УДК 004.5

В. Р. Сенченко, О. А. Бойченко, А. В. Бойченко
Інститут проблем реєстрації інформації НАН України
вул. М. Шпака, 2, 03113 Київ, Україна

Дослідження методів і технологій інтеграції онтологічної моделі з реляційними даними

Проведено дослідження методів і технологічних рішень щодо інтеграції онтологічної моделі з реляційними даними — Data Mapping. Основною метою є прискорення та зменшення вартості побудови онтологічних моделей систем, які оперують з розподіленими даними в гетерогенному середовищі. Запропоновано методологію застосування методів Data Mapping для систем обробки розподілених даних на прикладі побудови онтологічної моделі системи моніторингу Державного бюджету України. Показано напрями семантичної оптимізації SPARQL-запитів, які дозволяють поліпшити якість отриманих даних.

Ключові слова: *онтологічна модель, реляційна модель, OWL, SQL, SPARQL, контроль державного бюджету, Protégé 5.*

Актуальність

На сучасному етапі функціонування вже існуючих аналітичних систем основною проблемою стає не стільки питання отримання потрібного для прийняття рішення зрізу даних (з одного джерела), скільки питання інтеграції даних, що розподілені в гетерогенному середовищі з різних джерел і потрібних для прийняття збалансованих рішень [14–17]. Зважаючи на те, що більшість аналітичних систем зіштовхнуться саме з питаннями інтеграції множини джерел даних, це цілком зрозуміло. Наприклад, на верхньому рівні представлення знань предметної області розроблено онтологічну модель у вигляді RDF-графа, а величезна кількість накопичених в аналітичній системі даних зберігаються у вигляді таблиць реляційної бази даних. Тобто потрібні методи та програмні засоби, які можуть поєднати онтологічні моделі та реляційні дані. Отже, створення та застосування універсальних механізмів взаємодії між онтологічної моделлю та реляційною моделлю даних є актуальною науковою задачею і займає свою значущу роль у вирішенні загальної проблеми інтеграції даних — Data Mapping [1]. Data Mapping у сучасних комп'ютерних науках — це процес інтеграції даних, тобто находження відповідностей між різними моделями представлення даних, проте поєднаних загальною семантикою даних певної предметної області.

Встановлення відповідностей між онтологіями та реляційними базами даних, крім суто наукового, має великий практичний інтерес, оскільки дозволяє створювати системи, які оперують широким спектром моделей даних. Розробка та застосування методів і програмних засобів Data Mapping дозволяє подолати такі труднощі як:

— інтеграція даних, представлених різними моделями — онтологічними та реляційними для видобутку знань;

— взаємодія систем, в основі яких лежать застарілі моделі даних, із сучасними онтолого-орієнтованими системами;

— усунення дублювання даних при побудові систем, заснованих на знаннях.

Мета

Основною метою статті є дослідження методів і технологій інтеграції онтологічної моделі з реляційними даними, які дозволяють прискорити терміни розробки аналітичних систем, що побудовані на різних моделях даних, і підвищити їхню ефективність за рахунок використання вертикально інтегрованих рішень для побудови мережі понять, використовуючи знання, які зберігаються в об'єктно-орієнтованій структурі, що дають практичний ефект від застосування складних технологій.

Методика досліджень

На даний час теоретичною основою інтеграції різнотипних моделей даних (Data Mapping) є напрям, який визначається як Ontology-based data integration [2]. Теоретичним і практичним розвитком цього напряму є підхід Ontology-Based Data Access (OBDA), який інтегрує онтологічні моделі, що представлені у вигляді RDF-графів, з реляційними даними [3]. В підході OBDA концептуальний шар формується онтологією предметної області, яка визначає загальний словник предметної області, домен, описує структуру джерел даних, характер відносин між сутностями та аксіомами, а також за рахунок використання правил виводу дозволяє отримувати нові знання, спираючись на аксіоми. Метод інтеграції різнотипних моделей даних OBDA дозволяє виконувати запити до кількох неоднорідних джерел даних. Ефективність методу OBDA щодо інтеграції різноманітних даних залежить від версії дескриптивної логіки (DL — description logics), яку використовує програмна платформа OBDA. Як відомо, виразні можливості мови опису DL впливають на якість логічного виводу при зверненні до бази знань. Тому за цим показником розрізняють дві версії OBDA [4]:

— інтеграційну платформу OBDA з конверторами доступу до реляційних баз даних: деякі мови опису, такі як DL-Lite (і OWL 2 QL), дозволяють зменшити споріднені запити за онтологіями до запитів першого порядку над стандартними реляційними базами даних;

— інтеграційну платформу OBDA з двигунами каталогу: інші DL, що охоплюють логіку в сімействі EL (OWL 2 EL), Horn-SHIQ і Horn-SROIQ, підтримують зниження даних і можуть використовуватися з двигунами каталогу.

На рис. 1 наведено схему взаємодії різнотипних моделей даних на основі технології Ontology-Based Data Access.

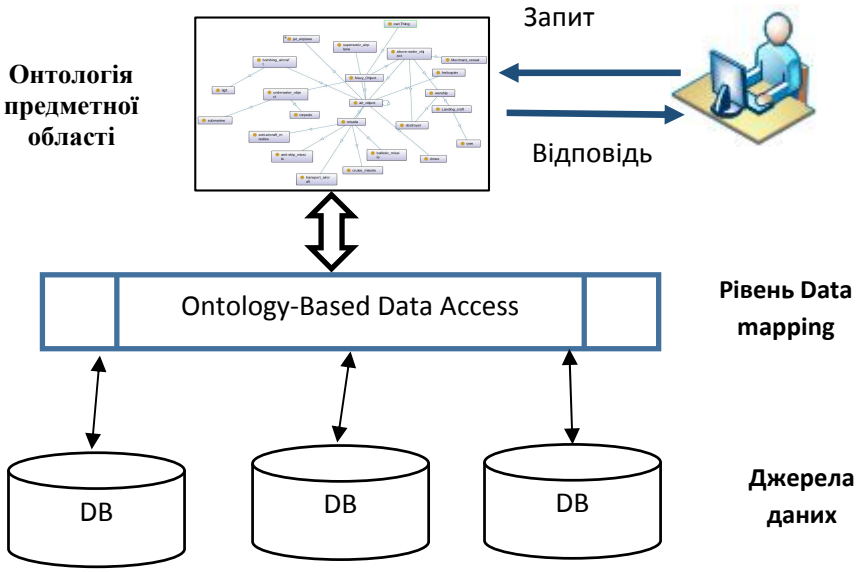

Рис. 1. Схема взаємодії онтології з реляційною моделлю даних за допомогою Ontology-Based Data Access

Нагадаємо, що база знань предметної області — SD (subject domen), яка побудована на мові дескрипційних логік, являє собою поєднання тверджень, які містяться в ABox(SD) та TBox(SD) [5]:

Knowledge base (SD) = TBox(SD) + ABox(SD),

де TBox(SD) — опис ієрархії класів <Class hierarch> предметної області — SD; ABox(SD) — множина властивостей, якими описуються класи та їхні екземпляри цієї ПрО — SD.

Для отримання знань користувачу системи потрібно формувати DL-запит до онтології, тобто до множини ABox(SD), де саме й зберігаються знання, які містяться в екземплярах онтології у вигляді конкретних значень властивостей класів онтології і можуть бути як числовими, так і логічними даними. Бажано, щоб онтологія предметної області містила якомога більше конкретних екземплярів класів з описом усіх можливих фактів для поліпшення якості логічного виводу. Але практично цього дуже складно досягти, оскільки конкретні дані можуть розміщуватися в різних СУБД і бути територіально розподіленими. Крім того, для перетворення реляційних даних в екземпляри онтології потрібні значні фінансові та часові витрати.

Виходом з цієї ситуації є застосування саме інтеграційної платформи OBDA, яка складається з конверторів доступу онтології до реляційних даних БД [6]. Як відомо, при відпрацюванні DL-запиту машина логічного виводу розшукує дані з простору, що сформований на базі ABox(SD) (онтологічна модель), і видає користувачеві відповідь на DL-запит.

Користувач системи формулює запит q у словнику даної онтології TBox(SD). Онтологічна модель підключається до джерел даних через декларативну специфікацію, надану в термінах відображення, включаючи класи та їхні властивості. Для отримання потрібних даних з реляційної СУБД необхідно конверту-

вати логічний DL-запит (онтологічній моделі) в SQL-запит реляційної СУБД за допомогою відображення, який створено експертом домену або автоматично вилучено. Це означає, що, по-перше, об'єктна структура реляційної СУБД (структура класів) повинна відповідати ієрархії класів — TBox(SD) (онтологічній моделі), а, по-друге, множина атрибутів реляційної моделі даних повинна співвідноситися з множиною властивостей ABox(SD) онтологічної моделі.

Базовою платформою для розробки онтологій при проектуванні аналітичних систем, що засновані на знаннях, обрано редактор Protégé 5 [7]. Для інтеграції онтологічних і реляційних моделей даних у редакторі Protégé 5 застосовується платформа Ontop, яка є конкретною реалізацією методу інтеграції даних на основі підходу Ontology-Based Data Access [8]. Платформа OBDA підтримує всі рекомендації W3C, що пов'язані з OBDA: OWL 2 QL, R2RML, SPARQL, SWRL та режими управління OWL 2 QL у SPARQL. Крім того, програмна платформа Ontop підтримує всі основні реляційні бази даних з відкритим кодом, що є дуже важливою умовою для зменшення вартості створюваних систем. Онтологія Ontop дозволяє використовувати RDFS та OWL 2 QL як онтологічні мови, що гарантують можливість переформатування SPARQL-запитів за онтологічною моделлю на еквівалентні SQL-запити до реляційних баз даних.

З точки зору перспектив застосування в нових проектах, Ontop — це відкрита система Ontology-Based Data Access, яка розроблена Free University of Bozen-Bolzano (Вільний університет міста Бозен-Больцано, Італія) та розповсюджується за вільною ліцензією Apache [9].

Програмна платформа Ontop реалізує методи взаємодії онтології з реляційними структурами та розглядає реляційні бази даних як віртуальні графи RDF, пов'язуючи сутності онтології (класи та властивості) з табличною організацією даних через відповідні відображення класів у онтологіях з класами таблиць реляційної бази даних, а властивості класів онтології — з атрибутами таблиць. При такому підході віртуальний графічний RDF-масив даних може бути запитаний за допомогою мови SPARQL шляхом перекладу SPARQL-запитів у SQL-запити, використовуючи механізми формування запитів у реляційних базах даних.

Розглянемо застосування програмної платформи Ontop на прикладі системи аналізу держбюджету України. База даних системи моніторингу держбюджету України складається з реляційних таблиць, які містять звіти Державного казначейства про виконання дохідної і видаткової частин як державного, так і міських бюджетів, а також регіональний розріз. Крім того, база даних містить дані про кредитування та заборгованості бюджетних установ [10, 11].

Слід нагадати, що процес семантичного моделювання із застосуванням Web Ontology Language складається з наступних етапів [17]:

1) визначення сутностей базових понять — класів, сутностей, категорії (<Active Ontology>, <Entities>, <Classes>), які описують SD;

2) визначення множини властивостей, які описують властивості сутності SD і дозволяють, у кінцевому розумінні, створювати базу знань предметної області;

3) встановлення відносин між сутностями предметної області та їхніми властивостями із застосуванням механізмів формування предикатів (<Object Properties>);

4) встановлення числових або логічних обмежень, які використовуються при опису властивостей екземплярів (<Individuals>) бази знань;

5) формування бази знань із застосуванням механізму опису екземплярів бази знань та її наповнення (<Individual by class>).

Сучасні інформаційні технології та інструментальні засоби дозволяють суттєво прискорити сам процес створення онтологічної моделі предметної області — **Knowledge base (SD)**. Так, наприклад, для отримання структури БД системи моніторингу держбюджету України [10], яка працює під управлінням СУБД Oracle 11g, застосовувався інструментальний CASE-засіб Toad for Oracle 12 [12]. Завдяки цьому CASE-засобу отримано реальну об'єктно-орієнтовану структуру БД з відповідними таблицями та їхніми реквізитами (рис. 2). Крім того, CASE-засіб Toad for Oracle 12 дозволяє представити структуру у вигляді або XML-файлу або SQL-скриптів, які можуть використовуватися у подальшому для автоматизованого формування моделі онтології. У редакторі Protégé 5 це можна виконати за допомогою інструментальної панелі <Tools>, яка дозволяє отриману об'єктно-орієнтовану структуру БД перетворити на відповідну онтологію (класи, властивості, відношення), використовуючи наявні процедури.

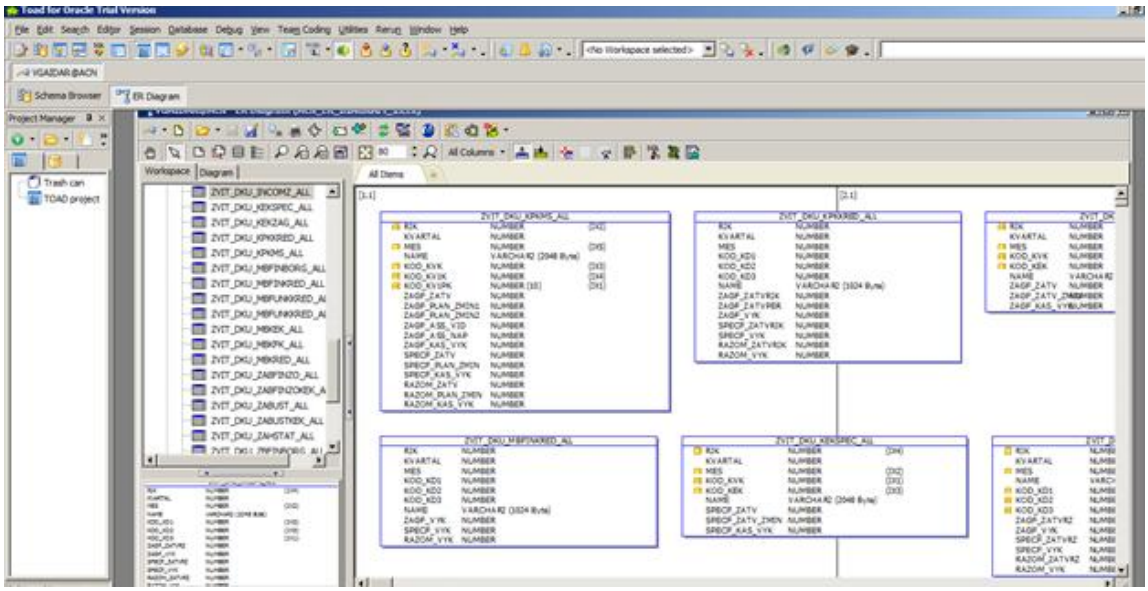

Рис. 2. Отримання структури БД системи аналізу держбюджету України
за допомогою CASE-засобу Toad for Oracle 12

Отже, в результаті використання зазначених технологій і CASE-засобів, отримано модель онтології системи моніторингу держбюджету України, яка представлена у вигляді RDF-графа на рис. 3. Відповідно до семантики предметної області онтологічна модель системи моніторингу держбюджету України — **Tbox(SD)** містить чотири суперкласи — <Class hierarch>:

1) <BUDGET_REVENUES> (виконання доходної частини бюджету) — складається з класів — назв звітів Держказначейства, які співвідносяться з відповідними реляційними таблицями БД системи моніторингу держбюджету України у напрямку «ДОХОДИ». Наприклад, клас <ZVIT_DKU_INCOM_ALL> моделі он-

тології співвідносяться з аналогічною назвою реляційної таблиці БД загального звіту про виконання доходів держбюджету, клас <ZVIT_DKU_INCOMM_ALL> — співвідносяться з назвою таблиці загального звіту про виконання місцевих бюджетів, клас <ZVIT_DKU_INCOMREG_ALL> співвідносяться зі звітом про виконання доходів місцевих бюджетів у регіональному розрізі тощо;

2) <BUDGET_EXPENDITURES> (виконання видатків бюджету — загального та спеціального фондів, зведеного бюджету за різними бюджетними класифікаторами, а також видатків місцевих бюджетів, включаючи регіональний розріз) — складається з класів — назв звітів Держказначейства, які співвідносяться з відповідними реляційними таблицями БД системи моніторингу держбюджету України у напрямку «ВИДАТКИ». Так, клас моделі онтології <ZVIT_DKU_DBKEK_ALL> співвідноситься з аналогічною назвою реляційної таблиці БД про звіт виконання видатків держбюджету за кодами економічної класифікації, а клас моделі онтології <ZVIT_DKU_EXPENSKVREG_ALL> співвідноситься з назвою реляційної таблиці звіту про виконання місцевих видатків у регіональному розрізі. Аналогічно визначено інші класи;

3) <CREDIT> (кредитування) — складається з класів — назв звітів Держказначейства про кредитування державних установ відповідно до програм і кодів бюджетної класифікації, які співвідносяться з відповідними реляційними таблицями БД системи моніторингу держбюджету України у напрямку «КРЕДИТУВАННЯ», наприклад, клас <ZVIT_DKU_DBFINKRED_ALL>, <ZVIT_DKU_DBFUNKKRED_ALL>, ZVIT_DKU_DBKRED_ALL> та інші;

4) <DEBT_BUDGET> (боргові зобов'язання) — складається з класів — назв звітів Держказначейства про боргові зобов'язання державних установ, наприклад, класи <ZVIT_DKU_DBFINBORG_ALL>, <ZVIT_DKU_DBFINBORG_ALL_BKP>, <ZVIT_DKU_MBFINBORG_ALL> та інші.

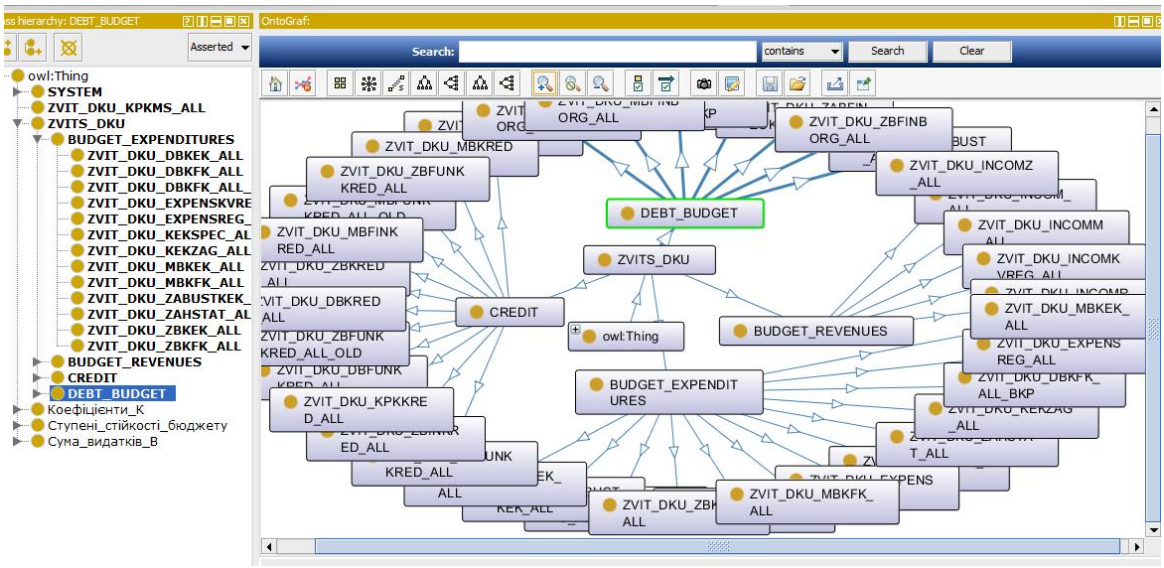

Рис. 3. RDF-граф онтології системи моніторингу держбюджету України

Створення моделі онтології, окрім формування таксономії предметної області **Tbox(SD)**, обов'язково передбачає процес визначення підмножини властивостей

— **Abox(SD)**, саме за допомогою яких і формуються SPARQL-запити будь-якої складності [13, 14]**.** Зрозуміло, що ефективність SPARQL-запитів до БД цілком залежить від якості моделі властивостей <Datatype Properties> онтології. Слід нагадати, що модель властивостей <Datatype Properties> повинна співвідноситися з атрибутами відповідних реляційних таблиць БД. Отже, для формування відносно повної моделі онтології було використано близько 100 атрибутів реляційних таблиць БД, які складають підмножину <Datatype Properties> моделі онтології бюджетного процесу. Для зручності використання такої кількості властивостей підмножина OWL:DataPropertty згрупована за семантичною ознакою у вигляді ієрархічної структури та супроводжуються коментарями (рис. 4).

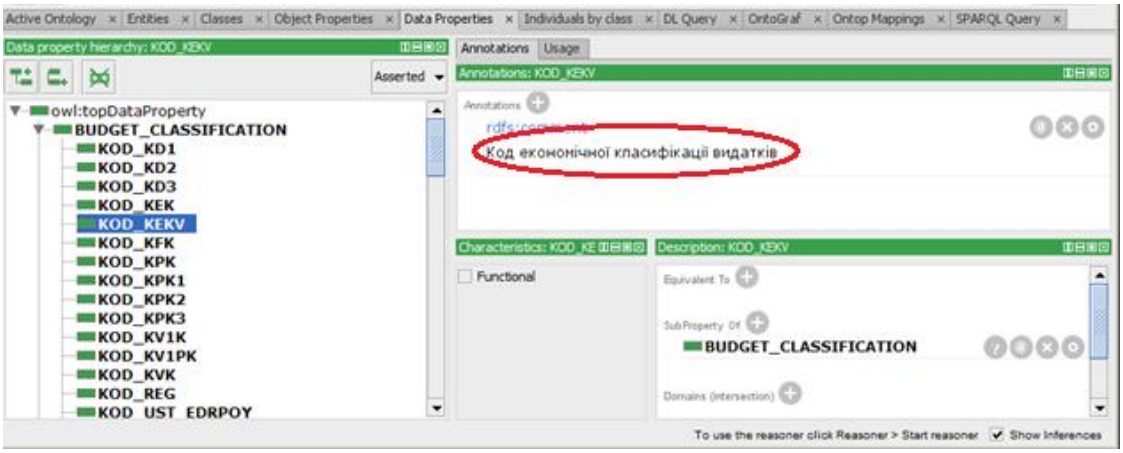

Рис. 4. Приклад відображення OWL:DataPropertty
для моделі онтології моніторингу бюджетного процесу

У редакторі Protégé 5 платформа Ontop доступна через інтерфейс командного рядка <Ontop Mapping> (рис. 5). Плагін Ontop забезпечує графічний інтерфейс для різних ключових функціоналів, пов'язаних з OBDA: редагування відображення, виконання SPARQL-запитів, перевірка послідовності онтології, автоматичне завантаження онтології і відображення з бази даних, імпорт та експорт відображення R2RML.

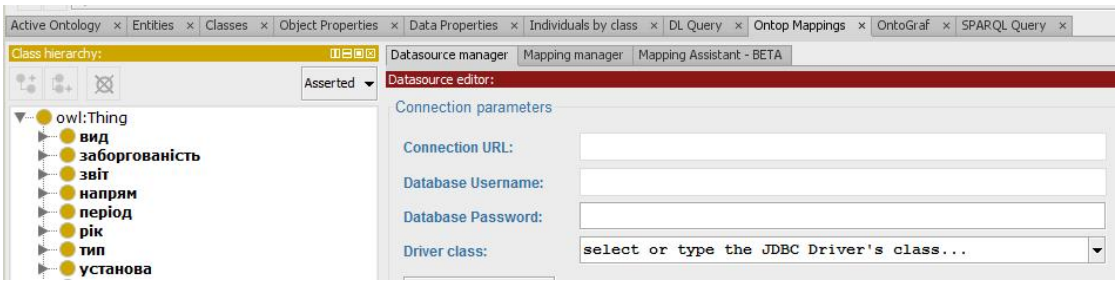

Рис. 5. Головне меню інтеграційної платформи Ontop

Для формування запиту до даних в інструментальній панелі <Datasource manager> (редактора Protégé 5) спочатку необхідно налаштувати драйвер для звернення до СУБД, яка використовується у проекті, та заповнити поля персональни-

ми даними. Оскільки БД системи моніторингу держбюджету України працює під управлінням СУБД Oracle 11g, у полі <Driver class> (панелі <Datasource manager>) обираємо відповідний драйвер **<oracle.jdbc.driver.OracleDriver>** та заповнюємо поля <Database Password> <Database Username> персональними даними (рис. 6).

Рис. 6. Налаштування драйверу та персональних даних на звернення до БД

Далі виконується безпосередньо процедура звернення до обраного ресурсу БД, і платформа Ontop переключається в режим <Mapping Assistant>. Як можна бачити з рис. 7, при коректному під'єднанні до ДБ, Ontop зчитує модель онтології, а у правому фреймі з'являється можливість вибору класу онтологічної моделі та їхніх властивостей, за якими має формуватися SPARQL-запит. Наприклад, обрано клас <ZVIT_DKU_BDKEK_ALL> — звіт про виконання видатків держбюджету за кодами економічної класифікації. У центральному фреймі <Mapping Assistant> розкривається поле для формування запиту мовою SQL, використовуючи візуальний редактор формування запиту.

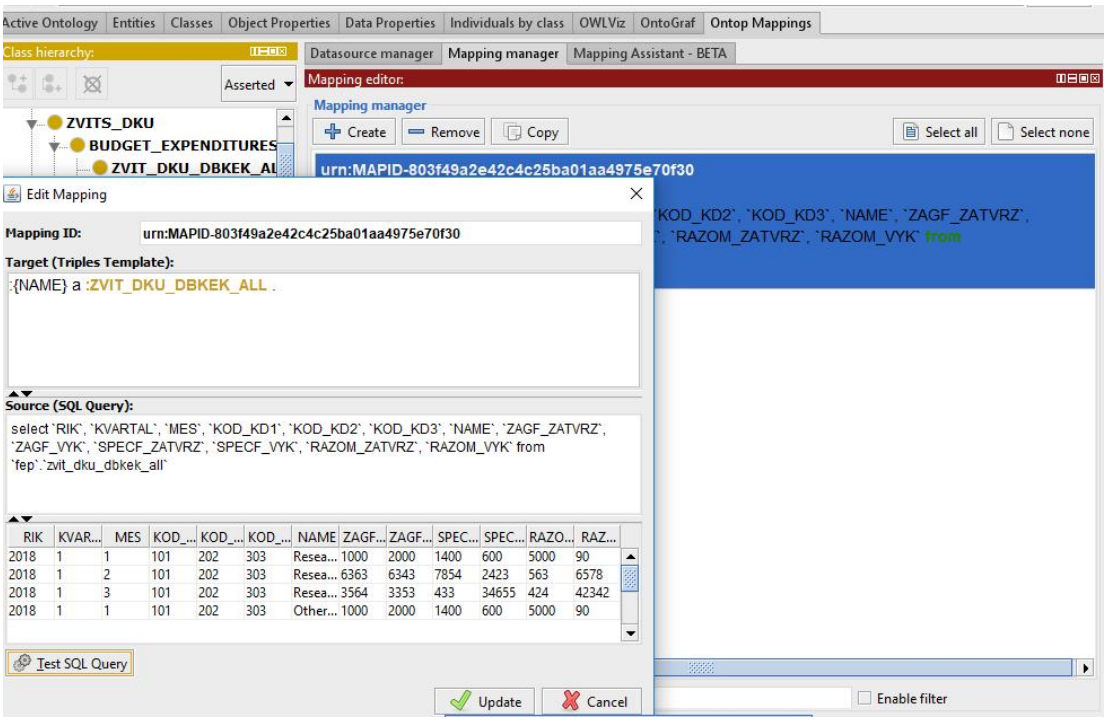

Рис. 7. Формування SPARQL-запиту

Приклад скрипту згенерованого SPARQL-запиту наведено нижче.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ? ZVIT_DKU_BDKEK_ALL?type ?VGAIDAR.KLASS_ANALIZ
?object ?subject ?position_label ?department ?pemail
WHERE {
?infoResource code:hasPublicationVenue ?pubVenue.
?pubVenue rdf:type zvit: ZVIT_DKU_BDKEK_ALL.
?infoResource rdf:type ?anyType.
?anyType rdfs:label ?type.
?infoResource vivo:informationResourceInKOD ?id.
?KOD rdf:type vivo:KOD.
?KOD vivo:linkedclass ?classURI.
?classURI rdfs:label ?class.
?infoResource rdfs:label ?infoResource_label.
OPTIONAL { $classURI foaf:KOD_KEK ?object }.
OPTIONAL { $classURI foaf:NAME_GALUS ?subject }.
OPTIONAL { $classURI vivo:KVARTAL_AN ?pemail }.
?classURI vivo:personInPosition ?position.
?position vivo:positionInOrganization $organization.
?organization rdfs:label ?department.
?position rdfs:label ?position_label.
}
```

Програмний застосунок Ontop <Mapping Assistant> перетворює SPARQL-запит онтологічної моделі нв SQL-запит реляційної бази даних. Згенерований SQL-запит уже може бути виконаний драйвером СУБД Oracle 11g, який повертає відповідь у вигляді зрізу даних (рис. 8).

Слід зауважити, що отримані дані не обов'язково ефективні. Це пояснюється тим, що вони можуть містити надмірні самоз'єднання, які не завжди коректно можуть поєднувати логічні правила складного SPARQL-запиту. Тому для підвищення продуктивності SPARQL-запитів, слід застосовувати метод семантичної оптимізації запитів.

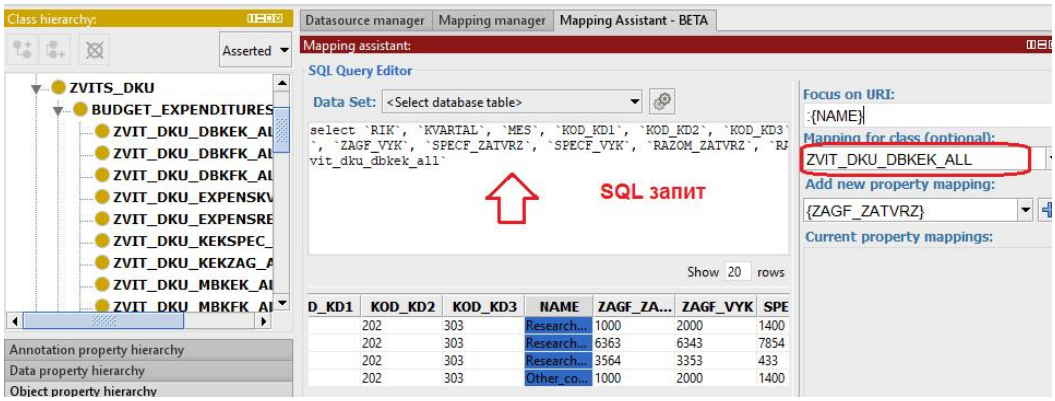

Рис. 8. Результати виконання SQL-запиту до БД системи моніторингу державного бюджету

Отже, семантична оптимізація — дуже важлива фаза формування коректного SPARQL-запиту, оскільки надлишковість даних, що отримані в процесі виконання SQL-запиту до реляційної БД, часто ускладнює аналіз даних кінцевим користувачем. Одним із засобів, за допомогою якого можна виконувати семантичну оптимізацію, є Semantic Query Optimization — закладка <Ontop> на інструментальній панелі редактора Protégé 5. Закладка <Ontop> містить методи семантичного аналізу онтології, наприклад, перевірку невідповідності щодо непересічних і функціональних властивостей створеної онтології, або аналіз наявності порожніх понять і ролей у створеній онтології та інші (рис. 9) Семантична оптимізація виконується за рахунок потужної програми логічного виводу Reasoner — Ontop 1.18.1. [9], яка поширюється за ліцензією The Apache Software License, Version 2.0.

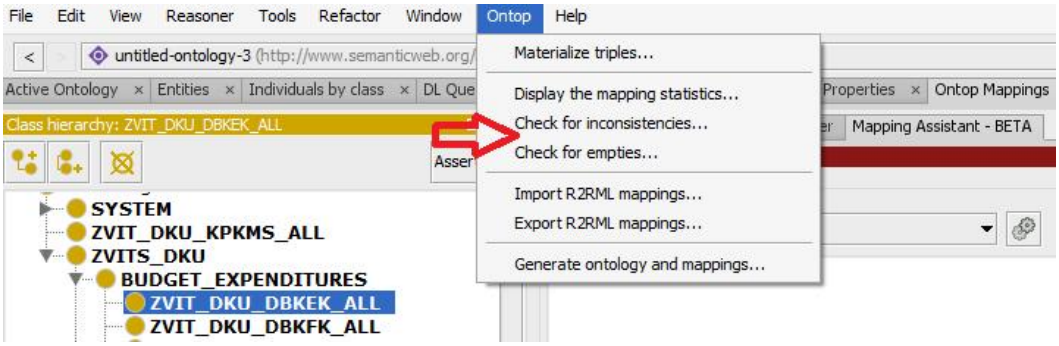

Рис. 9. Вибір процедури семантичної оптимізації засобами Ontop

Програма логічного виводу Ontop 1.18.1. доповнює платформу Ontop, додаючи інтуїтивно зрозумілий конструктор візуальних запитів.

Висновки

За результатами описаних досліджень одержано такі основні результати:

— запропоновано методологію застосування методів Data Mapping для систем обробки розподілених даних (на прикладі побудови онтологічної моделі системи моніторингу Державного бюджету України). Надано результати виконання SPARQL-запиту до реляційних даних бюджетного процесу під СУБД Oracle 11g;

— показано необхідність вертикальної інтеграції технологій та інструментальних засобів для побудови мережі понять, використовуючи знання, що зберігаються в об’єктно-орієнтованій структурі БД для створення ефективної аналітичної системи, заснованої на різних моделях даних;

— наведена інтеграційна технологія дозволяє як прискорити сам процес створення онтологічної моделі, так і оптимізувати SPARQL-запити, що суттєво підвищує ефективність аналітичної системи в цілому.

Запропонована методологія застосування методів Data Mapping для систем обробки розподілених даних може використовуватися в інформаційно-аналітичних системах органів державної влади та науково-освітніх закладах.

1. Zohra Bellahsene, Angela Bonifati, Fabien Duchateau3 and Yannis Velegrakis. On Evaluating Schema Matching and Mapping. URL: https://perso.liris.cnrs.fr/angela.bonifati/pubs/BellahseneBDV11.pdf

2. Ontology-based Approach for Information Fusion, Anne-Claire Boury-Brisse. URL: https://pdfs.semanticscholar.org/9cb1/8a7f733f45b8ec7ba53ce159bd567164d96f.pdf

3. Diego Calvanese, Ontology-Based Data Access:From Theory to Practice. URL: http://www.inf.unibz.it/~calvanese/presentations/BDA-2012-obda-calvanese.pdf

4. Kontchakov Roman, Mariano Rodr´ıguez-Muro, Zakharyaschev Michael, Ontology-Based Data Access with Databases:A Short Course. URL: http://www.dcs.bbk.ac.uk/~roman/papers/RW-Chapter.pdf

5. OWL 2 Web Ontology Language Document Overview (Second Edition) W3C. Recommendation 11 December 2012. URL: http://www.w3.org/TR/owl2-overview/

6. RDF Model Theory. URL: https://www.w3.org/TR/2002/WD-rdf-mt-20020429/

7. Protégé — Free, open-source ontology editor and framework for building intelligent systems. URL: http://protege.stanford.edu

8. OntoStudio. URL: http://www.semafora-systems.com/en/products/ontostudio/

9. Ontop a platform to query DBs as Virtual RDF Graphs using SPARQL. URL: https://sourceforge.net/projects/ontop4obda/

10. Сенченко В.Р., Коваль О.В., Діденко О.О. Система моніторингу державного бюджету України. *Research bulletin of National Technical University of Ukraine «Kiev Polytechnic Institute»*. 2013. № 6(92). С. 39–51.

11. Сенченко В.Р. Методи та технології обробки даних у системі моніторингу та аналізу держбюджету: зб. наук. праць «Інформаційні технології та спеціальна безпека». 2016. № 1. С. 50–64. ISSN 2414-5947.

12. Toad for Oracle. URL: https://www.quest.com/products/toad-for-oracle/

13. Ontop: Answering SPARQL queries over relational databases, Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebria, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, Guohui Xiao. URL: http://eprints.bbk.ac.uk/15625/1/main.pdf

14. Calvanese D., Cogrel B., Komla-Ebri S., Kontchakov R., Lanti D., Rezk M., Rodriguez-Muro M. and Xiao G. Ontop: Answering SPARQL queries over relational databases. Semantic Web. URL: http://www.semantic-web-journal.net/system/files/swj1004.pdf

15. Бойченко А.В. Дослідження та розробка моделей предметних областей у задачах сценарного аналізу. AIIS-2016 International scientific youth school «Systems and measures of artificial intelligence». 2016. С. 14–17.3-9

16. Christian Ferber and Martin Hepp. Using sparql and spin for data quality management on the semantic web. In:Business information systems: Springer, 2010. P. 35–46.

17. Senchenko V.R., Koval A.V., The technology of semantic modeling for knowledge management system in environment Protégé. Материалы XVII международной научно-практической конференции «Информационные технологии и безопасность». Киев, 2017. Вып. 17. С. 211–234.

18. Novogrudska R.L., Globa L.S., Koval O.V., Senchenko V.R. Ontology for Applications Development. Chapter 2 «Ontology in Information Science». Book edited by Ciza Thomas. Print ISBN 978-953-51-3887-7. Published: March 8, 2018. P. 29–53: URL: http://dx.doi.org/10.5772/intechopen.74042

Надійшла до редакції 02.08.2018