УДК 004.6

А.Ю. Яцишин

Національний технічний університет України «Київський політехнічний інститут», Київ

ПРОЕКТУВАННЯ ГІБРИДНИХ СХОВИЩ ДАНИХ З ВРАХУВАННЯМ СТРУКТУРОВАНОСТІ ДАНИХ

Розглянуто питання проектування гібридних сховищ даних з врахуванням структурованості даних. Введено поняття мультибазових сховищ даних як розширення гібридних сховищ даних.

Ключові слова: гібридні сховища даних, мультибазові сховища даних, проектування сховищ даних, структурованість даних

Вступ

Питання проектування сховищ даних завжди було важливим етапом створення інформаційних систем, адже від швидкодії сховища даних суттєво залежить швидкодія системи в цілому. Крім того, структура сховища даних впливає на функціональність та гнучкість системи.

Постановка задачі

Автором у роботах [8-11] запропоновано підхід до автоматизованого проектування гібридних сховищ даних на базі генетичного алгоритму, з врахуванням інформації про запити, що надходять до системи. За таких умов забезпечувалося поєднання швидкодії реляційних баз даних з гнучкістю багатовимірних баз даних. Отримані автором наукові результати дозволяють побудувати сховище даних, яке автоматично перепроектовується під конкретні популяції запитів, розподіляючи дані між реляційною та багатовимірною базою даних.

Однак цей підхід не враховує ті випадки, в яких система ще не отримувала запитів користувачів чи в силу певних причин ці дані недоступні.

У такому випадку постає питання проектування гібридних сховищ даних на основі інформації про дані, які можна отримати безпосередньо з джерел даних.

Однією з важливих характеристик даних, яка може слугувати ознакою для їх розподілу даних, є структурованість даних. У цьому дослідженні розглядається питання автоматизованого проектування гібридних сховищ даних з врахуванням структурованості даних.

Огляд існуючих підходів та досліджень

У статті [5] наведено наукові результати, які є важливими для визначення структурованості сховищ даних і побудови її математичного опису.

Стаття присвячена питанням дослідження реальних та тестових наборів даних RDF (Resource Description Framework). В ній пропонується формальне визначення структурованості даних наборів даних RDF, а також пропонується загальний метод генерації тестового набору даних RDF, що змінюється залежно від вимірів, структурованості та обсягу реальних та тестових наборів даних RDF.

Наведемо базові співвідношення для визначення структурності даних [5].

Визначення 1 (за [5]). Системою типів T є множина:

$$\tau = \{T \mid \exists(s, w3type, T) \in D\} \quad (1)$$

Визначення 2 (за [5]). Набором екземплярів I типу T в системі типів D є множина:

$$I(T, D) = \{s \mid \exists(s, w3type, T) \in D\} \quad (2)$$

Визначення 3 (за [5]). Набором властивостей P типу T в системі типів D є множина:

$$P(T) = \{p \mid s \in I(T, D), \exists(s, p, o) \in D\} \quad (3)$$

Визначення 4 (за [5]). Кількість входжень властивості p в набір екземплярів I(T,D) (позначимо через OC(p,I(T,D))) визначається таким чином:

$$OC(p, I(T, D)) = \\ = |\{s \mid s \in I(T, D), \exists(s, p, o) \in D\}| \quad (4)$$

Визначення 5 (за [5]). Структурованість для одного типу даних визначається за формулою:

$$CV(T, D) = \frac{\sum_{\forall p \in P(T)} OC(p, I(T, D))}{|P(T)| \times |I(T, D)|} \quad (5)$$

Формула (5) описує структурованість лише для одного типу даних. Для того, щоб врахувати декілька типів, використовується зважена сума.

Визначення 6 (за [5]). Структурованість набору даних $CH(\tau, D)$ визначається за формулою:

$$CH(\tau, D) = \sum_{\forall T \in \tau} WT(CV(T, D)) \times CV(T, D). \quad (6)$$

$$WT(CV(T, D)) = \frac{|P(T)| + |I(T, D)|}{\sum_{\forall T' \in \tau} |P(T')| + |I(T', D)|} \quad (7)$$

Із цього співвідношення випливає, що існує взаємозв'язок між обсягом даних і структурованістю набору даних, тобто зміна розміру даних може вплинути на структурованість та відповідно зміна структурованості даних може вплинути на обсяг набору даних.

У статті [5] стверджується, що в RDF можуть бути представлені різноманітні набори даних, починаючи від структурованих даних (наприклад, DBLP) до неструктурованих даних (наприклад, у Вікіпедії / DBpedia). Структурованість даних традиційно є одним з ключових чинників при ухваленні рішення про відповідний формат представлення даних (наприклад, для структурованих реляційних і XML, для частково структурованих даних).

Вибір, в свою чергу, багато в чому визначає організацію даних (наприклад, з використанням теорії залежностей і нормальних форм для реляційної моделі та XML). Це має вирішальне значення при прийнятті рішення про те, як індексувати ці дані (наприклад, B^+-дерево індексів для реляційної моделі і нумерації на базі індексів XML).

Структурованість також впливає на те, як здійснюється доступ до даних (наприклад, за допомогою SQL для реляційних і XPath / XQuery для XML). Іншими словами, структурованість даних пронизує всі аспекти керування даними і відповідно продуктивність СКБД зазвичай вимірюється за даними з очікуваним рівнем структурованості (наприклад, в тесті TPC-H для реляційної моделі і XMark для даних XML).

Основною перевагою RDF є саме те, що він може бути використаний для подання даних для всіх можливих рівнів структурованості - від неструктурованих до структурованих. Однак така гнучкість RDF пов'язана з її головним недоліком : при нечіткому визначенні структурованості даних неможливо задати типи даних для системи керування баз даних RDF, оскільки це робиться апріорно на базі інформації про структурованість даних.

Інтуїтивно зрозуміло, що рівень структурованості даних D по відношенню до типу T визначається тим, наскільки добре дані екземпляра в D відповідають типу T. Розглянемо, наприклад, набори даних D та D' трійок RDF в табл 1.

Для простоти припустимо, що тип T з цих трійок має властивості name, office та ext. Якщо кожен об'єкт (суб'єкт) в D встановлює значення для більшості (якщо не всіх) властивостей T, то всі об'єкти в D мають досить схожі структури, які відповідають T.

Таблиця 1

Приклади трійок RDF

Набір даних D	Набір даних D'
(person0, name, Eric)	(person3, name, Timmy)
(person0, office, BA7430)	(person3, major, C.S.)
(person0, ext, x4402)	(person3, GPA, 3.4)
(person1, name, Kenny)	(person4, name, Stan)
(person1, office, BA7349)	(person4, GPA, 3.8)
(person1, office, BA7350)	(person5, name, Jimmy)
(person1, ext, x5304)	(person5, GPA, 3.7)
(person2, name, Kyle)	
(person2, ext, x6282)	

У цьому випадку можна сказати, що D має високу структурованість по відношенню до T, що так і є для набору даних D з табл. 1. Розглянемо набір даних D_M , який складається з об'єднання трійок $D \cup D'$ у табл. 1. Для наочності розглянемо тип TM з властивостями major та GPA, на додаток до властивостей T.

Набір D_M має низьку структурованість по відношенню до T_M. Щоб зрозуміти чому це так, зверніть увагу, що тип T_M об'єднує спільно об'єкти з перекриттям властивостей. Таким чином, поки всі об'єкти в D_M мають значення для імені властивості, перші три особи (що належать до набору даних D) встановили значення тільки для властивостей office та ext, а останні три особи (що належать до набору даних D') мають значення для властивостей *major* та *GPA*.

У роботі [1] розглядається питання побудови схем для інтеграції та трансляції схем для структурованих та слабкоструктурованих даних. Автори представили дві мови визначення схем і проілюстрували їх можливості для вираження структурованих та слабкоструктурованих даних.

У дисертації [2] розглядається питання інтеграції структурованих даних та тексту та використання реляційної СУБД для моделювання інвертованого індексу, що дозволяє інтегрувати структуровані дані і текст.

Метою статті [3] є опис підходу MOM IS (Mediator environment for Multiple Information Sources) до інтеграції і виконання запитів до декількох різнорідних джерел інформації, які містять структуровані і слабкоструктуровані дані.

У роботі [4] розглядаються аспекти виконання запитів до слабкоструктурованих даних. Згідно з цією роботою, є два визначення, якими можна описати слабко-структуровані дані.

"*Частково-структуровані дані це дані, які описують самі себе*". Відповідно до цього визначення структури даних зберігаються разом зі своїми даними як метадані за допомогою міток. Ці мітки являють собою семантику кожного елемента даних. Крім того, дані значення пов'язані один з одним за допомогою вбудованої ієрархії, яка являє собою природний зв'язок між елементами даних.

"*Частково-структуровані дані є даними, що не мають схеми*". Тобто немає фіксованої, жорсткої схеми, якій повинні слідувати дані .

Зазначені особливості дозволяють слабкоструктурованим даним бути достатньо гнучкими щоб містити нерегулярні структури. Це дає змогу структурі даних змінюватися швидко і непередбачувано.

Визначення слабкоструктурованих даних засновані на тих порушеннях даних, які вони можуть представляти. Наприклад:

"*Дані, які представлені деякими закономірностями (це не зображення або текст), але, можливо, не настільки слабкоструктуровані, як деякі реляційні дані або дані ODMG*"

В роботі [3] визначено напівструктуровані дані як "*Дані, які є нерегулярними або які відображають тип та структурну неоднорідність, оскільки вона не може відповідати жорсткій, заздалегідь визначеній схемі*". Жодне з двох визначень не є точним. Більш того, немає чіткого формального визначення слабкоструктурованих даних, з яким би всі дослідники погодилися.

У контексті дослідження [7], використовується друге визначення:

"*Слабкоструктуровані дані - це дані, які є нерегулярними або які відображають неоднорідність структури та типу, оскільки вона не може відповідати жорсткій, заздалегідь визначеній схемі"*.

Це визначення демонструє гнучкість частково структурованих даних в їх здатності зберігати нерегулярні дані без жорсткої визначеної схеми.

Слабкоструктуровані дані з точки зору структури мають такі ключові характеристики:

- структури даних є нерегулярними та неявними і мають частковий характер;
- слабкоструктуровані дані є більш гнучкими.

Наведені вище характеристики призводять до визначення частково структурованих даних як класу даних, який має певний ступінь нерівномірності в її структурі.

Характеристики частково структурованих даних з точки зору схеми можна описати таким чином:

- використовуються апріорні, а не апостеріорні схеми;
- використовуються індикативні, а не обмежувальні структури;
- схеми можуть не враховуватися.
- схеми можуть суттєво змінюватися;
- типи даних елементів є еклектичними та не є точними;
- немає чітких відмінностей між елементом схеми та даних.

Дослідження [7] розширює класифікації XML-документів на три категорії (високоструктуровані, слабкоструктуровані, неструктуровані) замість двох (структуровані та неструктуровані) залежно від ступеня структурованості даних, що містяться в документі .

У цій класифікації документи можуть бути класифіковані таким чином:

- Високоструктуровані документи. Такі документи містять лише високоструктуровані дані, тобто дані, які мають чітку структуру або організацію. Ці дані відповідають реляційній або об'єктноорієнтованій моделі даних.
- Слабкоструктуровані документи містять лише частковоструктуровані дані.
- Неструктуровані документи містять лише неструктуровані дані.

У статті [6] було розширено стандартну модель об'єктів баз даних (модель ODMG) та її мову запитів (OQL) для інтеграції слабкоструктурованих даних зі структурованими даними. Автори представляють їхню реалізацію розширеної моделі ODMG і мову запитів в систему під назвою Ozone.

У роботі показано, що Ozone добре підходить для обробки гібридних даних - даних, що є частково структурованими і частково слабко-структурованими. Виключне використання структурованої моделі даних для гібридних даних зумовлює відсутність багатьох переваг моделі слабкоструктурованих даних. З іншого боку, використання виключно слабкоструктурованої моделі даних виключає строгу типізацію та ефективні механізми реалізації структурованої частини даних. Підхід, запропонований у роботі [6], що базується на гібридній моделі даних, надає переваги як структурованих, так і слабкоструктурованих даних. Цікавою особливістю цього підходу є те, що він також дозволяє структуровані дані розглядати як слабкоструктуровані дані для здійснення навігації по структурованих даних без повного знання

структури. З іншого боку, він дозволяє розглядати слабкоструктуровані дані як структуровані дані, дозволяючи стандартним додаткам отримувати доступ до слабкоструктурованих даних.

У роботі [6] було дане таке визначення частковоструктурованих даних: «*Частково-структуровані дані - це гібридні дані, частково структуровані і частково слабкоструктуровані. Вони містять точки входу з структурованих даних частково-структурованих даних і навпаки*".

Частковоструктурований документ XML містить в його ієрархічній структурі хоча б один високоструктурований і як мінімум один слабко-структурований елемент, де:

- Кожне суб-дерево з коренем у високоструктурованому елементі може містити комбінацію високоструктурованих суб-елементів та/або слабкоструктурованих суб-елементів, які є вузлами цього суб-дерева.
- Кожне суб-дерево з коренем у слабкоструктурованому елементі, складається тільки з слабкоструктурованих суб-елементів як його вузлів.

Такий опис частково-структурованих даних та документів XML дає підставу для визначення частковоструктурованого документу XML як гібриду високо структурованих і частково структурованості даних. XML як формат документа є досить гнучким, щоб містити дані різного рівня структурованості.

В результаті аналізу існуючих рішень бачимо, що пропоновані рішення не враховують багатовимірної компоненти гібридного сховища даних. Однак вони дають корисні вказівки щодо визначення структурованості даних, а також опису властивостей структурованих даних.

Можна зробити висновок, що на сьогодні немає досліджень, які б розглядали питання автоматизованого проектування гібридних сховищ даних з врахуванням структурованості даних, отже дане дослідження є актуальним.

Концепція мультибазових сховищ даних

Як бачимо з аналізу існуючих рішень, для зберігання даних з різною структурованістю потрібні різні бази даних.

Звісно, можна мати одну, наприклад, реляційну базу даних. Однак в ній недоцільно зберігати слабкоструктуровані дані, оскільки такі дані не можуть бути коректно описані за допомогою теорії залежностей та нормальними формами через відсутність чітко описаної структури.

З іншого боку, часто виникає необхідність зберігати дані (в переважній більшості випадків не високоструктуровані) в багатовимірній базі даних. Вона зумовлена тим, що існує багато комерційних рішень (наприклад, звітності), які базуються на базах даних класу OLAP.

Крім того, можемо бачити, що зберігати неструктуровані дані, наприклад бінарні файли, в базі даних також недоцільно : суттєво зменшується швидкість доступу до даних – це стосується як реляційних, так і багатовимірних баз даних. Для таких випадків використовуються такі технології, як RBS для Microsoft SQL Server, які дозволяють зберігати файли безпосередньо на файловій системі, а база даних містить лише посилання на ці файли. Але, очевидно, опиратися на використання вбудованих систем не практично, в зв'язку з тим, що можуть використовуватися різні СКБД, і тому доступ до цих даних має здійснюватися із загальної системи керування сховищем даних.

З вищенаведених міркувань пропонується:

а) використати вже описану в [8-11] концепцію гібридних сховищ даних;

б) розширити цю концепцію за рахунок наявності в сховищі бази даних XML та прямого використання файлової системи.

Це розширення дасть нам змогу розміщувати структуровані дані в реляційній базі даних, слабкоструктуровані (особливо, коли є ієрархічні залежності та складні суми) – в багатовимірній, частково-структуровані поділяти між реляційною базою даних та базою даних XML, а неструктуровані дані зберігати у вигляді файлів файлової системи. При цьому за допомого системи керування гібридних сховищ даних буде забезпечений доступ до інформації не залежно від того, де вона розміщена.

У зв'язку з вищенаведеним визначимо мультибазове сховище даних як розширення гібридних сховищ даних (рисунок)

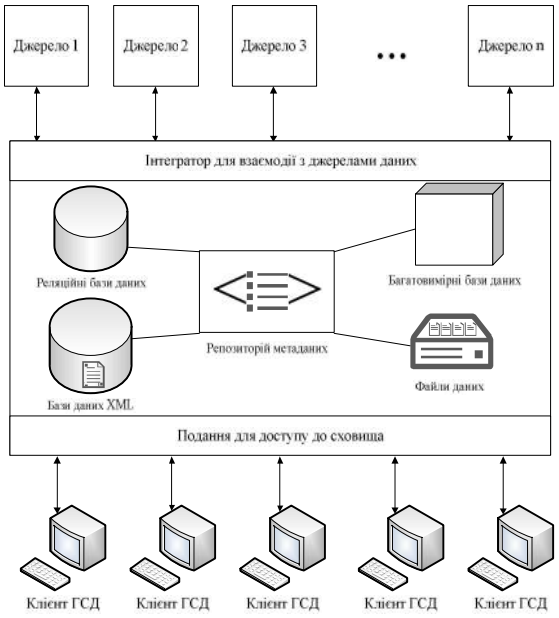

Рисунок. Архітектура сховища даних згідно концепції мультибазових сховищ даних

Визначення 7. Мультибазове сховище даних – це сховище даних, яке включає такі компоненти:

- реляційні бази даних, які на етапі проектування є призначеними для зберігання високоструктурованих даних, а на етапі оптимізації можуть містити й інші дані;
- багатовимірні бази даних, які на етапі проектування призначені для слабкоструктурованх даних, а на етапі оптимізації можуть містити і інші дані;
- бази даних XML, які на етапі проектування зберігають частково-структуровані дані, а на етапі оптимізації можуть містити й інші дані;
- файли файлової системи, які на етапі проектування розміщуються у файловій системі, а на етапі оптимізації не беруть в ній участі;
- сховище метаданих, яке найчастіше представляється набором XML файлів і зберігає інформаціїю про дані, що зберігаються в базах даних;
- інтегратор джерел даних, який виконує початкову ініціалізацію сховища та здійснює взаємодію з джерелами даних. Взаємодія з джерелами даних полягає у відслідковуванні змін даних та метаданих, що відбуваються у джерелах та застосуванні цих змін відповідно до налаштувань сховища даних;
- подання для доступу до сховища, які надають уніфікований доступ до сховища даних. Уніфікованість дає змогу користувачам єдиним звертатися до даних, що зберігаються у сховищі незалежно від їх фізичного та логічного розташування.

Визначимо задачу проектування мультибазових сховищ даних наступним чином.

Задані множини атрибутів розділених файлів S, файлів XML X, відношення у реляційній базі даних R, виміри багатовимірної бази даних D та міри M. Крім того, відоме порогове значення частот доступу до даних.

Спроектувати мультибазове сховище даних, визначивши області сховища даних A, таблиці T та атрибути B. Знайти такі значення ознак розміщення L_a, індексування I_c, матеріалізації M_c, на яких значення цільової функції часу виконання запиту буде мінімальним серед всіх можливих наборів значень цих змінних.

Оскільки мультибазове сховище даних є розширенням гібридного, то можна здійснити його проектування на основі запитів, згідно підходу, описаного у [8]. Проте цей підхід не враховує структурованість даних.

Структурованість даних може суттєво вплинути на дисковий простір, який займає сховище даних, бо у випадку неструктурованих та слабкоструктурованих даних вони зберігаються розріджено. Це приводить до падіння швидкодії сховища даних в силу її залежності від операціїй введення – виведення.

Для врахування структурованості даних при проектуванні мультибазових сховищ даних необхідно визначити процедури розрахунку структурованості даних різних джерел та розподілу даних різного рівня структурованості у сховищі даних.

Розрахунок структурованості даних джерел даних

Розрахунок структурованості ST_s різних джерел даних здійснимо із використанням співвідношень (8)-(22).

Для реляційних баз даних $ST_s=1$. Покажемо це.

Через D позначимо всю базу даних, через T – таблицю бази даних, через τ - набір таблиць бази даних. В такому випадку $|P(T)| = A_j$ - кількість атрибутів таблиці j, $|I(T,D)| = r_j$ -, кількість рядків таблиці j, а кількість входжень всіх атрибутів у всі рядки (площа таблиці)

$$\sum_{\forall p \in P(T)} OC(p, I(T,D)) = A_j r_j \quad - \qquad (8)$$

Виходячи з того, що всі екземпляри набору (рядки) мають всі властивості (атрибути), коваріація $CV(T,D) = \frac{A_j r_j}{A_j r_j} = 1$, з чого отримуємо:

$$ST_S = CH(\tau, D) = \\ = \sum_{j=1}^{n_j} \frac{A_j + r_j}{\sum_{j'=1}^{n_j} A_{j'} + r_{j'}} \times 1 = 1 \qquad (9)$$

Для розділених файлів $CH(\tau, D) = CV(T,D) = 1$, оскільки маємо лише 1 таблицю, яка є заповненою.

Для багатовимірних баз даних у якості структурованості можна взяти зважену суму наповненостей по кожній мірі:

$$ST_S = CH(\{M\}, S) = \\ = \sum_{\forall M \in \{M\}} CV(D, M) \times WT(CV), \qquad (10)$$

де M – міри джерела S; D_M – всі виміри, на яких визначені елементи для мірі M.

При цьому:

$$CV(\{D_M\}, M) = \frac{\sum_{\forall D \in D(M)} E_D^M}{\prod_{\forall D \in D(M)} |D|}, \qquad (11)$$

де E_D^M - кількість клітинок, які визначені по виміру D міри M;

$$WT = \frac{\sum_{\forall D \in D(M)} |D|}{\sum_{\forall M' \in S} \sum_{\forall D \in D(M')} |D|} \quad (12)$$

При цьому для повністю заповненої бази:

$$\sum_{\forall D \in D(M)} E_D^M = \prod_{\forall D \in D(M)} |D|, \quad (13)$$

тому аналогічно $ST_S=1$, а для пустої $ST_S=0$.

Для файлів даних XML можемо скористатися визначеннями (1)-(6) напряму, причому якщо у двох наборів даних (що містяться чи в двох тегах одного файлу, чи в двох різних файлах) є спільні підтеги (атрибути), то їх розраховують як один з'єднаний набір, а загальна структурованість всього джерела розраховується так:

$$ST_S = CH(D,S) = \sum_{\forall D \in S} CH(\tau_D, D) \times \times WT(CH(\tau_D, D)) \quad . \quad (14)$$

При цьому коефіцієнт зваженості визначається так:

$$WT = \frac{\sum_{\forall T \in \tau_D} (|P(T)| + |I(T,D)|)}{\sum_{\forall D' \in S} \sum_{\forall T \in \tau_D} (|P(T)| + |I(T,D)|)} \quad (15)$$

Для нерозподілених файлів $ST_S=0$.

Отже, розрахунок структурованості здійснюється таким чином:

- Для реляційних баз даних та плоских розподілених файлів;
- Для багатовимірних баз даних структурованість може бути визначена для кожної міри за формулою (11), при цьому сильно структуровані дані підлягають перерозподілу у реляційну базу даних, а неструктуровані – у базу даних XML;
- . Нерозподілені файли (якщо такі є) мають ST=0.

Класифікація структурованих даних

Базуючись на проаналізованих дослідженнях, виділимо чотири категорії структурованих даних:

1. Ідеально структуровані дані. Для таких даних ST=1, оскільки множина атрибутів кожного елемента набору даних збігається з множиною всіх атрибутів набору

$$\forall I \in D : A(I) = A(D) \quad . \quad (16)$$

2. Сильно структуровані дані. Для таких даних:

$$\frac{|A(D)| \times |I| - |I|}{|A(D)| \times |I|} < ST < 1 \quad . \quad (17)$$

При цьому існує хоча б один елемент набору даних, множина атрибутів якого дорівнює множині атрибутів набору:

$$\exists I \in D : A(I) = A(D) \quad . \quad (18)$$

3. Слабкоструктуровані дані: Для таких даних об'єднання множин атрибутів всіх елементів набору даних дорівнює множині всіх атрибутів набору, але при цьому жодна з множин атрибутів елементів набору включена строго в множину всіх атрибутів набору:

$$\forall I \in D : \bigcup_{I \in D} A(I) = A(D)$$
$$\forall I \in D : A(I) \subset A(D) \quad (19)$$

4. Частково-структуровані дані. Такі дані є по суті змішаним типом між структурованими та слабкоструктурованими. Для таких даних, як і в слабкоструктурованих, всі множини атрибутів елементів набору строго включені в множину всіх атрибутів набору даних, але існує така підмножина атрибутів, що на них ці дані є ідеально чи сильно структурованими:

$$\forall I \in D : \bigcup_{I \in D} A(I) = A(D)$$
$$\forall I \in D : A(I) \neq A(D)$$
$$\forall I \in D : \exists \{A\} \subset A(I) \quad (20)$$

Для таких даних структурованість може бути різною (залежно від ситуації), але у всіх випадках

$$ST > \frac{1}{|A(D)|} \quad . \quad (21)$$

5. Неструктуровані дані. Такі дані представляються одним чи декількома елементами без атрибутів:

$$|I| > 0, |A(D)| = 0 \quad (22).$$

Прикладами різних типів структурованих даних є: структуровані - база паспортів (всі паспорти мають однакову та чітку структуру); частково-структурованих – база медичних карток (можуть заповнюватися по-різному залежно від кожного пацієнта, однак є певні обов'язкові поля), неструктуровані – текстові дані (взагалі не мають структури).

Розподіл структурованих даних у сховищі даних

На основі інформації про структурованість даних можна покращити алгоритм ініціалізації сховища, представлений у [8], розподіливши дані у сховище таким чином:

1. Дані з ідеальною чи високою структурованістю розміщуються в реляційній базі даних. При такому розміщенні ці дані адаптуються під частий доступ та типові запити.

2. Слабкоструктуровані дані, особливо якщо вони містять довідники ієрархічної структури чи відносно яких є представлення, що розраховують багаторівневі суми, розміщуються в багатовимірній

базі даних. При такому розміщенні ці дані адаптуються під аналіз та гнучкі запити.

3. Частково-структуровані дані діляться на структуровану та слабкоструктуровану частини, після чого структурована частина розміщується у реляційній базі даних, а слабкоструктурована – у базі даних XML.

4. Неструктуровані дані чи дані, структурованість яких не може бути визначена, розміщуються у файлах безпосередньо у файловій системі.

Отже, для мультибазових сховищ даних у загальному випадку доступно два критерії розподілу даних – за структурованістю даних. Цей критерій який застосовується при попередньому проектуванні сховища даних та за інформацією про запити, який застосовується у процесі роботи сховища даних.

Слід зазначити, що запропонований у цій статті критерій розподілу даних є особливо важливим у випадках, коли інформація про запити, що будуть оброблятися системою, невідома, оскільки інших критеріїв розподілу даних немає.

Висновки

Розглянуто основні дослідження, які стосуються роботи з структурованими даними.

Досліджено питання класифікації структурованих даних, їх властивостей, визначення структурованості даних та аспекти їх зберігання в сховищі даних.

Запропоновано концепцію мультибазових сховищ даних як розширення гібридних сховищ даних, описано розрахунок структурованості джерел даних (реляційних, багатовимірних, файлів XML, розділених та нерозділених файлів). Описано процедуру розподілу даних згідно з їх структурованістю.

У подальших наукових дослідженнях варто розглянути питання використання комбінованого підходу до проектування мультибазових сховищ даних на основі інформації про дані джерел та запити до сховища.

Список літератури

1. Catriel Beeri. Schemas for Integration and Translation of Structured and Semi-Structured Data [Текст]. Database Theory — ICDT'99/ Catriel Beeri.,Tova Milo - 1999

2. David A. Grossman.Integrating Structured Data and Text [Текст]. Journal of the American Society of Information Science/ David A. Grossman - 1997

3. S. Bergamaschi. Semantic Integration of Semistructured and Structured Data Sources. [Текст]. ACM SIGMOD Record - Volume 28 Issue 1, March 1999 /. S. Bergamaschi, S. Castano, M. Vincini - 1999

4. Serge Abiteboul - Querying Semi-Structured Data [Текст]. ICDT '97 Proceedings of the 6th International Conference on Database Theory / Serge Abiteboul - 1997

5. Songyun Duan. Apples and oranges: a comparison of RDF benchmarks and real RDF datasets[Текст]. SIGMOD '11 Proceedings of the 2011 international conference on Management of data / Songyun Duan, Anastasios Kementsietsidis, Kavitha Srinivas, Octavian Udrea – New York -2011.

6. Tirthankar Lahiri.- Ozone: Integrating Structured and Semistructured Data[Текст] DBPL '99 Revised Papers from the 7th International Workshop on Database Programming Languages: Research Issues in Structured and Semistructured Database Programming/ Tirthankar Lahiri, Serge Abiteboul, Jennifer Widom – London - 2000

7. Yasser Abdel Kader. A Performance Analysis of a Hybrid Relational-XML Approach to Store Partially-structured Data [Текст]/. Yasser Abdel Kader - University of Sheffield, Department of Computer Science - 2007

8 Томашевський В.М.Математична модель задачі проектування гібридних сховищ даних з врахуванням структур джерел даних // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. /Томашевський В.М.,Яцишин А.Ю. – К.:Век+, – 2011. – № 53. – 211с.

9. Томашевський В.М. Особливості проектування гібридних сховищ даних з врахуванням джерел даних. Інформатика, управління та обчислювальна техніка / Томашевський В.М.,Яцишин А.Ю., - К:Вект – 2011.

10.Яцишин А.Ю. Застосування генетичного алгоритму для проектування гібридних сховищ даних // Вісник Національного університету „Львівська політехніка", секція "Інформаційні системи та мережі",/ Яцишин А.Ю. – Львів: - 2011

11. Яцишин А.Ю. Підходи та алгоритми проектування гібридних сховищ даних [Текст]. Вісник Національного університету „Львівська політехніка", секція "Інформаційні системи та мережі" / Яцишин А.Ю - м.Львів – 2010

Стаття надійшла до редколегії 2.02.2012

Рецензент: д-р.техн.наук, проф., С.Д. Бушуєв, Київський національний університет будівництва і архітектури, Київ.