приложения. – К.: Наукова думка, 1979. – 199 с. 7. Крылова Т. Н. Интерференционные покрытия. – Л.: Машиностроение, 1973. – 224 с. 8. Путилин Э .С. Оптические покрытия: учеб. пособие. – СПб.: СПбГУ ИТМО, 2010. – 227с. 9. Риттер Э. Плёночные диэлектрические материалы для оптических применений // Физика тонких плёнок. − М.: Мир, 1978. − Т. 8. − С. 7–27.

Ю. Болюбаш
Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж

МЕТОДИ ОПРАЦЮВАННЯ ВЕЛИКИХ ДАНИХ У ФЕДЕРАТИВНОМУ СХОВИЩІ ДАНИХ

Подано визначення Великих даних та описано основні характеристики. Подано моделі асоціацій між сутностями та характеристиками для різних категорій Nosql баз даних. Подано модель федеративного сховища Великих даних. Для представлення великих даних використано простір даних, який дає змогу працювати з різнотипними даними. Проте основною операцією інтеграції є не консолідація, а федералізація, що дає змогу зменшувати ємнісну складність запитів. Розроблено метод обміну різнотипними даними та приведення реляційних даних до моделі "сутність–характеристика". Апробовано розроблені методи і алгоритми.

Ключові слова: Великі дані, NoSQL, інформаційна модель даних.

There is given the Big data defenition and described the main characteristics. The models associations between entities and properties for the different categories Nosql databases "entity-characteristic" is constructed. For the presentation of Big data space there is used data that allows to work with heterogeneous data. However, the main operation is federalisation but no consolidation of integration, This allows capacitive reduce the complexity of requests. The method of heterogeneous data sharing and bringing to relational data model "entity-characteristic" was created. Were tested developed methods and algorithms.

Key words: Big data, NoSQL, data model.

Вступ

Застосування систем територіального розвитку сприяє швидкому поширенню знань, навичок та найкращих практик у певних географічних межах, таких як місто, регіон, країна тощо. Для комплексного аналізу інформації на рівні регіону необхідно:

- зберігати і керувати інформацією розміром у петабайти;
- опрацьовувати інформацію з реляційних, багатовимірних баз даних, бази даних XML і NoSQL, структурованих і слабкоструктурованих текстових файлів, баз геоданих, медіафайлів тощо;
- аналізувати різнотипну інформацію, використовуючи як консолідаційний, так і федеративний підхід до її отримання.

Процес побудови узагальненої (комплексної) моделі регіону ускладнюється різноманітністю моделей даних, а також через наявність різних рівнів агрегації даних. Однією з популярних технологій для розроблення систем територіального управління є Великі дані.

Аналіз літературних джерел та постановка задачі дослідження

Великі дані є терміном, який використовується для ідентифікації наборів даних, з якими не можна впоратися з використанням існуючих методології та програмних засобів через їх великий

розмір і складність. Такі дослідники, як М. Гілбернт [1], С. Стрініваса та ін. розробили методики і програмні засоби для передавання даних або видобування інформаційних гранул з Великих Даних (колекції об'єктів, які зазвичай формуються для атрибутів з числовими і які розташовані поряд через їх схожість, функціональну або фізичну суміжність). Методи машинного навчання та візуалізації даних дають змогу опрацювати та графічно подати результати аналізу даних великих обсягів (мільойони кортежів). Проте нерозв'язаною задачею залишається задача побудови відображення між моделями даних різних джерел. В роботах Alejandro Maté та Lucentia Research Group [2] обґрунтовано використання багатовимірної моделі для представлення Великих даних та побудови відображення в реляційну модель. Проте у випадку використання бази даних ключ-значення як однієї з джерел даних застосування багатовимірної моделі неприйнятне. Vinayak Borkar [3], Yingyi Bu [4] пропонують використовувати об'єктно-орієнтований підхід, проте обмеженням є кількість зв'язків між об'єктами.

Отже, єдиного підходу до опрацювання Великих даних не знайдено.

Аналізуючи можливість використання Великих даних у системах територіального розвитку, отримуємо:

- великий набір сутностей: особи, місця, організації (фізичні, юридичні), дати, природні ресурси (річки, ліси, озера), рекреаційний фонд (історичні пам'ятки, санаторії), законодавчі акти та звіти;
- база даних особливостей: документи для інтелектуального аналізу даних, онологічні терміни, словники даних, які дають змогу пов'язати деякі об'єкти.

Грунтуючись на цій інформації, з метою її подальшого аналізу, слід вирішити питання, які сутності і в який спосіб пов'язані між собою.

Тому задача розроблення методів та засобів опрацювання Великих даних у системах регіонального розвитку є актуальною.

Великі дані (Big Data) в інформаційних технологіях за визначенням К. Лінч, Д. Леней – набір методів та засобів опрацювання структурованих і неструктурованих різнотипових динамічних даних великих обсягів з метою їх аналізу та використання для підтримки прийняття рішень [5]. Є альтернативою традиційним системам управління базами даних і рішенням класу Business Intelligence. До цього класу належать засоби паралельного опрацювання даних (NoSQL, алгоритми MapReduce, Hadoop) [6].

Визначальними характеристиками для Великих даних є [5, 7, 8] обсяг (volume, в сенсі величини фізичного обсягу), швидкість (velocity в сенсах як швидкості приросту, так і необхідності високошвидкісної обробки та отримання результатів), різноманіття (variety, в сенсі можливості одночасної обробки різних типів структурованих і слабкоструктурованих даних).

Основними проблемами, які виникають під час обробки даних, є відсутність методів аналізу, придатних до застосування через їх різнотиповість (для регіону – це і числові дані, і геодані, слабкоструктуровані звіти тощо), потреба у значних людських ресурсах для підтримання процесу аналізу даних, висока обчислювальна складність наявних алгоритмів аналізу та стрімке зростання обсягу зібраних даних. Вони призводять до постійного зростання часу аналізу даних навіть при регулярному оновленні апаратних засобів серверів, а також необхідності роботи із розподіленими базами даних, можливості яких за більшістю методів аналізу даних не використовуються ефективно.

Отже, виникає задача розроблення ефективного методу аналізу даних, що може застосовуватись до розподілених баз даних різних предметних областей (рис. 1). Тому для регіону доцільно розробляти методи та засоби роботи з Великими даними та використання їх для аналізу.

Основний матеріал

Яскравим прикладом Великих даних є масив даних, що описує функціонування регіону.

1) **обсяги інформації у петабайти:**

оцифровані книжки бібліотек (у Львівській області налічується 1356 бібліотек з бібліотечним фондом у 18 227,4 тис. примірників, що примірно дорівнює 17,4 петабайт інформації, кількість GPS-сигналів від автомобілів кожної транспортної компанії тощо)

статистичні дані офіційних установ у Львівській області (1849 сіл, 44 міста, 34 смт. та 1 селище) щорічно 1 терабайт інформації,

2) **необхідність обробки структурованої** (бази даних відділв охорони здоров'я, податкової інспекції тощо) **та неструктурованої інформації** (статистична звітність (населення та міграція, ринок праці, освіта, охорона здоров'я, доходи та умови життя, соціальний захист, населенні пункти та житло)),

3) **он-лайн аналіз даних** для швидкого доступу до детальних даних.

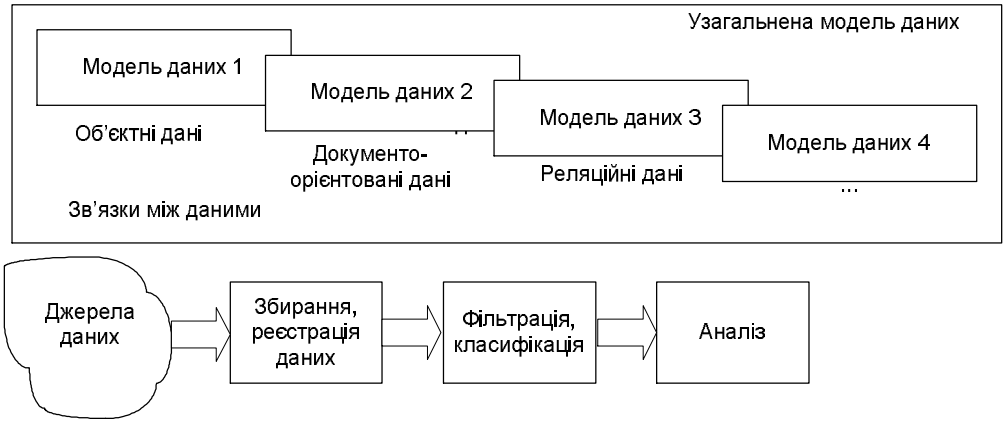

Рис. 1. Постановка задачі дослідження

Отже, у підсумку існує:

- великий набір сутностей: особи, місця, організації (фізичні, юридичні), дати, природні ресурси (річки, ліси, озера), рекреаційний фонд (історичні пам'ятки, санаторії), законодавчі акти та звіти;
- величезна база даних особливостей: документи для інтелектуального аналізу даних, онологічні терміни, словники даних, які дозволяють зв'язати деякі об'єкти.

Грунтуючись на цій інформації, ми повинні вирішити, які сутності і як пов'язані між собою.

Отже, інформаційна модель Великих даних – це

$$BigD = \langle e, f, a \rangle,$$

де сутності $e \hat{I} E$, характеристики $f \hat{I} F$, асоціації $a \to n_{e,f}$ між сутностями e та характеристиками f.

Формально можемо поділити усі об'єкти на такі категорії:

- сутності e,
- характеристики f,
- асоціації між сутностями e та характеристиками f.

Наприклад:

- Ім'я e згадується у документі f,
- Термін f з'явився у документі e.

Нехай також визначено:

- множину сутностей E;
- множину характеристик F;
- для кожних e і f зазначено номер асоціацій між e і f як $n_{e,f}$.

Від реляційної моделі модель "сутність-характеристика" відрізняється наявністю асоціації з значенням в інтервалі від 0 до 1 (реляційна модель є підвидом моделі "сутність-характеристика" зі значенням асоціації 1 для кожної сутності та асоційованої з нею характеристики).

Загальна кількість сутностей визначається як $|E|$, загальна кількість характеристик є потужністю множини F : $|F|$. Також опишемо:

- для кожної характеристики f множину $e(f) = \{e \in E : n_{e,f} > 0\}$ усіх асоційованих з f сутностей;
- для кожної сутності e множину $f(e) = \{f \in F : n_{e,f} > 0\}$ усіх асоційованих з e характеристик.

Опишемо ці якісні представлення у кількісному вигляді.

Для цього скористаємося аналогом опису міри TF-IDF у текстових документах [9].

У подібних ситуаціях, коли у нас є кілька сутностей, пов'язаних з харакетристикою, використаємо кількісне представлення інформації, тобто кількість бінарних запитань (так, ні), які необхідно задати, щоб знайти потрібний об'єкт. Загалом, якщо ми знаємо, що невідомий об'єкт належить множині, що складається з N елементів, можемо поділити цей набір на дві половини і, задаючи двійкові питання, з'ясувати, до якої половини належить шуканий об'єкт. Отже, тоді кількість об'єктів становитиме $\frac{N}{2}$. Продовжимо далі таку саму процедуру: задамо друге питання, для чого поділимо виділену половину ще на дві половини. Отже, після двох запитань матимемо $\frac{N}{4}$ об'єктів, серед яких є шуканий. Після трьох запитань матимемо $\frac{N}{8}$. Загалом після відповіді на q бінарних запитань матимемо множину з $N \cdot 2^{-q}$ елементів, що містить необхідний об'єкт [10].

Коли множина складатиметься з одного елемента, ми точно визначимо необхідну нам альтернативу. Кількість бінарних запитань для пошуку характеристики для N альтернатив: $N \cdot 2^{-q} = 1$, або $q = \log_2(N)$.

Так само можна описати сутності. Маємо $|E|$ сутностей з кількістю інформації $\log_2(|E|)$. Коли ми знаємо, що якась сутність асоційована з характеристикою (маємо $|e(f)|$ сутностей асоційованих з характеристикою f), то кількість питань дорівнює $\log_2(|e(f)|)$. Отже, той факт, що сутність e пов'язана з характеристикою f, дає змогу зменшити кількість питань до

$$k = \log_2(|E|) - \log_2(|e(f)|) = \log_2\left(\frac{|E|}{|e(f)|}\right).$$

Загальна важливість характеристики f для сутності e визначається як $\log_2\left(\frac{|E|}{|e(f)|}\right)$ з фактором важливості $1 + \log_2(n_{e,f})$. Результуюча кількість питань визначається як

$$I(e, f) = (1 + \log_2(n_{e,f})) \cdot \log_2\left(\frac{|E|}{|e(f)|}\right)$$

Ця формула є одним з варіантів в термінах частоти термінів – так званою зворотною частотою документа TF-IDF [9, 10]. Для кожної сутності e маємо кількість питань $I(e, f)$ для різних характеристик f. Значення важливості необхідно нормалізувати:

$$V(e, f) = \frac{(1 + \log_2(n_{e,f}))\chi \log_2\left(\frac{|E|}{|e(f)|}\right)}{\sqrt{\sum_{j \in f(e)} \left((1 + \log_2(n_{e,j}))\chi \log_2\left(\frac{|E|}{|e(j)|}\right)\right)^2}}$$

Для кожної сутності e є вага $V(e, f)$. Отже, як міру близькості між двома об'єктами E_1 і E_2 вважатимемо відстань між відповідними векторами $(V(e_1, f), V(e_2, f), ...)$.

У звичайній евклідовій відстані $d(a,b) = \sqrt{(a_1 - b_1)^2 + ...}$ додаються квадрати різниць. Отже, для кожної ваги $V(e, f)$, що репрезентує кількість відповідей "так"-"ні", матимемо

$$d(e_1, e_2) = \sum_{f \in F} |V(e_1, f) - V(e_2, f)|.$$

Ця відстань залежить від кількості характеристик: наприклад, якщо на додаток до документів ми зберігаємо їх копії, відстань збільшується вдвічі. Щоб уникнути цієї залежності, відстань $d(e_1, e_2)$, як правило, нормалізується в інтервалі [0,1] через ділення на максимально можливе значення цієї відстані.

Порівнюючи сутності за кожною характеристикою, отримаємо відстань між ними, подану як

$$D(e_1, e_2) = \frac{\sum_f |V(e_1, f) - V(e_2, f)|}{\sum_f \max(V(e_1, f), V(e_2, f))}.$$

Носій даних у моделі "ключ-значення" (інша назва – колонкова БД) описується кортежами вигляду:

$$KV = \{< k, v >\},$$

де k – ключ, який набуває унікальних значень у кожній парі, v – значення, що відповідає цьому ключу. Ключі можуть бути складеними (major або minor), значення підтримує практично необмежену семантику, e « k; f « v.

Сигнатура моделі виглядає як :

$$O = \langle p, s \rangle,$$

де p – операція проекції за атрибутами (ключ або значення), s – селекції атрибутів (вибір значення за ключем, ключів за значенням, ключів за значенням предків). Перераховані операції належать до категорії читання [86 – 87].

Приклади реальних операцій читання:

- `get(key),`
- `multiGet,`
  - `також MultiGetIterator, StoreIterator (за major key)`
  - `Subrange (keyFirst, keyLast)`
  - `Depth.CHILDREN_ONLY`
- `multiGetKeys`
  - `Subrange (keyFirst, keyLast)`
  - `Depth.CHILDREN_ONLY`

Прикладом СУБД колонкового типу є Cassandra.

Носій моделі *"об'єкт-документ"* описується кортежами вигляду:

$$OD = \{< f_0, < f_1 : e_1, f_2 : e_2, f_n : e_n, f_{n+1} : d_1, f_2 : d_2, f_{n+l} : d_l >\},$$

де f_0 – ідентифікатор документа; $f_1 .. f_m$ – характеристики (атрибути) документа; $e_1 .. e_m$ – атомарні значення характристик $f_1 .. f_m$, $d_1 .. d_l$ – посилання на інші документи, $d_i = e(f_i)$.

Операції цієї моделі є об'єктні.

Операція визначення вузлів елемента

$$v(f_i) = \{C\} \cup \{f_{0_i} \mid i = \overline{1,n}\} \cup \{e(f_i) \mid i = \overline{0, n+l}\},$$

де C – колекція документів f_{0_i}.

Операція визначення значень вузлів:

$$v(f_i) = \{n_{e_j, f_i} \mid i = \overline{1,n},\ j = \overline{0,m} + l\},$$

де e_j – значення атрибутів f_i.

Також визначено відношення над елементами носія.

Відношення “елемент-елемент” визначаються між документами та колекцією:

$$OD \times C \to EE .$$

Відношення “елемент-атрибут”:

$$f_i \times OD \to EA .$$

Відношення “елемент-посилання”:

$$f_i \times d_j \to ER.$$

Відношення “елемент-дані” визначаються так:

$$f_i \times e_j \to ED.$$

Прикладами СУБД цього типу є MongoDB та CouchDB.

Графову модель даних подано як:

$$O = \langle ID, A, z, r \rangle ,$$

де ID – множина ідентифікаторів, вузлів графа; A — множина позначених спрямованих дуг (p,l,c), $p,c \in ID$, l – “ рядок-мітка”, запис (p,1,c) означає, що між вузлами p та c є зв’язок l; z – функція, що відображає кожний вузол $n \in ID$ у конкретне значення складеного або атомарного типу, $z : n \to v$; V – особливий кореневий вузол графу.

Основною метою перетворення даних на модель “сутність-характеристика” є забезпечення можливості опрацювання даних будь-якої структури.

Метод перетворення даних на модель “сутність-характеристика” полягає у перерахунку важливості характеристики для сутності, а також фізичному перетворенні схеми даних на пару “сутність–характеристика”.

Схему перетворення даних на модель “сутність-характеристика” подано на рис. 2.

Для перетворення графової моделі на модель “сутність-харакетристика” важливо визначити вагу зв’язку між елементами. Оскільки першим параметром моделі є характеристика, другим – зв’язок, а третім – сутність, то перетворення між моделями полягатиме у числовому вираженні асоціації між елементами RDF-моделі (рис. 3).

Для апробації розроблених методів та визначення методики аналізу фінансових показників регіону розроблено відповідне програмне забезпечення. Для реалізації сховища даних вибрано 64-розрядну платформу (x64). Узагальнене сховище даних складається з таких компонент:

– Реляційна база даних (Microsoft SQL Server Database Services, Oracle Database, MySQL,PostgreSQL) у 64-розрядному виконанні;

– Багатовимірна база даних (Microsoft SQL Server Analysis Services або Hyperion Essbase) у 64-розрядному виконанні;

– “Ієрархічна” база даних (MongoDB);

– Система керування федеративним сховищем даних, яка є окремою програмою, розробленою спеціально для забезпечення функціонування сховища даних, та містить файлове сховище.

Для розроблення системи керування федеративним сховищем даних було використано платформу Microsoft.Net, мову C# та середовище розробки Visual Studio. Бібліотека класів, що поставляється з .net та мова високого рівня C# , а також методологія RAD (швидкого розроблення застосувань), на якій побудовано середовище розробки Visual Studio, дає змогу швидко створювати застосування, орієнтовані на бази даних.

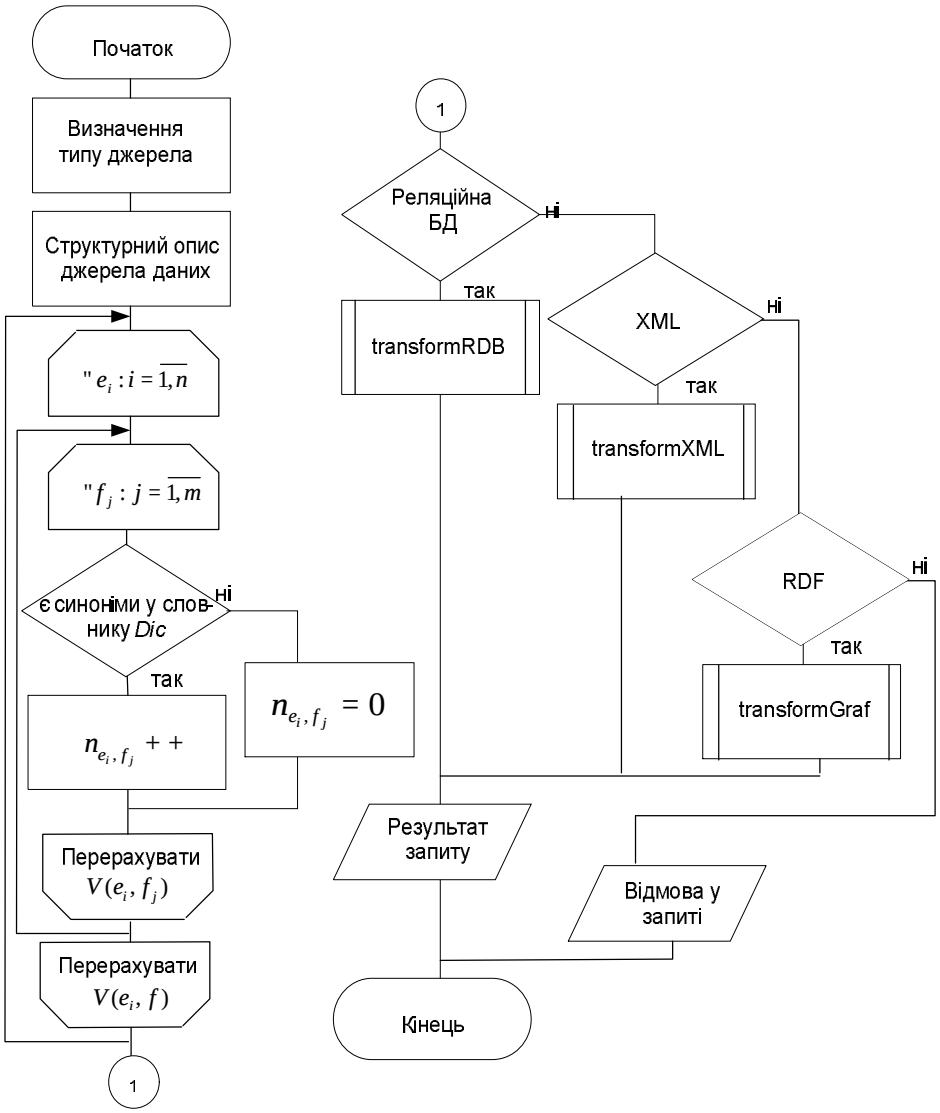

Рис. 2. Схема перетворення даних на модель "сутність-характеристика"

Насамперед здійснюється федеративне опрацювання даних з джерел. Аналізуємо відносну кількість об'єктів або документів, наявних у джерелах даних, до загальної кількості об'єктів, які потрапили до федеративного сховища. У таблиці наведено структуру детермінованих схем БД з деталізацією за областями в обсязі, достатньому для прогнозування процесів розвитку регіону.

У ній також зазначено структурованість даних і розміщення у сховищі даних (порядковий номер області та тип БД).

1	2	3
Джерело/область/таблиця	Відстань $V(e, f)$	Розміщення
Джерело даних MFUVUDB	**0.154224**	-
MFUVUDB.t_aspnet_Roles	0	RDB
MFUVUDB.t_mfuvu_Settlements	1	RDB
MFUVUDB.t_vw_mfuvu_list_sti_correspond	0.666942	RDB,XML
MFUVUDB.t_vw_mfuvu_Users	0.401167	RDB,XML
Джерело даних TransBudgDB	**0.628414**	-
TransBudgDB.t_VXXDDMMYY	0.961365	RDB
TransBudgDB.t_D_BUDG_LOCAL_DET	0.888913	RDB

Продовження таблиці

1	2	3
TransBudgDB.t_D_ECON_CRED	0.9	RDB,XML
TransBudgDB.t_D_FIN	0.916667	RDB,XML
TransBudgDB.t_INCDET	1	RDB
TransBudgDB.t_VW_EXPENSES_LVL1	0.5	RDB,XML
Область даних FUPortalDB	**0.325218**	-
FUPortalDB.t_Users	1	RDB
FUPortalDB.t_Applications	0.666667	RDB,XML
FUPortalDB.t_IsAuthCorrespond	1	RDB
FUPortalDB.t_Articles	0.375858	RDB,XML
FUPortalDB.t_Sections	1	RDB
FUPortalDB.t_People	0.227109	RDB,XML

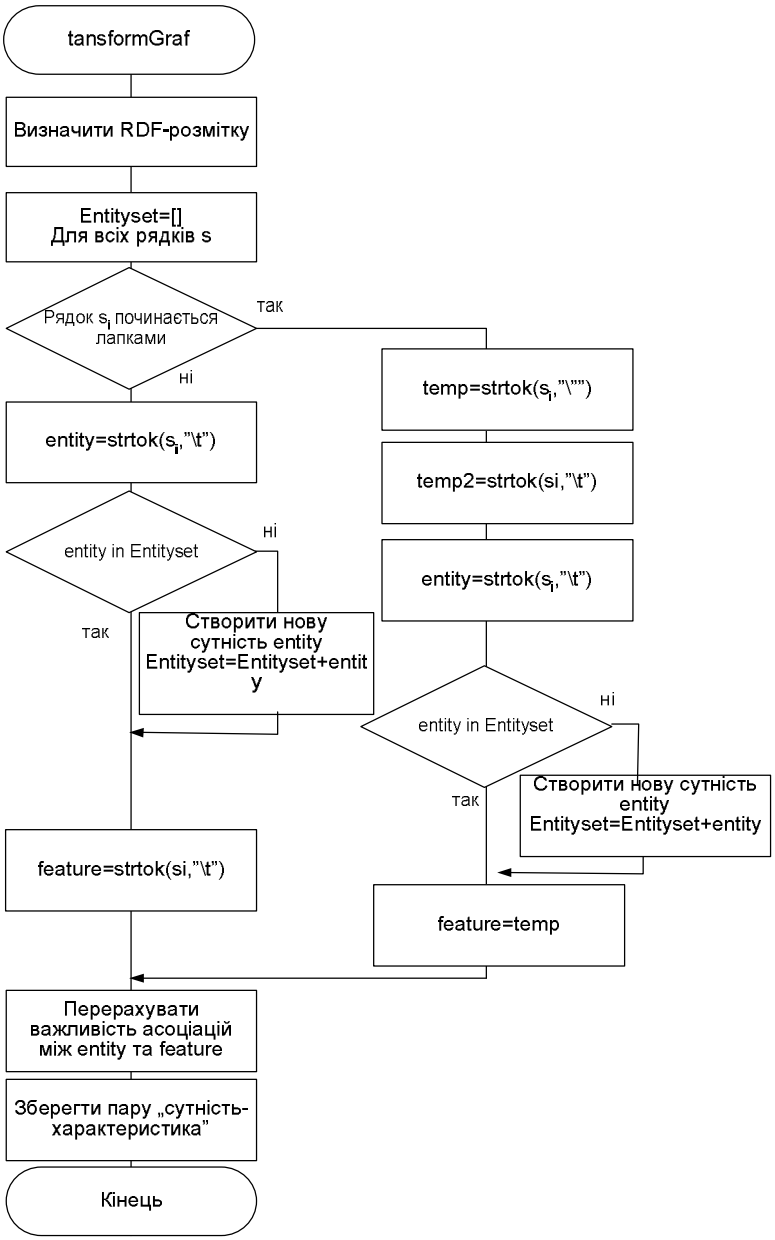

Рис. 3. Схема конвертації графової БД з врахуванням моделі "сутність-характеристика"

Розроблено засоби трансформації запитів у різних моделях даних. Результат трансформації подано на рис. 4.

	Інтегратор				DocumentDB			
вид запиту	select	select…join	insert	delete	select	select…join	insert	delete
РБД (Microsoft SQL Server Database Services)	98	95	93	93	98	92	94	94
XML (XBase)	86	82	-	-	-	-	-	-
NoSQL (mongoDB)	91	86	84	84	89	82	81	81

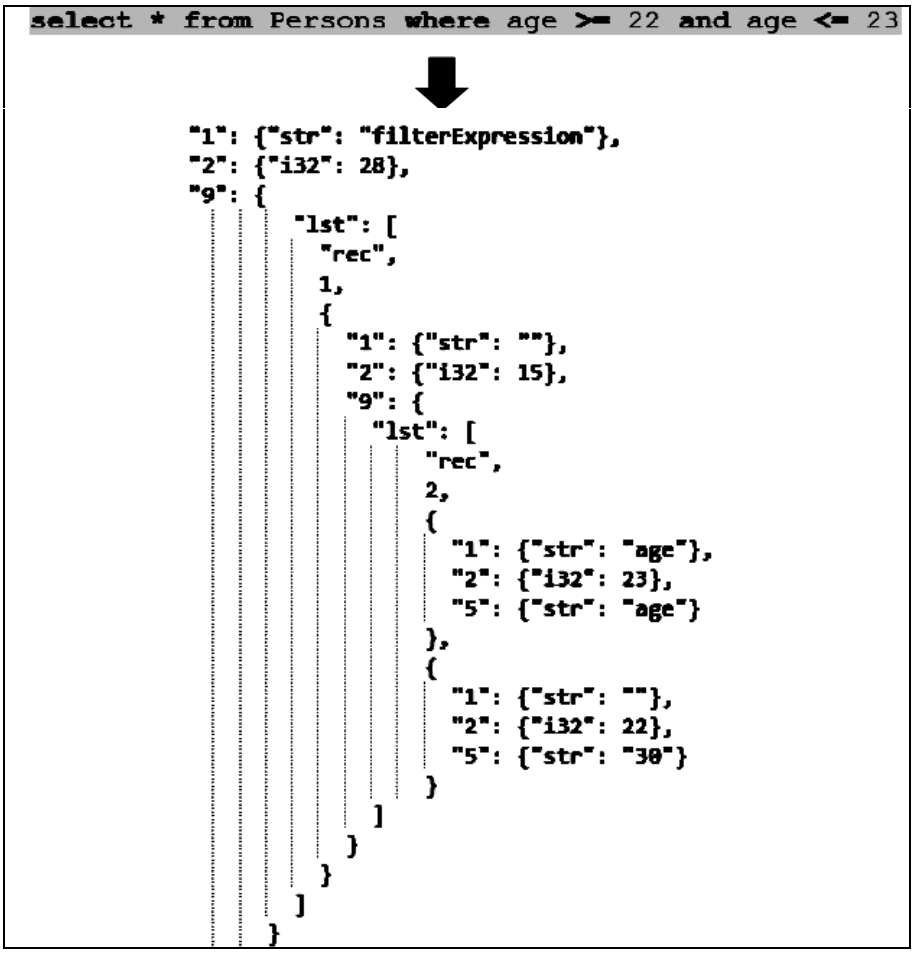

Рис. 4. Схема трансформації запитів у різних моделях даних

Наступний крок – це аналіз правильність перетворення запитів різних типів

З цією метою порівнювалися розроблені засоби в системі Інтегратор із середовищем DocumentDB, у якому є можливість формування запитів мовою SQL та Nosql. Результати порівняння подано в таблиці у відсотковому значенні правильно поданих запитів. Загальна кількість запитів, які тестували, по 50 запитів кожної категорії. Правильність формування запитів перевіряли експертно.

Висновки

1. Проаналізовано проблему подання та опрацювання різнотипових джерел даних. Обґрунтовано актуальність вирішення цієї проблеми на основі введення Великих даних, що дало змогу виділити невирішені раніше задачі з опрацювання та консолідації даних з наперед невідомих джерел.

2. Розроблено модель Великих даних "сутність-характеристика", яка дає змогу організовувати структуровані та слабкоструктуровані дані і на відміну від багатовимірної моделі не містить надлишковості.

3. Розвинено метод інтеграції даних завдяки попередньому визначенню структури даних та узгодження семантики, що на відміну від методів інтеграції даних на рівні сховища даних дало змогу інтегрувати дані з джерел з наперед невідомою структурою даних, і що, своєю чергою, дало змогу підвищити ефективність подальшого аналізу Великих даних.

1. Martin Hilbert, Big Data for Development: From Information – to Knowledge Societies, SSRN Scholarly Paper No. ID 2205145). Rochester, NY: Social Science Research Network; http://papers.ssrn.com/abstract=2205145(2013). 2. Maté, A., Peral, J., Ferrández, A., Gil, D., & Trujillo, J. (2016). A hybrid integrated architecture for energy consumption prediction. Future Generation Computer Systems. 3. Borkar, Vinayak, Michael J. Carey, and Chen Li. "Inside Big Data management: ogres, onions, or parfaits?." Proceedings of the 15th international conference on extending database technology. ACM, 2012. 4. Bu, Yingyi, et al. "HaLoop: efficient iterative data processing on large clusters." Proceedings of the VLDB Endowment 3.1-2 (2010): 285-296. 5. D.LANEY The Importance of 'Big Data': A Definition. Gartner (2012). 6. M. Beyer, Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data. Gartner. Archived from the original on 10 July 2011 (2011). 7. N. Shakhovska, Y. Bolubash: Big Data Model "entity and characteristics" / EconTechMod, 2015, Vol. 4, No. 2, 51–58. 8. Шаховська Н. Б. Модель Великих даних "сутність-характеристика" / Н. Б. Шаховська, Ю. Я. Болюбаш // Вісник Нац. ун-ту "Львівська політехніка". – 2015. – № 814 : Інформаційні системи та мережі. – С. 186–196. – Бібліографія: 11 назв. 9. Di Ciaccio A., Coli M., Angulo Ibanez J. M. (eds.): Advanced Statistical Methods for the Analysis of Large Data. Springer, Berlin (2012). 10. Fang L., Sarma A.D., Yu. C., Bohannon P.: Rex: explaining relationships between entity pairs. Proc. VLDB Endowment 5(3), 241–252 (2011).