

**Моделювання
потоків даних:
нотації та
принципи
моделювання**

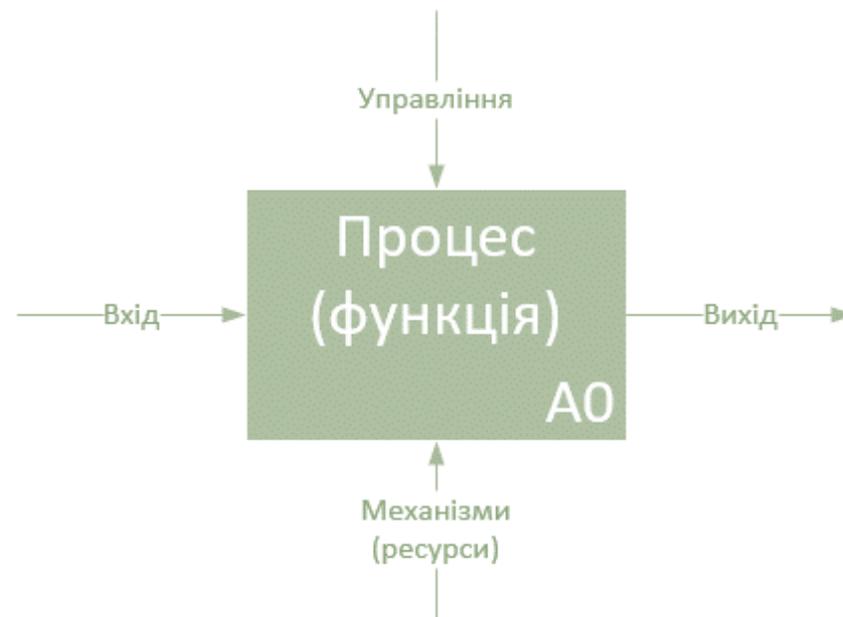
Структурно-функціональний підхід



визначення функціоналу майбутньої системи



побудови функціональної моделі, з врахуванням вхідних-вихідних потоків та опису коли і яким чином функції виконуються й управляються



Який наступний крок?

Що таке модель DFD?

- ✓ наочно відображає перебіг інформації в межах процесу чи системи
- ✓ для зображення вхідних та вихідних даних, точок зберігання інформації та шляхів її пересування між джерелами та пунктами доставки у таких діаграмах застосовуються стандартні фігури
- ✓ застосовуються для аналізу існуючих та моделювання нових систем

Використання і особливості DFD діаграм

Створені моделі потоків даних організації можуть бути використані при вирішенні таких завдань, як:

-  визначення існуючих сховищ даних (текстові документи, файли, Система управління базою даних - СУБД);
-  підготовка до створення моделі структури даних організації, так звана ERD-модель (IDEF1X);
-  визначення та аналіз даних, необхідних для виконання кожної функції процесу;
-  виділення основних і допоміжних бізнес-процесів організації

Основні елементи

DFD – Data Flow Diagram – діаграма потоків даних

- ✓ Потоки даних
- ✓ Сховища даних (накопичувачі)
- ✓ Зовнішні сутності
- ✓ Процеси

Потоки даних (DFD)

- ✓ **Абстрації**, які використовуються для моделювання передачі інформації з однієї частини системи в іншу
- ✓ Зображаються у вигляді **іменованої стрілки**, орієнтація яких вказує напрям руху інформації

Процес (DFD)



Продуктування вихідних потоків з вхідних у відповідності з дією, яка задається **іменем процесу**



Ім'я процесу повинно відповідати на запитання “Що робити?” (наприклад, отримати документи, розрахувати вартість, перевірити наявність у БД тощо)



Унікальний **номер** (можна разом з номером діаграми)

Сховище даних (DFD)

-  На певних ділянках діаграми визначаються **дані**, які зберігаються у пам'яті між процесами
-  **Ім'я сховища** повинно визначати його зміст

Зовнішня сутність



Матеріальний об'єкт **зовні** контексту системи, який є **джерелом** або **споживачем інформації**. Наприклад, склад товарів, клієнт



Сутності **не приймають участь** у процесах всередині системи

Додаткові елементи DFD



Словники даних – каталоги елементів даних, які використовуються в DFD



Мініспецифікації опрацювання – DFD –процеси нижнього рівня (алгоритми описання задач, які виконуються процесами)

Нотація діаграми

Гейн та Сарнсон

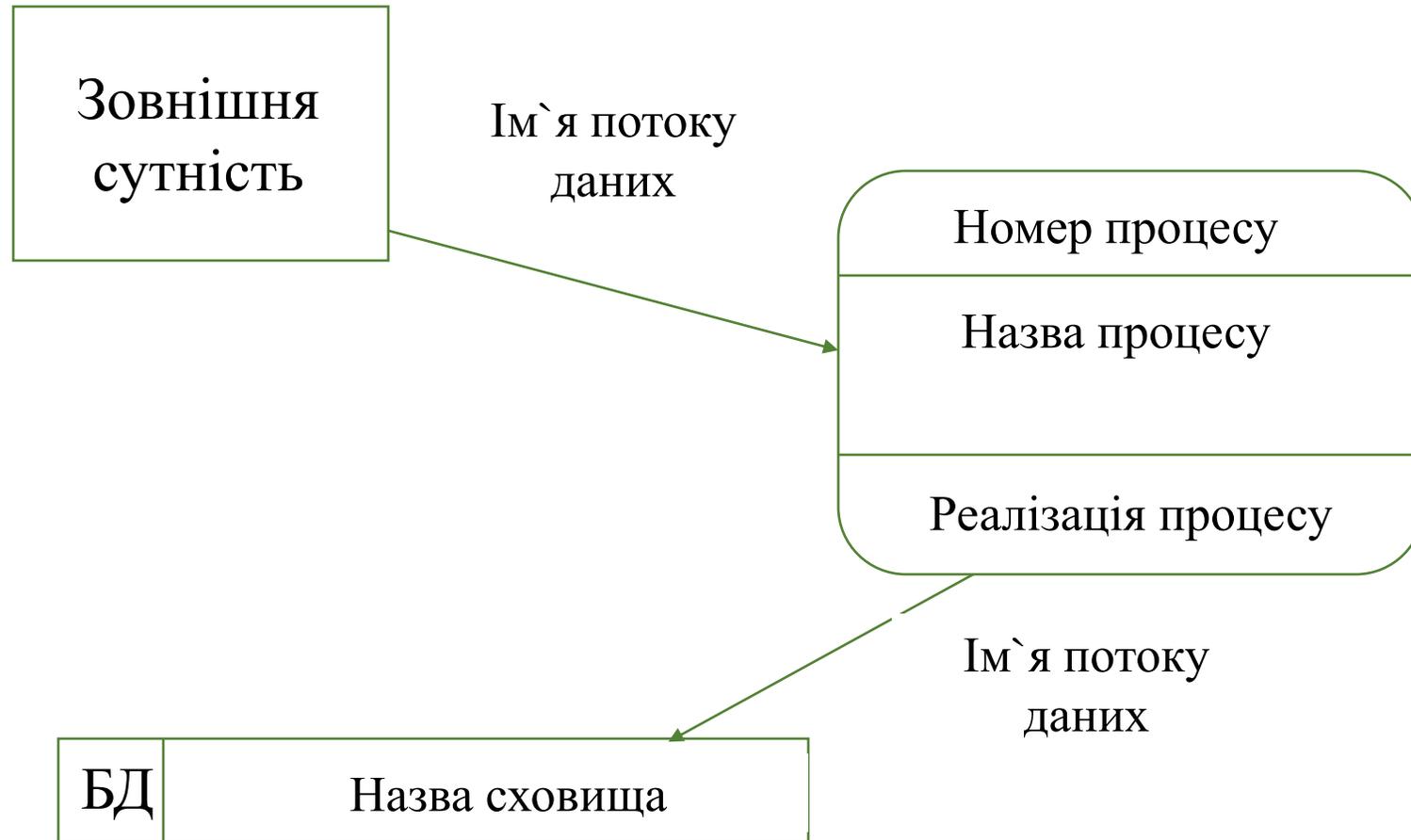


**Йордана-Коуда
Йордана де Марко**

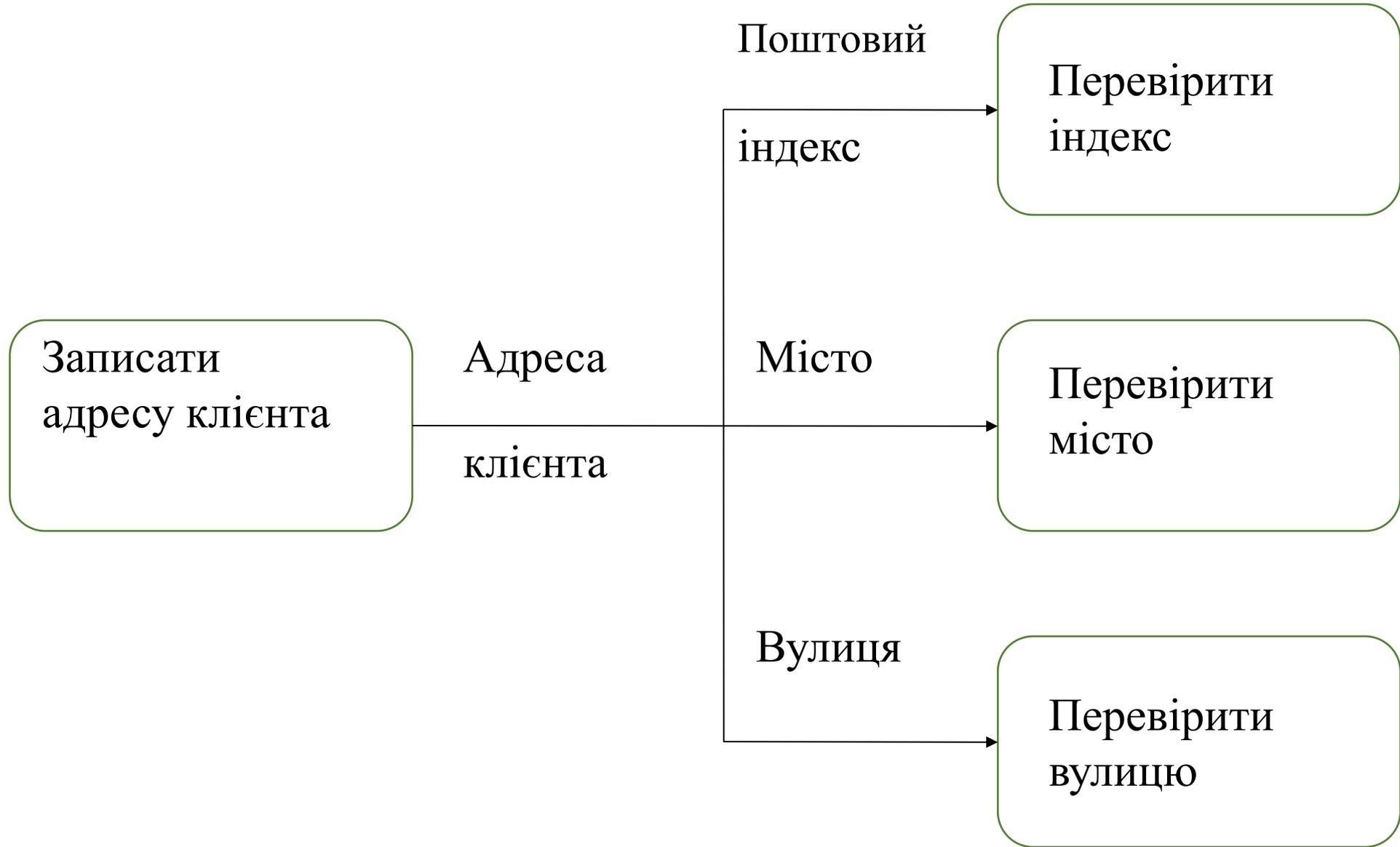


Модель потоків даних – діаграми DFD (Data Flow Diagram)

Нотація Гейна-Сарнсона



Компоненти DFD-діаграм



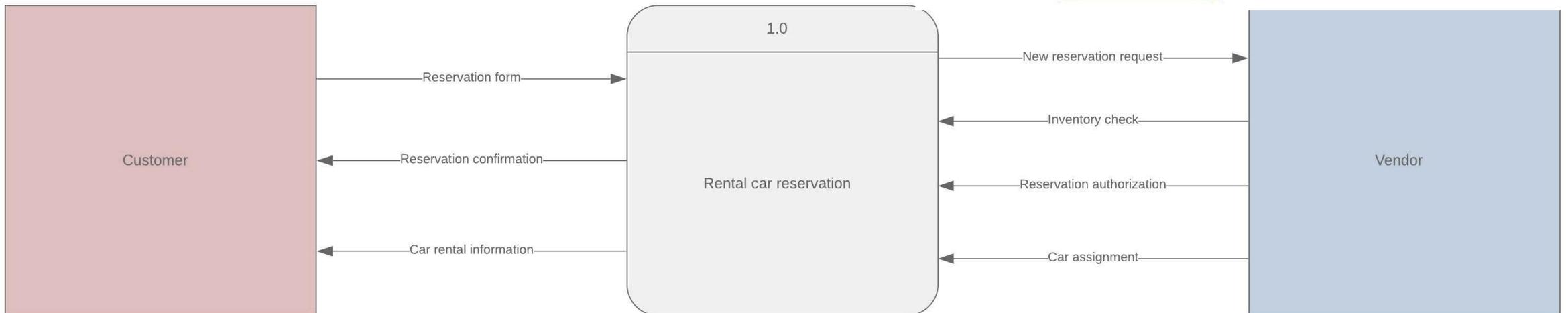
Правила та поради щодо побудови діаграм DFD

- ✓ Кожен процес має супроводжуватися як мінімум одним вхідним та одним вихідним потоком
- ✓ У кожне сховище повинен впадати щонайменше один потік даних і щонайменше один — витікати
- ✓ Дані, що зберігаються у системі, повинні проходити через процес
- ✓ Кожен процес діаграми DFD повинен вести або до іншого процесу, або до сховища даних

Рівні та шари DFD-моделей



Контекстний рівень



Особливості контекстної DFD-моделі

Доцільність побудувати кількох контекстних діаграм з ієрархією

- ✓ наявність великої кількості зовнішніх сутностей (десять і більше)
- ✓ розподілена природа системи
- ✓ багатofункціональність системи з уже сформованим або виявленим угрупованням функцій в окремі підсистеми



контекстна діаграма верхнього рівня містить не єдиний головний процес, а набір підсистем, з'єднаних потоками даних



контекстні діаграми наступного рівня деталізують контекст і структуру підсистем

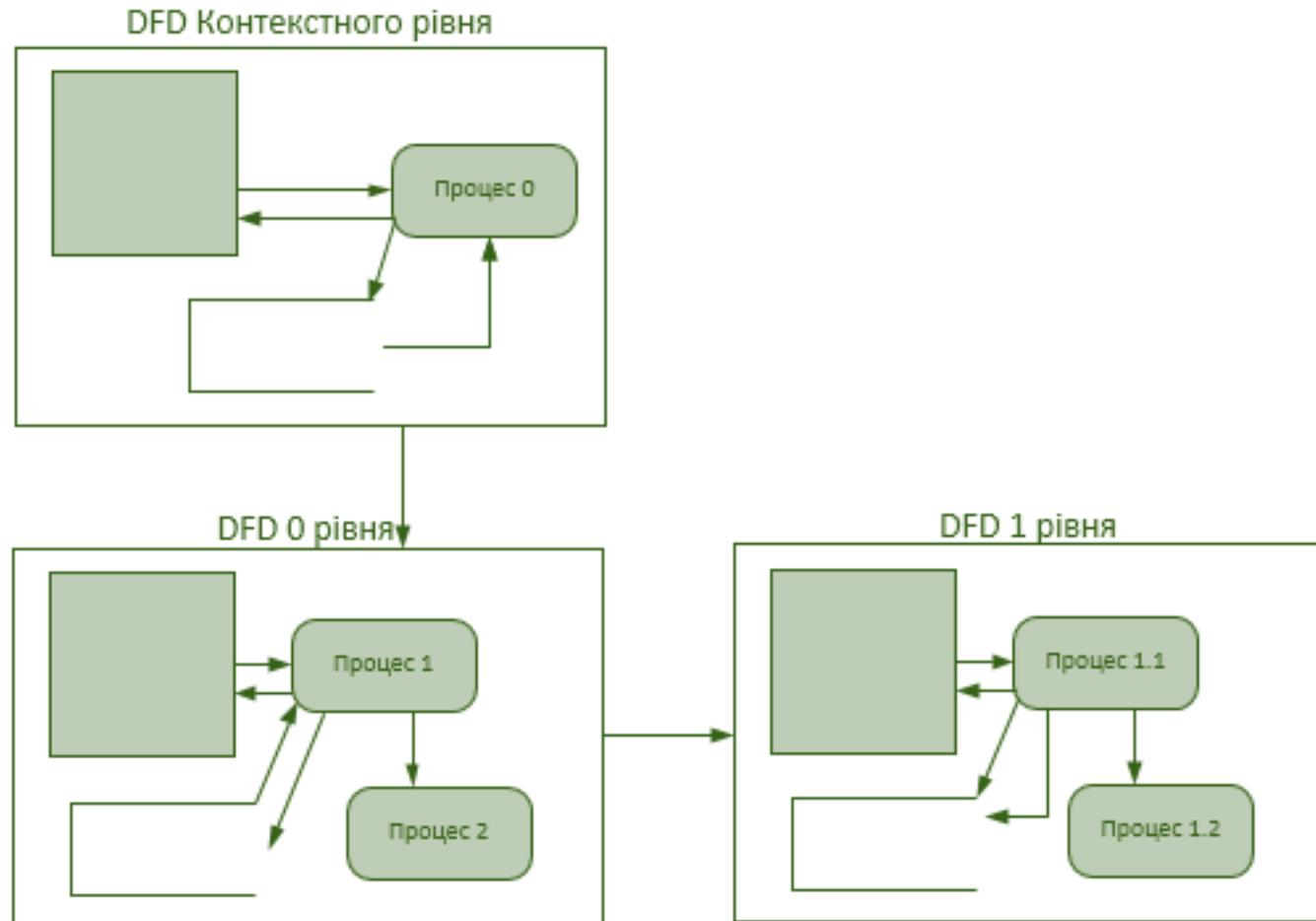
Декомпозиція контекстної DFD-моделі



декомпозиція – побудова діаграми нижчого рівня



кожен процес може бути деталізований за допомогою окремої діаграми або мініспецифікації



Діаграми DFD нижнього рівня

- ✓ від 3 до 7 процесов на діаграмі
- ✓ не показуються несуттєві для цього рівня деталі
- ✓ декомпозиція потоків даних здійснюється одночасно з декомпозицією процесів
- ✓ для багатофункціональних або розподілених систем контекстна діаграма може включати кілька блоків
- ✓ на самому нижньому рівні визначаються специфікації процесів

Правила деталізації процесів



правило балансування

при деталізації процесу дочірня діаграма в якості зовнішніх джерел / приймачів даних може мати тільки ті компоненти (підсистеми, процеси, зовнішні сутності, накопичувачі даних), з якими має інформаційну зв'язок відповідний процес на батьківській діаграмі;



правило нумерації

при деталізації процесів повинна підтримуватися їх ієрархічна нумерація



правило семи процесів

для того, щоб діаграма легко читалася, кількість функцій на діаграмі не повинно бути більше семи

Специфікація процесу

документ, який детально описує логіку процесу. **Містить:**



номер процесу



списки вхідних і вихідних даних



тіло процесу – докладний алгоритм функції, що перетворює вхідні потоки даних у вихідні

є кінцевою вершиною ієрархії моделі DFD

Специфікація процесу

- ✓ процес має **2-3** вхідних/вихідних потоків
- ✓ процес виконує **єдину логічну функцію**
- ✓ описання логіки процесу займає **20-30 рядків**
- ✓ процес може бути описаний у вигляді послідовного **алгоритму**

Словник даних

У словнику даних визначається **структура і зміст** усіх потоків даних і накопичувачів даних, які присутні на діаграмах:

Для кожного потоку в словнику зберігаються:



ім'я потоку



тип потоку



атрибути потоку

Як створити діаграму DFD



Введення та виведення інформації

Більшість процесів і систем починаються з введення інформації із зовнішнього джерела та закінчуються виведенням даних в іншу сутність чи базу



Створення контекстної схеми

Додайте вузол одиничного процесу та з'єднайте його з необхідними зовнішніми сутностями. Цей вузол символізує узагальнений процес, через який проходить інформація від введення до виведення



Розширення контекстної схеми до DFD 1-го рівня

Повинна включати кілька вузлів процесів, а також основні бази даних та всі зовнішні сутності



Поглиблення діаграми DFD до 1-го рівня і далі

Процеси, що входять до складу діаграми DFD 0-го рівня, можна так само розбити на докладніші підпроцеси. Коли діаграма досягне бажаного рівня деталізації, можна закінчити з поділом



Перевірка остаточної схеми

Чи логічно все викладено? Чи всі сховища даних на місці? Фінальний варіант схеми повинен давати чітке уявлення про те, як улаштована ваша система.

Повнота та несуперечність діаграми DFD

Повнота моделі

- ✓ Відсутні “висячі” процеси



Несуперечність моделі:

- ✓ Немає потоків які пов'язують дві зовнішні сутності
- ✓ Немає потоків, які пов'язують зовнішню сутність і сховище
- ✓ Немає потоків, що поєднують два сховища даних

Переваги та недоліки DFD

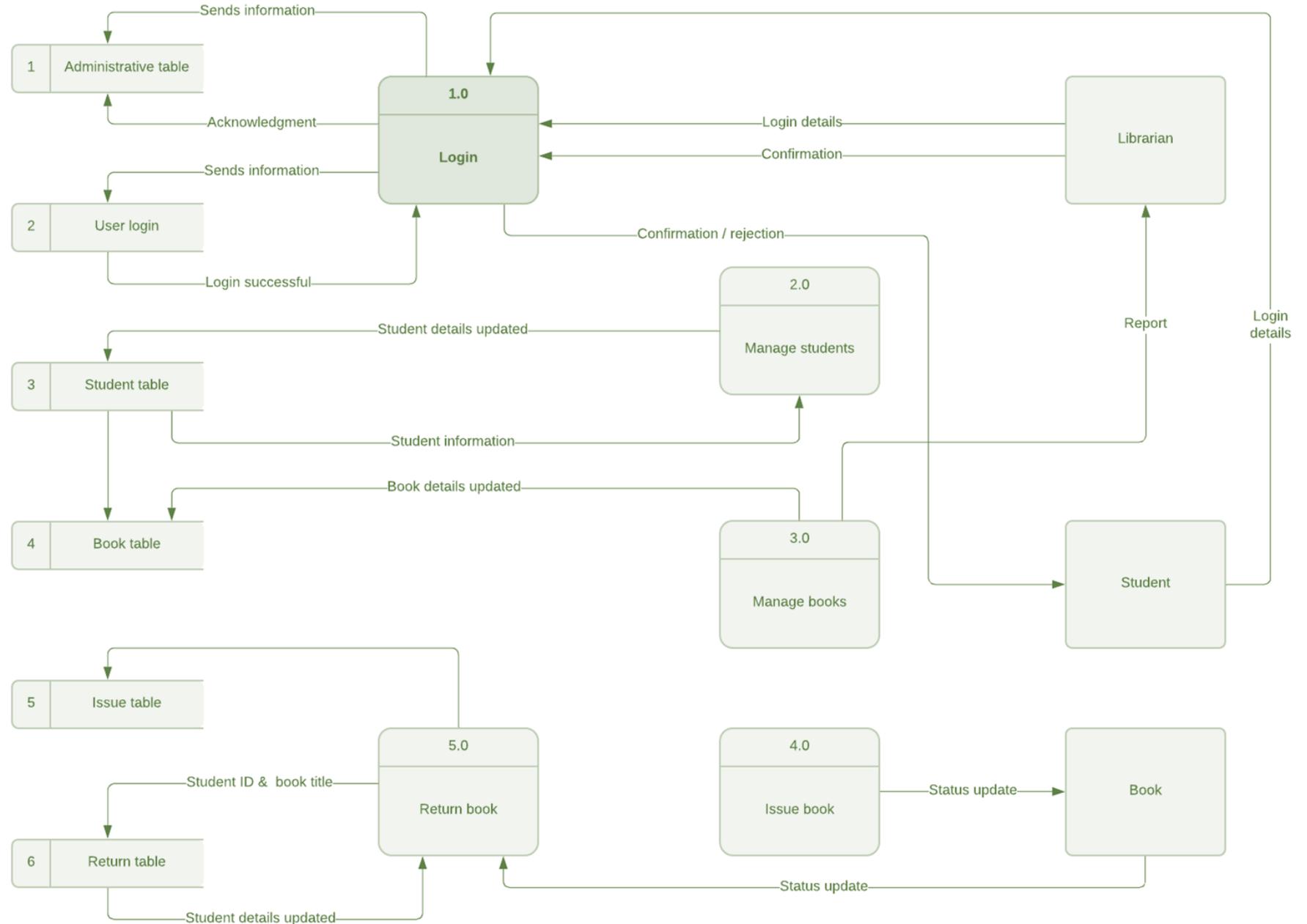
Переваги:

- ✓ Однозначне визначення зовнішніх сутностей
- ✓ Проектування “зверху-вниз”
- ✓ Повна функціональна специфікація системи

Недоліки:

- ✓ Штучне введення управляючих процесів
- ✓ Відсутність аналізу часових параметрів перетворення даних

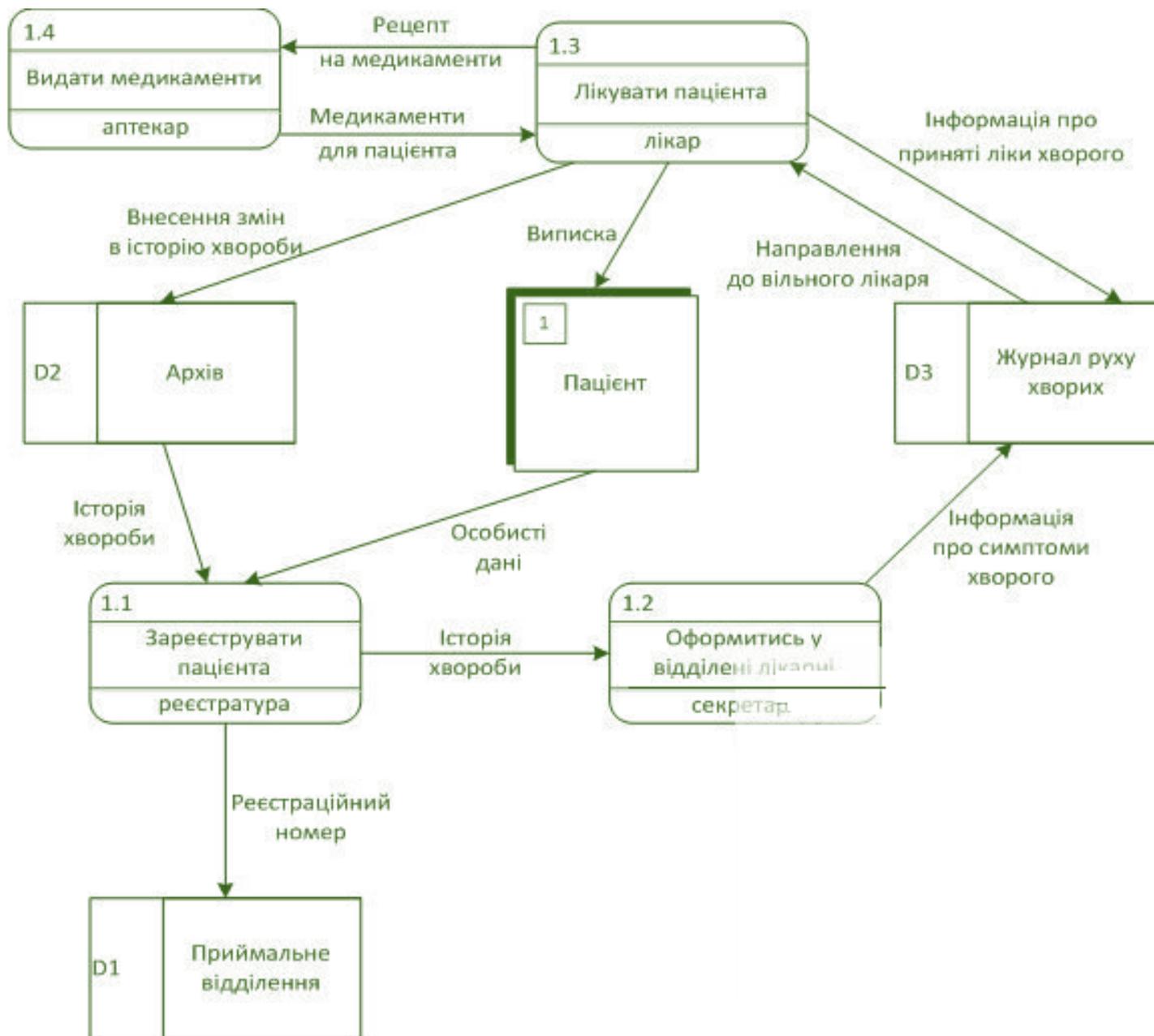
Перевірка моделі



Перевірка моделі: контекстний рівень



Перевірка моделі: 0 рівень



Ключові аспекти

- ✓ Суть DFD моделювання
- ✓ Основні елементи моделі
- ✓ Нотації моделі
- ✓ Графічне представлення елементів
- ✓ Правила побудови





**Методологія
моделювання
IDEF 1X: модель
даних**

Структурно-функціональний підхід



визначення функціоналу майбутньої системи



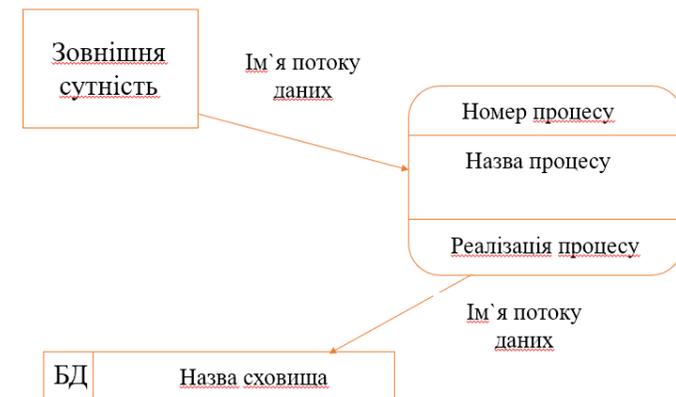
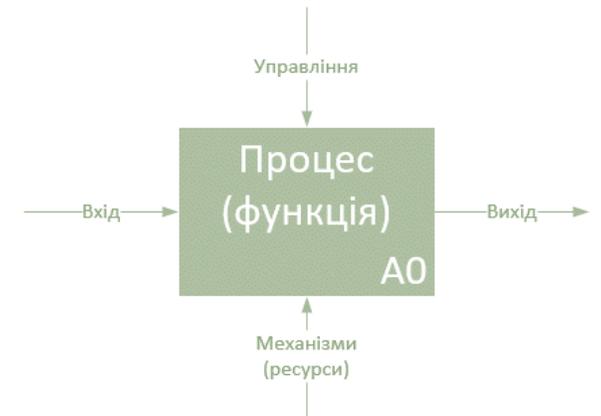
побудови функціональної моделі, з врахуванням вхідних-вихідних потоків та опису коли і яким чином функції виконуються й управляються



моделювання потоків даних

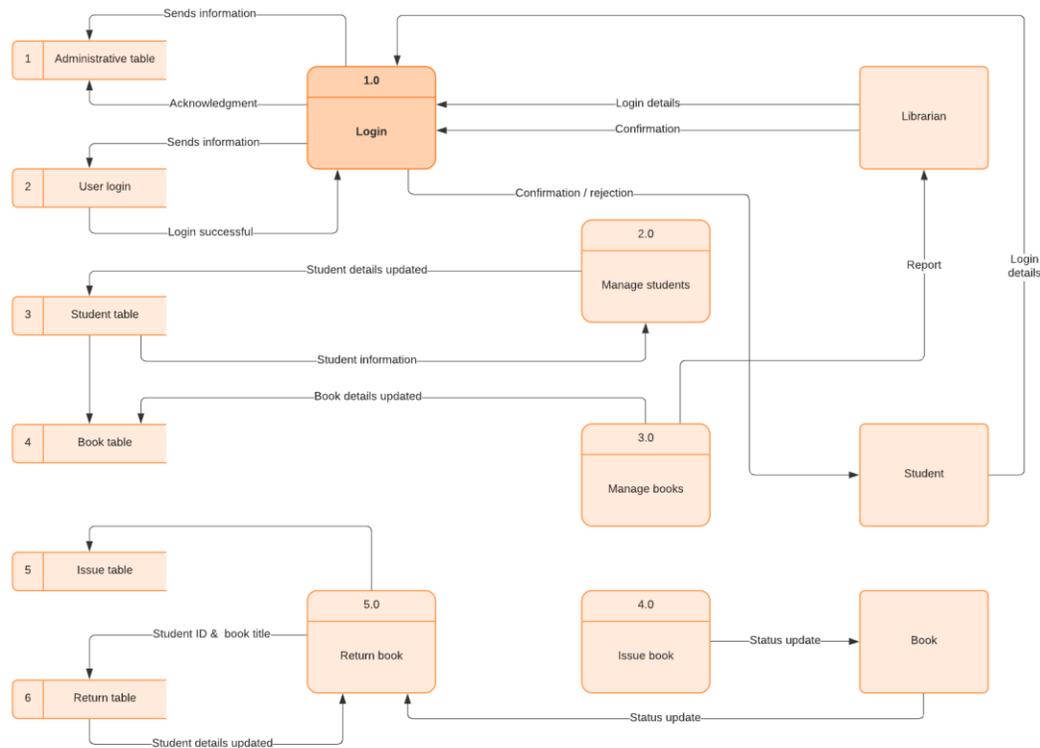


Який наступний крок?



Моделювання даних

Модель «сутність-зв'язок» – моделює дані, які необхідні при виконанні процесів



сутності



атрибути



відносини

Що таке ER-діаграма?

Схема "сутність-зв'язок" (також ERD або ER-діаграма) - це різновид блок-схеми, де показано, як різні "сутності" (люди, об'єкти, концепції і так далі) пов'язані між собою всередині системи.

Застосовуються:



для проектування та налагодження реляційних баз даних



дослідження та розробки програмного забезпечення та інформаційних систем для бізнесу

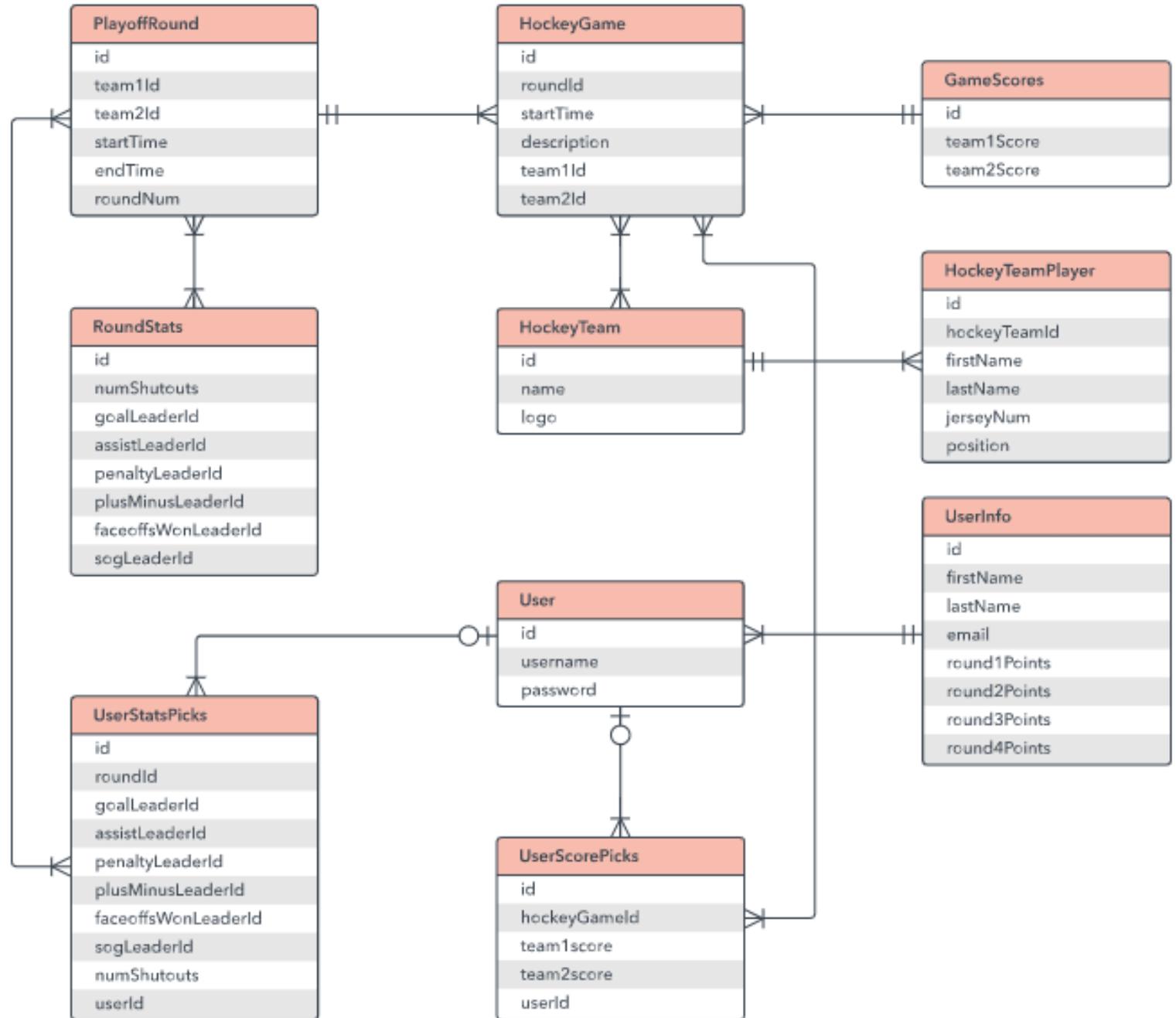
ER-діаграми (або ER-моделі) покладаються на стандартний набір символів, включаючи прямокутники, ромби, овали та сполучні лінії для відображення сутностей, їх атрибутів та зв'язків. Ці діаграми влаштовані за тим самим принципом, як і граматичні структури: сутності виконують роль іменників, а зв'язку — дієслів.

Приклад ER-діаграми

Сутність



Екземпляр сутності



Рівні деталізації

-  **Концептуальна модель даних** – схема найвищого рівня з мінімальною кількістю подробиць. Перевага цього підходу полягає у можливості відобразити загальну структуру моделі та всю архітектуру системи. Менш масштабні системи можуть обійтися без цієї моделі. І тут можна відразу переходити до логічної моделі
-  **Логічна модель даних** містить більш детальну інформацію, ніж концептуальна модель. На цьому рівні визначаються докладніші операційні та транзакційні сутності. Логічна модель залежить від технології, у якій застосовуватиметься
-  **Фізична модель даних** - на основі кожної логічної моделі даних можна скласти одну чи дві фізичні моделі. В останніх має бути достатньо технічних подробиць для складання та впровадження самої бази даних

Обмеження застосування



Тільки реляційні дані. Слід чітко розуміти, що мета ER-діаграм – показати зв'язки та відносини між елементами, тому вони відображають лише реляційну структуру



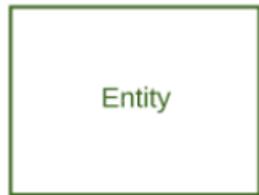
Тільки для структурованих даних. Дані мають бути чітко розбиті на поля, стовпці та рядки, інакше користі від ER-діаграми буде мало. Це стосується і частково структурованих даних, оскільки лише деякі з них будуть придатними для роботи.



Складність інтеграції з існуючою базою даних. Застосування ER-моделей для інтеграції з існуючою базою даних — непросте завдання через різницю в архітектурах

Нотація Чена: символи ERD-сутностей

Під поняттям «сутності» маються на увазі об'єкти чи поняття, які мають важливу інформацію. З погляду граматики, вони, зазвичай, позначаються іменниками, наприклад, «товар», «клієнт», «заклад» чи «акція»



**Незалежна
(сильна)
сутність**

Не залежить від інших сутностей і часто називається «батьківською», тому що у неї в підпорядкуванні зазвичай є слабкі сутності. Незалежні сутності супроводжуються первинним ключем, що дозволяє ідентифікувати кожен екземпляр сутності



**Залежна
(слабка)
сутність**

Сутність, що залежить від сутності іншого типу. Не супроводжується первинним ключем і не має значення у схемі без своєї батьківської сутності.

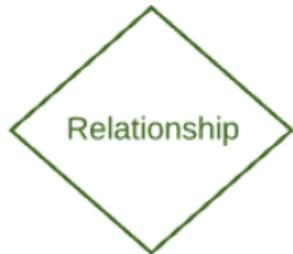


**Асоціативна
сутність**

Поєднує екземпляри сутностей різних типів. Також містить атрибути, характерні зв'язків між цими сутностями

Нотація Чена: символи ERD-зв'язків

Зв'язки використовуються у схемах «сутність-зв'язок» для позначення взаємодії між двома сутностями. Граматично зв'язки, як правило, виражаються дієсловами, наприклад, "призначити", "закріпити", "відстежити", і несуть корисну інформацію, яку неможливо отримати, спираючись лише на типи сутностей



Зв'язок

Відношення між сутностями



Слабкий зв'язок

Зв'язок між залежною сутністю та батьківською

Нотація Чена: Символи ERD-атрибутів

ERD-атрибути характеризують сутність, дозволяючи користувачам краще розібратися у пристрої бази даних. Атрибути містять інформацію про сутності, виділені в концептуальній ER-діаграмі



Атрибут

Характеризує сутність, і навіть відносини між двома чи більше елементами



Багатозначний атрибут

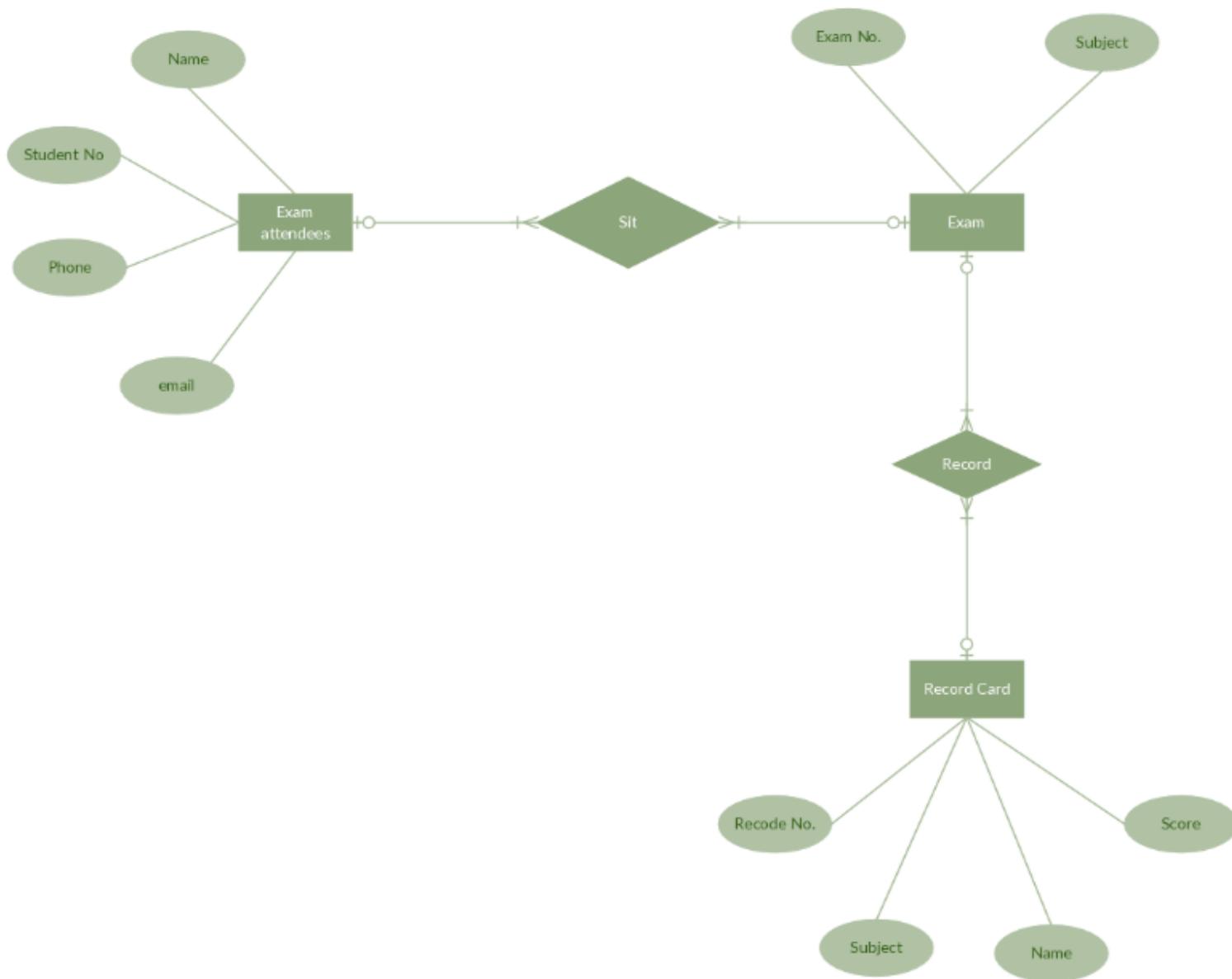
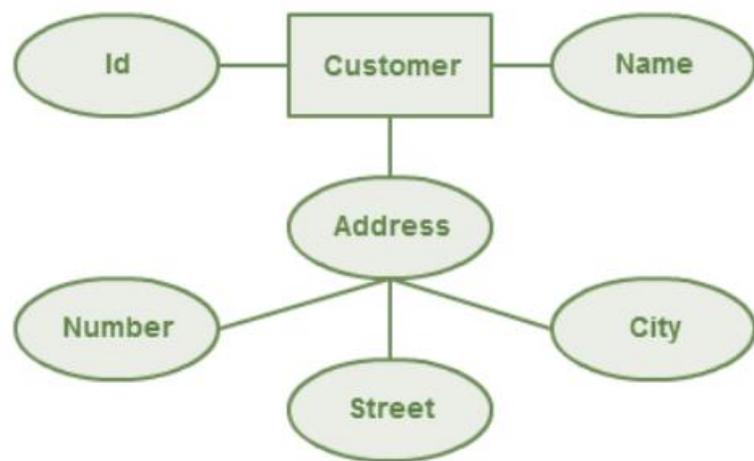
Атрибут, якому може бути надано кілька значень



Похідний атрибут

Атрибут, значення якого можна обчислити, спираючись на значення пов'язаних з ним атрибутів

Нотація Чена



Нотація Мартіна (нотація гусячих лапок)



У цій нотації **атрибути** сутностей записуються прямо всередині символу **прямокутника**, що означає **сутність**.



Діаграма виходить компактнішою



Відносини більше не зображуються як ромбів, а записуються як тексту над стрілкою



Категоризація атрибутів

Ключі – один із способів категоризації атрибутів. Застосовуються з метою максимально ефективно зв'язати між собою різні таблиці в базі даних

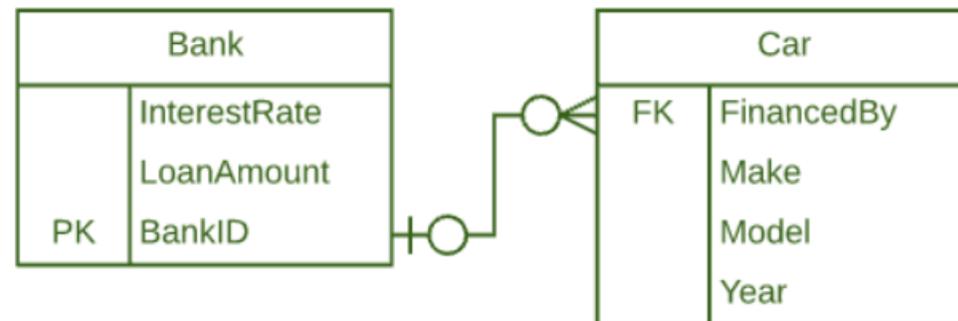


Первинний ключ — це атрибут чи поєднання атрибутів, що ідентифікують один конкретний екземпляр сутності



Зовнішній ключ створюється щоразу, коли атрибут прив'язується до сутності у вигляді одиничного чи множинного зв'язку

Наприклад, позику на кожен окремий автомобіль може бути виданий тільки одним банком, тому як зовнішній ключ FinancedBy («ким видано позику») у таблиці Car (автомобіль) використаний основний ключ BankId («ідентифікатор банку»). При цьому ідентифікатор BankId може бути зовнішнім ключем відразу для кількох автомобілів



Типи даних

Під типом мається на увазі тип даних у відповідному полі таблиці. Однак це також може бути тип сутності, тобто опис її складових. Наприклад, по суті «книга» будуть такі типи: «автор», «назва» та «дата публікації»

Entity
Field
Field
Field

Entity	
Key	Field
Key	Field
Key	Field

Entity	
Field	Type
Field	Type
Field	Type

Entity		
Key	Field	Type
Key	Field	Type
Key	Field	Type

Додається на етапі проектування

Типи зв'язків

✓ Під **кардинальністю** мається на увазі максимальна кількість зв'язків, яка може бути встановлена між екземплярами різних сутностей

✓ **Ординальність** – показує мінімальну кількість зв'язків між примірниками двох сутностей

✓ **Кардинальність та ординальність** відображаються на сполучних лініях згідно з вибраним форматом нотації



One



Many



One (and only one)



Zero or one



One or many

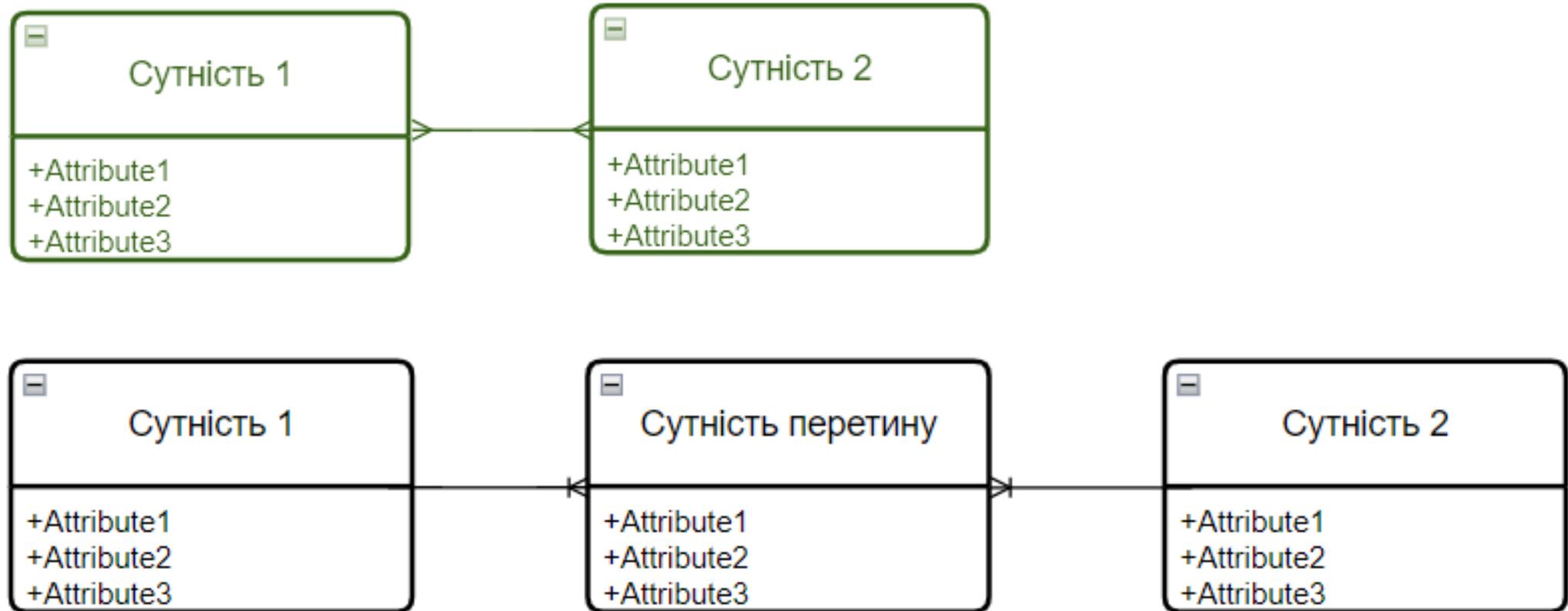


Zero or many

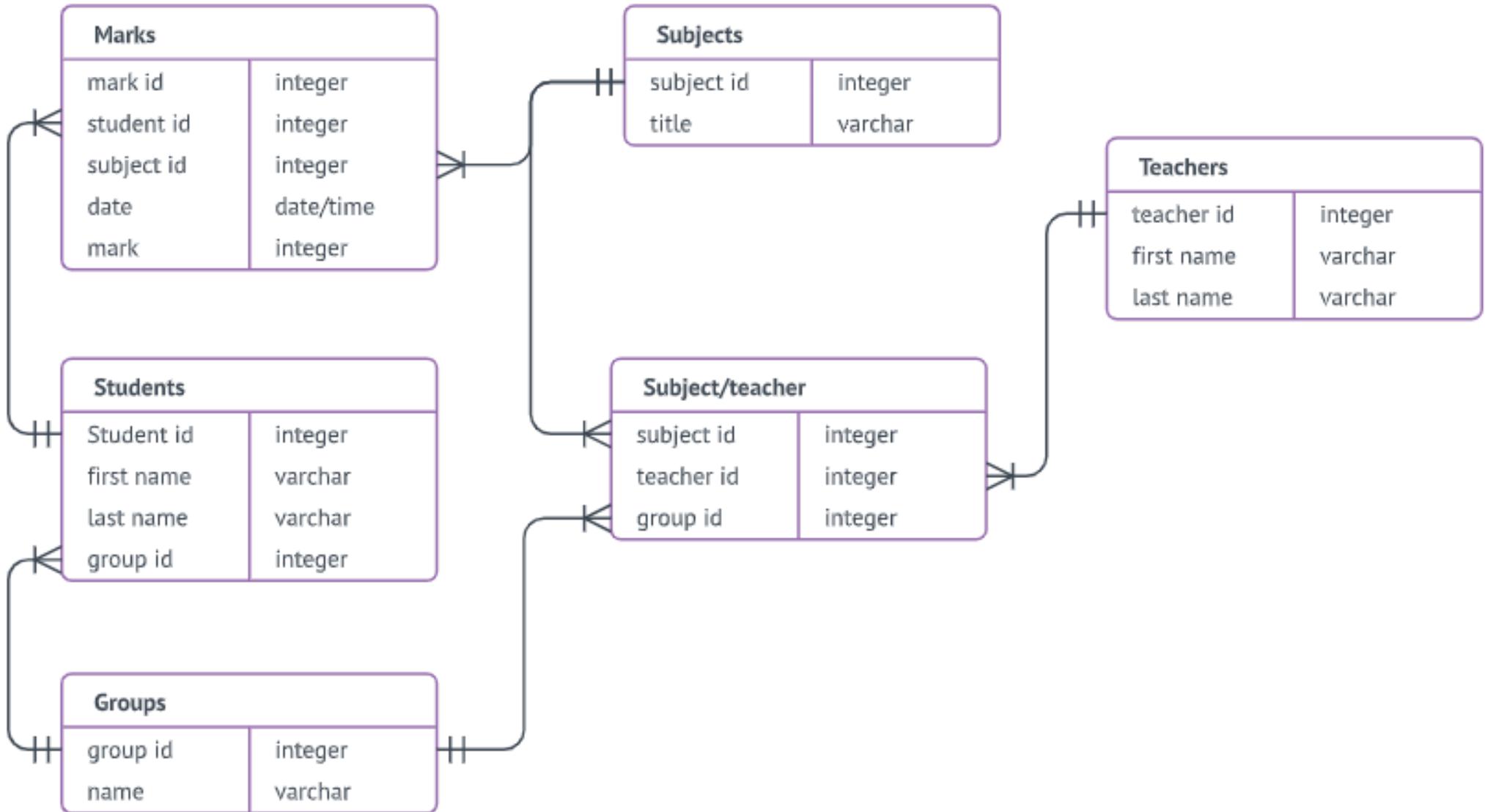
Етапи створення ER-діаграми

- ✓ **Встановіть сутності на основі виділених сховищ** (вони, як правило, позначаються іменниками, наприклад, "автомобіль", "банк", "студент", "товар" і так далі)
- ✓ **Визначтеся зі зв'язками** (вони показують, як сутності взаємодіють між собою)
- ✓ **Додайте атрибути** (вони ілюструють конкретні характеристики сутності, фокусуючи увагу на важливій інформації у контексті моделі)
- ✓ **Додайте сутності перетину**

Сутності перетину



Приклад ER-діаграми



Ключові аспекти

- ✓ Стандарт моделювання даних?
- ✓ Суть ER-моделі
- ✓ Нотації моделі
- ✓ Типи зв'язків
- ✓ Правила побудови

