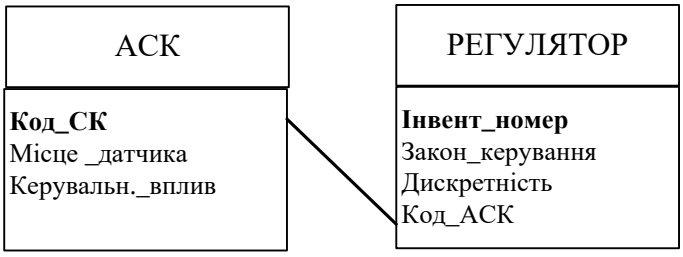
6. Таблиця АСК

Назва поля	Код_АСК	Місце_датчика	Керув_вплив
Тип поля	Числовий	Текстовий	Текстовий

Таблиця РЕГУЛЯТОР

Назва поля	Інвентар_номер	Закон_керування	Дискретність	Код_АСК
Тип поля	Числовий	Текстовий	Текстовий	Числовий

7.



8.

Таблиця	АСК	АСК	РЕГУЛЯТОР	РЕГУЛЯТОР
Поле	Місце датчика	Керувальн_вплив	Дискретність	Закон керування
Умова відбору			Аналоговий	П

Контрольне завдання №5

Розглянути предметну область РЕМОНТ ТЕХНІЧНИХ ЗАСОБІВ АВТОМАТИЗАЦІЇ з сутностями ПРИСТРІЙ та СЛЮСАР КВП і А.

Умова! - Кожний пристрій може бути відремонтований одним слюсарем.

- Один слюсар може ремонтувати декілька пристроїв.
- Бухгалтерія заводу надає кожному працівнику окремий табельний номер.

1). Визначити для кожної сутності не менше ніж 3 атрибути.

2). Визначити тип зв'язку між сутностями.

3). Створити ER – діаграму у нотації Чена.

4). Записати схеми відношень.

5). Визначити первинні ключі та ключ зв'язку.

6). Визначити необхідні таблиці, поля і типи даних для створення БД у MS Access.

7). Побудувати схему даних.

8). Вказати умови створення запиту для того, щоб визначити, які назви та інвентарні номери пристроїв відремонтував слюсар Петренко А.А. за період з 10.03.2012р на теперішній час, вказати його розряд.

Контрольне завдання №6

Розглянути предметну область АВТОМАТИЗАЦІЯ ВИРОБНИЦТВА з сутностями ПРИСТРІЙ та АПАРАТ.

1). Визначити для кожної сутності не менше ніж 3 атрибути.

2). Визначити тип зв'язку між сутностями.

3). Створити ER – діаграму у нотації Чена.

4). Записати схеми відношень.

5). Визначити первинні ключі та ключ зв'язку.

6). Визначити необхідні таблиці, поля і типи даних для створення БД у MS Access.

7). Побудувати схему даних.

8). Вказати умови створення запиту для того, щоб вивести назви та інвентарні номери показувальних та реєструвальних пристроїв, встановлених для контролю параметрів технологічного апарату з інвентарним номером 254 (вказати цех його розташування).

Контрольне завдання №7

Розглянути предметну область НАУКОВЕ СПІВРОБІТНИЦТВО з сутностями НАУКОВЕЦЬ та НАУКОВЕ ДОСЛІДЖЕННЯ .

Умова! - Кожний науковець може приймати участь тільки в одному науковому дослідженні.

- Бухгалтерія закладу надає кожному науковцю окремий табельний номер

1). Визначити для кожної сутності не менше ніж 3 атрибути.

2). Визначити тип зв'язку між сутностями.

3). Створити ER – діаграму у нотації Чена.

4). Записати схеми відношень.

5). Визначити первинні ключі та ключ зв'язку.

6). Визначити необхідні таблиці, поля і типи даних для створення БД у MS Access.

7). Побудувати схему даних.

8). Вказати умови створення запиту для того, щоб визначити ПІПб науковців, які приймають участь у дослідженні за темою №56 (вказати керівника дослідження, термін закінчення), мають вчений ступінь доктора наук та більше 120 публікацій.

ЗМІСТ

ПЕРЕДМОВА…….. 3
1. ОСНОВНІ ПОНЯТТЯ ПРО БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ…….. ..6
1.1. Поняття про інформацію, дані та базу даних……. ..6
1.2 Система керування базами даних. Автоматизовані інформаційні системи…… ..8
1.3. Структура та властивості СКБД…….. .13
1.4. Класифікація АІС…….. .15
1.5. Інформаційні системи. Основні типи, функції та компоненти……. .22
Запитання та завдання для самостійної роботи……. .28
2. МОДЕЛІ ДАНИХ……... .29
2.1. Поняття про модель даних…… .29
2.2. Класифікації моделей даних. Трирівнева схема подання даних……... .29
2.3. Інфологічна модель предметної області……. .38
2.4. Подання інфологічних моделей у вигляді *ER* – діаграм…… .45
2.5. Види даталогічних моделей……. .58
Запитання та завдання для самостійної роботи……. .64
3. ПРОЄКТУВАННЯ РЕЛЯЦІЙНИХ БАЗ ДАНИХ……... .67
3.1. Поняття реляційної бази даних. Структурні характеристики відношень..…….. .67
3.2. Типи даних……. .71
3.3. Ключі відношень. Встановлення зв’язків між відношеннями…….. .74
3.4. Цілісність та достовірність даних…… .81
3.5. Операції з відношеннями…….. .82
3.6. Нормалізація відношень……... .86

3.7. Створення схеми даних…… .95
3.8. Узагальнений алгоритм проєктування реляційної бази даних……. .97
Запитання та завдання для самостійної роботи…… 113
4. ПРИКЛАДИ ВИКОРИСТАННЯ БАЗ ДАНИХ У ЗАДАЧАХ АВТОМАТИЗАЦІЇ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ…… 115
4.1. Створення БД для визначення режимних параметрів екструзії полімерів при зміні сировини……... 115
4.1.1. Дослідження технологічної системи та розробка моделі даних… 115
4.1.2. Опис алгоритму визначення режимних параметрів…… 119
4.2. Створення інформаційно-дослідницької бази даних моделей та систем керування екструдером……. 126
4.2.1. Схема даних……. 128
4.2.2. Приклад бази даних…… 129
4.3. Створення бази даних для вибору технічних засобів автоматизації... 132
4.3.1. Визначення інформаційних об’єктів предметної області……... 133
4.3.2. Визначення первинних ключів та ключів зв’язку. Схема даних… 138
СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ….. 144
ДОДАТКИ…….. 145
Додаток 1. Блок тестів до трьох тем курсу…… 145
Додаток 2. Приклади комплексних контрольних завдань та їх виконання 154