

УДК 371.13.001.76

Мельник В.М. к.ф.-м.н. доц., Клімчук Н.О.

Луцький Національний технічний університет

ОСНОВНІ КОНЦЕПЦІЇ ОБ'ЄКТНО-ОРІЄНТОВАНИХ БАЗ ДАНИХ ТА ЇХ ПРОБЛЕМАТИКА

Мельник В.М., Клімчук Н.О. Основні концепції об'єктно-орієнтованих баз даних та їх проблематика. У статті розкрито основні вимоги, яким повинна задовольняти база даних, щоб бути кваліфікованою як об'єктно-орієнтована база даних. Було розглянуто проблематику та протиріччя, що виникають на шляху узгодження специфікації об'єктно-орієнтованої моделі та пошуку єдиної системи керування об'єктно-орієнтованою базою даних. Також проаналізовано існуючі вимоги до об'єктно-орієнтованих баз даних.

Ключові слова: властивість, клас, специфікація моделі даних, системи керування, об'єктно-орієнтовані бази даних.

Мельник В.М., Клімчук Н.А. Основные концепции объектно-ориентированных баз данных и их проблематика. В статье раскрыты основные требования, которым должна удовлетворять база данных, чтобы быть квалифицированной как объектно-ориентированная база данных. Была рассмотрена проблематика и противоречия, возникающие на пути согласования спецификации объектно-ориентированной модели и поиска единой системы управления объектно-ориентированной базой данных. Также проанализированы существующие требования к объектно-ориентированным базам данных.

Ключевые слова: свойство, класс, спецификация модели данных, системы управления, объектно-ориентированные базы данных.

Melnyk V., Klimchuk N. Basic concepts of object-oriented databases and their problems. The article shows the basic requirements that must satisfy the database to be classified as an object-oriented database. We consider problems and conflicts that arise in the way of agreeing the specification of object-oriented model and the search for a unified system of management of object-oriented database. Also, we analyzed existing requirements for object-oriented databases.

Keywords: property, class, specification data model, management systems, object-oriented databases.

Постановка проблеми. Об'єктно-орієнтовані бази даних (ООБД) – це нове, третє, покоління баз даних, що привертає все більшу увагу як дослідників і розробників, так і потенційних користувачів з прикладних областей. Це зумовлено тим, що багато інформаційних систем вже «доросли» до такого рівня, що вимагають використання відповідних баз даних, які зможуть справлятися зі складно структурованими даними.

Системи другого покоління – реляційні бази даних, не можуть задовольнити цієї потреби. Для прикладу візьмемо видавничу систему за допомогою якої клієнт може створити макет газети, а потім роздрукувати її. Для роботи такого додатку потрібно зберігати фрагменти тексту, графіку, піктограми і багато інших типів елементів даних, що зустрічаються у більшій частині гіпертекстових середовищ. Підтримка таких елементів даних системами другого покоління зазвичай пов'язана з немалими труднощами. [8]

Напрямок ООБД виник порівняно давно. Публікації з'являлися вже в середині 1980-х років. Однак найбільш активно цей напрямок розвивається в останні роки [4, с.194]. З одного боку це пояснюється розвитком і впровадженням в практику об'єктно-орієнтованого підходу в цілому. Але з іншого боку, інтуїтивно ясно, що максимальний ефект можна отримати саме від використання об'єктно-орієнтованих баз даних. Утім, на даний час не сформована єдина думка про те, що собою являє об'єктно-орієнтована система, не кажучи уже про об'єктно-орієнтовані бази даних. Немає загальної згоди відносно якоїсь одної існуючих моделей даних, відсутня теоретична база.

Як зазначають багато дослідників, своїм успіхом реляційна модель даних багато в чому зобов'язана тому, що спиралася на строгий математичний апарат теорії множин, відносин і логіки першого порядку [4, с.197]. Основні труднощі об'єктно-орієнтованого моделювання даних виникають через те, що такого розвиненого математичного апарату, на який могла б спиратися загальна об'єктно-орієнтована модель даних, не існує. Великою мірою і досі несформована базова об'єктно-орієнтована модель. З іншого боку, в роботах Майєра стверджується, що загальна об'єктно-орієнтована модель даних в класичному сенсі і не може бути визначена з причини непридатності класичного поняття моделі даних до парадигми об'єктної орієнтованості [4, с.198].

За допомогою об'єктно-орієнтованих баз даних вирішуються дві проблеми. По-перше, складні інформаційні структури виражаються в них краще, ніж у реляційних базах даних, а по-друге, усувається необхідність транслювати дані з того формату, що підтримується в системах керування базами даних (СКБД). Наприклад, у реляційній СКБД розмірність цілих чисел може становити 11 цифр, а у використовуваній мові програмування – 16. Програмістові прийдеється враховувати цю ситуацію [1].

Проте суттєвим недоліком об'єктно-орієнтованих баз даних є їхні тісні зв'язки із застосовуваною мовою програмування. До даних, що зберігаються в реляційній СКБД, можуть звертатися будь-які додатки, наприклад, Java-об'єкт, поміщений в об'єктно-орієнтовану базу даних, буде становити інтерес лише для додатків, написаних на Java [1].

До теперішнього моменту нам невідома будь-яка мова програмування ООБД, яка була б спроектована цілком заново, починаючи з нуля. Природним підходом до побудови такої мови було використання (з необхідними розширеннями) деякої існуючої об'єктно-орієнтованої мови. Практично усі розробники усвідомлюють, що існує потреба підтримки не тільки мови програмування ООБД, а й розвиненої мови запитів. Система повинна підтримувати легко освоюваний інтерфейс, прямо доступний кінцевому користувачеві в інтерактивному режимі [4, с. 202].

Аналіз досліджень. Проаналізувавши літературу на тему баз даних, в розділах з описами моделей даних – ієрархічна, реляційна, об'єктно-орієнтована моделі даних, останній приділяється найменше уваги. Це наслідок того, що відсутня теоретична база, не говорячи вже про єдину специфікацію об'єктно-орієнтованої моделі даних.

У період з 1989 по 1995 рр. авторські групи, включаючи відомих фахівців в області баз даних, підготували і опублікували три документи, які відображали точки зору авторів щодо перспектив розвитку технології баз даних. З легкої руки авторів ці документи отримали назву маніфестів, що, загалом, відображало їх суть: у кожному з документів проголошувався набір ідей і вимог, на яких, на думку авторів, повинні бути базуватися системи баз даних наступного покоління. Саме ці три документи – Маніфести, було взято за теоретичну базу для написання даної статті.

Цікаво відзначити відмінності між колективами авторів кожного з маніфестів. «Маніфест систем об'єктно-орієнтованих баз даних» (Перший маніфест) написаний академічними дослідниками; майже всі вони є професорами різних університетів. Стиль Першого маніфесту – дуже м'який і помірно рекомендаційний.

Автори документу «Системи баз даних третього покоління: Маніфест» (Другого маніфесту) були представниками індустрії. Другий маніфест написаний в більш жорсткому стилі і багато в чому спрямований на захист інвестицій великих компаній-виробників програмного забезпечення SQL-орієнтованих СУБД. Звичайно, Другий маніфест багато в чому являв собою реакцію індустрії на революційні пропозиції Першого маніфесту.

«Третій маніфест» був одночасно найбільш консервативним і найбільш радикальним. Консервативність Третього маніфесту полягає в тому, що його автори усіма силами стверджують необхідність і достатність використання в системах баз даних наступного покоління класичної реляційної моделі даних. Радикальність полягає в тому, що автори повністю заперечують підходи, пропоновані в перших двох маніфестах, розцінюючи їх як необґрунтовані, погано опрацьовані, надлишкові і навіть шкідливі (за винятком однієї спільної ідеї про потреби забезпечення розвиненої системи типів); фактично, автори повністю відкидають технологію, створену індустрією баз даних за останніх 25 років, і пропонують повернутися до витоків реляційної моделі даних, тобто до початкових статей Е. Кодда [2].

В 1991 році була сформована група ODMG – Object Database Management Group (група управління об'єктними базами даних). Дана група об'єднувала розробників об'єктних СУБД та інших зацікавлених організацій для розробки стандарту об'єктно-орієнтованих СУБД. В 1993 році був опублікований стандарт ODMG-1. Остання версія стандарту була випущена в 1999 році – ODMG-3. Архітектура об'єктної моделі в стандарті ODMG-3 передбачала наявність єдиного сховища даних, до якого користувач може здійснювати різні види доступу. Сховище даних доступно з мови визначення даних, мови запитів і ряду мов маніпулювання даними.

Центральною в архітектурі є модель даних, що представляє організаційну структуру, в якій зберігаються всі дані, керовані ООСУБД. Мова визначення об'єктів, мова запитів і мови маніпулювання розроблені таким чином, що всі їхні можливості спираються на модель даних. Архітектура допускає існування різноманітних реалізаційних структур для зберігання модельованих даних, але важливою вимогою є те, що всі програмні бібліотеки і всі інструментальні засоби забезпечуються в об'єктно-орієнтованих рамках і повинні зберігатися в узгодженні з даними.

Проте, об'єктна модель, що представлена в стандарті ODMG-3 має ряд недоліків, зокрема: об'єктна модель недостатньо обґрунтована. Виникає відчуття, що в цю модель прагнули помістити властивості реалізаційних моделей всіх комерційних ООСУБД, що існували до моменту написання стандарту; поділ мов визначення даних, об'єктних запитів і мов маніпулювання, звичайно, відображає прагнення розробників стандарту до ортогоналізації моделі, але дуже ускладнює її цілісне розуміння; мова об'єктних запитів є простою з концептуальної точки зору, але дуже складною для використання початківцями.

На ринку ООСУБД присутні три заслужені системи, кожна з яких існує близько 20 років, – Versant компанії Versant Corporation, ObjectStore, що належить з 2003 р. компанії Progress Software, і Objectivity/DB компанії Objectivity Inc. У всіх цих системах підтримується об'єктно-орієнтована модель даних ODMG-3, але архітектури систем значно різняться.

На сьогоднішній день ведуться багаточисленні експериментальні розробки: люди реально будують системи. Деякі з них являються тільки прототипами, а деякі – вже стали комерційними продуктами. З кожним роком збільшується число публікацій і реалізованих комерційних та експериментальних систем. Та це ніяк не впливає на прогресування процесу формування єдиної моделі об'єктно-орієнтованих баз даних.

На даний момент, коли теперішнє суспільство технічно ще не готове перейти до використання об'єктно-орієнтованих баз даних та середовищ їх управління, потрібно реально оцінити та проаналізувати можливості існуючих СУБД. Можливо, це зможе сприяти подальшій еволюції реляційних баз даних до нового, третього покоління – об'єктних баз даних.

Виходячи з цього, **мета** статті полягає у аналізі вимог та властивостей, які представлені у трьох Маніфестах, володіючи якими база даних та середовище керування базами даних вважатимуться об'єктно-орієнтованими. На основі визначених властивостей і вимог у даній статті, буде проводитись дослідження існуючих баз даних та їх середовищ створення на предмет об'єктної орієнтованості.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Об'єктно-орієнтовані системи беруть свій початок з об'єктно-орієнтованих мов програмування. Основна ідея, що об'єднує ці дві області, полягає в тому, щоб відгородити користувача від конструкцій, що пов'язані з апаратним забезпеченням, таких як біти і байти. Замість цього, користувач має справу з об'єктами і операціями над цими об'єктами, які більше відповідають своїм аналогам в реальному світі. [6, с.944]

Посилаючись на вміст Першого Маніфесту, властивості і вимоги, якими повинна володіти об'єктно-орієнтована база даних, можна розділити на три категорії:

1. **обов'язкові** («золоті» правила) (властивості, що мають забезпечувати у системі об'єктно-орієнтованість);
2. **необов'язкові** (ті, які будуть покращувати систему);
3. **відкриті** (можливість вибору проектувальником рішення певного завдання різними способами).

Таблиця 1 – Категорії властивостей та характеристик ООБД, запропоновані у «Маніфесті систем об'єктно-орієнтованих баз даних»

Обов'язкові можливості («золоті» правила)	
1.	Система повинна бути СУБД:
1.1	стабільність;
1.2	управління вторинною пам'яттю;
1.3	паралелізм;
1.4	відновлюваність;

1.5 засоби забезпечення незапланованих запитів.
2. Система повинна бути об'єктно-орієнтованою, використовуючи при цьому сучасні об'єктно-орієнтовані мови програмування: <ul style="list-style-type: none">2.1 складні об'єкти;2.2 ідентифікація об'єктів;2.3 інкапсуляція;2.4 типи чи класи;2.5 успадкування;2.6 перекриття методів спільно з пізнім зв'язуванням;2.7 розширюваність;2.8 обчислювальна повнота.

Продовження таблиці 1

Необов'язкові можливості
1. множинне успадкування 2. перевірка і виведення типів 3. розподілюваність 4. проектні транзакції 5. версії
Відкриті можливості
1. парадигма програмування 2. системне представлення 3. система типів 4. однорідність

Система об'єктно-орієнтованих баз даних повинна задовольняти двом критеріям: повинна бути СУБД і при цьому являться об'єктно-орієнтованою системою, на максимально можливому ступені знаходитись на рівні сучасних об'єктно-орієнтованих мов програмування. Реляційні системи баз даних не задовольняють правилам з пункту 2.1 по пункт 2.8. [7]

У другому Маніфесті представлено три основні принципи, якими потрібно керуватися при створенні систем третього покоління.

Принцип 1. Крім традиційних послуг по управлінню даними, СУБД третього покоління забезпечать підтримку більш «багатих» структур об'єктів і правил. Більш «багата» структура об'єктів характеризує засоби, що необхідні для збереження і маніпулювання нетрадиційними елементами даних – тексти, зображення, просторові дані. Творцям додатків потрібно забезпечити можливість задавати групу правил, що стосуються елементів даних, записів і наборів.

Принцип 2. СУБД третього покоління повинні включати в себе СУБД другого покоління. Іншими словами, системи другого покоління внесли вирішальний вклад в двох областях: непроцедурний доступ та незалежність даних. І ці досягнення не повинні бути відкинени системами третього покоління. Проте існує точка зору, що є додатки, яким ніколи не знадобиться виконувати запити через властиву їм простоту доступу до СУБД.

Принцип 3. СУБД третього покоління повинні бути відкритими для інших підсистем. Інакше кажучи, будь-яка СУБД, що розраховує на широку сферу застосування, повинна бути оснащена мовою четвертого покоління (4GL), різноманітними інструментами підтримки прийняття рішень, дружнім доступом із багатьох мов програмування, дружнім доступом із популярних підсистем, інтерфейсами з графічними бізнес-пакетами, можливістю запуску додатків з бази даних на другій машині і з розподіленої СУБД.

Крім цього, представлені 13 пропозицій, в яких вимоги до нових систем обговорені більш детально:

1. Система типів СУБД третього покоління повинна бути «багатою» і різноманітною.
2. Наслідування – хороша ідея.
3. Функції, в тому числі процедури і методи баз даних, і інкапсуляція – хороші ідеї.
4. Унікальні ідентифікатори (UID) записів повинні задаватися СУБД тільки в тому випадку, коли недоступний визначений користувачем первинний ключ.
5. Правила (тригери, обмеження) стануть одною із ключових характеристик майбутніх систем.
6. Всі програмуючі доступи до баз даних повинні виконуватись через не процедурну мову доступу високого рівня.
7. Повинно бути по крайній мірі два способи специфікації наборів: за допомогою перерахування членів і шляхом використання мови запитів для задання членів.
8. Суттєвою є наявність оновлюваних уявлень.
9. Показники продуктивності не мають нічого спільного з моделями даних і не повинні в них проявляться.

10. СУБД третього покоління повинні бути доступні для різноманітних мов програмування високого рівня.
11. Мова з підтримкою стабільних даних – хороша ідея.
12. SQL, як універсальний спосіб вираження запитів.
13. Запити і відповіді на них повинні формувати нижній рівень комунікацій між клієнтом і сервером.

Третій маніфест стосується майбутнього систем управління даними і СУБД. Назва маніфесту обумовлена тим, що він слідує за двома попередніми маніфестами. Автори даного маніфесту твердо впевнені в тому, що в будь-якій спробі рухатись вперед, для того щоб витримати випробування часом, потрібно відкинути мову запитів SQL та реляційну модель взагалі. Проте вказано що потрібно робити із спадщиною SQL та реляційною моделлю. Автори Третього маніфесту винесли на розсуд читачів власні конкретні приписи та заборони, а також вельми наполегливі пропозиції, що стосуються систем управління даними і СУБД.

Перед тим як представити винесені пропозиції у Третьому маніфесті, доцільно уточнити певні поняття, які трактуються авторами в даному документі.

Поняття **домен** і **тип даних** – синонімічні поняття.

Узагальнено значення домену називають **скалярними значеннями** або просто **скалярами**.

Терміни **операція** і **функція** – синонімічні.

Кортеж – це елемент відносин, тобто рядок таблиці.

Таблиця 2 – Приписи, заборони та пропозиції, що представлені у Третьому Маніфесті

Приписи	Заборони	Наполегливі пропозиції
1	2	3
Реляційної моделі		
1. Домени 2. Типізовані скаляри 3. Скалярні оператори 4. Реальне уявлення 5. Істинність значень 6. Конструктор типів 7. Оператор порівняння 8. Кортежі 9. Відносини 10. Скалярні і кортежні змінні 11. Змінні відносин 12. Транзакції 13. Операції створення/руйнування 14. Реляційна алгебра 15. Імена і явні значення відносин 16. Реляційні функції 17. Присвоєння відносин і кортежів 18. Порівняння 19. Обмеження цілісності 20. Предикати відносин і баз даних 21. Каталог	1. Ніякого впорядкування атрибутів 2. Ніякого впорядкування кортежів 3. Ніяких дублікатів кортежів 4. Ніяких невизначених значень 5. Ніяких помилок, пов'язаних з порожніми відносинами 6. Ніяких конструкцій внутрішнього рівня 7. Ніяких операцій на рівні кортежів 8. Ніяких зіставних стовпців 9. Ніяких придушень перевірки доменів 10. Не SQL	1. Можливі ключі для похідних 2. Системно-генеровані ключі 3. Посилальна цілісність 4. Висновок можливих ключів 5. Запити з квотою 6. Транзитивне замикання 7. Параметри кортежів і відносин 8. Значення за замовчуванням 9. Міграція SQL

22. Розробка мови		
-------------------	--	--

Продовження таблиці 2

1	2	3
Ортогональні		
1. Перевірка типів на стадії виконання 2. Просте спадкування (умовне) 3. Множинне успадкування (умовне) 4. Обчислювальна повнота 5. Явні границі транзакцій 6. Вкладені транзакції 7. Агрегати і порожні множини	1. Ніяких ідентифікаторів об'єктів 2. Ніяких «публічних змінних екземпляра» 3. Ніяких «захищених змінних екземпляра» або «друзів»	1. Спадкування типів 2. Конструктори типів-колекцій 3. Конвертація в і з відносин 4. Однорівневе зберігання

Висновки та перспективи подальшого дослідження. Отже, зважаючи на вищевказані пропозиції та вимоги Маніфестів, можна дійти до загального висновку, що об'єктно-орієнтовані бази даних повинні зберігати всі можливості об'єктно-орієнтованої мови програмування і забезпечувати створення і довготривале зберігання об'єктів у зовнішній пам'яті. Об'єктно-орієнтовані СУБД повинні забезпечувати довготривале зберігання об'єктів, адекватне внутрішнє представлення об'єктів, що відносяться до різних типів даних, паралелізм в обробці різних типів об'єктів, виконання всіх методів, успадкованих об'єктом. Такі бази даних називають об'єктно-орієнтованими в повному розумінні об'єктної парадигми.

В об'єктних СУБД велике майбутнє, перш за все, тому, що структура об'єктних баз даних більш точно відповідає предметній області, чим структура реляційних баз даних. Надзвичайно важливо прийти до спільної згоди щодо визначення системи об'єктно-орієнтованих баз даних та єдиної специфікації об'єктної моделі баз даних, оскільки це нове покоління, що може прийти на заміну реляційній моделі. На шляху до цієї мети, виникає багато проблем, серед них можна виділити три характерні риси сучасного стану досліджень у цій галузі:

- відсутність загальноприйнятої моделі даних;
- відсутність єдиної формальної теорії;
- активна експериментаторська діяльність.

Одною із переваг ООСУБД являється їх можливість інтеграції з об'єктно-орієнтованими мовами програмування. В результаті всі складові інформаційної системи зможуть конструюватись з єдиних позицій і в єдиному стилі.

В даний час перспектива об'єктних моделей баз даних дуже велика. Існує багато галузей діяльності людини в якій саме об'єктні моделі можуть якісно і з меншою проблематичністю допомогти. Та, на жаль, об'єктно-орієнтовані бази даних ще не настільки використовувані, щоб перерости і витіснити реляційні бази даних. Не останню роль у практичній відсутності досліджень з ООБД у нашій країні відіграє часто невинуватна критика об'єктно-орієнтованого програмування в цілому.

1. <http://www.ua5.org/database/118-obktno-orntovan-bazi-danikh.html>
2. Дейт К.Дж. Введение в системы баз данных. - К.: Диалектика, 1998. - 784 с.
3. <http://www.simulation.kiev.ua/dbis/lection30.html>
4. Кузнецов С.Д. Основы современных баз данных.
5. <http://ukrbukva.net/print/page,1.35438-Manifest-sistem-ob-ektno-orientirovannyh-baz-dannyh.html>
6. Хомоненко А.Д. Базы данных: учебник для вузов по технич. и экономич. спец. / [А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев] ; под ред. А. Д. Хомоненко. - Санкт-Петербург : КОРОНА принт, 2002. - 672 с.
7. «Манифест систем объектно-ориентированных баз данных» М. Атkinson, Ф. Бансилон, Д. ДеВитт, К. Диттрих, Д. Майер, С.Здоник, Перевод: М.Р. Когаловский, Источник: журнал Системы Управления Базами Данных # 4/1995, издательский дом «Открытые системы» Новая редакция: Сергей Кузнецов, 2009 г.
8. Системы баз данных третьего поколения: Манифест. (Второй Манифест)
9. Третий Манифест, Х. Дарвин и К. Дэйт, Перевод: М.Р. Когаловский, Источник: журнал Системы Управления Базами Данных #1/1996, Издательский дом «Открытые системы».

