УДК 004

Н.Б. Шаховська
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

ПРОСТОРИ ДАНИХ: ГНОСЕОЛОГІЯ, КОНЦЕПЦІЇ ТА ТЕНДЕНЦІЇ РОЗВИТКУ

Проаналізовано проблеми, що виникають під час роботи з розрізненими джерелами з використанням сховищ даних та баз даних. Уведено модель простору даних як засобу інтеграції та опрацювання даних з розрізнених джерел.

Problems which arise up during work with separate sources with the use of depositories information and databases are analysed. Described model of space of information as mean of integration and working of information from separate sources.

Вступ

Як відомо, для зберігання та опрацювання даних використовують різні засоби: бази даних, сховища даних, оперативні сховища даних.

Система управління базами даних (реляційна, постреляційна тощо) забезпечує загальний репозиторій для зберігання і опрацювання структурованих даних. СУБД підтримує набір взаємозв'язаних послуг і дає змогу розробникам зосередитись на специфічних проблемах їх застосувань, а не на завданнях, які виникають за потреби в узгодженому і ефективному управлінні великими об'ємами даних. Проте СУБД вимагають, щоб всі дані знаходилися під єдиним адміністративним управлінням і відповідали єдиній схемі. У відповідь на задоволення цих обмежень СУБД можуть забезпечити розвинені засоби маніпулювання даними і обробки запитів із зрозумілою і строгою семантикою, а також строгі транзакційні гарантії оновлень, паралельного доступу і довготривалого зберігання (так звані властивості "ACID").

Сховища даних краще пристосовані до зберігання та аналітичного опрацювання великих обсягів даних і переважно є інтеграцією реляційної та багатовимірної моделей (Корпоративна інформаційна фабрика Біла Інмона, шина Ральфа Кімбола, Зведення даних корпорації TDAN) [2, 3]. Вони мають розвинені засоби інтеграції даних з різних джерел та дають змогу працювати як з деталізованою, так і агрегованою інформацією, що, своєю чергою, зменшує час опрацювання таких запитів. Також завдяки операційним сховищам даних існує можливість збереження оперативної інформації, актуальність якої вимірюється секундами (наприклад, зчитування давачів приладів), завантаження потокових даних та прийняття на їх основі оперативних рішень.

Проте і бази, і сховища даних дають змогу опрацьовувати деталізовані та інтегровані дані, що побудовані на основі наперед допустимих моделей даних. У випадку роботи у Всесвітній мережі з величезною кількістю ресурсів (прикладами таких задач є туристичний бізнес – збирання інформації про місця відпочинку, її інтеграція та зберігання у внутрішніх базах даних, геоінформаційні системи – сьогодні ще не розроблено єдних стандартів подання такої інформації, а її збирання також проходить із джерел з наперед не відомими моделями даних) неможливо визначити, які саме моделі даних використовуватимуться. Тому за допомогою тільки баз даних та сховищ даних не можна організувати ефективної взаємодії між усіма об’єктами у цих предметних областях. Розробники часто зустрічаються з набором слабкозв'язаних джерел даних і тому повинні кожного разу вирішувати низькорівневі завдання управління даними. До таких завдань належать забезпечення можливостей пошуку і запиту даних; дотримання правил, обмежень цілісності, угод про іменування і т.д.; відстежування походження даних; забезпечення доступності, відновлення і контролю доступу; керований розвиток даних і метаданих.

Традиційні СУБД представляють тільки одну точку (хоч і дуже важливу) в просторі рішень управління даними. Важливою точкою є "системи інтеграції даних". Насправді, системи інтеграції даних і обміну даними традиційно призначаються для підтримки багатьох інших служб в системах просторів даних. Особливість полягає у тому, що в системах інтеграції даних потрібна семантична інтеграція до того, як можуть бути забезпечені які-небудь інші послуги. Тому, хоч і відсутня єдина схема, якій відповідають всі дані, система повинна знати точні взаємозв'язки між елементами, що використовуються в кожній схемі. В результаті для створення системи інтеграції даних потрібна значна попередня робота.

Простір даних розглядають як нову абстракцію управління даними [1]. Як ключова задача робіт у області управління даними використовується платформа підтримки просторів даних (DataSpace Support Platforms, DSSP). DSSP забезпечує набір взаємозв'язаних послуг і гарантує розробникам можливість концентруватися на специфічних проблемах їх додатків, а не на завданнях, що повторюються, виникають при потребі узгодженої і ефективної роботи з взаємозв'язаними, але роздільно керованими даними

На відміну від СУБД, в ядрі DSSP потрібна підтримка декількох моделей даних, щоб природним чином підтримувалося якомога більше типів учасників.

Тому стаття присвячена побудові логічної структури простору даних, виділенню задач із забезпечення єдиного засобу зберігання та опрацювання даних.

1. Аналіз останніх досліджень

Перші статті з опрацювання різних джерел даних та використання цих даних в єдиній предметній області з’явилися у 2005 р. Роботи [2, 3] та задекларували проблеми, які привели до необхідності введення такої абстракції даних, як простір даних. Серед них:

1. Інтеграція тексту, даних, коду і потоків (частково вирішена завдяки введенню Коддом 12-ти правил побудов сховищ даних та їх подальшої модифікації [12]).
2. Забезпечення можливості багатоконтрольності даних (у концепції сховищ даних вирішувалася через процедури витягання, перетворення та завантеження даних (extract, transform and load – ETL) [3], а у випадку Інтернет із тисячами джерел інформації, поданої у різних форматах, ETL не придатна для використання).
3. Створення простих способів аналізу, узагальнення, пошуку і огляду електронних підбірок мультимедійної інформації, включаючи розробку стандартів опису метаданих (на даний момент нема єдиних стандартів опису та опрацювання мультимедійної інформації).
4. Підтримка неточних та невчасних (тих, що надходять із запізненням або в неочікуваному порядку) даних та реалізація неточних запитів. Останніми роками достатньо інтенсивно досліджується окремий випадок цієї проблеми, так звані top-K-запити та неточні запити. Знову ж таки, розроблені моделі та технології працюють лише із визначеними моделями даних (зокрема реляційною – Fquery [12], Fuzzy Grouping та Fuzzy Lookup [13]). Стосовно невчасних даних взагалі відсутні методи, які б дозволяли не тільки фіксувати факт запізнення даних, але й на основі цих тимчасово відсутніх даних приймати рішення (ведуться роботи в області машин для обробки подій, проте на сьогодні задекларовані лише проблеми, які призводять до задачі маніпулювання потоками. Розробки, здійснені у середині 90-х років в області активних баз даних (Active DBMS, ADBMS), залишилися невикористаними через великий час відклику запитів та неврахування тимчасової відсутності даних [14]).
5. Релевантність відповіді повинна залежати від користувача і від контексту. Потрібне середовище для накопичення і використання відповідних метаданих. Є наробки щодо визначення типу користувача на основі записів у журналі доступу до ресурсів [14].
6. Проблема інтеграції даних, зокрема надоперативних (частково вирішена оперативними сховищами даних – дані збираються та оперативно опрацьовуються, але залежать від зовнішньої структури) та частково структурованих (частково вирішено засобами пошуку неструктурованих даних [10]).

7. Використання природних мов запитів до баз даних (так звані системи з природномовним інтерфейсом) [9] – передбачають формування запитів до системи у вигляді запитальних речень природною мовою.

8. Підтримка систем обробки потокових даних (наприклад, система Postgres Майкла Стоунбрейкера, високорівнева мова “STREAMSQL” з вбудованими орієнтованими на потоки примітивами і операціями).

9. Повинна бути можливість ефективного зберігання, доступу і модифікації інформації про стан, а також її комбінування з реальними потоковими даними.

10. Для інтеграції в системі повинна використовуватися однорідна мова для роботи з усіма різновидами даних.

Як бачимо, майже в усіх вказаних напрямках ведуться дослідження, але вони є неінтегровані, не передбачають єдиного опрацювання та жорстко прив’язані до моделі даних, що є цілком неприйнятним у контексті просторів даних. Тому проблема формалізації просторів даних є актуальною.

2. Основний матеріал

2.1. Формальний опис простору даних

Отже, *простір даних* **DS** – це множина даних, поданих у різних моделях (баз даних **DB**, сховищ даних **DW**, статичних веб-сторінок **Wb**, неструктурованих даних **Nd**, графічних та мультимедійних даних **Gr**), локальних сховищ та індексів (**ODW**), а також засобів інтеграції (**Int**), пошуку (**Se**) та опрацювання інформації (**Wo**), об’єднаних середовищем управління моделями (**EM**).

DS=<**DB**, **DW**, **ODW**, **Wb**, **Nd, Gr**, **Int**, **Se**, **Wo, EM**>

Моделі даних, що підтримуються у просторі даних, утворюватимуть ієрархію відповідно до їх виразної потужності [1]: реляційна, багатовимірна, об’єктно-реляційна моделі, розширена мова розмітки інформації (Extensible Markup Language — XML) зі схемою, середовище опису ресурсів (Resource Description Framework – RDF), стандартний засіб опису зв’язків між об’єктами даних – онтології, описані за допомогою Web Ontology Language – OWL, структурований текст (у тому числі HTML), неструктурований текст (рис. 1).

Кожен учасник простору даних підтримує деяку модель даних і деяку мову запитів, відповідну цій моделі. "Запит" до такої моделі даних відповідає тому, що звичайно підтримується у файлових системах відносно їх директорій: зіставлення імен, пошук в діапазоні дат, сортування за розміром файла і т.д. На наступному рівні DSSP повинна підтримувати модель даних мультимножини слів для ефективного пошуку необхідної інформації за ключовими словами, отже, ми отримаємо деяку можливість бачення вмісту учасників простору даних. Нижче рівня моделі мультимножини слів в ієрархії може розташовуватися модель напівструктурованих даних, заснована на позначених графах.

За наявності деякого середовища ключова проблема полягає в знаходженні методів інтерпретації запитів різними мовами для учасників, що підтримують деякі моделі. Точніше, проблема полягає в переформулюванні запиту, поданому складною мовою, для джерела, яке підтримує слабкішу модель даних, і навпаки, переформулюванні запиту, поданого простою мовою, для джерела, яке підтримує виразніші модель даних і мову запитів (наприклад, запит за ключовими словами до реляційної бази даних).

Однією з основних служб простору даних є каталогізація елементів даних учасників. Каталог **CG** – це реєстр ресурсів даних, що містить основну інформацію про кожний з них: джерело, ім'я, місцеположення в джерелі, розмір, дата створення і власник і т.д. Каталог є інфраструктурою для більшості інших сервісів простору даних, але він також може підтримувати базовий, призначений для користувача, інтерфейс проглядання простору даних.

DB, DW, Wb, Nd, Gr⇒CG.

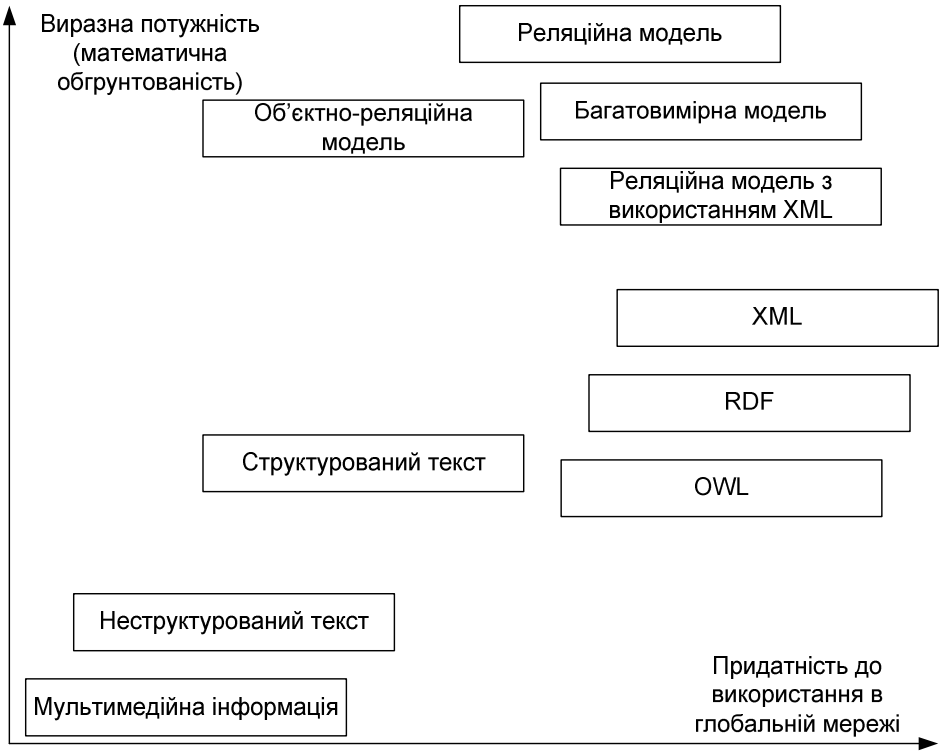
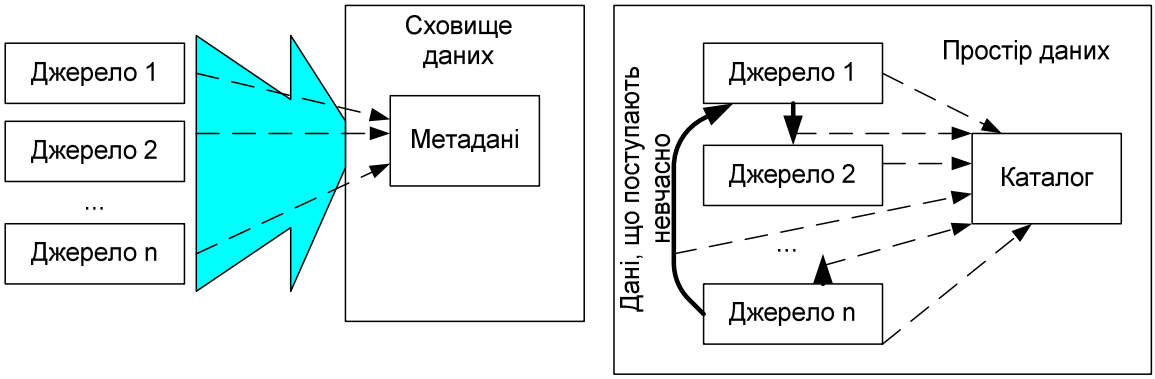


Рис. 1. Придатність моделей даних до підтримки мов запитів та до використання в глобальній мережі

Він не тільки містить описову інформацію (тобто виконує роль метаданих), але й зберігає для кожного учасника схему джерела, статистичні дані, швидкість зміни, точність, можливості відповідей на запити, інформацію про власника і дані, про політику доступу і підтримку конфіденційності. Оскільки джерела простору даних фізично не переносять у нього інформацію та можуть обмінюватись між собою інформацією, то у каталозі необхідно зберігати дані і про зв'язки між джерелами.

Відмінності між поданням джерел даних у метаданих та каталозі схематично подані на рис. 2.



а *б*

Рис. 2. Подання даних у метаданих у сховищі даних (а); каталозі простору даних (б)

На рис. 2 джерелами даних сховища даних є бази даних (реляційні, об'єктно-реляційні або багатовимірні) та оперативні сховища даних, а джерелами простору даних є об'єкти, подані у довільній моделі. Тому у каталозі необхідно також вказувати тип джерела та засоби його опрацювання (програмні продукти, стандарти передачі тощо).

Зв'язки у каталозі можуть зберігатися у вигляді:

- метаданих,
- перетворень запитів,
- графів залежності,
- текстових описів
- тощо.

Приклад схеми каталогу поданий на рис. 3. Залежно від усієї реалізації простору даних для каталогу можна використовувати відношення реляційної моделі, XML-файли, програмні модулі тощо.

Поверх каталога розміщене середовище управління моделями, яке дає змогу створювати нові зв'язки і маніпулювати існуючими зв'язками (наприклад, об'єднувати або інвертувати відображення, зливати схеми і створювати єдині представлення декількох джерел).

Для ідентифікації та роботи з неоднорідними колекціями в просторі даних можна використовувати глобальну схему імен (Uniform Resource Identifiers – URI) як механізм посилань на глобальні константи, щодо яких є деяка угода між декількома постачальниками даних.

Важливою компонентою простору даних є компонента зберігання і індексування (**ODW**) для досягнення таких цілей:

- для створення асоціацій між об'єктами даних від різних учасників;
- для вдосконалення доступу до джерел з обмеженими власними засобами доступу;
- для забезпечення можливості виконання деяких запитів без доступу до реального джерела даних;
- для підтримки високого рівня доступності і відновлення.

Засоби індексування повинні володіти високим рівнем адаптивності до неоднорідних середовищ. Результатом локального зберігання та індексування є запит, що може повернути, наприклад, рядок в текстовому файлі, елемент шляху до файла, значення в базі даних, елемент схеми або тег в XML-файлі. Важливими аспектами індексу є те, що, по-перше, він визначає інформацію *для всіх* учасників, коли деякі значення входять до декількох джерел даних (у деякому розумінні це узагальнює ідею індексів з'єднання). По-друге, індекс повинен справлятися з різноманітністю посилань на об'єкти предметної області, наприклад, з різними способами опису адміністративної одиниці.

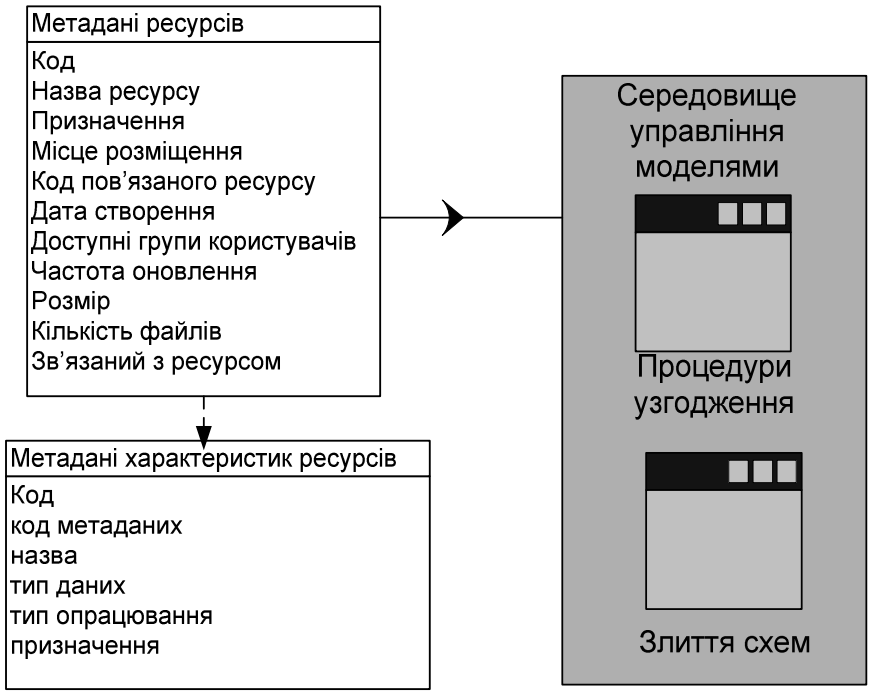


Рис. 3. Схема каталогу простору даних

Отже, зв’язок між каталогом **CG**, середовищем управління моделями **EM** та локальним сховищем та індексами **ODW** можна подати як функцію

EM(**CG**)⇒**ODW.**

Чим більше моделей здатне «розрізнити» середовище управління, тим точнішою буде інфорація в **ODW** і тим ефективніше можна буде проводити процедури інтеграції, пошуку та опрацювання даних у просторі даних **DS**.

Оскільки одним із ключових питань простору даних є питання інтеграції, то розглянемо стандарти інтеграції.

Інтеграція інформаційних систем на основі веб-служб **Int** пов'язана з використанням чотирьох ключових стандартів [4]:

- Розширена мова розмітки інформації — Extensible Markup Language (XML). Описує інформацію, що пересилається по Інтернету. Запит на одержання яких-небудь даних чи виконання певних дій іншим застосуванням вимагає наявності способів передачі параметрів і одержання назад певних результатів. При використанні веб-служб ця інформація описується за допомогою мови XML, що є міжнародним загальноприйнятим стандартом для опису довільних даних, якими, своєю чергою, можуть обмінюватися інформаційні системи.
- Простий протокол доступу до об'єкта — Simple Object Access Protocol (SOAP). Цей стандарт описує протокол виклику веб-служби (віддалений процес доступу до послуг/інформації деякої прикладної системи). У типовій ситуації взаємодії система однієї організації може викликати систему іншої організації, використовуючи протокол SOAP. Запит, що зазвичай містить ту чи іншу форму бізнес-документа, посилається ініціатором до запитуваної системи. Остання приймає запит, і вхідний документ, який міститься в запиті, обробляється. У результаті запитана система генерує відповідь, що повертається ініціатору взаємодії. Ініціатор також інформується про статус (успіх або інше) запиту.
- Мова опису веб-служб — Web Services Description Language (WSDL). Це мова, яка ґрунтується на стандарті XML, що визначає спосіб доступу до веб-служб. Вона описує функціональні можливості веб-служб і групує операції взаємодії у певні інтерфейси, що задають способи виконання операцій і ті параметри, які повинні бути на вході і виході.
- Універсальний метод опису, виявлення та інтеграції — Universal Description, Discovery and Integration (UDDI). Технологія UDDI надає засоби, за допомогою яких можна зробити так, щоб будь-які застосування чи послуги, описані в термінах веб-служб, можуть бути розпізнані іншими застосуваннями та/або організаціями. Тобто це стандарт створення регістра, використовуючи який можна описати організації і послуги, які вони надають, у вигляді, доступному для динамічного виявлення і взаємодії.

Інтеграція на основі веб-сервісів має декілька рівнів. На рівні даних програмні застовування можуть обмінюватись інформацією. Цей рівень передбачає інтеграцію даних і є найпростішим. Наступний рівень – об’єктна взаємодія. Тут йдеться про те, що програмне застосування, розташоване на одному сервері, може запускати програмні процеси на іншому. Третій рівень інтеграції – інтеграція на рівні стандартної семантики. На цьому рівні сервіси можуть “спілкуватися спільною мовою”, обходячи технологічні розбіжності. Один сервіс може звертатись до іншого із “запитом на виконання покупки”, “запитом на виконання пошуку”, “запитом на отримання статистики” та ін. На цьому рівні інтеграції сервіси потребуватимуть лише стандартизації семантики, тобто, під словами “покупка”, “пошук” і “статистика” вони повинні розуміти одне й те саме. Якщо семантичних розбіжностей між ними немає, інтеграція не має особливих труднощів. Тобто, використовуючи специфікацію WSDL, програмне застосування може “говорити” системно-незалежною мовою. З одного боку, системна незалежність застосувань постає з використання мови XML при створенні WSDL-описів, а з іншого – специфікація SOAP дає змогу взаємодіяти серверному та клієнтському застосуванням. Потрібно лише надати вхідні дані, а турботи про те, яким чином доставити їх додатку на обробку та повернути її результати назад, протокол SOAP повністю бере на себе.

Водночас існують і деякі недоліки технології веб-сервісів. У [16] зазначено основні недоліки: неоднозначність специфікації SOAP, недостатня безпечність та недостатня швидкість роботи веб-сервісів.

Проте простори даних не є підходом до інтеграції даних; швидше, це підхід співіснування даних. Мета підтримки простору даних полягає в забезпеченні базового набору функцій між всіма джерелами даних, а не в їх інтеграції. Наприклад, DSSP може забезпечити між всіма своїми джерелами даних пошук за ключовими словами, аналогічно тому, що забезпечують існуючі пошукові системи в десктопах. При потребі в складніших операціях, таких як запити в реляційному стилі, аналіз даних (data mining) або моніторинг яких-небудь джерел, можна докласти додаткові зусилля до тіснішої інтеграції цих джерел в інкрементній манері "оплати поточних рахунків" ("pay-as-you-go").

DSSP повинні працювати з даними і застосуваннями в різноманітних форматах, доступних від багатьох систем через різні інтерфейси. Від DSSP потрібна підтримка всіх даних простору даних, без яких-небудь винятків (як це буває при використанні СУБД). Тому однією із ключових задач побудови простору даних є визначення виразної потужності запитів із **Se.** Цей компонент повинен забезпечувати такі можливості:

(1) Запит про довільні дані: У користувачів повинна бути можливість запиту будь-якого елемента даних, незалежно від його формату і моделі даних. Спочатку DSSP повинні підтримувати для кожного учасника запити за ключовими словами. У міру того, як ми одержимо більше інформації про учасника, ми повинні поступово почати підтримувати складніші запити. Система повинна підтримувати плавне перемикання між запитами за ключовими словами, переглядом і структурованими запитами. Зокрема, при видачі відповідей на запит за ключовими словами (або на структурований запит) повинні пропонуватися додаткові інтерфейси запитів, що дають змогу користувачу удосконалити свій запит.

(2) Стуктуровані запити: Запити в стилі баз даних повинні підтримуватися на основі загальних інтерфейсів (тобто схем-посередників), що забезпечують доступ до декількох джерел, чи можуть адресуватися до конкретного джерела даних (з використанням його власної схеми) з наміром отримання відповідей і від інших джерел (як в системах управління одноранговими даними — Peer-Data Management System) [6]. Запити можуть формулюватися різноманітними мовами (і на основі різних моделей даних) і повинні, по можливості, якнайкраще переформулювати на інші моделі даних і схеми, забезпечуючи точні і наближені семантичні відображення.

(3) Запити до метаданих: У системі повинен підтримуватися широкий спектр запитів до метаданих. Повинні забезпечуватися можливості:

- отримання даних про джерело відповіді або про те, як ця відповідь була виведена або обчислена;
- забезпечення тимчасових позначок на елементах даних, які брали участь в обчисленні відповіді;
- визначення елементів даних у просторі даних, що можуть залежати від заданого елемента даних, і підтримка гіпотетичних запитів (тобто Що *б змінилося, якби я видалив елемент даних X?)*;
- запити джерел і рівня невірогідності відповіді.

DSSP повинні також підтримувати запити на встановлення місцеположення даних, відповідями на які є джерела даних, а не конкретні елементи даних. Наприклад, система повинна бути в стані відповідати на запити *Де я можу знайти дані про Чернівецьку область?* або В яких джерелах є атрибут "призначення"?. Аналогічно, за наявності XML-документа повинна бути можливість вибрати XML-документи зі схожою структурою і відповідні XML-перетворення. Нарешті, за наявності фрагмента схеми або опису Web-сервісу повинно бути можливо знайти в просторі даних схожі фрагменти.

(4) Моніторинг: Всі перераховані служби пошуку і запиту даних повинні також підтримуватися в інкрементній формі, що застосовується у реальному часі до потокових або змінних джерел даних. Моніторинг може бути організований у вигляді процесу без стану, в якому елементи даних розглядаються окремо, або у вигляді процесу із станом, в якому аналізується декілька елементів даних. Наприклад, фільтрація повідомлень – це процес без станів, а віконне агрегатне обчислення – це процес із станами. Служба інкрементного моніторингу може забезпечити додаткові функції виявлення складних подій і генерації сигналів.

Хоча DSSP забезпечує засоби інтегрованого пошуку, запиту, оновлення і адміністрування просторів даних, ті самі дані часто можуть бути доступні для читання і оновлення через власний інтерфейс системи, що безпосередньо управляє даними. Тому, на відміну від СУБД, DSSP не має повного контролю над своїми даними.

Засоби опрацювання даних **Wo** повинні підтримувати:

- Видобування даних (Data mining) – асоціативні правила, дерева рішень, генетичні алгоритми тощо;
- Засоби аналізу даних (Online Analytical Processing – OLAP) – реляційний OLAP (Relational OLAP – ROLAP), багатовимірний OLAP (Multidimensional OLAP – MOLAP), гібридний OLAP (Hybrid OLAP – HOLAP), динамічний OLAP (Dynamic OLAP – DOLAP);
- Засоби природномовного пошуку – побудова нечітких запитів, запитів у вигляді природних питань, запитів до метаданих;
- Засоби підбору контенту на основі аналізу характеристик користувача;
- Засоби миттєвого аналізу даних (наприклад, визначення причин підвищення тиску у котлах за значеннями давачів приладів та пропонування методів усунення неполадок).

Схему зв'язку між елементами сховища даних подано на рис. 4.

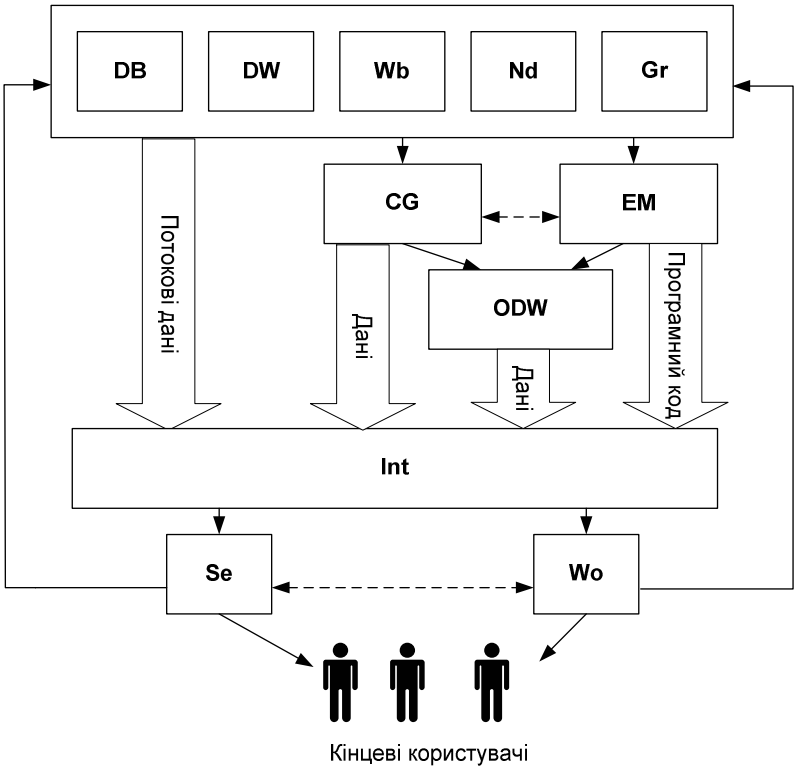


Рис. 4. Схема зв'язку між елементами сховища даних

Отже, треба виділити такі особливості просторів даних [7]:

- Простори даних складаються з широкої різноманітності форматів та інтерфейсів і усі без винятку формати даних повинні підтримуватися;
- Дані у просторі даних повністю не контролюються;
- Передбачається інтеграція тексту, даних, коду і потоків;
- Підтримка структурованих, текстових, просторових, темпоральних, мультимедійних, процедурних даних; тригерів; потоків і черг даних як рівноправних компонентів;
- Простори даних повинні забезпечувати вбудовану підтримку неточних даних. Повинна бути можливість задання неточних запитів, і процесор запитів повинен відноситися до цього як до додаткового джерела неповноти і неточності;

- Відповіді на запити повинні залежати від профілю користувача. Відповідь на запит експерта повинна відрізнятися від відповіді на запит новачка. Релевантність відповіді теж повинна залежати від користувача і від контексту;
- Система повинна знати точні взаємозв'язки між елементами, що використовуються у кожній схемі;
- DSSP пропонує рівні обслуговування та методи отриманні приблизних відповідей;
- DSSP повинен запропонувати інструменти і шляхи створення щільнішої інтеграції даних в просторі в міру необхідності.

Можуть забезпечуватися різні рівні послуг з обробки запитів до DSSP, і в деяких випадках вони можуть повертати якнайкращі з можливих приблизні відповіді. Наприклад, якщо деякі джерела даних стають недоступними, DSSP може забезпечити найкращий з можливих результат на основі даних, доступних під час виконання запиту.

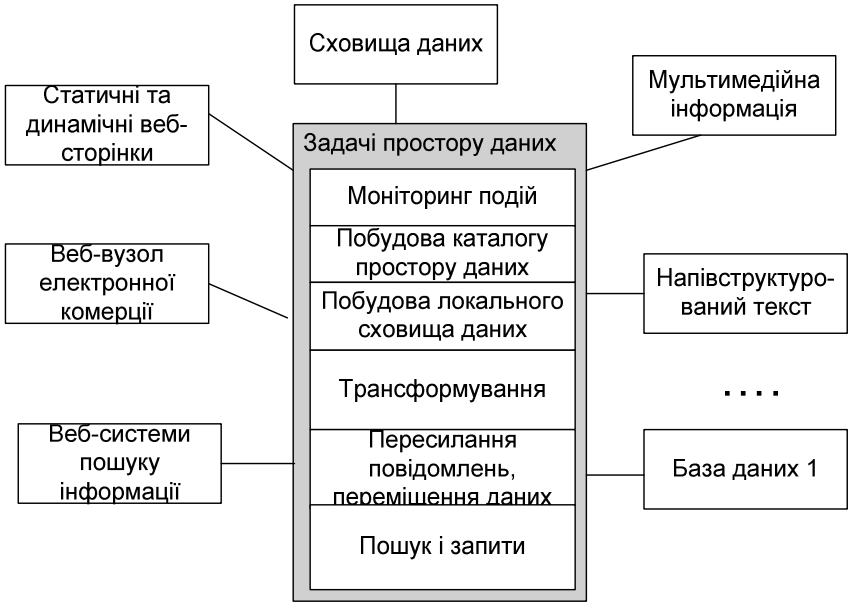


Рис. 5. Об'єкти простору даних та його задачі

Простір даних підпорядковується загальним методам адміністрування.

Для кращого розуміння принципів побудови простору даних побудуємо концептуальну модель простору даних в галузі туризму.

2.2. Концептуальна модель простору даних в галузі туризму

2.2.1. Об'єкти простору даних

Опишемо об'єкти простору даних в галузі туризму. Для простору даних необхідна інтергація інформації про такі об'єкти:

- Місцеві органи управління – надають інформацію про відпочинкові, рекреаційні та оздоровчі ресурси, а також правила їх експлуатації; лінії сполучення, особливості місцевості тощо.
- Туристичне агенство – надають інформацію про себе, про послуги, що вони надають.
- Адміністративні одиниці – описуються через інфорацію місцевих органів управління, а також через відклики попередніх відвідувачів.
- Особа (відпочиваючий) – надає інформацію про себе, про умови, які він хоче отримати, ціни тощо.

Залежно від типу об'єкта інформація може зберігатися у різних моделях та надходити з різних джерел.

- Туристичне агентство – база даних, динамічний веб-сайт з базою даних, розміщеною на Веб-сервері;
- Адміністративна одиниця – сховище даних;

- Особа – веб-сайт, база даних, текстові дані тощо,
- Відпочинковий ресурс – база даних, веб-сайт.

Оскільки специфікою галузі туризму є подання інформації в Інтернет у вигляді реклами, замовлень тощо, то на найвищому рівні ієрархії моделей даних знаходяться колекції іменованих ресурсів з базовими властивостями – розмір, дата створення і тип (наприклад, зображення JPEG, база даних MYSQL).

Галузь туризму накопичить мільйони даних протягом всього лише декількох років. Через велику кількість розрізнених ресурсів жодна людина не зможе знати ні все сховище повністю, ні те, що означає кожен файл. Людям, що звертаються до цих даних, особливо, тим, які не входять до сховища даної групи, знадобиться зведений реєстр основних атрибутів файлів, таких як період часу, до якого відноситься даний файл, географічний район тощо. Коли врешті-решт інформація знайдена, постає питання її узгодження та інтерпретації.

Незабаром таким групам потрібно буде об'єднуватися з іншими групами для створення туристичних просторів даних регіонального або національного масштабу. Їм потрібно буде якомога простіше імпортувати свої дані в стандартних форматах і з глибиною деталізації (частина файлу або декілька файлів). Користувачі федеральних просторів даних можуть захотіти побачити колекції даних, що належать різним групам федерації, наприклад, всі описи та відгуки відпочиваючих щодо опису певного природного ресурсу. Для швидкого пошуку в таких колекціях можуть знадобитися локальні копії або додаткові індекси.

Усі об’єкти простору даних, залежно від моделей даних що вони використовують та методів опрацювання інформації, об’єднуються в учасників.

2.2.2. Учасники простору даних

Простір даних повинен містити всю інформацію, необхідну для в галузі туризму, незважаючи на формат і місцерозташування цієї інформації, а також моделювати розвинений набір зв'язків між репозиторіями даних. Отже, ми моделюємо простір даних як набір *учасників* і зв'язків.

Наведемо приклад простору даних в галузі туризму із врахуванням засобів інтеграції (рис. 6).

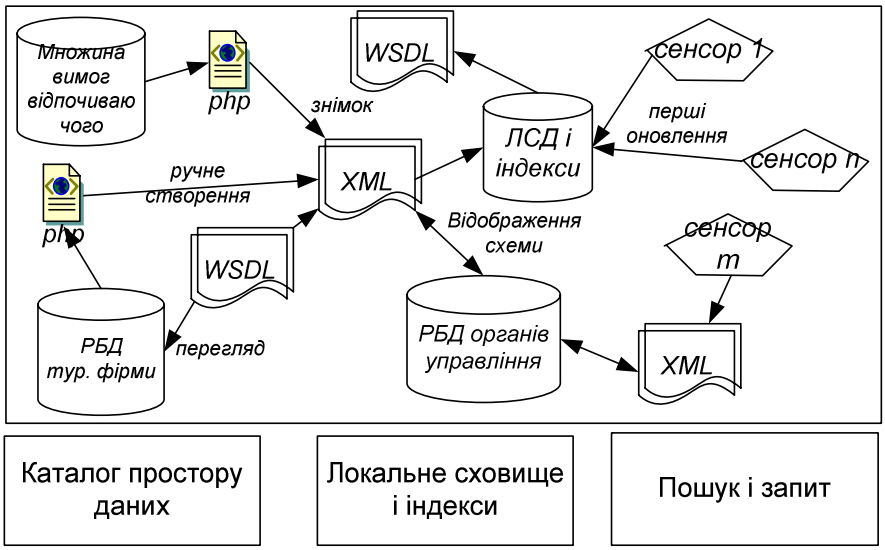


Рис. 6. Приклад простору даних в галузі туризму і компоненти системи простору даних

Учасниками простору даних є індивідуальні джерела даних: вони можуть бути реляційними базами даних туристичних фірм (*РБД тур. фірми*) та органів управління (*РБД органів управління*), репозиторіями *XML*, текстовими базами даних (наприклад, вимоги відпочиваючого – *множина вимог відпочиваючого*), Web-сервісами (*WSDL*) і пакетами програмного забезпечення (наприклад, статичні чи динамічні сторінки *php*). Вони можуть зберігатися у локальному сховищі даних та індексах (*ЛДС і індекси*) або бути потоками даних (локально керованими системами потоків даних) – на рис. 6 подані за допомогою стрілок, або навіть сенсорними установками (*сенсор і*).

Деякі учасники можуть підтримувати мови запитів (наприклад, бази даних туристичних фірм, органів управління), а інші – бути неінтелектуальними і підтримувати лише обмежені інтерфейси для формулювання запитів (структуровані файли, Web-сервіси) – вимоги відпочиваючих. Учасники можуть бути дуже структурованими (наприклад, реляційними базами даних), напівструктурованими (XML, колекції коду) або повністю неструктурованими. Деякі джерела підтримуватимуть традиційні операції оновлення (бази даних), інші – допускають тільки додавання (в цілях архівації), а треті можуть бути повністю незмінюваними (статичні сторінки, напівструктурований або неструктурований текст).

Наведемо приклади учасників простору даних у галузі туризму.

Напівструктурований текст

Приклад текстового файла з вимогами відпочиваючого:

Місце відпочинку – Гурзуф, кількість осіб – 4, харчування – так. Екскурсії – так, загальна сума – від 1000 грн. до 1200 грн.

Отже, необхідна наявність процедур відображення цієї інформації в Internet (наприклад, з використанням php та MySql), пошуку у напівструктурованому тексті та запису знайдених структурних одиниць даних до внутрішньої бази даних. Важлива роль при побудові простору даних належатиме методам інформаційного пошуку (Information Retrieval). Важливо те, що в складному просторі даних користувачі часто не знають, що саме вони шукають і як інтерпретувати результати. Тому важливо, щоб вони могли ефективно візуалізувати результати пошуку і запитів для поліпшення спрямованості своїх досліджень простору даних. Тут стануть в нагоді сучасні методи з галузі візуалізації інформації (Information Visualization).

Реляційна база даних

Приклад бази даних однієї туристичної фірми (реалізована в СУБД MS Access 2003):

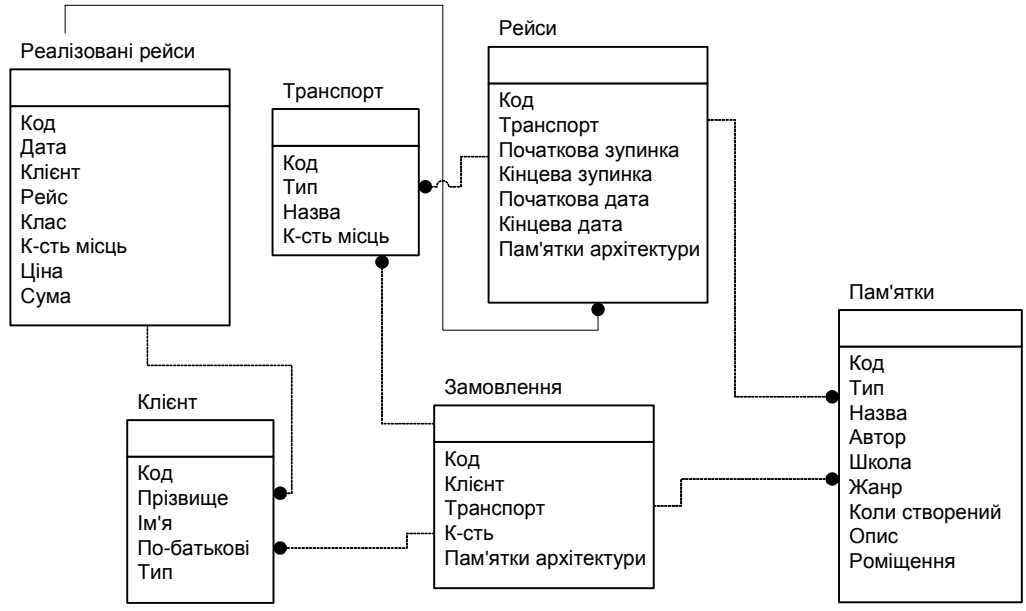


Рис. 7. Можлива логічна модель даних туристичного агентства

Отже, необхідна наявність процедур пересилання даних з локальної бази даних до серверної та у зворотному напрямку (можна використати php, java тощо), а також засоби опису веб-служб (наприклад, WSDL).

DSSP забезпечує декілька взаємозв'язаних служб над простором даних, деякі з яких є узагальненням компонентів, підтримуваних в традиційній СУБД. На відміну від СУБД, в DSSP не передбачається наявність повного контролю над даними в просторі даних. Натомість, DSSP дає змогу управляти даними системам-учасникам, але забезпечує новий набір служб поверх всіх цих систем, автономно підтримуючи їхні потреби. Крім того, для однієї галузі туризму може бути декілька DSSP, які обслуговують один і той самий простір даних – у деякому розумінні, у DSSP може бути своє власне уявлення про конкретний простір даних.

Локальне сховище даних

Наведена на рис. 8 схема містить інтегровану інформацію про об’єкти галузі туризму.

Локальне сховище повинне об’єднувати великі обсяги інформації. Тому для його реалізації доцільно вибрати MS SQL Server 2005, Oracle XE (безкоштовний для розробників) тощо.

2.2.3. Особливості простору даних у галузі туризму

Простори даних туристичних сфер для різних адміністративних одиниць можуть вкладатися одним до іншого (наприклад, простір даних району вкладається до простору даних області), і вони можуть перекриватися (наприклад, простір даних в галузі туризму перекривається просторами даних оздоровчо-лікувальної, історичної сфери та сфери управління природними ресурсами). Тому в просторі даних повинні міститися правила розмежування доступу. Прикладами таких розмежувань для простору даних в галузі туризму є:

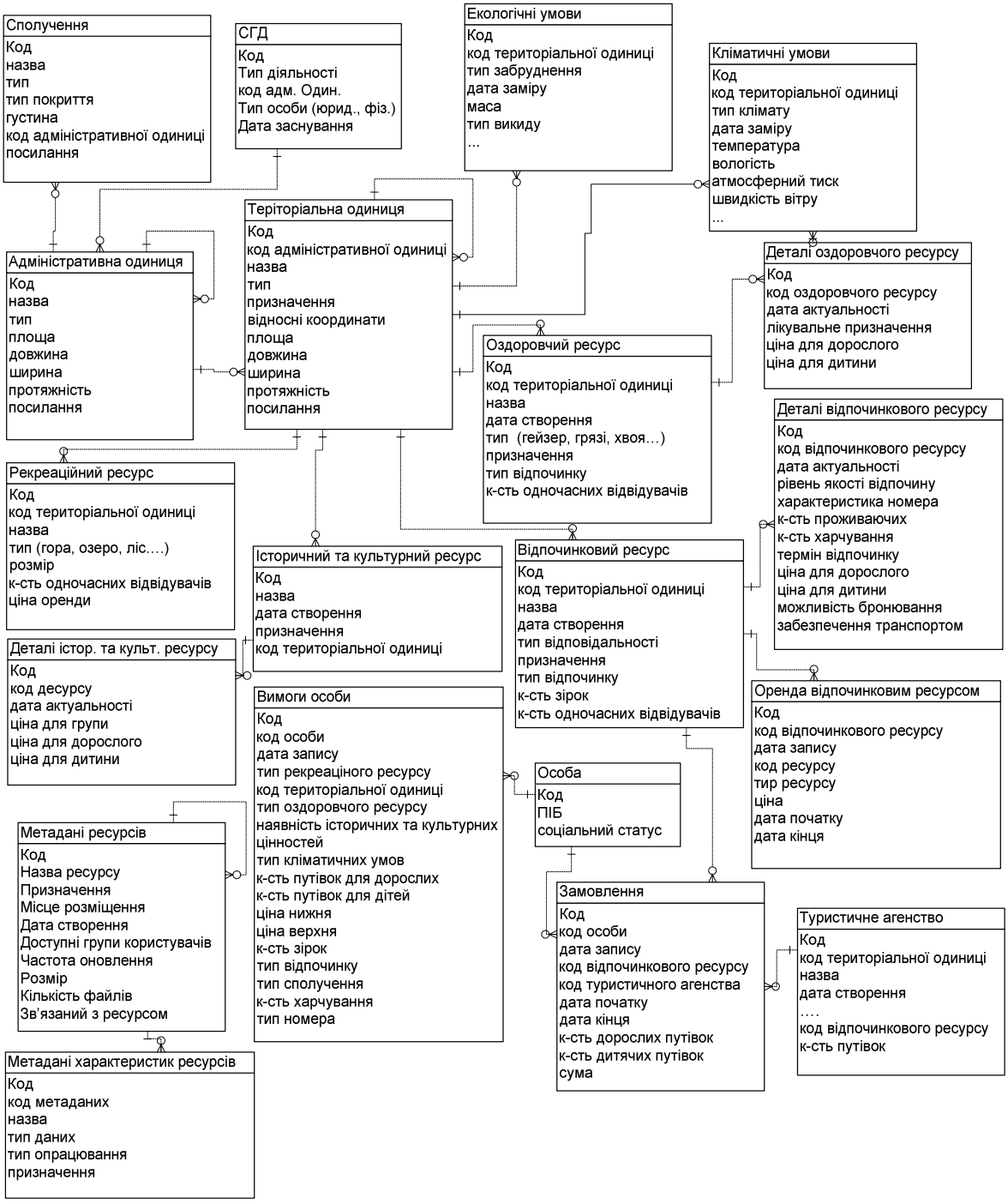


Рис. 8. Логічна модель даних локального сховища

- для учасників простору даних в галузі туризму надати можливість пошуку даних у просторах даних оздоровчо-лікувальної, історичної сфери та сфери управління природніми ресурсами;
- для учасників простору даних сфери управління природними ресурсами надати права блокування записів та встановлення властивості неактуальності для даних простору даних в галузі туризму;
- інше.

Висновки

Наведено фомальну модель простору даних та концептуальну модель простору даних в галузі туризму, що забзпечує взаємодію між джерелами іформації, поданої за допомогою різних моделей даних, з різними методами подання та опрацювання.

Наукова новизна полягає в уведенні формального опису простору даних та окресленні його основних завдань.

Практична цінність полягає у побудові простору даних у галузі туризму, виділенні основних об'єктів та учасників.

Подальші дослідження стосуватимуться формалізації методів інтеграції даних та пошуку неструктурованих, напівструктурованих та строго структурованих даних.

1. Кузнецов С. От баз данных к пространствам данных: новая абстракция управления информацией. – 2006, http://www.citforum.ru/database/articles/from_db_to_ds. 2. Дрюэк К. (Katherine Drewek). "Хранилища данных: сходство и различия подходов Билла Инмона и Ральфа Кимболла", 2005, http://www.b-eye-network.com/view/743 3. Dan Linstedt. Data Vaulttm overview the next evolution in data modeling. – 2005, http://www.tdan.com/i021hy01.htm. 4. Огляд технологій інтеграції інформаційних систем, 2006, http://www.microsoft.com/Ukraine/Government/Analytics/Integration Technologies/Overview.msp. 5. Кузнецов С. Пространства данных: исследовательский полигон или путь к новому поколению систем управления данными? http://synthesis.ipi.ac.ru/sigmod/ seminar/s20060420. 6 Donald Kossmann, Jens-Peter Dittrich. Personal Data Spaces. http://www.inf.ethz. ch/news/focus/res_focus/feb_2006/index_DE. 7. Garretts Summary of Principles of Dataspace Systems, http://aravaipa.eas.asu.edu/wiki/index.php/Garretts_Summary_of_Principles_of_Dataspace_Systems#Ove rview 8. ETH – Databases and Information Systems – iMeMex, www.dbis.ethz.ch/research/ current_projects/iMeMex 9. Processing of natural language queries to a relational database. Samsonova M, Pisarev A, Blagov M, http://www.cs.dartmouth.edu/~brd/Teaching/AI/Lectures/Summaries/ natlang.html 10. Основные концепции и подходы при создании контекстно-поисковых систем на основе реляционных баз данных. http://www.citforum.ru/database/articles/search_sys.shtml. 11. Особенности построения хранилищ данных. http://citforum.uar.net/seminars/cis99/sch.shtml/ 12. Kacprzyk J., Ziolkowski A. Database Queries with Fuzzy Linguistic Quantifiers // IEEE Transactions on Systems, Man, and Cybernetics. SMC-16, 1996. – P. 512-529. 13. Fuzzy Grouping в Microsoft SQL Server 2005 http://msdn.microsoft.com/msdnmag/issues/05/09/SQLServer2005/default.aspx. 14. Пелещишин А.М. Методи та алгоритми моделювання Web-систем // Вісник Держ. ун-ту "Львівська політехніка". – 2000. – № 406. – С. 199–211. 15. Черняк Л. Машины для обработки событий. – Открытые системы #09/2006, http://www.osp.ru/os/2006/09/3776498/_p1.html 16. Гіпертекстові технології, http://moodle. ukma.kiev.ua/mod/resource/view.php?id=1120.