**УДК 004.67**

# *Методологія створення сховищ даних: стандарти та моделювання*

***Георгій Асєєв,***
*доктор технічних наук, професор, завідувач кафедри інформаційних технологій ХДАК*

*Представлено основну частину термінології, уживаної при створенні електронних сховищ даних. Розглянуто стандарти моделей сховищ даних, названих схемами "зірка" і "сніжинка". Дано рекомендації з їхнього використання.*

У попередніх публікаціях [1—4] нами була розглянута методологія електронних сховищ даних (СД) і архівів документів. Для розуміння матеріалу, що далі викладається, дамо деякі пояснення використовуваної у статті термінології.

**Термінологія**. У недавно виниклій сфері СД існує проблема термінології. Розглянемо основні терміни [5].

*Сховище даних.* Це поняття трактується найбільш широко. Наведемо визначення Б. Інмона, що стало вже класичним: Сховище даних (Data Warehouse) це — *"предметно-орієнтований, інтегрований, незмінний, підтримуючий хронологію набір даних, організований для підтримки прийняття* рішень". Найчастіше під СД мається на увазі не тільки набір даних, а й вся технологія його використання. У представленій статті СД ми будемо розуміти тільки як набір даних, причому не єдиний, що використовується у рамках цієї технології.

*Вітрина даних.* Це поняття виникло дещо пізніше терміна "сховища даних", тому в деяких джерелах воно злито з поняттям СД. У цій публікації під вітриною (або кіоском) даних (Data Mart) ми будемо розуміти порівняно невелике СД, сконструйоване для використання якимсь підрозділом з однією істотною відмінністю від СД — у вітрині даних кінцевий користувач може створювати свої власні структури даних. Існує ще одна особливість у вітрин даних (ВД) — джерелом для більшості даних, що зберігаються в них, є СД. Це призводить до того, що при створенні ВД рідко використовують інструменти з очищення, денормалізації та уніфікації даних.

*Технологія сховищ даних.* Під цим терміном будемо розуміти технологію використання всіх об'єктів, пов'язаних із СД, як-то:

- сховища даних;
- вітрини даних;
- програмне забезпечення.

*Система підтримки прийняття рішень.* Термін "система підтримки прийняття рішень (DSS, СППР)" почали використовувати раніше виникнення концепції СД, але й дотепер він має дуже багато трактувань. Чимало авторів застосовують його для назви всієї системи в цілому, включаючи джерела даних, СД і засоби представлення та аналізу даних, цього ж дотримуватимемось і ми.

*Інформаційна система керівника.* Інформаційна система керівника (ІСК), на наш погляд, не зовсім вдалий переклад терміна Executive Information System (EIS). Справа в тому, що такі системи звичайно є засобом створення додатків без програмування, тому їх використовують не стільки керівники, скільки аналітики, котрі, звичайно, застосовуючи цей засіб, створюють додатки, якими потім користуються керівники.

*Засоби OLAP.* У більшості випадків під цим терміном розуміють зручну й красиву оболонку для навігації в багатомірних даних [6].

*Операційні БД (ОБД).* Цей термін означає наші старі, добрі бази даних (БД) і уведений для того, щоб підкреслити їхню істотну відмінність від БД, використовуваних для реалізації СД.

*Засоби аналізу.* У статті цей термін означає весь спектр додатків для кінцевого користувача, включаючи:

- ІС;
- СППР;
- засоби OLAP;
- інші спеціалізовані засоби аналізу, прогнозу і представлення даних.

*Інформаційна система нового покоління (ІСНП).* У даній публікації цей термін (ІСНП) уводиться для означення всієї системи, побудованої за технологією СД, включаючи джерела даних, сховище даних і засоби аналізу.

*Структура даних у ІСНП.* ОБД є основним, але не єдиним джерелом інформації. Не секрет, що її частина (іноді навіть істотна) зберігається у форматах, які не претендують на гучну назву "База даних". Найпоширенішим таким форматом є текстовий файл, а засобом доступу — файлова операційна система. Ці джерела даних називаються зовнішніми даними.

Дані, що надходять до СД, не використовують безпосередньо системи представлення та аналізу. Ці системи одержують дані з вітрин даних. Уведення проміжного поняття "вітрина даних" має ряд безсумнівних переваг:

- кінцевий користувач працює тільки з необхідними йому даними;
- підвищується безпека доступу до даних;
- структура даних відбиває вимоги кінцевого користувача;
- спрощується проектування даних;
- знижується навантаження на СД.

Слід відзначити, що вітрина даних є логічним продовженням СД і тому витрати технічного устаткування на їхню реалізацію мінімальні для невеликих СД. Надалі, зі збільшенням обсягу СД, їх можна легко перебудувати на іншу конфігурацію технічного устаткування. Для великих організацій технологія СД реалізується в ієрархічній схемі СД. Для СД верхнього рівня СД рівнем нижче є таким самим джерелом даних, як і ОБД.

*Розмірне моделювання.* У розмірному моделюванні прийнятий стандарт моделі, названий схемою "зірка" (star schema), що забезпечує високу швидкість виконання запиту за допомогою денормалізації та поділу даних. Неможливо створити універсальну денормалізовану структуру даних, що забезпечує високу продуктивність при виконанні будь-якого аналітичного запиту. Тому схема "зірка" будується так, щоб забезпечити найвищу продуктивність при виконанні одного найважливішого запиту або для групи схожих запитів.

Схема "зірка" звичайно містить одну велику таблицю, яка називається таблицею фактів (*fact table)*, поміщену в центр, і оточуючі її менші таблиці, названі таблицями розмірності (*dimensional table)*, з'єднані з таблицею фактів у вигляді зірки радіальними зв'язками. У цих зв'язках таблиці розмірності є батьківськими, таблиця фактів — дочірньою. Схема "зірка" може мати також консольні таблиці *(outrigger*

*table)*, приєднані до таблиці розмірності. Консольні таблиці є батьківськими, таблиці розмірності — дочірніми.

Перш ніж створити БД зі схемою типу "зірка", необхідно проаналізувати бізнес-правила предметної області з метою з'ясування центрального питання, відповідь на яке найважливіша. Всі інші потрібно об'єднати навколо цього основного питання і моделювання починати саме з нього. Дані, необхідні для відповіді на це питання, потрібно помістити в центральну таблицю моделі — таблицю фактів.

Описові схеми "зірка" і рекомендаціям з її застосування присвячені роботи, наприклад [5—7]. Її ідея полягає в тому, що створюють таблиці для кожного виміру, а всі факти уміщують в одну таблицю, яка індексується множинним ключем, складеним із ключів окремих вимірів. Кожен промінь схеми "зірка" задає, за термінологією Кодда, напрям консолідації даних за відповідним виміром.

*Розмірності.* У технології багатомірного моделювання розмірність — це аспект, у розрізі якого можна одержувати, фільтрувати, групувати та відображати інформацію про факти. Типові розмірності, що зустрічаються практично в будь-якій моделі:

- клієнт;
- продукт;
- час;
- географія;
- співробітник тощо.

Розмірності, як правило, мають багаторівневу ієрархічну структуру. Наприклад, розмірність ЧАС може мати таку структуру: РІК, КВАРТАЛ, МІСЯЦЬ, ДЕНЬ.

*Таблиця фактів.* Таблиця фактів центральна в схемі "зірка". Вона може складатися з мільйонів рядків і містити підсумовуючі або фактичні дані, які допоможуть відповісти на запитання, що виникають. У ній акумульовані дані, що могли би бути розосереджені по багатьох таблицях традиційних реляційних БД. Між таблицею фактів і таблицями розмірності встановлені ідентифікуючі зв'язки, при цьому первинні ключі таблиці розмірності мігрують у таблицю фактів як зовнішні ключі. У розмірній моделі напрями зв'язків явно не показуються — вони визначаються типом таблиць. Первинний ключ таблиці фактів цілком складається з первинних ключів усіх таблиць розмірності. Таблиці розмірності мають меншу кількість рядків, ніж таблиці фактів і містять описову інформацію. Ці таблиці дають змогу користувачеві швидко переходити від таблиці фактів до додаткової інформації.

Таблиця фактів є основною таблицею СД. Як правило, вона містить відомості про об'єкти або події, сукупність яких буде надалі аналізуватися. Звичайно, говорять про чотири типи фактів, що зустрічаються найчастіше. До них відносяться:

- факти, пов'язані з транзакціями (Transaction facts). Вони базуються на окремих подіях (типовими прикладами яких є телефонний дзвінок або зняття грошей з рахунку за допомогою банкомату);
- факти, пов'язані з "моментальними знімками" (Snapshot facts). Базуються на стані об'єкта (наприклад, банківського рахунку) у певні моменти часу, наприклад, на кінець дня або місяця. Типовими прикладами таких фактів є обсяг продажів за день або денний виторг;
- факти, пов'язані з елементами документа (Line-item facts). Базуються на тому або іншому документі (наприклад, рахунку за товар чи послуги) і містять докладну інформацію про елементи цього документа (наприклад, кількість, ціна, відсоток знижки);
- факти, пов'язані з подіями або станом об'єкта (Event or state facts). Представляють виникнення події без подробиць про неї (наприклад, просто факт продажу або факт відсутності такого без будь-яких подробиць).

Факти мають низку властивостей, на яких ми коротко зупинимося:

*Адитивні факти.* Адитивність визначає можливість підсумовування факта уздовж визначеної розмірності. Адитивні факти можна підсумовувати і групувати уздовж усіх розмірностей на будь-яких рівнях ієрархії.

*Напіваддитивні факти,* тобто ті, які можна підсумовувати уздовж визначених розмірностей і не можна — уздовж інших. Прикладом може служити залишок грошей на рахунку (або залишок товару на складі). Цю величину не можна підсумовувати уздовж розмірності ЧАС. Однак сума залишків на рахунках уздовж розмірності має сенс для аналізу.

*Неаддитивні факти* взагалі не можна підсумовувати. Приклад неаддитивного факту — відношення (наприклад, виражене у відсотках). Фахівці рекомендують моделювати неаддитивні факти таким чином, щоб зробити їх адитивними. Приміром, представити відсоток складовими його величинами.

*Таблиці покриття.* Таблиці покриття використовуються з метою моделювання сполучення розмірностей, для яких відсутні факти. Наприклад, потрібно знайти кількість категорій продуктів, що сьогодні жодного разу не продавалися. Таблиця фактів продажів не може відповісти на це запитання, оскільки вона реєструє тільки факти продажів. Для того, щоб модель давала можливість відповідати на подібні запитання, потрібна додаткова таблиця фактів (що, по суті, не містить фактів), яка й називається таблицею покриття.

Для прикладу, розглянемо факти, пов'язані з елементами документа (у даному випадку рахунку, виставленого за товар). Таблиця фактів, як правило, містить унікальний складений ключ, що поєднує первинні ключі таблиць вимірів. Найчастіше це цілочисельні значення або значення типу "дата/час" — адже таблиця фактів може містити сотні тисяч або навіть мільйони записів, і зберігати в ній повторювані текстові описи, як правило, невигідно — краще помістити їх у менші за обсягом таблиці вимірів. При цьому як ключові, так і деякі неключові поля мають відповідати майбутнім вимірам OLAP-куба. Крім цього, таблиця фактів містить одне або кілька числових полів, на підставі яких надалі будуть отримані агрегатні дані.

На більш високому рівні даних виникне надмірність. Наприклад, у травні 2009 року 31 день, значення 2009 буде повторено 31 раз. Оскільки виміри, як правило, займають 1—5 % усього простору, необхідного для збереження куба, така надмірність не викликає браку пам'яті. Крім того, централізована підтримка відновлення вимірів гарантує узгодженість. Таким чином, використання денормалізованих таблиць вимірів, необхідних для підтримки спрощеного формулювання запитів, які, до того ж, ефективно обчислюються, часто дає додаткові переваги.

БД має підтримувати використання вторинних таблиць розмірності, що називаються консольними, які можуть бути пов'язані тільки з таблицями розмірності, причому консольна таблиця в цьому зв'язку батьківська, а таблиця розмірності — дочірня. Зв'язок буває ідентифікуючим або неідентифікуючим. Консольна таблиця не може бути пов'язана з таблицею фактів. Її використовують для нормалізації даних у таблицях розмірності. Нормалізація даних корисна при моделюванні реляційної структури, але вона зменшує ефективність виконання запитів до СД. У розмірній моделі головною метою є забезпечення високої ефективності перегляду даних і виконання складних запитів. Коли консольні таблиці використовують у розмірній моделі для нормалізації кожної таблиці розмірності, модель називається "сніжинка". Цю схему застосовують для нормалізації схеми "зірка". Вона дещо скорочує надмірність у таблицях розмірностей. Однією з переваг є швидше виконання запитів про структуру розмірностей (запити типу "вибрати всі рядки з таблиці розмірності на певному рівні"), що дуже часто зустрічаються при аналізі даних, можуть затримувати його хід.

Однак головна перевага схеми "сніжинка" — не економія дискового простору, а можливість мати таблиці фактів із різним рівнем деталізації. Приміром, фактичні дані на рівні дня, а планові — на рівні місяця. У цих випадках окремі таблиці фактів створюються для можливих сполучень рівнів узагальнення різних вимірів. Це дає змогу домогтися вищої продуктивності, але часто приводить до надмірності даних і до значних ускладнень у структурі БД, у якій виявляється величезна кількість таблиць фактів. Їхнє збільшення у БД може виникати не тільки з множинності рівнів різних вимірів, а й через ту обставину, що в загальному випадку факти мають різну кількість вимірів. При абстрагуванні від окремих вимірів користувач має одержувати проекцію максимально повного гіперкуба, причому далеко не завжди значення показників у ній бувають результатом елементарного підсумовування. Таким чином, при великій кількості незалежних вимірів необхідно підтримувати значну кількість таблиць фактів, що відповідають кожному можливому сполученню обраних у запиті вимірів, що також спричиняє неощадливе використання зовнішньої пам'яті, збільшення часу завантаження даних у БД схеми "зірка" із зовнішніх джерел та ускладнення адміністрування.

Часто в процесі створення СД передбачається розроблення прототипу — невеликої системи для демонстрування користувачеві нових можливостей, щоб він, випробувавши систему в дії, оцінив: чи варто продовжувати розроблення, чи відкласти його на майбутнє.

Відзначимо, що навіть при наявності ієрархічних вимірів для підвищення швидкості виконання запитів до СД нерідко перевага віддається схемі "зірка". Однак не всі СД проектують за двома наведеними схемами. Так, досить часто замість ключового поля для виміру, що містить дані типу "дата", і відповідної таблиці вимірів сама таблиця фактів може містити ключове поле типу "дата". У цьому випадку відповідна таблиця вимірів просто відсутня.

У 2005—2010 роках, як вважають експерти відомої консалтингової фірми Meta Group, приблизно 90—95 % американських фірм, що входять до списку Fortune 1000 та використовують у своїй діяльності інформаційні технології, розгорнуть електронні архіви. Наймасштабніші СД, що існують на сьогодні, розгорнуті в телекомунікаційних компаніях і досягають обсягу 5 Tb. Обсяг інвестицій у технології СД на Заході становить декілька мільярдів доларів на рік і продовжує зростати.

Отже, формально корпоративне СД можна визначити, *як комплекс апаратно-програмних засобів і технологій створення архіву (масштабу галузі, або корпорації, або підприємства) документів в електронному вигляді*. Мета створення СД полягає в забезпеченні оперативного і повноцінного доступу до усіх документів, що зберігаються і надходять. Для цього потрібно вирішити два головних завдання: ввести масив наявних в архіві документів і забезпечити можливість оперативного повнотекстового доступу до електронних документів.

Типове СД, як правило, відрізняється від звичайної реляційної БД [9]. *По-перше*, звичайні БД призначені для того, щоб допомогти користувачам виконувати повсякденну роботу, тоді як СД — для прийняття рішень. Наприклад, продаж товару і виписування рахунку відбувається з використанням БД, призначеної для обробляння транзакцій, а аналіз динаміки продажів за кілька років, що уможливлює планування роботи з постачальниками, — за допомогою СД. *По-друге*, звичайні БД піддані постійним змінам у процесі роботи користувачів, а СД відносно стабільне: дані в ньому, зазвичай, обновляються відповідно до розкладу (наприклад, щотижня, щодня або щогодини — у залежності від потреб). В ідеалі, процес поповнення являє собою просте додавання нових даних за певний період часу без зміни колишньої інформації, що вже знаходиться в сховищі. І *по-третє*, звичайні БД найчастіше є джерелом даних, завантажених у сховище. Крім того, воно може поповнюватися за рахунок зовнішніх джерел, наприклад, статистичних звітів.

Дуже важливий основний принцип дії СД: раз занесені в СД дані потім багаторазово витягаються з нього і використовуються для аналізу. Звідси випливає одна з основних переваг використання СД у роботі підприємства — контроль за критично важливою інформацією, отриманою з різних джерел, як за виробничим ресурсом.

Відзначимо, що найуразливішим місцем використання СД на підприємстві є коректність його даних, отриманих із різних джерел. Дані перед завантаженням у СД потрібно або "очистити від шуму", або обробити методами нечіткої логіки, що допускає наявність суперечливих фактів. Наприклад, дані про підприємство-партнера можуть бути отримані від різних експертів, чиї оцінки інколи діаметрально протилежні [4].

Відзначимо також, що інтеграція у визначенні СД розуміється не як інтеграція інформації з усіх джерел функціональної діяльності підприємства, а у сенсі погодженого представлення даних з різних джерел за їхнім типом, розмірністю і змістовним описом. Це є інтеграція даних від бізнес-процесів, а не самих бізнес-процесів. Бізнес-процеси інтегруються в рамках корпоративної інформаційної системи (ІС) підприємства.

Важливо пам'ятати, що використання інформаційних технологій на базі СД припускає задачний підхід у його організації. СД створюється для вирішення конкретних, чіткого визначених задач аналізу даних. Їхнє коло може бути розширене згодом, але визначальним моментом у побудові СД є завдання аналізу даних, які слід вирішувати для досягнення цілей ваших бізнес-процесів.

*Список використаної літератури*


1. *Асєєв Г. Г.* Методологія електронного документообігу: підсистеми автоматизації діловодства і статичні архіви документів / Георгій Асєєв // Вісн. Кн. палати. — 2005. — № 3. — С. 24—27.
2. *Асєєв Г. Г.* Методологія електронного документообігу: динамічні архіви / Георгій Асєєв // Вісн. Кн. палати. — 2005. — № 11. — С. 22—25.
3. *Асєєв Г. Г.* Методи добування та знаходження знань у сховищах даних електронного документообігу / Георгій Асєєв // Вісн. Кн. палати. — 2008. — № 9. — С. 29—31.
4. *Асєєв Г. Г.* Методологія створення сховищ даних: проблема виявлення нового знання методами Knowledge discovery in databases / Георгій Асєєв // Вісн. Кн. палати. — 2008. — № 4. — С. 23—26.
5. *Асеев Г. Г.* Электронный документооборот : учебник / Г. Г. Асеев. — К. : Кондор, 2007. — 500 с.
6. *Raden N.* Star Schema / N. Raden. — Santa Barbara, CA : Archer decision sciences. 2005—2006. — Режим доступу : http://members.aol.com/nraden/str101.htm. — Назва з екрану.
7. *Mumick I. S.* Maintenance of data cubes and summary tables in a warehouse/ I. S. Mumick, D. Quass, B. S. Mumick. — Stanford University, Database group, 2006. — Режим доступу : http://www.db.stanford.edu/pub/papers/cube.maint.ps. — Назва з екрану.
8. *Федоров А.* Введение в OLAP. Ч. 2. Хранилища данных / А. Федоров, Н. Елманова. — Режим доступа : http://www.olap.ru/basic/OLAP_intro1.asp. — Загл. с екрана.