

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

ТЕКСТИ (конспект) лекцій з дисципліни

“Організація баз даних”
для студентів спеціальності 123 “Комп’ютерна інженерія”
усіх форм навчання
частина II

2020

Тексти (конспект) лекцій з дисципліни “Організація баз даних”
для студентів спеціальності 123 “Комп’ютерна інженерія” усіх форм
навчання /Укл.: Т.О. Паромова,. - Запоріжжя: ЗНТУ, 2020. – 26 с.

Укладачі: Т.О.Паромова, ст.викладач.

Рецензент: С.Д. Точилін, доцент, к.ф-м.н.

Відповідальний за випуск: Паромова Т.О.

Затверджено
на засіданні кафедри
“Комп’ютерні системи
та мережі ”
Протокол № 1
від “21” серпня 2020.

Рекомендовано до видання
НМК факультету комп’ютерних
наук і технологій
Протокол № 1
від “28” серпня 2020.

ЗМІСТ

1.	Інфологічне моделювання баз даних	4
1.1.	Основні визначення	4
1.2.	Класифікація атрибутів.....	8
1.3.	Класифікація сутностей.....	9
1.4.	Типи зв'язків.....	10
1.5.	Кардинальність зв'язку.....	13
1.6.	Ступінь участі (модальність)	15
1.7.	Створення ER-моделі.....	16
1.8.	Приклад створення ER-моделі	17
1.9.	Проблеми побудови ER-моделі	20
1.10.	Перетворення ER-моделі у відношення	21
1.11.	Вибір вибору конкретної СУБД.....	23
	ЛІТЕРАТУРА	26

1. Інфологічне моделювання баз даних

1.1. Основні визначення

Інфологічна модель (інформаційно-логічна модель) орієнтована на людину і не залежна від типу СУБД. Інфологічна модель - модель предметної області, яка визначає сукупності інформаційних об'єктів, їх властивостей і відносин між об'єктами, динаміку змін предметної області, а також характер інформаційних потреб користувачів. Інфологічна модель предметної області може бути описана моделлю "сутність-зв'язок" (моделлю Чена), в основі якої лежить розподіл реального світу на окремі помітні сутності, які знаходяться в певних зв'язках один з одним, причому обидві категорії - сутність і зв'язок покладаються первинними, невизначеними поняттями. Модель представляється у вигляді схеми.

Концептуальна модель - інформаційна модель предметної області в термінах конкретної СУБД, що містить повний набір даних і зв'язків між ними. В архітектурі СУБД являє проміжний між зовнішнім і внутрішнім рівень.

Схема бази даних - опис даталогічних моделей в термінах СУБД (часто використовується як синонім моделі даних).

Інформації відповідає інфологічне уявлення, яке розглядає питання, пов'язані з збереженням інформації, незалежно від подання в пам'яті комп'ютера. Даталогічне уявлення розглядає питання представлення даних в пам'яті комп'ютера.

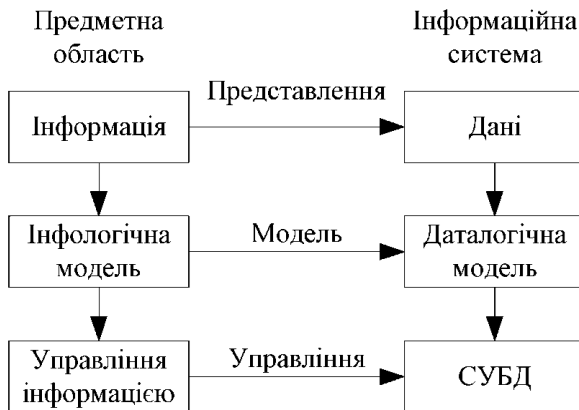


Рисунок 1.1 - Зв'язок між інформацією та даними

Вимоги до інфологічної моделі:

- 1 адекватність відображення предметної області;
- 2 відображення потреб всіх користувачів;
- 3 неможливість неоднозначності трактування моделі;
- 4 розширюваність - введення нових сутностей не повинно

приводить до кардинальних змін моделі;

- 5 легко сприйматися різними групами користувачів;
- 6 достатність інформації для подальшого проектування.

На практиці найчастіше використовується графічна інтерпретація моделі «СУТНІСТЬ-ЗВ'ЯЗОК» - ER -діаграма.

Основними конструктивними елементами інфологічних моделей є сутності, зв'язки між ними і їх властивості (атрибути).

СУТНІСТЬ - це клас однотипних об'єктів, інформація про яких повинна бути врахована в моделі, тобто СУТНІСТЬ - це будь-який помітний об'єкт, інформацію про який необхідно зберігати в базі даних. Сутностями можуть бути люди, місця, літаки, рейси, смак, колір і т. п. Прикладами сутностей можуть бути такі класи об'єктів як "Постачальник", "Співробітник", "Накладна".

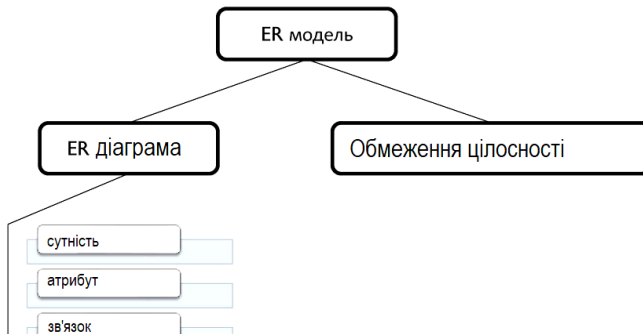


Рисунок 1.2 - Складові інфологічної моделі

Сутність дозволяє моделювати клас однотипних об'єктів. СУТНІСТЬ має ім'я, унікальне в межах системи, що моделюється. Ім'я виражається іменником в однині. Сутність має унікальне ім'я в межах модельованої бази даних. Оскільки сутність відповідає деякому класу однотипних об'єктів, то передбачається, що в системі існує багато екземплярів даної сутності. Об'єкт, якому відповідав би сутність, має набір атрибутів, що характеризують його властивості. При цьому на-

бір атрибутів повинен бути таким, щоб можна було розрізняти конкретні екземпляри сутності. Кожен тип сутностей ідентифікується ім'ям і списком властивостей. БД зазвичай містить багато різних типів сутностей. Різні розробником можуть виділяти різні сутності. Єдиною визначальною ознакою може бути те, що сутності - це іменники, наприклад, Студент, Викладач, Цех.

В якості сутності може бути обраний не тільки об'єкт, а й явище, процес, дію. Якщо автоматизується будь-який процес, то необхідно вказати властивості цього процесу, об'єкти беруть участь в ньому.

Необхідно розрізняти такі поняття, як тип сутності й екземпляр сутності. Тип сутності являє собою безліч об'єктів реального світу з однаковими властивостями.

Приклад сутності - це конкретний представник даної сутності. Приклади сутностей повинні бути помітні, тобто сутності повинні мати деякі властивості, унікальні для кожного екземпляра цієї сутності. ім'я.

Наприклад, типом сутності може бути МІСТО, а екземпляром - Запоріжжя, Київ і т.п.

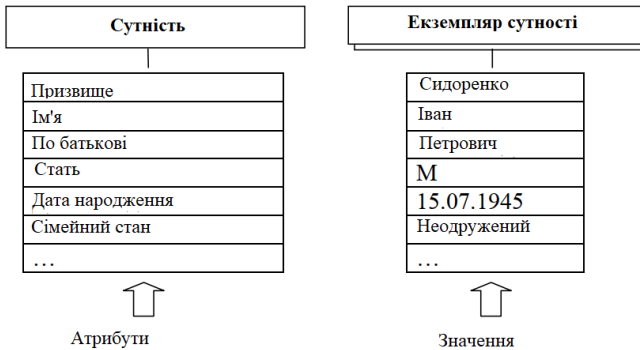


Рисунок 1.3 - Сутність та екземпляр сутності

На діаграмах ER-моделі СУТНІСТЬ подається у вигляді прямокутника, що містить ім'я сутності. При цьому ім'я сутності - це ім'я типу, а не деякого конкретного екземпляра цього типу. Для кращого розуміння ім'я сутності може супроводжуватися прикладами конкретних об'єктів цього типу. СУТНІСТЬ АЕРОПОРТ із об'єктами Бориспіль, Жуляни.

Властивості сутності описуються атрибутами. Атрибут - поименована характеристика сутності, властивість типу сутності або типу зв'язку. Наприклад, СУТНІСТЬ Студент може бути описаний атрибутами КодСтуд, Прізвище, Група, Спеціальність. Атрибути сутності містять значення, що описують кожен екземпляр СУТНІСТІ.

Найменування атрибута повинно бути унікальним для конкретного типу сутності, але може бути однаковим для різного типу сутностей (наприклад, КОЛІР може бути визначений для багатьох сутностей: СОБАКА, АВТОМОБІЛЬ, КВІТКА і т.д.).

Атрибути використовуються для визначення того, яка інформація повинна бути зібрана про СУТНІСТЬ. Прикладами атрибутів для сутності АВТОМОБІЛЬ є ТИП, МАРКА, НОМЕРНИЙ знак, колір і т.д. Найменування атрибута повинно бути виражено іменником в однині (можливо, з характеризують прикметниками).

Прикладами атрибутів сутності "Співробітник" можуть бути такі атрибути як "Табельний номер", "Прізвище", "Ім'я", "По батькові", "Посада", "Зарплата" і т.п.

Домен атрибута - безліч значень, які може приймати атрибут.

Ключ - мінімальний набір атрибутів, за значеннями яких можна однозначно знайти необхідний екземпляр сутності. Мінімальність означає, що виключення з набору ключових атрибутів будь-якого атрибута не дозволяє ідентифікувати СУТНІСТЬ за атрибутами, що залишилися.

Зв'язок - асоціація двох або більше сутностей. Одна СУТНІСТЬ може бути пов'язана з іншою сутністю або сама з собою.

Зв'язки - п - арні асоціації, що показують, яким чином сутності співвідносяться або взаємодіють між собою. Зв'язок може існувати між двома різними сутностями або між сутністю і їй же самій (рекурсивна зв'язок). Вона показує, як пов'язані екземпляри сутностей між собою.

Зв'язок зазвичай виражається дієсловом. Студент ВІВЧАЄ Дисципліну, Співробітники ПРАЦЮЮТЬ У Відділах.

Кожна зв'язок має два кінця і одне або два найменування. Найменування зазвичай виражається в невизначеному дієслівній формі: "мати", "належати" і т.п. Кожне з найменувань ставиться до свого кінця зв'язку. Іноді найменування не пишуть зважаючи на їх очевидності.

Зв'язок, який з'єднує дві сутності, також може мати атрибути. Зв'язок між сутностями може бути обов'язковим або необов'язковим.

Графічно зв'язок зображується у вигляді лінії, що з'єднує дві або більше сутностей.

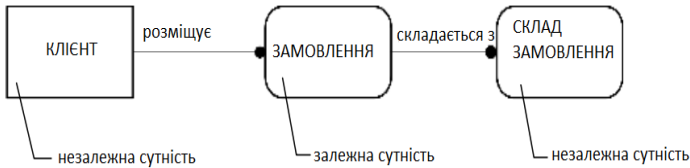


Рисунок 1.4 – Зображення залежних сутностей

1.2.Класифікація атрибутів

Атрибути СУТНОСТІ можна класифікувати за різними ознаками.

1. За кількістю складових

Простий атрибут складається з одного компонента з незалежним існуванням. Прості атрибути не можуть бути розділені на більш дрібні. Наприклад, атрибут Зарплата.

Складний атрибут складається з декількох компонентів, кожен з яких характеризується незалежним існуванням. Наприклад, атрибут Адреса може бути розбитий на атрибути Вулиця, Будинок, Квартира і т. п. Рішення про розбиття складеного атрибута на прості атрибути приймається замовником системи.

2. За кількістю значень, які може приймати атрибут

Однозначний атрибут може мати тільки одне значення, наприклад атрибут Пол, Дата народження.

Багатозначний атрибут може мати кілька значень для однієї сутності. Наприклад, атрибут Телефон може мати кілька значень - встановлено кілька телефонів в організації.

3. За можливістю зміни у часі

Крім того розрізняють атрибути:

Одиничні - атрибути, які можуть приймати єдине значення для даної сутності, наприклад, рік народження,

Статичні – атрибути, які не можуть змінювати своє значення з часом, наприклад, Рік народження;

Динамічні - атрибути, значення яких змінюється з часом, наприклад, Освіта

Умовні атрибути характеризують наявність даного атрибута у всіх сутностей даного класу.

Похідний атрибут представляє значення, похідне від значення пов'язаного з ним атрибута або деякого безлічі атрибутів, що належать деякому типу сутностей. Наприклад, Вік співробітника є похідною величиною від його Дати народження, атрибут вік обчислюється на підставі атрибута Дата народження. У деяких випадках атрибут є похідним від багатьох сутностей, наприклад атрибут Кількість працівників визначається в результаті підрахунку кількості сутностей Співробітник. Таким чином, похідний атрибут є обчислюваним атрибутом, як правило такі атрибути не зберігаються у базі даних, а обчислюються при необхідності за допомогою запитів.

1.3.Класифікація сутностей

Сутності можна класифікувати за різними ознаками.

Типи сутностей можна поділити на слабкі і сильні.

Слабкі (залежні) сутності залежать від якогось іншого типу сутностей (логічно чи фізично).

Сильні сутності не залежать від якогось іншого типу сутностей. Наприклад, сильна сутність Працівник, слабка сутність Діти. У Працівника є Діти. Слабкі сутності називають дочірніми, залежними або підлеглими. Сильні сутності - батьківськими сутностями - власниками або домінантними.

К.Дейт визначає три основні класи сутностей: стрижневий, асоціативні і характеристичні, а також підклас асоціативних сутностей - позначення.

Стрижнева сутність (стрижень) - це незалежна сутність. Наприклад сутність Студент, Викладач. Ці сутності існують незалежно від будь-яких інших сутностей. Їх атрибути будуть визначати інші сутності. Стрижневі сутності відображаються на ER-моделі прямокутниками.



Рисунок1.5 – Приклад зображення незалежної сутності

Асоціативна сутність (агрегатована) - це сутність, яка об'єднує деякі сутності і реалізує зв'язок між двома або більше сутностями або екземплярами сутності. Наприклад, сутність Іспит, її атрибути належать від сильних сутностей Викладач, Студент і Дисципліна.

Асоціативні сутності позначають іменниками, які утворюються від дієслів, та пояснюють природу зв'язку між сутностями.

Асоціації розглядаються як повноправні сутності:

- вони можуть брати участь в інших асоціаціях і позначеннях так само, як стрижневі сутності;
- можуть мати властивості, тобто мати не тільки набір ключових атрибутів, необхідних для вказівки зв'язків, а й будь-яке число інших атрибутів, які характеризують зв'язок.

Характеристична сутність (характеристика) уточнює властивості деякої іншої сутності. Необхідність в них виникає в зв'язку з тим, що сутності реального світу мають іноді багатозначні властивості. Наприклад, книга може мати кілька характеристик перевидання (доповнене, перероблене) і т. п. Всі екземпляри характеристики повинні бути обов'язково пов'язані з яким-небудь екземпляром сутності, яку вони характеризують. Існування характеристики повністю залежить від сутності, яка характеризується: зникає незалежна стрижнева сутність, то зникає і характеристична.

1.4. Типи зв'язків

Процес визначення сутностей з предметної області та їх опис є ключовим для створення будь-яких структур даних.

Сутності, визначені з предметної області, можуть бути пов'язаними між собою. Одна сутність може посилатися на іншу і навіть бути ознакою іншої сутності більш високого рівня (так, сутність «Товар» і «Продавець» можуть бути елементами або необхідними атрибутами сутності «Купівля товарів», сутність «Співробітник» - елементом або необхідним атрибутом сутності «Організація»).

Поняття зв'язок між сутностями відзначає наявність якої-небудь залежності, асоціації між сутностями - тобто наявність інформаційної або логічного зв'язку між об'єктами предметної області, що автоматизується.

Таким чином, зв'язок - осмислена асоціація між сутностями різних типів. Кожному типу зв'язку присвоюється ім'я, яке повинно описувати його функцію. Зв'язок – асоціація між сутностями, що

включає з кожного боку по одній сутності, що бере участь в зв'язку. Зв'язки вказують на окремі типи сутностей, що об'єднуються.

При побудові інфологічних моделей використовуються ER-діаграми (від англ. Entity-Relationship, тобто сутність-зв'язок). У них сутності зображуються позначеними прямокутниками, асоціації - позначеними ромбами або шестикутниками, атрибути - поміченими овалами, а зв'язки між ними - ненаправлені ребрами, над якими може проставлятися ступінь зв'язку (1 або буква, що заміняє слово "багато") і необхідне пояснення або у вигляді ромба з проставленням виду зв'язку. Тому що не існує стандартів на зображення ER-діаграм, то у літературі можна знайти багато варіантів зображення цих діаграм. Сприйняття зображення ER-діаграми у спосіб, який наведено на рис 6, ускладнено, тому частіше таку діаграму зображають способом, який наведено на рис 7. Такий вигляд має ER-діаграм наприклад у СУБД Access.



Рисунок 1.6 - Приклад зображення ER-діаграми зі винесенням атрибутів

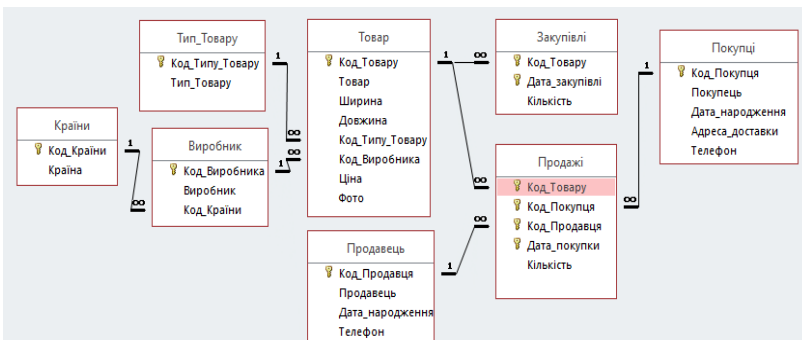


Рисунок 1.7 - Приклад зображення ER-діаграми без винесення атрибутів

Ступінь зв'язку (потужність)- кількість сутностей, які охоплені цією зв'язком.

Зв'язок називається бінарним, якщо в ньому беруть участь дві сутності (рис 8).

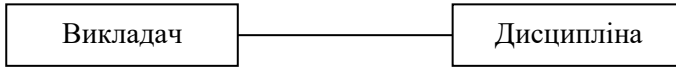


Рисунок 1.8- Приклад бінарної зв'язку між сутностями
Тернарний зв'язок – зв'язок між трьома сутностями (рис 9).



Рисунок 1.9 - Приклад тернарного зв'язку між сутностями

Рекурсивна зв'язок - зв'язок, в якій одні і ті ж сутності беруть участь кілька разів в одних і тих же зв'язках. В цьому випадку для вказівки призначення кожного зв'язку їм можуть присвоюватися рольові імена. Наприклад, на рис.3 приведено приклад зображення рекурсивного зв'язку, що зв'язує сутність ЛЮДИНА з нею ж самою. Кінець зв'язку з ім'ям СИН визначає той факт, що в одного батька може бути більше ніж один син. Кінець зв'язку з ім'ям БАТЬКО означає, що не в кожній людини можуть бути СИНИ.

Опис зображеної діаграми (рис.10) такий:

- кожна ЛЮДИНА є СИНОМ одного і тільки одного ЧОЛОВІКА;
- кожна ЛЮДИНА може бути БАТЬКОМ для одного або більше ЛЮДЕЙ.

На сутності накладаються деякі обмеження, які відображають вимоги реального світу. Наприклад, кожен об'єкт нерухомості має свого власника, на кожному підприємстві є хоча б один співробітник.



Рисунок 1.10 – Приклад рекурсивного зв'язку

1.5.Кардинальність зв'язку

Зв'язки між сутностями мають дві характеристики, які визначають характер їх взаємодії: кардинальність та модальність.

Показник кардинальності описує кількістю можливих зв'язків для кожної сутності-учасниці в обох напрямках. Найбільш поширеними є бінарні зв'язки з показниками кардинальності Один-До-Одного (1:1), Один-До -Багатьох (1:M), Багато-До-Багатьох (M:M). Між двома сутностями, наприклад, А і В, можливі чотири види зв'язків.

Перший тип - зв'язок **ОДИН-ДО-ОДНОГО** (1:1): в кожен момент часу кожному екземпляру сутності А відповідає 1 чи 0 екземпляр сутності В (рис. 11).

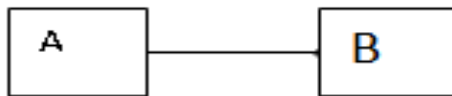


Рисунок 1.11- Зв'язок між сутностям А та В типу 1:1

Приклади: кожен викладач читає одну дисципліну і кожену дисципліну читає один викладач (рис. 12).

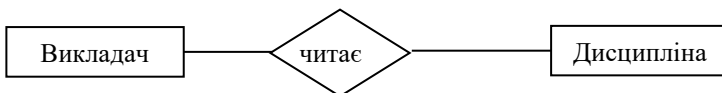


Рисунок 1.12 - Зв'язок між сутностям Викладач та Дисципліна 1:1

Студент може не "заробити" стипендію, по-лучити звичайну або одну з підвищених стипендій, але тільки одну стипендію.

Другий тип - зв'язок **ОДИН-ДО -БАГАТЬОХ** (1: M).

Зв'язок типу Один-До -Багатьох означає, що в кожен момент часу кожному екземпляру сутності А відповідає 0 чи N екземплярів

сутності В, та кожному екземпляру сутності В відповідає тільки один екземпляр сутності А. Це найбільш часто використовуваний тип зв'язку, якій дуже поширений у житті. Сутність з боку "один" називається батьківською, сутність з боку "багато" – дочірньою (рис 13).

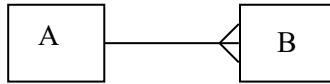


Рисунок 1.13 – Зв'язок між сутностями А та В типу 1:M

Приклад: кожному дисципліну читає кілька ВИКЛАДАЧІВ але кожен ВИКЛАДАЧ читає тільки одну ДИСЦИПЛІНУ (рис 14).

Квартира може пустувати, в ній може жити один чи кілька мешканців.

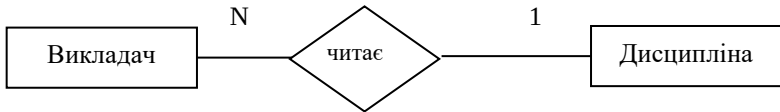


Рисунок 14 – Зв'язок типу 1:M між сутностями ДИСЦИПЛІНА та ВИКЛАДАЧ

Так як між двома сутностями можливі зв'язки в обох напрямках, то існує ще два типи зв'язку БАГАТО-ДО-ОДНОГО (M: 1) і БАГАТО-ДО-БАГАТЬОХ (M: N).

Зв'язок типу Багато-До-Багатьох означає, що кожен екземпляр сутності А може бути пов'язаний з декількома екземплярами сутності В, і кожен екземпляр сутності В може бути пов'язаний з декількома екземплярами сутності А (рис 14).

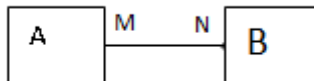


Рисунок 1.15 – Зв'язок між сутностями А та В типу M:N

Приклад: кожному дисципліну читає кілька ВИКЛАДАЧІВ та кожен ВИКЛАДАЧ читає кілька ДИСЦИПЛІН (рис 15).

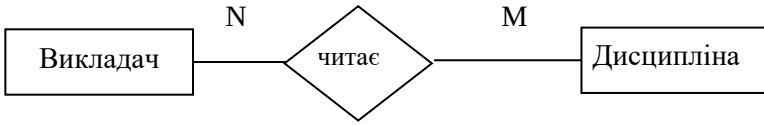


Рисунок 1.16 - Зв'язок M:N між сутностями Дисципліна та Викладач

Тип зв'язку Багато-До-Багатьох є тимчасовим зв'язком, допустимим на ранніх етапах розробки моделі предметної області. Надалі цей тип зв'язку повинен бути замінений двома зв'язками типу 1: 1 або 1: M шляхом створення проміжної асоціативної сутності.

1.6. Ступінь участі (модальність)

Ступінь участі визначає, чи залежить існування деякої сутності від участі в зв'язку деякої іншої сутності. Існує два варіанти участі сутності у зв'язку: повне і часткове.

Ступінь зв'язку є повною, якщо для існування деякої сутності потрібно існування іншої сутності, пов'язаної з нею певної зв'язком. В іншому випадку ступінь участі є частковою. Повну ступінь участі називають обов'язковою, а часткову необов'язковою.

Необов'язкова зв'язок виражається дієсловом МОЖЕ, а обов'язкова – ПОВИНЕН. Модальність " МОЖЕ " означає, що екземпляр однієї сутності може бути пов'язаний з одним або декількома екземплярами іншої сутності, а може бути і не пов'язаний ні з одним екземпляром., Тобто, не кожен екземпляр сутності повинен брати участь в зв'язку.

Модальність "ПОВИНЕН" означає, що екземпляр однієї сутності **обов'язково** пов'язаний не менше ніж з одним екземпляром іншої сутності, тобто кожен екземпляр даної сутності повинен брати участь в зв'язку.

Зв'язок може мати різну модальність з різних кінців. Наприклад, "СПІВРОБІТНИК може мати кілька ДІТЕЙ", "кожна Дитина повинна мати одного Батька", тобто з боку Співробітник зв'язок має необов'язковий характер, а з боку Дитина - обов'язковий.

При побудові БД в середовищі конкретної СУБД модальність зв'язку відбивається в тому, чи може атрибут приймати значення 0 або невизначене значення NULL.

Зліва направо: "кожен співробітник може мати кілька дітей".
Справа наліво: "Кожна дитина має при-належних рівно одному спів-робітнику". Зазвичай неов'язкова зв'язок з боку М.

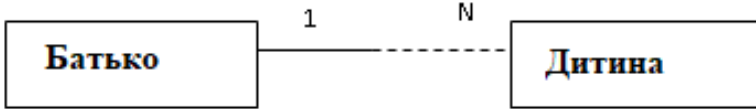


Рисунок 1.17 – Зв'язок між сутностями Батька та Дитина

Позначення обов'язкового і неов'язкового зв'язку може бути різним:

Обов'язкова	Необов'язкова
Суцільна лінія	Пунктирна лінія
Перпендикуляр в кінці зв'язку	Порожнє коло в кінці зв'язку

1.7. Створення ER-моделі

Концептуальне проектування - створення моделі предметної області, що не залежить від будь-яких фізичних умов реалізації.

Кожна концептуальна модель включає наступне:

- типи сутностей;
- типи зв'язків;
- атрибути;
- домени атрибутів;
- потенційні ключі;
- первинні ключі.

Проектування бази даних з використанням моделі "Сутність-Зв'язок" виконуються за наступними етапами.

1. Визначення типів сутностей

На підставі аналізу предметної області складається список сутностей, який може оновлюватись в процесі проектування. Як правило, сутності присутні в специфікації користувача у вигляді іменників. Слід проаналізувати, які сутності будуть стрижневими, а які асоціативними.

2. Визначення первинних ключів сутностей

Визначення атрибута, який однозначно визначить конкретний екземпляр сутності. Як правило, для простих сутностей цей атрибут являє собою числову послідовність з кроком 1. Для асоціативних сут-

ностей, що реалізують зв'язок між декількома незалежними сутностями, ключ є сукупністю зовнішніх ключів.

3.Визначення типів зв'язку

Встановлюється наявність зв'язків між сутностями, при цьому враховуються тільки ті зв'язки, які задовольняють вимогам замовника. У більшості випадків зв'язки є парними, тобто пов'язують дві сутності. Зв'язки присутні в специфікаціях як дієслів. Для гарантії їх відповідності зв'язкам реального світу їх необхідно ретельно їх вивчити, при виявленні неоднозначності уточнити вихідні дані

4.Визначення показника кардинальності та степеню участі

Необхідно визначити, до якого типу зв'язку відносяться встановлені зв'язки (1: 1, 1: М, М: М), та чи є вони обов'язковими. При наявності зв'язку типу М: М необхідно розщепити цей зв'язок, для чого ввести асоціативну сутність, яка реалізуватиме цей зв'язок.

5.Визначення атрибутів і зв'язування їх з типами сутностей і зв'язків

Для кожної сутності виділяються всі властивості, які згадані в опису предметної області. Перелік атрибутів кожної сутності може уточнятися у процесі проектування. Ці властивості відображаються в атрибутах, які описують сутність. Серед сутностей виділених на 1 етапі проектування сутності характеристики вводяться до складу стрижневих сутностей -асоціативних сутностей.

Асоціативні сутності реалізують зв'язок між сутностями, тому особливо важливо правильно визначити їх ключі. Як правило, ключі таких сутностей є складними, що складаються з декількох атрибутів

6.Рішення про виділення класів і підкласів

Для зменшення надмірності інформації часто необхідно виділити сутності в класи і підкласи, тому що підкласи дозволяють відобразити специфічні властивості окремих екземплярів сутностей. Наприклад, сутність Співробітник може бути поділена на підклас Викладачі, Студенти, Адміністрація.

1.8.Приклад створення ER-моделі

Необхідно спроектувати інформаційну систему оптової фірми. Система , що проектується, повинна виконувати наступні дії:

- 1 зберігати інформацію про покупців;
- 2 друкувати накладні на відпущені товари;

З стежити за наявністю товарів на складі.

З опису предметної області можна визначити три сутності Товар, Покупець і Склад, які описуються іменниками.

При купівлі товару покупцеві видається на-Накладна, в якій вказується найменування товару, його кількість і вартість.

Зв'язок між ТОВАРОМ і ПОКУПЦЕМ М: М при цьому обов'язкова з обох сторін, тому що не всі ТОВАРИ купує ПОКУПЕЦЬ і не всі ТОВАРИ можуть бути продані конкретному ПОКУПЦЕВІ.

Тому що зв'язок типу М:М не можливо реалізувати у реляційній базі даних, то необхідно розщепити цей зв'язок за допомогою додаткової сутності. При купівлі обов'язково оформляється НАКЛАДНА, тому сутність НАКЛАДНА розміщується між сутностями ПОКУПЕЦЬ і ТОВАР.



Рисунок 1.18 – Зв'язок між сутностями ПОКУПЕЦЬ та ТОВАР

При цьому зв'язок ПОКУПЕЦЬ_НАКЛАДНА 1: М $\Rightarrow$ кожен покупець може отримати кілька НАКЛАДНИХ, але кожна НАКЛАДНА виписується конкретному ПОКУПЦЕВІ. Обов'язковість зв'язку: ПОКУПЕЦЬ обов'язково отримує НАКЛАДНУ $\Rightarrow$ з боку ПОКУПЕЦЬ зв'язок обов'язковий, кожен ПОКУПЕЦЬ може отримати кілька НАКЛАДНИХ $\Rightarrow$ з боку накладної зв'язок необов'язковий.

У НАКЛАДНІЙ зазначаються ТОВАРИ, що зберігаються на СКЛАДІ $\Rightarrow$ зв'язок НАКЛАДНА – СКЛАД, зв'язок типу 1: М. При цьому з будь-якого СКЛАДУ може бути виписано багато НАКЛАДНИХ. $\Rightarrow$ зв'язок необов'язковий. Кожен ТОВАР у Накладній знаходиться на конкретному СКЛАДІ: зв'язок обов'язковий з боку СКЛАД і необов'язкова з боку НАКЛАДНА.

На СКЛАДІ зберігаються ТОВАРИ. СКЛАДІВ може бути багато, ТОВАРІВ теж багато. Зв'язок СКЛАД $\Rightarrow$ ТОВАР М: М, при чому зв'язок необов'язковий з обох сторін.

Таким чином, ER -діаграма має вигляд (рис 18). Між сутностями ТОВАР і НАКЛАДНА, ТОВАР і СКЛАД зв'язки типу М: М, це тимчасовий зв'язок, його потрібно розщепити на два зв'язку 1: М. Для чого ввести асоціативну сутність. Атрибутами цієї сутності будуть ключові атрибути сутностей, що зв'язуються, а також і власні атрибути.

Для зв'язку Товар-Накладна вводиться додаткова сутність Список товарів. Зв'язок між сутностями Накладна - Список товарів характеризують наступними чином: кожна Накладна обов'язково має кілька записів зі Списку-товарів але у НАКЛАДНУ включається рівно один СПИСОК_ТОВАРІВ , тому цей зв'язок обов'язковий з боку сутності Накладна, та необов'язковий з боку сутності СПИСОК_ТОВАРІВ.

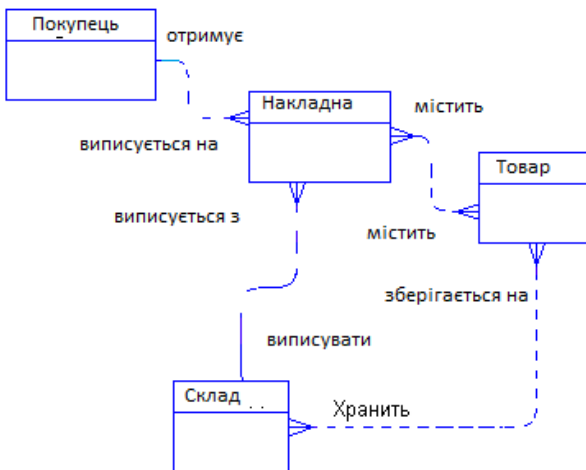


Рисунок 1.19 – Проміжна ER- діаграма предметної області

Також, ТОВАР може зберігатися на будь-якому СКЛАДІ і може бути проданий з будь-якого СКЛАДУ. Тому для розщеплення цього зв'язку вводиться сутність ТОВАР_на_Складі. Зв'язок між сутностями СКЛАД- ТОВАР_на_Складі типу 1:М: кожен ТОВАР зазнача-

ється у СПИСКУ_ТОВАРІВ один раз, але у СПИСКУ_ТОВАРІВ може зазначатися багато ТОВАРІВ.

Таким чином підсумкова ER - діаграма має вигляд, що поданий на рис 19.



Рисунок 1.20 – Результуюча ER-діаграма бази даних

1.9. Проблеми побудови ER-моделі

При недостатньому розумінні сутності опису предметної області та у наслідок встановлених надлишкових зв'язків може бути створена модель, яка не буде в повній мірі відображати зв'язки між реальними об'єктами. Визначають дефекти з'єднання, які виникають при неправильній інтерпретації змісту деяких зв'язків: дефекти розгалуження і дефекти розриву.

Дефекти розгалуження мають місце, коли модель вірно відображає зв'язку між сутностями, але шлях між окремими сутностями

певний неоднозначно. Цей дефект виникає в тому випадку, коли два або більше зв'язків типу 1: М виходять з однієї сутності.

Приклад: на факультеті займається багато студентів, до складу факультету входить багато груп (рис. 20). Ці зв'язки вірно відображають зміст предметної області, але при спробі з'ясувати, до якої групи займаються конкретні студенти, виникають проблеми. Із сутності Факультет виходять дві зв'язку 1: М.

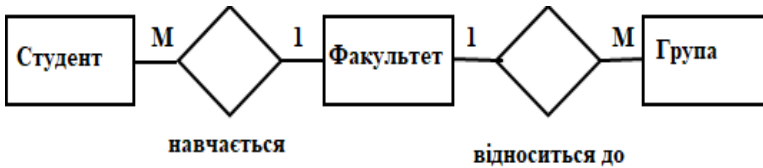


Рисунок 1.21 – Дефект розгалуження

Усунути цю проблему можна шляхом перебудови моделі для подання вірною взаємодії цих сутностей на підставі ретельного аналізу опису предметної області (рис.18)

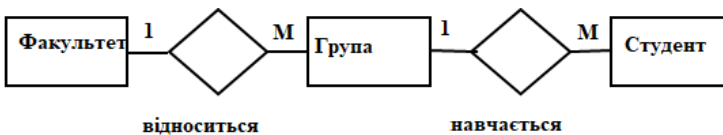


Рисунок 1.22 – Усунення дефекту розгалуження

1.10.Перетворення ER-моделі у відношення(датологічна модель)

Інфологічна модель дозволяє зрозуміти суть розроблюваної бази даних, але вона не підходить для безпосередньої реалізації структури БД з урахуванням вимог конкретної СУБД, яка обирається для реалізації системи. Для ER-моделі існує алгоритм однозначного перетворення в датологічну модель даних

Даталогічна модель - модель даних, орієнтована на обраний тип СУБД: внутрішня, концептуальна, зовнішня.

Алгоритм переходу від ER-моделі до датологічної моделі даних зазвичай зводиться до таких кроків:

1. Кожній сутності ставиться у відповідність відношення реляційної бази даних. Імена відносин можуть бути обмежені вимогами конкретної СУБД, вони обмежені по довжині і не повинні містити пробілів і деяких спеціальних символів.

2. Кожен атрибут сутності стає атрибутом відповідного відношення, на їх імена також можуть накладатися деякі обмеження (наприклад, багато СУБД не підтримують кирилицю). Для кожного атрибуту задається конкретний допустимий в СУБД тип даних і обов'язковість або необов'язковість даного атрибуту. На рис. 22, спрощено, показаний процес перетворення, з урахуванням обраної СУБД.

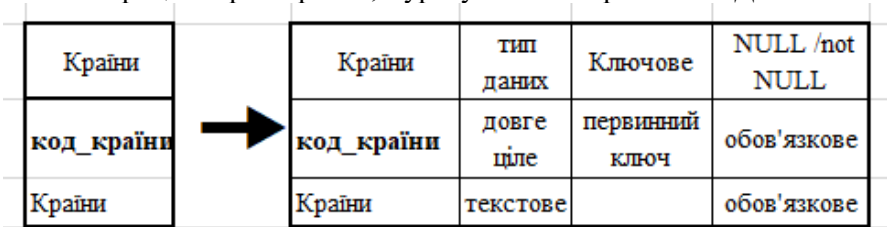


Рисунок 1.23 – Перетворення сутності ER-моделі у відношення

3. Первинний ключ сутності стає первинним ключем відповідного відношення (на рис. 22 первинний ключ виділено жирними літерами). Атрибути, що входять в первинний ключ відносини, автоматично отримують властивість обов'язковості і унікальності.

4. У кожне відношення, відповідне підпорядкованої сутності, додається набір атрибутів первинного ключа головної суті. Відносно, відповідному підпорядкованої сутності, цей набір атрибутів стає зовнішнім ключем.

Для необов'язкових типів зв'язку на фізичному рівні у атрибутів, відповідних зовнішньому ключу, встановлюється властивість допустимості невизначених значень. При обов'язковому типі зв'язку атрибути отримують властивість відсутності невизначених значень.

Сесія		Сесія	тип даних	Ключове	NULL /not NULL
код студента		код студента	довге ціле	первинний ключ	обов'язкове
код дисципліни		код дисципліни	довге ціле	первинний ключ	обов'язкове
дата		дата	дата		необов'язкове
оцінка		оцінка	числовий		необов'язкове

Рисунок 1.24 – Перетворення сутності ER-моделі у відношення з зовнішніми ключами

5. Після створення схеми БД і перевірки її коректності необхідно перевірити дотримання бізнес-правил, які визначаються замовником розробки. Бізнес-правила (БП) задають обмеження на значення даних в БД. Вони також визначають механізми, згідно з якими при зміні одних даних змінюються і пов'язані з ними дані в тій же або інших таблицях БД. Таким чином, бізнес-правила визначають умови підтримки БД в цілісному стані. На першому етапі проектування, тобто при описі предметної області визначаються і обмежувальні умови, правда, на даному етапі вони реалізуються вже в рамках реляційної моделі даних.

1.11. Вибір вибору конкретної СУБД

Вибір СУБД являє собою складну багатопараметричну задачу та є одним з важливих етапів при розробці БД. Обраний програмний продукт повинен задовольняти як поточним, так і майбутнім потребам підприємства, при цьому слід враховувати фінансові витрати на придбання необхідного обладнання, самої системи, розробку необхідного програмного забезпечення на її основі, а також навчання персоналу. Крім того, необхідно переконатися, що нова СУБД здатна принести підприємству реальні вигоди.

Можна виділити ряд критеріїв, за якими слід проводити вибір СУБД:

- 1 модель даних;
- 2 додаткові можливості;
- 3 особливості архітектури і функціональні можливості;
- 4 особливості розробки додатків;
- 5 продуктивність;

- 6 надійність;
- 7 вимоги до робочого середовища;
- 8 змішані критерії.

Модель даних Існує безліч моделей даних. Вибір моделі СУБД визначається тим, яка модель найкраще підходить для вирішення завдання.

Додаткові можливості. Чи існує підтримка тригерів та збережених процедур. Процедура - програма, яка зберігається на сервері і може викликатися клієнтом. Оскільки збережені процедури виконуються безпосередньо на сервері БД, забезпечується більш висока швидкість, ніж при виконанні тих же операцій засобами клієнта БД. Тригер - програма БД, що викликається всякий раз при модифікації таблиці. Тригери забезпечують перевірку будь-яких змін на коректність, перш ніж ці зміни будуть прийняті. Тригери дозволяють вести журналізацію всіх змін в БД.

Мобільність - це незалежність системи від середовища, в якій вона працює. Під середовищем розуміється апаратно-програмна платформа.

Масштабованість. При виборі СУБД необхідно враховувати, чи зможе ця система відповідати зростанню інформаційної системи, причому зростання може проявлятися в збільшенні числа користувачів, обсягу збережених даних і обсязі оброблюваної інформації.

Розподіленість. Прагнення до централізованого зберігання і обробки даних змушують використовувати розподілені БД. Різні системи мають різні можливості управління розподіленими БД.

Мережеві можливості. Багато системи дозволяють використовувати безліч мережевих протоколів і служб для роботи і адміністрування.

Особливості розробки додатків. Використовувані мови програмування та засоби розробки.

Продуктивність. Рейтинг TPC (Transactions per Cent) є одним з найпопулярніших і об'єктивних для аналізу продуктивності систем. Показник TPC - це відношення кількості запитів оброблених за якийсь проміжок часу до вартості всієї системи.

Можливості оптимізації запитів. Деякі СУБД здатні оптимізувати виконання складних запитів за рахунок генерації плану його виконання.

Надійність.

Можливості відновлення після збоїв.

Функції резервного копіювання. В результаті серйозних збоїв може бути частково пошкоджений або виведений з ладу носій даних, в цьому випадку відновлення можливе тільки з резервних копій.

Відкат змін. При виконанні транзакції застосовується просте правило - або транзакція виконується повністю, або не виконується взагалі. Це означає, що в разі збоїв, всі результати незакінчення транзакції вона повинні бути анульовані.

Підтримувані апаратно-програмні платформи.

Вимоги до обладнання. *Максимальний розмір пам'яті* Змішані критерії.

Якість і повнота документації. На жаль, не всі системи мають повну і детальну документацію.

Модель формування вартості. Вартість СУБД може істотно змінюватися в залежності від того, скільки користувачів буде з ним працювати.

Поширеність СУБД.

Чіткий і глибокий порівняльний аналіз на підставі перерахованих вище критеріїв допоможе раціонально вибрати відповідну СУБД для конкретного проекту. Перелік критеріїв допоможе усвідомити масштабність завдання і виконати її адекватну постановку. Слід зазначити, що за існуючою практикою рішення про використання тієї чи іншої СУБД приймає одна людина - зазвичай, керівник підприємства, а він може спиратися аж ніяк не на технічні критерії. Тут свою роль можуть зіграти такі фактори як рекламна розкрутка компанії-виробника СУБД, використання конкретних систем на інших підприємствах, вартість.

ЛІТЕРАТУРА

- 5.1. http://www.lessons-tva.info/edu/e-inf2/m2t4_2.html
- 5.2. С Робинсон., Microsoft Access., Учебный курс. Санкт-Петербург -Москва-Харьков-Минск: Питер, 2014. - 512с.
- 5.3. Т.Коннолли, К Бегг, А. Страчан. Базы данных, проектирование, реализация и сопровождение. Теория и практика. - Москва-Санкт-Петербург-Киев: 2014. - 1112с.
- 5.4. <http://info-comp.ru/obucheniest/258-access-2003-query.html>
- 5.5. Електронні уроки по Microsoft Access www.youtube.com
- 5.6. <http://mei06.narod.ru/sem5/database/shpora/q6.htm>
- 5.7. <https://studizba.com/lectures/10-informatika-i-programmirovanie/362-proektirovanie-baz-dannyh-na-sql/4880-10-kriterii-vybora-subd.html>
- 5.8. http://www.mstu.edu.ru/study/materials/zelenkov/ch_2_1.html - проектирование БД
- 5.9. http://www.bseu.by/it/tohod/lekcii4_6.htm проектирование
- 5.10. http://www.mstu.edu.ru/study/materials/zelenkov/ch_2_1.html
- 5.11. <https://studfile.net/preview/2592104/page:13/>
- 5.12. <https://studizba.com/lectures/10-informatika-i-programmirovanie/362-proektirovanie-baz-dannyh-na-sql/4881-11-perehod-k-relyacionnoy-modeli-dannyh.html>
- 5.13. <https://studizba.com/lectures/10-informatika-i-programmirovanie/362-proektirovanie-baz-dannyh-na-sql/4880-10-kriterii-vybora-subd.html>
- 5.14. <https://studfile.net/preview/4646713/page:7/>