

АНАЛІЗ ПІДХОДІВ ДО ВИРІШЕННЯ ПРОБЛЕМ ВІДОБРАЖЕННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОЇ МОДЕЛІ В РЕЛЯЦІЙНУ СТРУКТУРУ

Савчук Тамара, Давидов Сергій

Вінницький національний технічний університет

Анотація

За результатами проведеного аналізу було виявлено основні проблеми відображення об'єктно-орієнтованої моделі в реляційну структуру та систематизовано підходи до їх вирішення.

Abstract

According to the results of the analysis, the main problems of mapping the object-oriented model to the relational structure were identified and the approaches to their solution were systematized.

Вступ

Ключовим сутностями об'єктно-орієнтованої моделі є класи, які описують атрибути об'єктів (екземплярів), і міжкласові асоціації, що визначають можливість зв'язку між конкретними екземплярами. Саме ці сутності розглядаються як ключові, оскільки реляційна база даних, в якій об'єкти будуть зберігатись, не здатна природним чином підтримувати їх поведінку. Тобто зберігаються атрибути та зв'язки, а не методи примірників і не повідомлення, якими вони обмінюються. Таким чином, актуальним є проведення аналізу підходів до вирішення проблем відображення об'єктно-орієнтованої моделі в реляційну структуру [1].

При збереженні об'єктно-орієнтованої моделі в реляційну структуру виникає конфлікт між двома рівнями системи бази даних, оскільки, незважаючи на деяку схожість концепцій об'єктно-орієнтованої і реляційної методологій, система не здатна природним для себе шляхом звертатися до бази даних [1].

Для коректного відображення об'єктно-орієнтованої моделі на ієрархічну структуру потрібно визначитись з рішенням проблем об'єктно-реляційного відображення [2]:

- 1) ідентифікація об'єктів;
- 2) збереження окремого атрибута;
- 3) збереження масивів;
- 4) збереження посилань і зв'язків між об'єктами;
- 5) відображення класу;
- 6) відображення дерева успадкування;

Однозначно ідентифікувати об'єкт в додатку і його дані в реляційній таблиці є ключовим моментом в організації роботи з об'єктно-орієнтованою моделлю даних. Розрізняються два види ідентифікаторів: Об'єктні Ідентифікатори в прикладній програмі (OID) і Унікальні Об'єктні Ідентифікатори Збережених Об'єктів у базах даних (БД) (POID). POID грає роль не тільки ключа в таблиці даного класу, але він також використовується в якості посилання на ці об'єкти з боку інших класів у відповідних таблицях. Найкращим загальноновизнаним підходом ідентифікації об'єктів в реляційну структуру є організація POID у вигляді пари цілих чисел – ідентифікатора класу об'єкта і ідентифікатора примірника самого об'єкта в рамках даного класу. Даний підхід дозволяє,

окрім вирішення основної задачі, по унікальній ідентифікації об'єкта вирішувати ще одну важливу задачу – визначати клас об'єкту [3].

Базовою концепцією у рішенні проблеми збереження окремого атрибуту є збереження атрибуту класу у стовбцях реляційних таблиць. Для збереження значень атрибутів конкретного класу необхідно забезпечити доступ до значень цих атрибутів. При інкапсуляції об'єкта існує два загальних методи доступу: прямий доступ до атрибутів та доступ за допомогою спеціальних методів. При прямому доступі для зберігання даних певного атрибута класу використовується певний стовпець реляційної таблиці, при цьому не потрібно ніяких перетворень і дані передаються безпосередніми викликами відповідних методів [4], що є важливим при збереженні окремого атрибуту з об'єктно-орієнтованої моделі у реляційну структуру.

Для вирішення проблеми збереження масивів доцільно використовувати спосіб перетворення в двійковий потік. Дане перетворення здійснюється з використанням механізму перетворення даних об'єкта в бінарний потік. Переведення в двійковий потік неможливо, якщо внутрішні дані масиву використовуються для запитів до бази даних. Рішенням для зберігання внутрішніх елементів масиву може служити утворення окремих таблиць для масивів кожного з елементарних типів. Якщо в запитах до БД не використовується навігація по конкретних елементах, цілком доцільно використовувати спосіб перетворення в двійковий потік [5] при відображенні масивів у реляційній структурі.

При відображенні об'єктних посилань і зв'язків виникають проблеми їх збереження у реляційній структурі. Сама парадигма об'єктно-орієнтованої бази даних забезпечує навігацію по внутрішній структурі об'єктів, тому перетворення у двійковий потік не підходить для збереження двікових посилань і зв'язків. Для зв'язків потрібно забезпечити двосторонню здатність до навігації. Ключовим моментом у відображенні об'єктів на реляційні таблиці є представлення посилання на будь-який об'єкт у вигляді пари цілих чисел. Таким чином, атрибут, який є односторонньою посиланням, представляється в реляційних системах управління БД (РСУБД) двома стовпчиками, в які записуються відповідні ідентифікатори потрібного об'єкту-мети. Загальним правилом для відображення зв'язків між об'єктами в реляційній базі даних є опис реляційних зовнішніх ключів (foreign key) для забезпечення цілісності даних [6].

Найпростішим способом організації відображення класів у реляційні таблиці є відображення (TABLE mapping). При цьому, атрибути одного класу зберігаються в окремій таблиці, де відсутні зайві атрибути, що не відносяться до класу, який розглядається. Іншим способом відображення класу є метод підмножини (SUBSET mapping), коли для зберігання даних певного класу в реляційній таблиці використовується тільки частина полів. Серед способів організації відображення класів у реляційній таблиці виділяють метод надмножини (SUPERSET mapping), що передбачає збереження тільки частини необхідної для даного класу інформації в окремій таблиці СУБД, тобто збереження даних класу в кількох таблицях [7].

При відображенні дерева успадкування існує проблема визначення стратегій таблиць для збереження атрибутів кожного екземпляру класу. Для вирішення означеної проблеми використовується три варіанти стратегії відображення в РСУБД ієрархії успадкування класів:

1) Відображення з фільтрацією, що полягає в підтримці однієї глобальної таблиці для кожної ієрархії класів. Спочатку в таблицю потрапляють всі атрибути кореневого класу, а потім для кожного нового спадкоємця в таблицю додаються специфічні для цього спадкоємця атрибути.

2) Горизонтальне відображення, яке розбиває глобальну таблицю ієрархії на кілька таблиць. При цьому, в кожній такій таблиці будуть знаходитися всі атрибути конкретного

класу, в тому числі і успадковані.

3) Вертикальне відображення, що відображує ієрархію спадкування найбільш природним чином. Для кожного класу визначається окрема таблиця, в якій зберігаються тільки специфічні для даного класу атрибути, а між таблицями в ієрархії визначаються зовнішні ключі, за якими в подальшому забезпечується їх зв'язок (join) для маніпулювання даними [8].

Таким чином, було виявлено основні проблеми відображення об'єктно-орієнтованої моделі в реляційну структуру та систематизовано підходи до їх вирішення.

Список використаних джерел:

1. Проблема отображения объектной модели в хранимую реляционную структуру. [Электронный ресурс]. – Режим доступа: <http://www.ict.nsc.ru/ws/YM2002/4752/>
2. Способы отображения объектов в реляционных базах данных. [Электронный ресурс]. – Режим доступа: <https://studfiles.net/preview/895257/>
3. Khoshafian S. Object Identity // Copeland G. OOPSLA Conference, 1986.
4. Khoshafian, Franklin, Carey. Storage management for persistent complex objects / Information Systems, vol.15, no.3, p. 303-320, 1990.
5. Kent W. A rigorous model of object reference, identity and existence / JOOP June, 1991.
6. Martin Rennhackkamp. Extending Relational Databases / DBMS Magazine, December 1997
7. К.В. Антипин, В.В. Рубанов. Проблемы организации объектно-ориентированного доступа к реляционным базам данных / Труды Института Системного Программирования, вып. 2. 2000 г.
8. Khoshafian, Franklin, Carey. Storage management for persistent complex objects / Information Systems, vol.15, no.3, p. 303-320, 1990.