

Кафедра управління інформаційною безпекою

Спеціальність «Кібербезпека», 5 курс навчання

ЛЕКЦІЯ. ЛОГІЧНЕ ПРОЕКТУВАННЯ БАЗ ДАНИХ

План:

1. Етапи логічного проектування.
2. Спрощення концептуальної моделі.
3. Методика перетворення ER-діаграм в реляційні структури.
4. Перевірка відношень за допомогою правил нормалізації.
5. Перевірка відповідності відношень вимогам транзакцій користувачів.
6. Перевірка підтримки цілісності.
7. Приклад створення логічної моделі бази даних

Література

1. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. К.: КНУБА, 2005. – 204 с.
2. Гайна Г.А. Організація баз даних і знань. Мови баз даних: Конспект лекцій.–К.:КНУБА, 2002. – 64 с.
3. Гайна Г.А., Попович Н.Л. Організація баз даних і знань. Організація реляційних баз даних: Конспект лекцій.–К.:КНУБА, 2000. – 76 с.

1. Етапи логічного проектування.

Логічне проектування виконується для певної моделі даних. Для реляційної моделі даних логічне проектування полягає у створенні реляційної схеми, визначенні числа і структури таблиць, формуванні запитів до БД, визначенні типів звітних документів, розробці алгоритмів обробки інформації, створенні форм для вводу і редагування даних в БД і рішенні цілого ряду інших задач. Концептуальні моделі за певними правилами перетворюються в логічні моделі даних. Коректність логічних моделей перевіряється за допомогою *правил нормалізації*, які дозволяють переконатися в структурній узгодженості, логічній цілісності і мінімальній збитковості прийнятої моделі даних. Модель також перевіряється з метою виявлення можливостей *виконання транзакцій*, які будуть задаватися користувачами. Проектування являє собою циклічний процес. Етапи логічного проектування наведені на рис. 1.

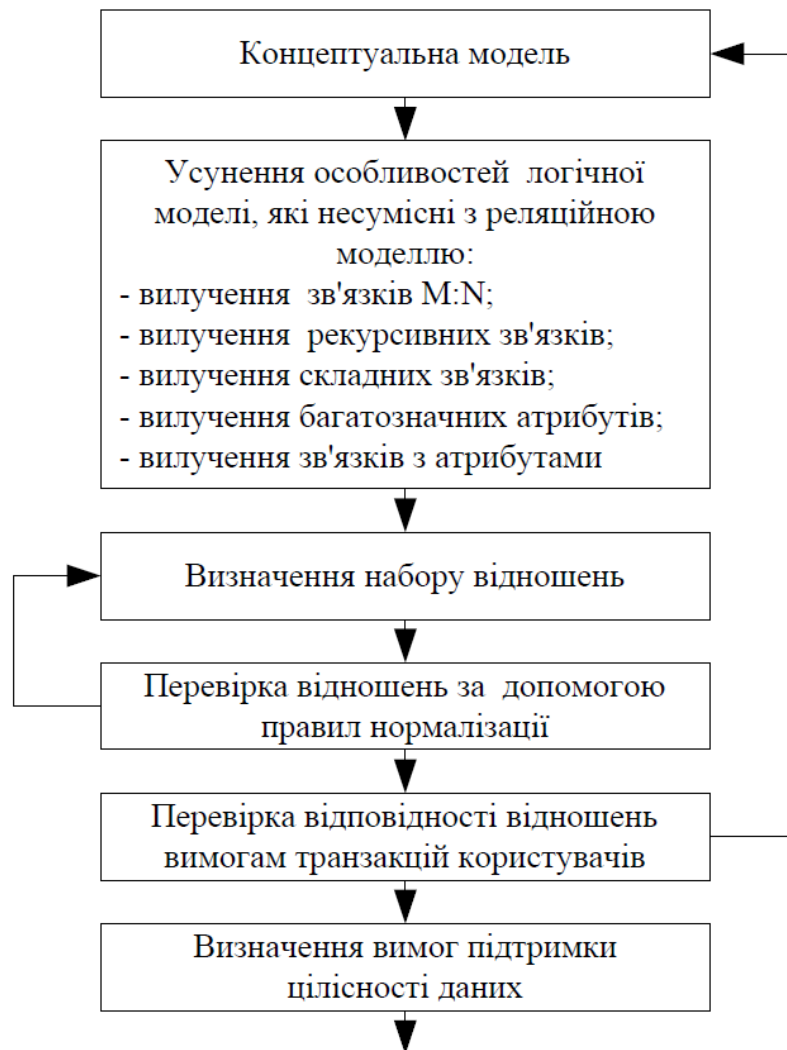


Рис. 1. Етапи логічного проектування бази даних

2. Спрощення концептуальної моделі.

Першим кроком спрощення концептуальної моделі є попередні перетворення з метою усунення зв'язків, які є несумісними з реляційною моделлю.

На цьому етапі виконуються такі операції:

- вилучення двосторонніх зв'язків M:N;
- вилучення складних зв'язків;
- вилучення багатозначних атрибутів;
- вилучення рекурсивних зв'язків;
- вилучення зв'язків з атрибутами.

Вилучення двосторонніх зв'язків "багато до багатьох"

Перетворення зв'язку "багато до багатьох" виконується шляхом введення проміжної сутності із заміною одного зв'язку M:N двома зв'язками 1:N з новою сутністю.

Приклад. Викладач може викладати багато Дисциплін, одну Дисципліну викладає багато Викладачів (рис. 2).

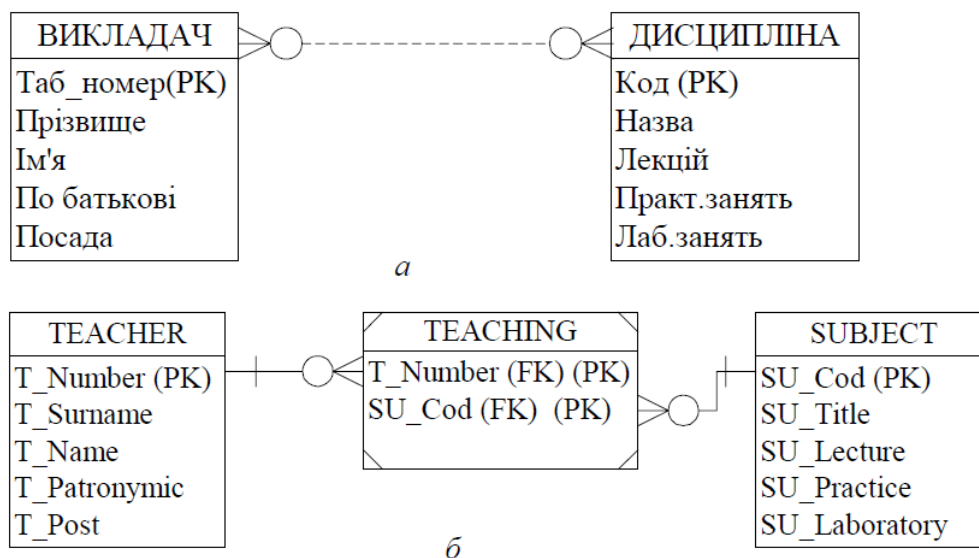


Рис. 2. Перетворення зв'язку "багато до багатьох": а – зв'язок M:N; б – результат перетворення – два зв'язку 1:N

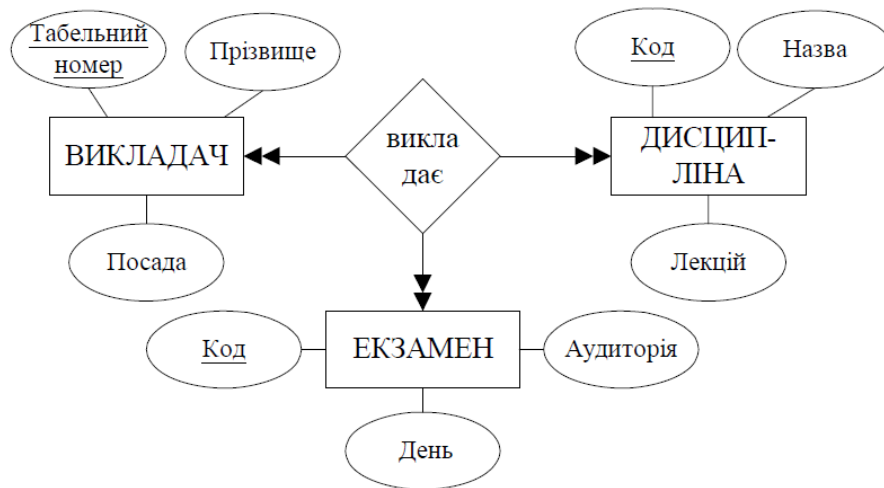
У результаті перетворення отримана нова сутність, яка є слабкою і залежить від двох інших сутностей. Її первинний ключ складається з первинних ключів двох сутностей, а кожен атрибут окремо є вторинним ключем.

Вилучення складних зв'язків

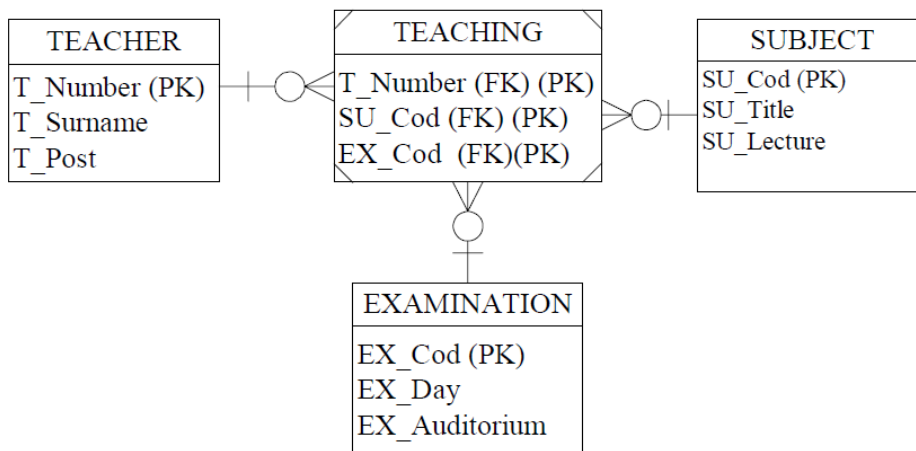
Для вилучення складних зв'язків виконуються такі операції:

- у модель вводиться нова сутність;
- складний зв'язок замінюється бінарними зв'язками "один до багатьох" зі знов створеною сутністю;
- кількість бінарних зв'язків дорівнює ступеню складності зв'язку.

Приклад. Викладач може викладати багато Дисциплін, одну Дисципліну викладає багато Викладачів. З Дисципліни Викладач проводить Екзамен (рис. 3).



а



б

Рис. 3. Представлення тернарного зв'язку бінарними зв'язками: а – складний зв'язок *Викладає*; б – декомпозиція складного зв'язку на три двосторонні зв'язки і нову сутність *Екзамен*

Вилучення багатозначних атрибутів

Якщо в концептуальній моделі даних присутній багатозначний атрибут, то може бути виконана декомпозиція цього атрибуту для визначення деякої сутності.

Приклад. Студент може мати декілька телефонів (рис. 4).

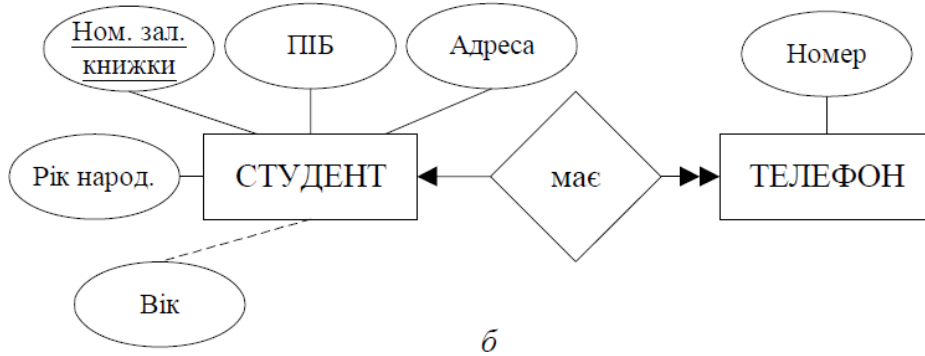
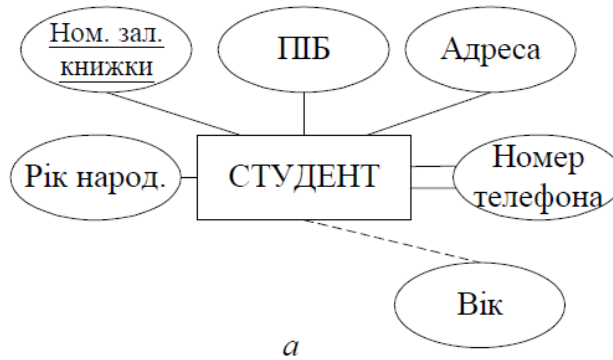


Рис. 4. Вилучення багатозначного атрибуту: а – сутність *Студент* з багатозначним атрибутом *Номер телефону*; б – нова сутність *Телефон*

Вилучення рекурсивних зв'язків

На етапі спрощення концептуальної моделі рекурсивні зв'язки 1:1 і 1:M (рис. 5) можуть бути перетворені у одне відношення. У випадку, коли є необов'язкова сутність з боку "багато" для зв'язку 1:M для зменшення пустих значень створюється нове відношення. Зв'язок M:N перетворюється на дві сутності (рис.6).

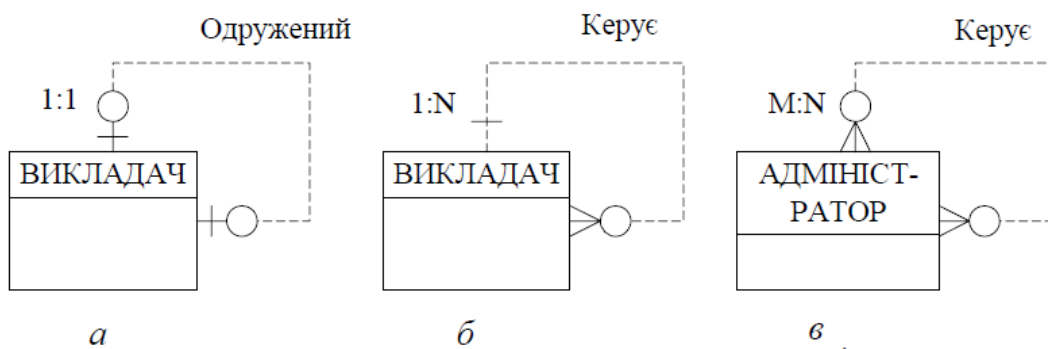


Рис. 5. Види рекурсивних зв'язків: а – 1:1; б – 1:N; в – M:N

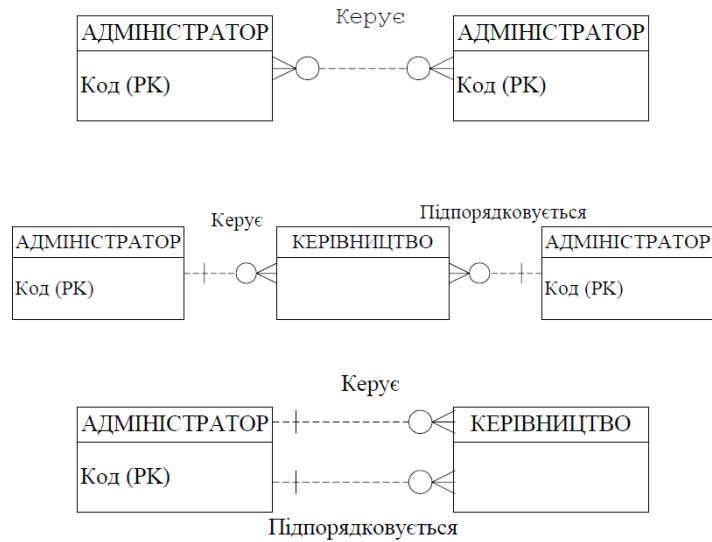


Рис. 6. Етапи послідовного перетворення рекурсивного зв'язку M:N

Вилучення зв'язків з атрибутами

Вилучення зв'язків з атрибутами виконується шляхом додавання у модель нової сутності для відношення M:N з атрибутами зв'язку. Для відношення 1:M атрибути зв'язку передаються у сутність "багато" без створення нової сутності.

Приклад. Розглянемо сутності *Студент-Дисципліна* (рис.7).

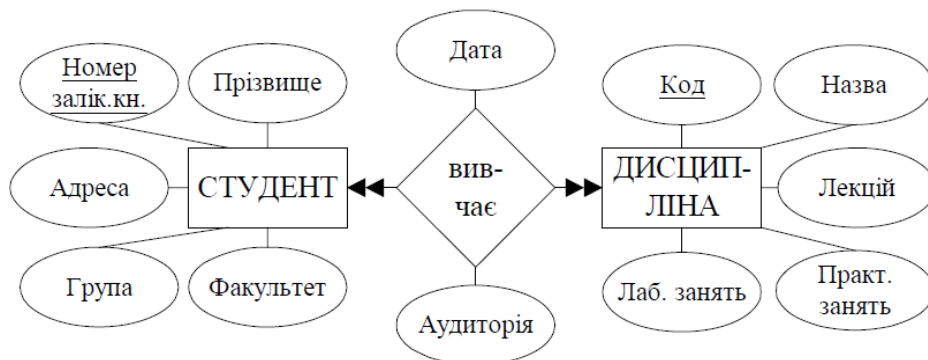


Рис. 7. Зв'язок "багато до багатьох" з атрибутами зв'язку Дата і Аудиторія
У результаті перетворення зв'язку з атрибутами отримано реляційну схему (рис. 8).

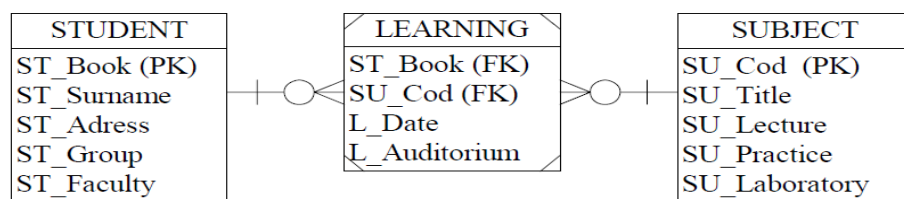


Рис. 8. Представлення атрибутів зв'язку *L_Date* і *L_Auditorium* у новому відношенні *Learning*

Спрощення концептуальної моделі передбачає також вилучення збиткових зв'язків. Збиткові зв'язки характеризуються тим, що одна і та ж інформація може бути отримана не тільки через них, але і через інші зв'язки.

Після спрощення в концептуальній моделі можуть бути присутні тільки такі елементи:

- об'єкти і атрибути;
- зв'язки типу 1:1 і 1:M;
- зв'язки типу суперклас-підклас.

3. Методика перетворення ER-діаграм в реляційні структури

Для ER-моделі існує алгоритм однозначного перетворення її в реляційну модель даних.

Розглянемо правила перетворення ER-моделі в реляційну модель.

Сутності і атрибути

Для кожної сутності створюється відношення, кожен атрибут сутності стає атрибутом відповідного відношення.

Для *сильних сутностей* первинний ключ сутності стає PRIMARY KEY (PK) відповідного відношення.

Приклад. Розглянемо сутність *Студент* (рис. 9).

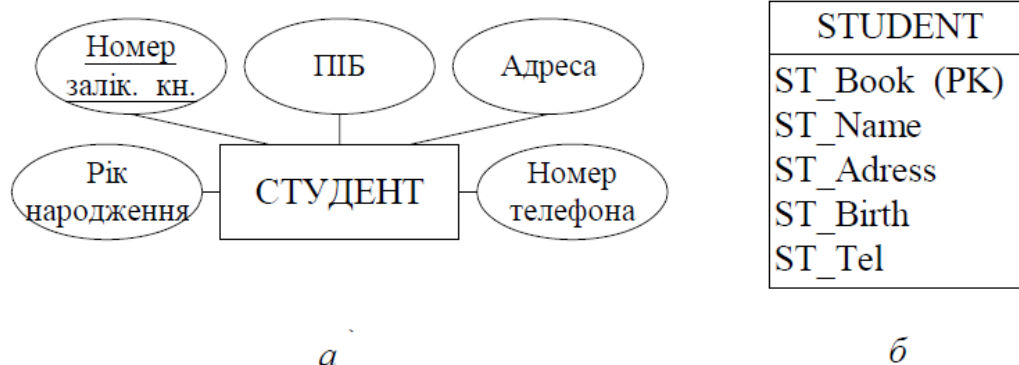


Рис. 9 . Перетворення сутності Студент (а) у відношення Student (б)

Для *слабких сутностей* первинний ключ частково або повністю залежить від ключа сутності володаря (декількох володарів), тобто PK визначається тільки тоді, коли визначені всі PK сутностей володарів.

Приклад. Розглянемо перетворення сильної сутності *Студент* і слабкої сутності *Нагорода* (рис. 10).

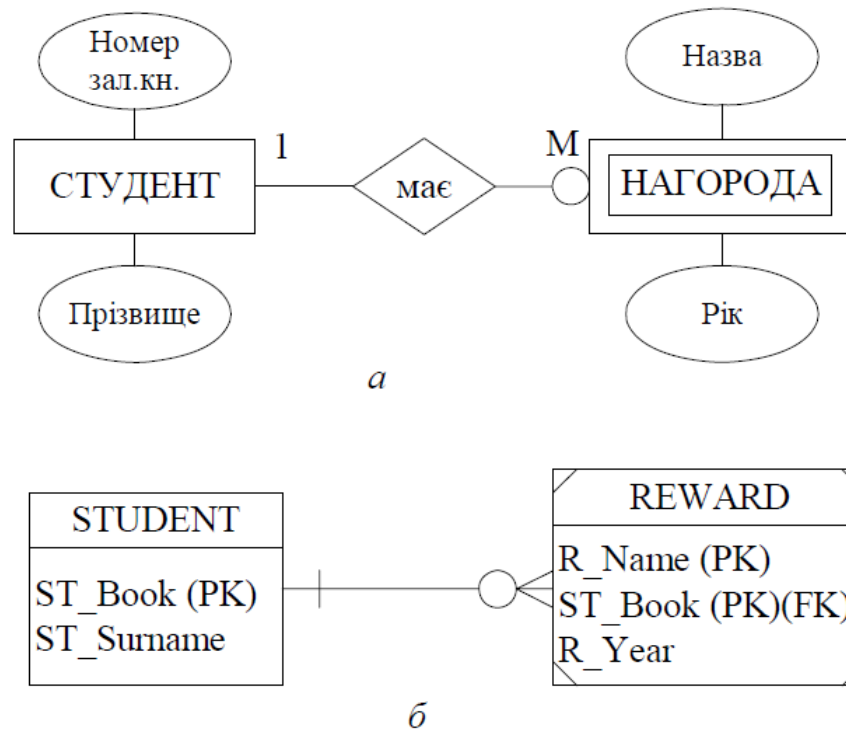


Рис. 10. Перетворення сильної сутності Студент і слабкої сутності Нагорода з ідентифікуючим зв'язком між ними (а) у зв'язані відношення Student і Reward (б)

Зв'язки

Після перетворення концептуальної моделі залишаються такі типи зв'язків:

- "один до одного";
- "один до багатьох";
- рекурсивні зв'язки;
- суперклас – підклас.

Для кожного типу зв'язку залежно від умов зв'язування існують свої різновиди. Зв'язки між відношеннями в реляційній моделі реалізуються шляхом використання первинних і зовнішніх ключів.

Зв'язки "один до одного"

В концептуальних моделях даних визначають такі обмеження ступеня участі сутностей:

- обов'язкова участь для обох сутностей;
- обов'язкова участь для однієї сутності;
- необов'язкова участь для обох сутностей.

Залежно від обмежень перетворення на реляційну модель будуть різні.

Приклад. Розглянемо можливі варіанти перетворення зв'язку між сутностями *Викладач* і *Дисципліна* (рис. 11).

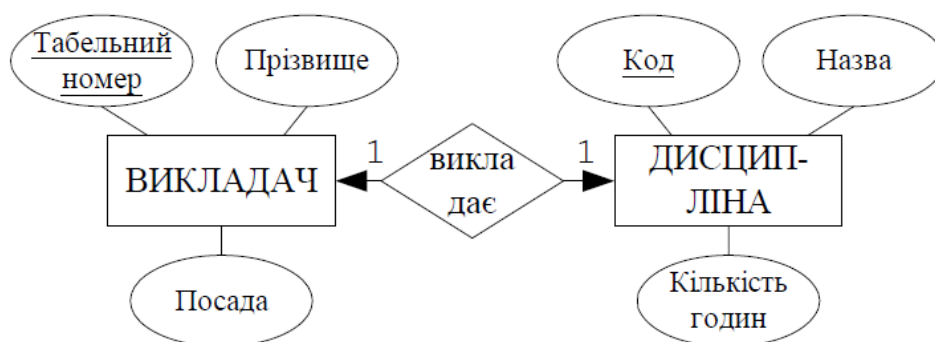


Рис. 11. Зв'язок 1:1 між сутностями *Викладач* і *Дисципліна*

1. *Обов'язкова участь для обох сутностей*

Припустимо, що кожен викладач обов'язково викладає одну дисципліну і кожну дисципліну обов'язково викладає один викладач. В цьому випадку реляційна структура буде складатися з одного відношення і мати один з таких варіантів (рис. 12).

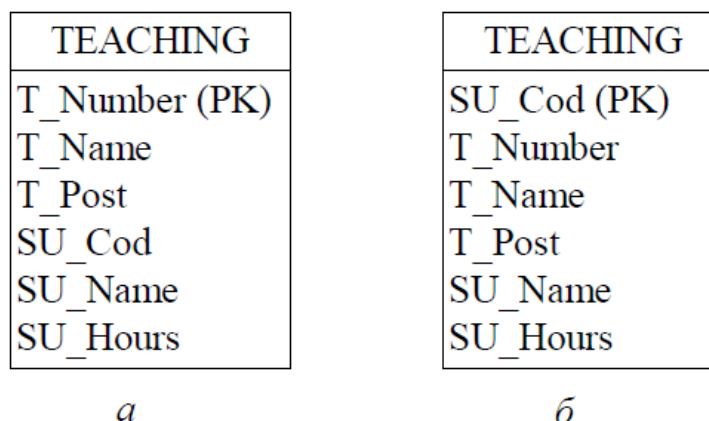


Рис. 12. Варіанти відношень для перетворення зв'язку 1:1 з обов'язковою участю для обох сутностей

2. *Обов'язкова участь для однієї сутності*

Припустимо, що кожен викладач обов'язково викладає одну дисципліну, а за кожною дисципліною необов'язково закріплений викладач. В цьому випадку сутність, яка є необов'язковою (*Дисципліна*) виступає в якості батьківської сутності, а обов'язкова сутність визначається як дочірня (*Викладач*). Реляційна структура показана на рис. 13.

Зовнішній атрибут *SU_Cod* також може бути ключем для *Викладача*.

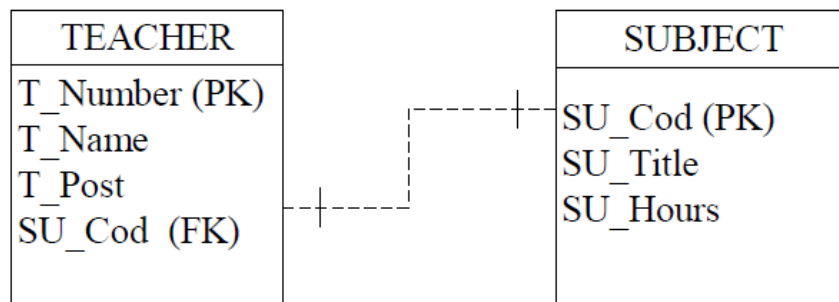


Рис. 13. Перетворення зв'язку 1:1 з обов'язковою участю сутності *Викладач* і необов'язковою участю сутності *Дисципліна*

3. Необов'язкова участь для обох сутностей

Припустимо, що необов'язково кожен викладач викладає дисципліни і необов'язково за кожною дисципліною закріплений викладач. В цьому випадку можливі три реляційні структури (рис. 14).

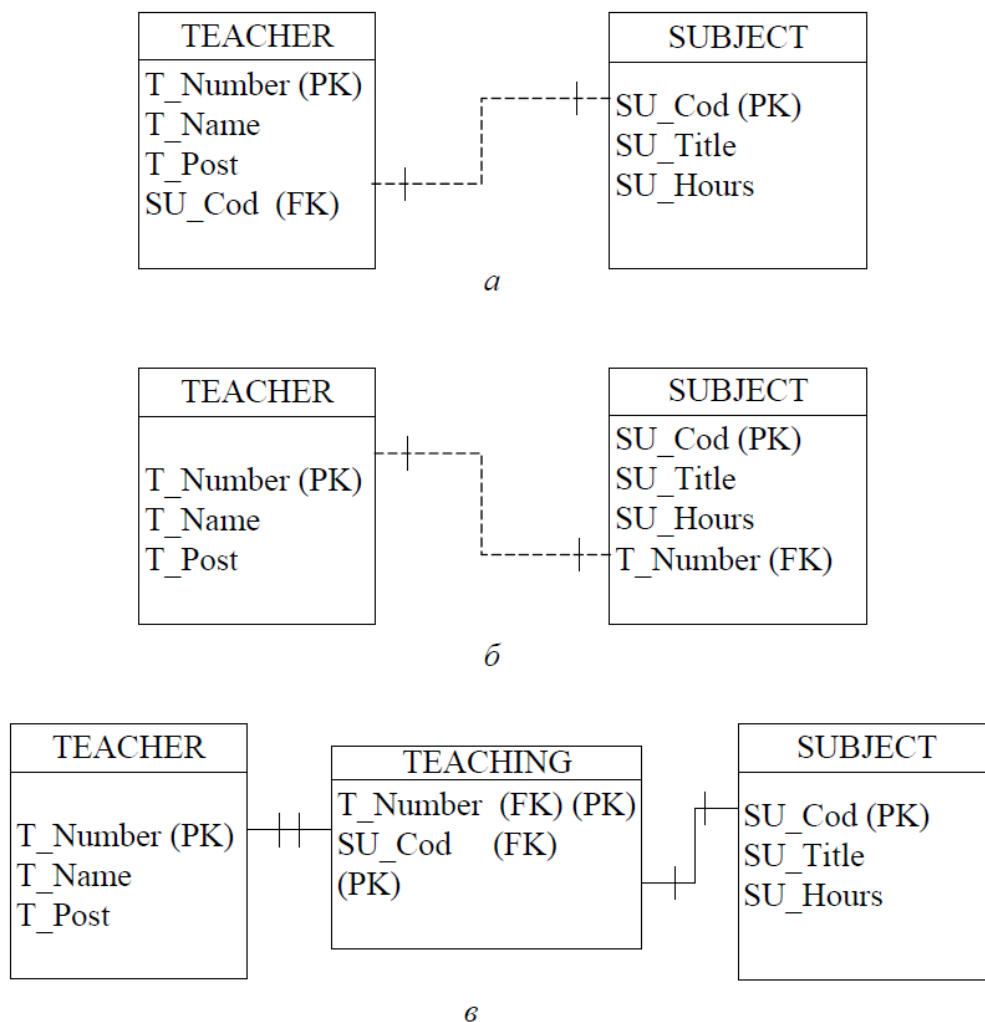


Рис. 14. Варіанти реляційних схем відношень для перетворення зв'язку 1:1 з необов'язковою участю для обох сутностей

Зв'язки "один до багатьох"

У кожне відношення, яке відповідає підлеглий (дочірній) сутності, додається набір атрибутів основної (батьківської) сутності, який складає первинний ключ основної сутності. У відношенні, що відповідає підлеглий сутності, цей набір атрибутів стає зовнішнім ключем (FOREIGN KEY, FK).

Для моделювання необов'язкового типу зв'язку у атрибутів, що відповідають зовнішньому ключу, встановлюється властивість допустимості невизначених значень (NULL). У разі обов'язкового типу зв'язку атрибути набувають властивості відсутності невизначених значень (NOT NULL).

Приклад. Розглянемо можливі варіанти перетворення зв'язку між сутностями *Викладач* і *Дисципліна* (рис. 15).

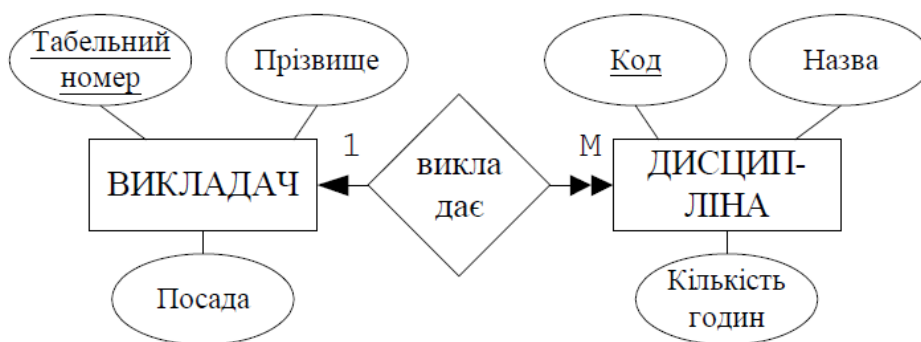


Рис. 15. Зв'язок 1:М між сутностями *Викладач* і *Дисципліна*

1. Необов'язкова участь сутності *Викладач* і обов'язкова участь сутності *Дисципліна* (рис. 16).

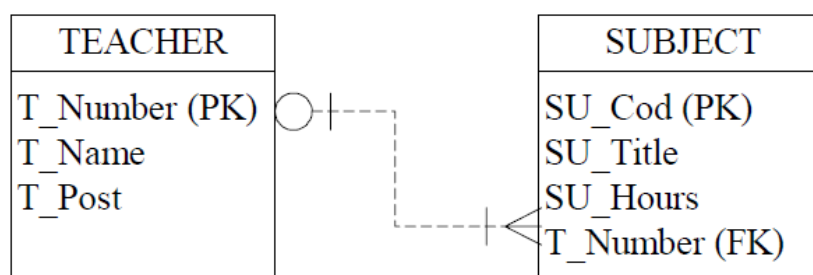


Рис. 16. Перетворення зв'язку 1:М з необов'язковою участю сутності *Викладач* і обов'язковою участю сутності *Дисципліна*

2. Необов'язкова участь сутності *Викладач* і необов'язкова участь сутності *Дисципліна* (рис. 17).

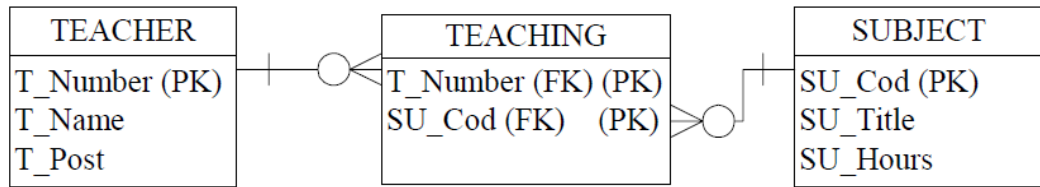


Рис. 17. Перетворення зв'язку 1:М з необов'язковою участю сутності *Викладач* і необов'язковою участю сутності *Дисципліна*

Зв'язки "багато до багатьох"

Для кожного зв'язку M:N необхідно створювати додаткове відношення, яке представляє цей зв'язок і включати в нього всі атрибути, які входять в склад цього зв'язку. Копії атрибутів первинного ключа сутностей, які беруть участь у зв'язку, передаються у нове відношення для використання в якості зовнішніх ключів. Ці зовнішні ключі утворюють також первинний ключ нового відношення.

Приклад. Розглянемо можливі варіанти перетворення зв'язку між сутностями *Викладач* і *Дисципліна* (рис. 18).

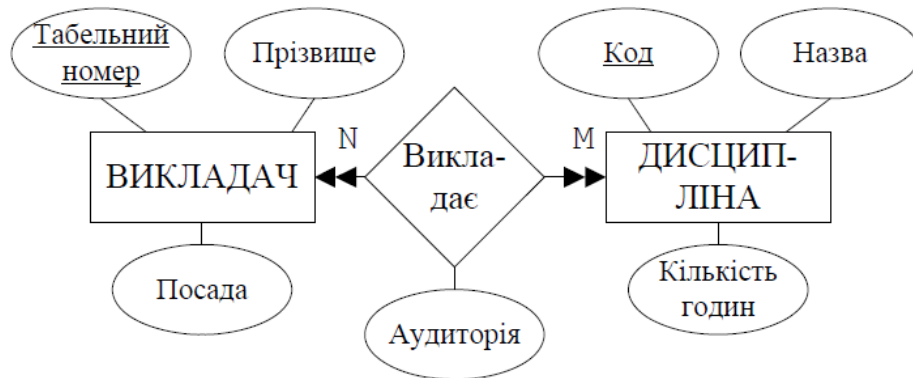


Рис. 18. Зв'язок N:М між сутностями *Викладач* і *Дисципліна*

В цьому випадку існує єдина схема перетворення (рис. 19).

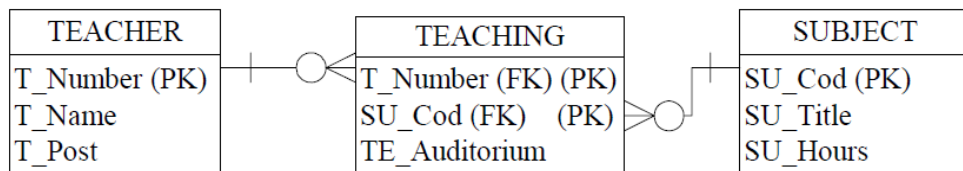


Рис. 19. Перетворення зв'язку N:М у реляційну схему

Для рекурсивних зв'язків 1:1 виконуються правила

визначені раніше для зв'язку між двома сутностями 1:1. Для рекурсивного зв'язку 1:1 з обов'язковою участю двох сторін, реляційна схема представляється

у вигляді одного відношення з двома копіями первинного ключа (див. рис. 12). Одна копія відповідає зовнішньому ключу. Для рекурсивного зв'язку 1:1 з обов'язковою участю тільки однієї сторони створюється або одне відношення, або нове відношення, яке відображає цей зв'язок (див. рис. 13). Для рекурсивного зв'язку 1:1 з необов'язковою участю обох сторін створюється нове відношення (див. рис. 14).

Для складних типів зв'язків створюється відношення, яке відображає цей зв'язок і включає всі атрибути, які входять в склад цього зв'язку. Копії атрибутів первинного ключа сутностей, які беруть участь у зв'язку, передаються у нове відношення для використання в якості зовнішніх ключів. Ці зовнішні ключі утворюють також первинний ключ нового відношення (див. рис. 3).

Для багатозначного атрибуту створюється нове відношення, яке відповідає багатозначному атрибуту, і в це нове відношення передається первинний ключ сутності для використання в якості зовнішнього ключа (див. рис. 4).

Зв'язки "суперклас – підклас"

Для виконання перетворення зв'язку типу суперклас – підклас у реляційну модель необхідно враховувати також обмеження ступеня участі у зв'язку (Mandatory або Optional) і обмеження неперетинання (And або Or). Можливі чотири сполучення, перетворення яких дає чотири реляційні схеми. На схему також впливає те, чи беруть участь підкласи в різних зв'язках, кількість сутностей в зв'язку і т.ін. Діапазон можливих варіантів рішення є достатньо великим і конкретна схема вибирається в кожному конкретному випадку з урахуванням багатьох факторів.

Приклад. Розглянемо суперклас *Викладач*, який має атрибути *Табельний номер*, *Прізвище*, *Посада*. Підкласами суперкласу виступають об'єкти *Професор*, *Доцент*, *Асистент* (рис. 20). Кожен екземпляр підкласу може бути екземпляром суперкласу, тобто суперклас може мати свої екземпляри (Optional). Кожен викладач обов'язково належить тільки одному підкласу (Or). Ця діаграма перетворюється в реляційну схему відношень показану на рис. 21. Зв'язок між відношеннями виконується за допомогою ключа суперкласу (*Табельний номер*).

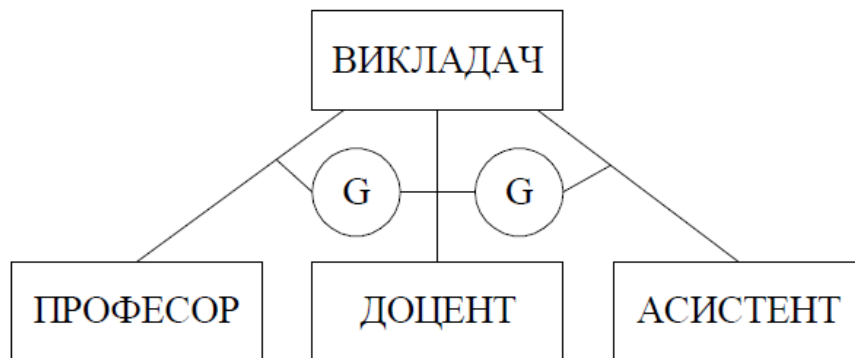


Рис. 20. Зв'язок суперклас – підклас з обмеженнями *Optional* і *Or*

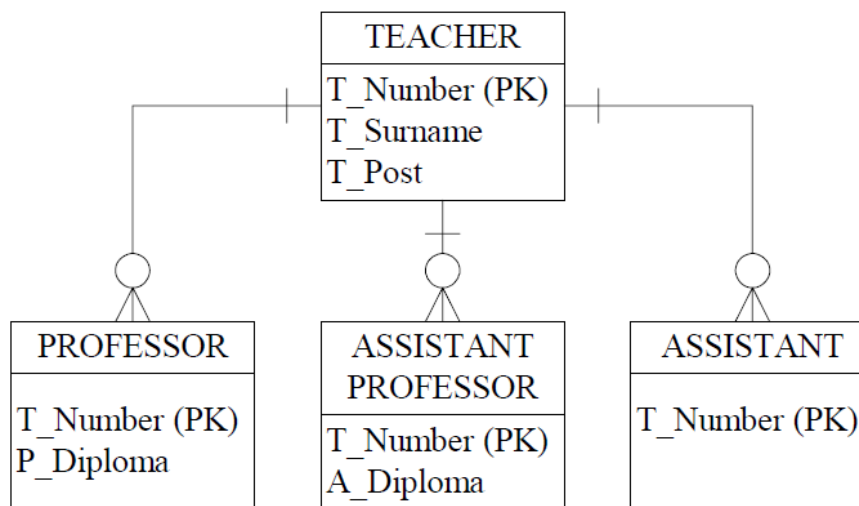


Рис. 21. Реляційна схема, яка відповідає попередньому зв'язку суперклас – підклас

Приклад. Розглянемо суперклас *Студент*, який має атрибути *Номер залікової книжки*, *Прізвище*, *Група*. Підкласами суперкласу виступають об'єкти *Очна*, *Заочна*, *Вечірня* і *Дистанційна форми навчання* (рис. 22). Кожен екземпляр підкласу є одночасно екземпляром суперкласу (*Mandatory*). Кожен студент може належати до декількох підкласів, тобто одночасно може займатися на різних формах навчання (*And*). Ця діаграма перетворюється в наступну реляційну схему відношень (рис. 23).

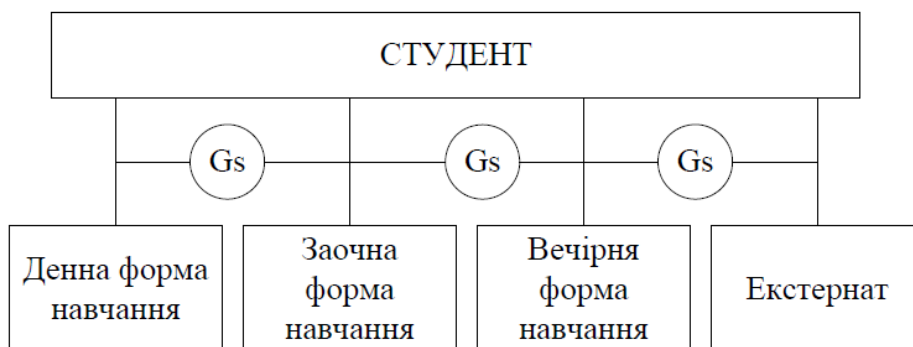


Рис. 22. Зв'язок суперклас–підклас з обмеженнями *Mandatory* і *And*

STUDENT
ST_Book (PK)
ST_Surname
ST_Group
ST_Confront
ST_Correspond
ST_Nightclasses
ST_Distance

Рис. 23. Реляційна схема, яка відповідає попередньому зв'язку суперклас– підклас

4. Перевірка відношень за допомогою правил нормалізації

Створений на попередніх етапах набір відношень логічної моделі БД повинен бути перевірений на коректність об'єднання атрибутів у кожному відношенні. Перевірка виконується шляхом застосування до кожного відношення процедури послідовної нормалізації. Нормалізація гарантує, що отримана модель не буде мати протиріччя і буде мати мінімальну збитковість. Атрибути в результаті нормалізації будуть згруповані відповідно до існуючих між ними логічних зв'язків. Для забезпечення коректності логічної моделі, у разі виявлення відношень, які не відповідають вимогам нормалізації, необхідно повернутися на попередні етапи проектування і перебудувати помилково створені елементи моделі.

Приклад. В результаті проектування отримано відношення показана на рис. 24.

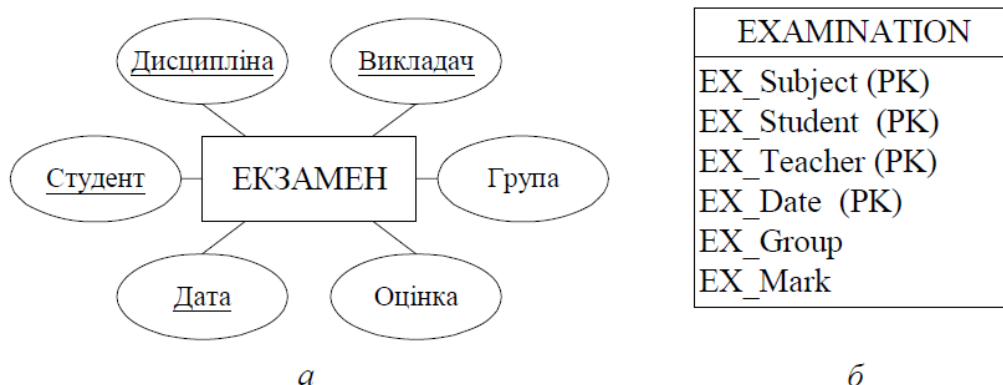


Рис. 24. Перетворення сутності *Екзамен* (а) у відношення *Examination* (б)

При дослідженні даного відношення були виявлені такі функціональні залежності:

Дисципліна, Викладач, Студент, Дата → *Оцінка Студент* → *Група*

У наведеній схемі існують аномалії і необхідно продовжити нормалізацію. В результаті декомпозиції вихідного відношення буде отримана схема показана на рис. 25.

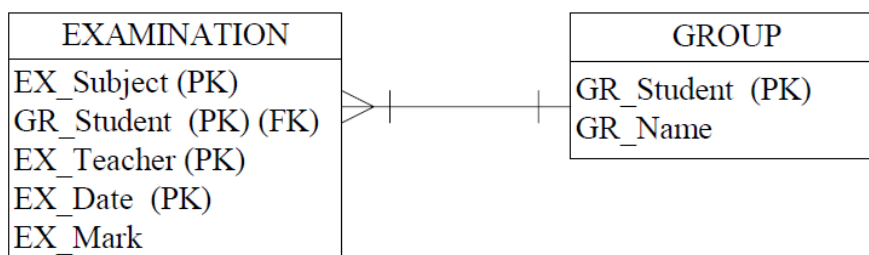


Рис. 25. Реляційна схема, яка відповідає сутності *Екзамен*

5. Перевірка відповідності відношень вимогам транзакцій користувачів

Перевірка полягає в нанесенні безпосередньо на ER-діаграму всіх шляхів, які потрібні для виконання кожної з транзакцій. Якщо таким чином вдається виконати всі транзакції, то перевірка на цьому завершується. У протилежному випадку необхідно повернутися до попередніх етапів і перевірити, а у разі потреби і змінити ті фрагменти моделі, які не відповідають необхідній роботі транзакцій.

Якщо в результаті перевірки будуть виявлені області, які не беруть безпосередньої участі у роботі транзакцій, то можливо їх видалення з моделі.

6.6. Перевірка підтримки цілісності

Обмеження цілісності запобігають появі в БД суперечливих даних. Вирішення цієї проблеми на стадії проектування полягає у такому:

- наявність обов'язкових і необов'язкових значень даних для атрибутів (NULL, NOT NULL);
- наявність обмежень для доменів атрибутів (визначення області значень або діапазону значень);
- цілісність сутностей (обов'язкова наявність Primary Key в кожному відношенні);

- посилкова цілісність (зв'язування таблиць за допомогою Foreign Key);
- обмеження предметної області (бізнес правила), які реалізуються як засобами БД, так і на рівні застосувань.

У табл. 1 наведені правила зовнішнього ключа для відношення "один до багатьох" для сильної сутності.

Таблиця 1

Підтримка посилкової цілісності для сильної сутності

Тип зв'язку	Вимоги до зовнішнього ключа
Обов'язкова наявність значень відповідних екземплярів у батьківській і залежній таблицях	NOT NULL ON DELETE RESTRICT ON UPDATE CASCADE
Необов'язкова наявність значень відповідних екземплярів у батьківській і залежній таблицях	NULL ALLOWED ON DELETE SET NULL ON UPDATE CASCADE
Обов'язкова наявність значень відповідних екземплярів у залежній таблиці і необов'язкова наявність значень в батьківській таблиці	NULL ALLOWED ON DELETE SET NULL ON DELETE RESTRICT ON UPDATE CASCADE
Обов'язкова наявність значень відповідних екземплярів у батьківській таблиці і необов'язкова наявність значень в залежній таблиці	NOT NULL ON DELETE RESTRICT ON UPDATE CASCADE

Для слабкої сутності використовуються ті ж самі правила за винятком обмежень на зовнішній ключ: NOT NULL, ON DELETE CASCADE, ON UPDATE CASCADE.

6. Приклад створення логічної моделі бази даних

У попередній лекції (*Концептуальне проектування баз даних п.4*) розроблено концептуальний проект бази даних для предметної області ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД (ВНЗ). ER-діаграма відображає всі бізнес правила, які в свою чергу визначають сутності, атрибути, зв'язки і т.д.

Наступним етапом проектування бази даних є створення логічної моделі бази даних на основі створеної ER-моделі (рис. 21). Створення логічної моделі бази даних виконується шляхом застосування правил перетворення ER-діаграми в логічну модель (рис. 26).

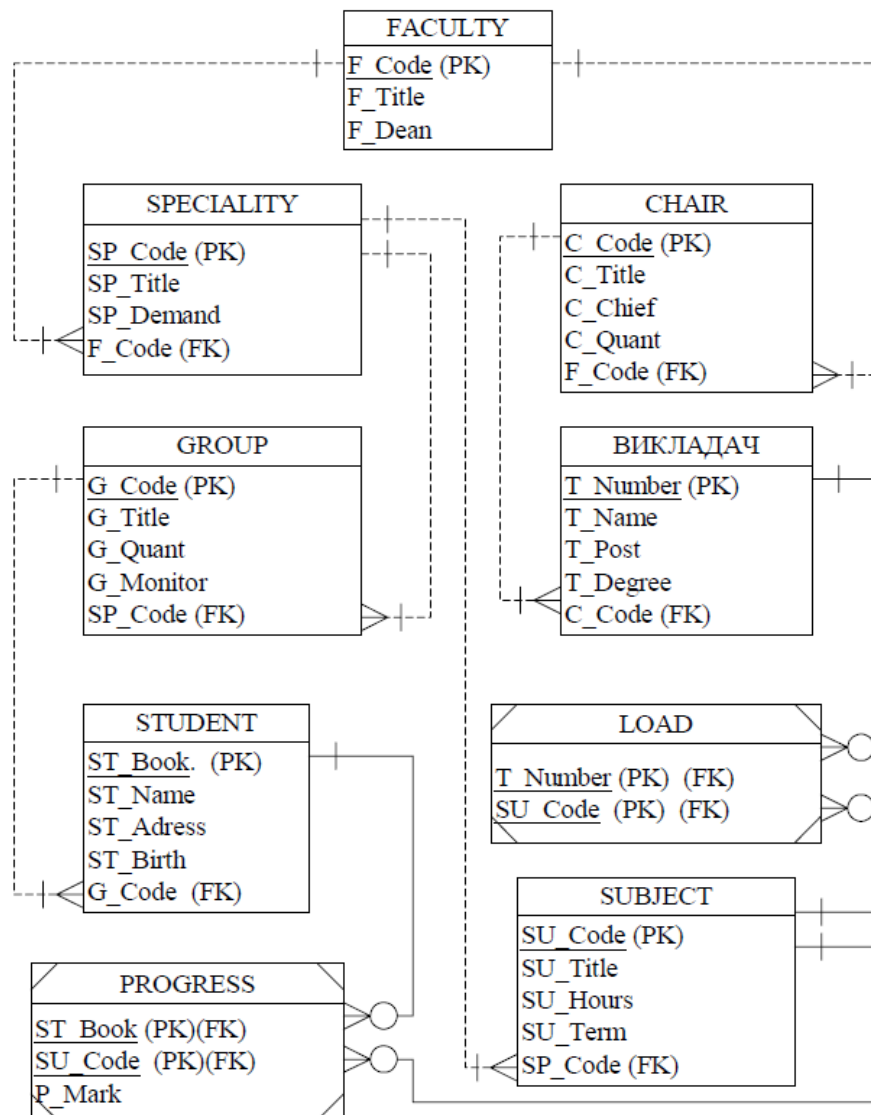


Рис. 26. Логічна модель бази даних для предметної області ВИЩІЙ НАВЧАЛЬНИЙ ЗАКЛАД

Після створення логічної моделі даних реляційна схема аналізується на коректність об'єднання атрибутів в одному відношенні. Перевірка коректності виконується шляхом застосування послідовної нормалізації до кожного з відношень. Метою цієї перевірки є отримання гарантій того, що схема бази даних щонайменше знаходиться в 3-й нормальній формі або в нормальній формі Бойса-Кодда. Якщо ця умова не виконується, то необхідно повернутися на попередні етапи проектування і перебудувати помилково створені

фрагменти моделі. Перевірка логічної моделі бази даних ВНЗ показує, що реляційна схема знаходиться в 4-й нормальній формі й корегування моделі не потрібно.

Після перевірки логічної моделі за допомогою правил нормалізації система аналізується на предмет виконання транзакцій користувачів, які задаються на початкових етапах проектування. У разі неможливості виконання певних транзакцій необхідне корегування моделі бази даних (див. рис. 1).

Подальша перевірка моделі вимагає перевірки підтримки цілісності даних. На основі матеріалів прикладу можна тільки визначити, що підтримується посилкова цілісність. Всі інші перевірки, включаючи і перевірку транзакцій користувачів, вимагають більш детально опрацьованого проекту бази даних.

Правила перетворення ER-діаграми в логічну модель наведені в е-матеріалі *«Правила перетворення ER-діаграми в логічну модель»* дистанційного курсу *«Проектування баз даних та знань для інформаційно-комунікаційних систем»* у віртуальному університеті ЛДУ БЖД.

Контрольні запитання

1. Що називається логічним проектуванням?
2. Яка інформація є вихідною для логічного проектування?
3. Перелічити етапи логічного проектування.
4. Які зв'язки ER-діаграм не підтримуються в реляційній схемі?
5. Як відображаються сильні і слабкі сутності та їх атрибути в реляційну схему?
6. Як відображається необов'язковість зв'язку між сутностями в реляційну схему?
7. Як відображається зв'язок "багато до багатьох" в реляційну схему?
8. Як відображаються складні зв'язки і зв'язки з атрибутами в реляційну схему?
9. Як відображаються рекурсивні зв'язки в реляційну схему?
10. Як відображається зв'язок "один до одного" залежно від рівня участі сутностей в реляційну схему?

11. Як відображається зв'язок "один до багатьох" залежно від рівня участі сутностей в реляційну схему?
12. Як відображається зв'язок суперклас–підклас залежно від ступеня участі сутностей і обмеження неперетинання в реляційну схему?
13. Навіщо потрібно виконувати перевірку реляційної схеми на відповідність правилам нормалізації?
14. Як виконується перевірка реляційної схеми на відповідність вимогам транзакцій користувачів?
15. Як перевіряється цілісність реляційної бази даних?
16. Навести приклади створення логічної моделі бази даних.

Розробив: к.т.н., доц. Полотай О.І.