**Практична робота №9** **Отримання схеми реляційної бази даних з діаграм класів**

**Мета:** поглиблене вивчення діаграм класів для отримання схеми реляційної бази даних**.**

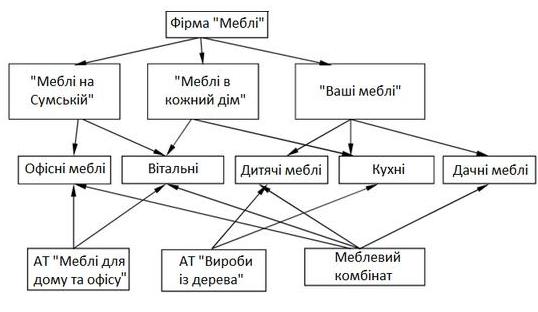
**На період** карантину в дистанційній формі навчання на надані запитання потрібно надати письмові відповіді, надіславши їх на електронну адресу викладача. Файл надавати з іменем у форматі

**PI<Номер групи><Номер лекції / практичної / лабораторної>[-<Номер завдання>][літера позначення типу роботи L – лекція, P –практична, R – лабораторна]<Прізвищеанглійською>**. Наприклад, **PI3104L**buts.doc. Відповіді на запитання повинні бути не довгими і змістовними. Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності відповідей-"близнюків" відповідь буде зараховуватися першому за часом надсилання.

**Строк виконання цієї роботи ІПЗ-31, ІПЗ-32 11.05.2020**

**Завдання:**

1. Опрацювати теоретичні відомості. Перевірити засвоєння Вами матеріалу на контрольних запитаннях.
2. Розробити діаграму класів для створення реляційної бази за наданою нижче інформаційною схемою, де представлений зв’язок фірми з філіалами (магазинами), для яких визначені типи товарів та постачальники цих товарів. Мінімальна кількість атрибутів: код, назва, адреса (для підприємств), ціна за комплект (для товарів). Роботу виконати в Umbrello UML Modeller (в разі відсутності роботу виконувати з іншим доступним засобом UML-моделювання)



1. До діаграми додати нотатки, де записати своє прізвище та групу, а також номер ЛР.
2. Для відсилки викладачу збережіть проект з іменем Вашої ПР та розширенням xmi.
3. Оформите звіт, в якому надайте скріншоти створеної діаграми. Файл зі звітом викласти на платформу коледжу до репозиторію викладача та копію відіслати на електронну адресу викладача.

**Контрольні запитання**

1. Які мовні конструкції і в якому вигляді використовуються в UML?
2. Які вирази кратності (множинності) недоцільно використовувати в діаграмі класів для представлення реляційної БД?
3. Які обмеження цілісності можуть зазначатися у діаграмах класів?

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

Реляційна модель даних у переважній більшості випадків цілком достатня для моделювання будь-яких даних. Однак проектування бази даних в термінах схеми відносин на практиці може викликати великі труднощі, тому що в цій моделі спочатку не передбачені механізми опису семантики предметної області. З цим пов'язана поява семантичних моделей даних, які дозволяють описати конкретну предметну область набагато ближче до інтуїтивного розуміння і, в той же час, достатньо формальним чином.

**Семантична модель даних** є абстрактною схемою, яка показує, як збережені символи співвідносяться с реальним світом. це інформаційна модель найвищого рівня абстракції. Така модель створюється без орієнтації на будь-яку конкретну СУБД та модель даних.

Історія систем автоматизації проектування баз даних (CASE-засобів) починалася з автоматизації процесу малювання діаграм, перевірки їх формальної коректності, забезпечення засобів довготривалого зберігання діаграм та іншої проектної документації. Звичайно, комп'ютерна підтримка роботи з діаграмами дуже корисна для проектувальника БД. Наявність електронного архіву проектної документації допомагає при експлуатації, адмініструванні та супроводі бази даних. Але система, яка обмежується підтримкою малювання діаграм, перевіркою їх коректності та зберіганням, нагадує текстовий редактор, що підтримує введення, редагування і перевірку синтаксичної коректності конструкцій деякої мови програмування, але існуючий окремо від компілятора. Переважна більшість подібних систем, представлених на ринку, забезпечує автоматизоване перетворення діаграмних концептуальних схем баз даних, представлених у тій чи іншій семантичної моделі даних, в реляційні схеми, представлені найчастіше мовою SQL. Автоматично таке перетворення робити недоцільно, оскільки в типовій схемі SQL-орієнтованої БД можуть міститися визначення багатьох об'єктів (обмежень цілісності загального вигляду, тригерів і збережених процедур і т.д.), які неможливо згенерувати автоматично на основі концептуальної схеми. Тому на завершальному етапі проектування реляційної схеми знову потрібно ручна робота проектувальника.

Існує багато різних підходів до семантичного моделювання баз даних. В останні 10 років одним з найбільш популярних мов семантичного моделювання є UML, і діаграми класів успішно застосовуються саме для таких цілей.

Мовний механізм діаграм класів за своєю суттю не дуже відрізняється від істотно раніше впровадженого в практику мовного механізму ER-діаграм . Тим не менш, проектування реляційних баз даних у середовищі UML дає одна суттєва перевага: можна виконати весь проект створення інформаційної системи на основі одного загального інструменту, але це не виключає трансформування діаграми класів до ER-діаграм. З погляду на спілкування із замовником програмної системи для з’ясування та уточнення вимог до системи доцільно подавати всі вимоги із застосуванням діаграм, виконаних за єдиною формальною мовою.

**Модель «сутність-зв'язок» (ER-модель)** (англ. *Entity-relationship model або entity-relationship diagram*) — модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. В ER-моделі дані представлені у вигляді компонентів (сутностей), які пов'язані між собою певними зв'язками, які виражають залежності і вимоги між ними. Сутності можуть мати різні властивості (атрибути), які характеризують їх. ER-модель зазвичай реалізується в вигляді баз даних. У разі реляційної бази даних, в якій зберігаються дані в таблицях, кожен рядок кожної таблиці являє собою один екземпляр сутності. Деякі поля даних в цих таблицях вказують на індекси в інших таблицях. Такі поля є покажчиками фізичної реалізації зв'язків між сутностями.

UML розроблено і розвивається консорціумом OMG (Object Management Group) і має багато спільного з об'єктними моделями, на яких заснована технологія розподілених об'єктних систем CORBA, і об'єктною моделлю ODMG (Object Data Management Group).

Мова UML активно застосовується для проектування реляційних БД за допомогою діаграм класів. У межах термінології UML діаграмою класів називається діаграма, на якій показаний набір класів, зв’язків між ними, а також може містити коментарі та обмеження. Приклад діаграми класів, розроблений у середовищі CASE-системи Enterprise Architect, поданий на рис. 1.

Рис. 1. Діаграма класів структури реляційної бази

Подана діаграма містить два об’єкта «Специальность» та «Студент», між якими існує зв’язок (асоціація) за принципом «головний – підпорядкований». При побудові даної діаграми використовувалися такі об’єкти: клас (таблиця), атрибут (поле, яке має ім’я та тип даних, а також може мати значення null, атрибути unique та primary key), зв’язки між класами (зв’язки між таблицями з використанням первинного та зовнішнього ключа, також встановлюється тип зв’язку між таблицями, що у термінах UML називається кратністю), обмеження для забезпечення структурної та посилальної цілісності даних. Розглянемо визначення та вищезазначені поняття для визначення їх тотожності.

Під класом у мові UML розуміється іменований опис сукупності об’єктів, що володіють однаковою структурою, властивостями та відносинами з іншими об’єктами. У кожного класу є ім’я, що унікально відрізняється від імен інших класів. Атрибутом класу називається іменована властивість, що описує безліч значень, які можуть приймати екземпляри цієї властивості. Клас може мати будь-яке число атрибутів або не мати жодного. Властивість, яка виражається через атрибут, є властивістю модельованої сутності, загальною для всіх об’єктів даного класу. Імена атрибутів розташовуються у розділі класу, розташованому під ім’ям класу. У діаграмі класів можуть брати участь зв’язки трьох категорій: залежність (dependency), узагальнення (generalization) та асоціація (association).

При моделюванні структури реляційних баз даних найбільш важлива тільки третя категорія зв’язків (асоціація), тому зупинимося на ній докладніше.

Асоціацією називається структурний зв’язок, який показує, що об’єкти одного класу певним чином пов’язані з об’єктами іншого або того ж самого класу. Допускається створення асоціацій, що пов’язують відразу n класів (вони називаються n-арними асоціаціями). Графічно асоціація зображується у вигляді лінії, що з’єднує клас сам із собою або з іншими класами.З поняттям асоціації пов’язано поняття кратності (множинності) ролі (multiplicity). Це характеристика, яка вказує, скільки об’єктів класу з даною роллю можуть брати участь у кожному примірнику асоціації. Найбільш поширеним способом завдання кратності ролі асоціації є вказівка конкретного числа або діапазону. Наприклад, вказівка діапазону «1 .. \*» повідомляє про те, що всі об’єкти класу з даною роллю повинні брати участь у деякому екземплярі даної асоціації, і в кожному примірнику асоціації повинен брати участь хоча б один об’єкт, при цьому верхня межа не задана (\*).

У діаграмах класів можуть зазначатися обмеження цілісності, які повинні підтримуватися у модельованій базі даних. В UML допускаються два способи визначення обмежень: неформально на природній мові або формально на мові об’єктних обмежень OCL (Object Constraints Language). Мова OCL призначена, головним чином, для визначення обмежень цілісності даних, відповідних моделі. З точки зору забезпечення цілісності бази даних найбільш важливі засоби визначення інваріантів класів.

Отже, мова UML, яка має майже універсальні можливості для моделюванні різних об’єктів, процесів, інформаційних систем та іншого програмного забезпечення, може застосовуватися для проектування реляційних баз даних через використання діаграми класів. З точки зору проектування реляційних БД, можливості діаграми класів не надто відрізняються від можливостей ER-діаграм, а їх зовнішній вигляд майже однаковий.

Перехід від діаграмного подання концептуальної схеми бази даних до її реляційної схемі не залежить від різновиду використовуваних діаграм. Зокрема, методика, розроблена для класичних діаграм «Сутність-Зв'язок» (Entity-Relationship), практично завжди придатна для діаграм класів UML. Ця методика широко відома, її вивчають разом із методикою проектування баз даних. Тим не менш, видається доцільним навести кілька рекомендацій, які тісно пов'язані зі специфікою діаграм класів.

**Рекомендація 1.** Перш, ніж визначати в класах операції, подумайте, що можна зробити з цими визначеннями в середовищі цільової реляційної СУБД (РСУБД). Якщо в цьому середовищі підтримуються збережені процедури, то, можливо, деякі операції можуть бути реалізовані саме за допомогою такого механізму. Але якщо в середовищі РСУБД підтримується механізм визначаються користувачами функцій, він може виявитися більш підходящим.

**Рекомендація 2.** Пам'ятайте, що порівняно ефективно в РСУБД реалізуються тільки асоціації видів "один-до-багатьох" і "багато-до-багатьох". Якщо в створеній діаграмі класів є асоціації "один-до-одного", то слід замислитися про доцільність такого проектного рішення. Реалізація в середовищі РСУБД асоціацій з точно заданими кратностямі (множинностями) ролей можлива, але вимагає визначення додаткових тригерів, виконання яких знизить ефективність.

**Рекомендація 3.** Для технології реляційних БД агрегатні і особливо композитні асоціації є неприродними. Подумайте про те, що ви хочете отримати в реляційної БД, оголосивши деяку асоціацію агрегатною. Швидше за все, нічого.

**Рекомендація 4.** У специфікації UML йдеться, що, визначаючи односпрямовані зв'язку, ви можете сприяти ефективності доступу до деяких об'єктів. Для технології реляційних баз даних підтримка такого оголошення викличе додаткові накладні витрати і тим самим знизить ефективність.

**Рекомендація 5.** Не зловживайте можливостями об'єктної мови обмежень OCL, яка використовується в UML для формалізації звичайних мовних конструкцій, які розглядалися в ході вивчення UML. Діаграми класів UML – це хороший і потужний інструмент для створення концептуальних схем баз даних, але, це не виключає, а можливо і не може замінити .

Не можна сказати, що проектування баз даних на основі семантичних моделей у будь-якому випадку прискорює і / або спрощує процес проектування. Все залежить від складності предметної області, кваліфікації проектувальника та якості допоміжних програмних засобів. Але в будь-якому випадку етап діаграмного моделювання забезпечує наступні переваги:

* на ранньому етапі проектування до прив'язки до конкретної РСУБД проектувальник може виявити і виправити логічні огріхи проекту, керуючись наочним графічним поданням концептуальної схеми;
* остаточний вигляд концептуальної схеми, отриманої безпосередньо перед переходом до формування реляційної схеми, а може бути, і проміжні версії концептуальної схеми повинні стати частина документації цільової реляційної БД; наявність цієї документації дуже корисна для супроводу і особливо для зміни схеми БД в зв'язку з зміненими вимогами;
* при використанні CASE-засобів концептуальне моделювання БД може стати частиною всього процесу проектування цільової інформаційної системи, що може сприяти правильній структуризації процесу, ефективності та підвищення якості проекту в цілому.

В контексті проектування реляційних БД структурні методи проектування, засновані на використанні ER-діаграм, та об'єктно-орієнтовані методи, засновані на використанні мови UML, розрізняються, головним чином, лише термінологією. ER-модель концептуально простіше UML, в ній менше понять, термінів, варіантів використання. І це зрозуміло, оскільки різні варіанти ER-моделей розроблялися саме для підтримки проектування реляційних БД, і ER-моделі майже не містять можливостей, що виходять за межі реальних потреб проектувальника реляційної БД.

Мова UML належить об'єктному світу. Цей світ набагато складніше реляційного світу. Оскільки UML може використовуватися для уніфікованого об'єктно-орієнтованого моделювання всього, що завгодно, в ньому міститься маса різних понять, термінів і варіантів використання, абсолютно надлишкових з точки зору проектування реляційних БД. Якщо виокремити з загального механізму діаграм класів те, що дійсно потрібно для проектування реляційних БД, то ми отримаємо в точності ER-діаграми з іншого нотацією і термінологією.

**Рекомендована література**

1. Чен П.П. Модель "сутність-зв'язок" – крок до єдиного поданням даних. СУБД, N3, 1995 р.
2. Вендров А.М. CASE-технології. Сучасні методи і засоби проектування інформаційних систем. М., Фінанси і статистика, 1998.
3. Вендров А.М. Проектування програмного забезпечення економічних інформаційних систем. М., Фінанси і статистика, 2000.
4. Фаулер М., Скотт К. UML у короткому викладі. Застосування стандартного мови об'єктного моделювання. М., Мир, 1999.
5. Буч Г. Об'єктно-орієнтований аналіз та проектування з прикладами додатків на C + +. 2-е вид. М., Видавництво Біном, СПб., Невський діалект, 1999.
6. Буч Г., Рамбо Д., Джекобсон А. Мова UML: керівництво користувача. М., ДМК, 2000