УДК 004.9

М. В. Микитюк

ОСОБЛИВОСТІ РОЗРОБКИ БАЗИ ДАНИХ МОВОЮ JAVA

Вінницький національний технічний університет

Анотація

Розглядаються особливості розробки програмного забезпечення мовою Java з використанням баз даних, розглянуто основні засоби та їх використання, обґрунтовані висновки на основі проведеного дослідження розробки баз даних.

Ключові слова:

Java, СУБД, база даних, Hibernate, JDBC, SQL.

Abstract

The article considers Java software development features using databases. It reviews the key facilities of them and their usage. That review had led to the conclusion of database development.

Keywords:

Java, DBMS, database, Hibernate, JDBC, SQL.

Ефективне управління даними в системах промислової автоматизації стало однією з ключових проблем 21-го століття, так як компанії у всіх галузях промисловості прагнуть підвищити ефективність і збільшити обсяги виробництва. Це може бути керований перехід від ручного збору та аналізу даних до автоматизованих систем; або управління все більш великими обсягами “живих” даних, отриманих в рамках сучасних, більш складних, продуктів автоматизації.

Загальна вимога для всіх цих додатків - використання баз даних, для обробки та зберігання отриманих даних. Java дозволяє працювати з багатьма видами СУБД, такими як MySQL, PostgreSQL, Oracle, Microsoft SQL Server та інші[1]. Для цього вона використовує JDBC (Java Database Connectivity) драйвер. Якщо спробувати визначити JDBC простими словами, то JDBC це опис інтерфейсів і деяких класів, які дозволяють працювати Java з базами даних. Головним принципом архітектури є уніфікований (універсальний, стандартний) спосіб спілкування з різними БД. Тобто з точки зору програми мовою Java спілкування з Oracle або PostgreSQL не повинно відрізнятися. Самі SQL-запити можуть відрізнятися за рахунок різного набору функцій для дат, рядків та інших типів даних, але алгоритм і набір команд для доставки запиту на SQL-сервер та отримання даних від сервера відрізнятися не повинні[2, 6].

Наш додаток не повинен “думати” над тим, з якою базою він працює - всі бази повинні виглядати для нього однаково. Але при всьому бажанні, внутрішній принцип передачі даних для різних СУБД різний. Правила передачі байтів для Oracle відрізняється від правил передачі байтів для MySQL і PostgreSQL.

Як випливає з рисунка 1, додаток працює з абстракцією JDBC у вигляді набору інтерфейсів, а реалізація для кожного типу СУБД використовується власна. Ця реалізація називається "JDBC-драйвер". Для кожного типу СУБД використовується свій драйвер[3].

Така система дозволяє завантажити драйвер для конкретної СУБД і одноманітно використовувати його компоненти за рахунок того, що звернення до них відбувається не безпосередньо, а через інтерфейси.

Тобто, додаток в принципі не розрізняє, звертається він до Oracle чи PostgreSQL - всі звернення йдуть через стандартні інтерфейси, за якими "ховається" різна реалізація.

Сучасні програми, написані мовою Java, як правило, працюють з СУБД не на пряму, а використовують Java Persistence Application Programming Interface (JPA). JPA - це API, який був доданий до складу платформ Java SE та Java EE, починаючи з п’ятої версії, для того, щоб було зручно зберігати об'єкти у базу даних і отримувати їх назад. Існує велика кількість ORM-бібліотек (ORM - Object-Relational Mapping) для Java, які реалізують специфікацію JPA.

Одна з популярних ORM-бібліотек - Hibernate ORM. На даний момент Hibernate є проектом компанії RedHat. Сервер додатків WildFly і JBoss також використовують Hibernate у якості ORM.

Hibernate ORM використовує об'єктно-орієнтовану мову запитів Hibernate Query Language (HQL) для написання запитів до сутностей, які зберігаються у базі даних[4, 6].

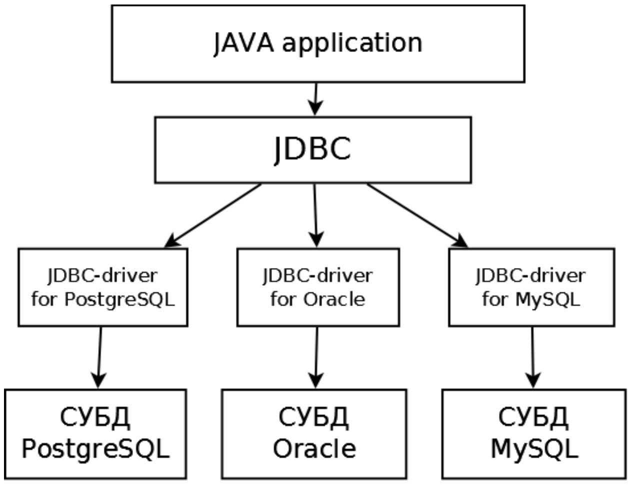

Рисунок 1 - Архітектура JDBC

Hibernate не тільки забезпечує автоматичне генерування Java-класів на основі таблиць БД (а також приведення базових типів Java до типів SQL), а й надає механізми формування запитів та вибірок даних (рис. 2). Також він може істотно знизити час на розробку, яка раніше виконувалася шляхом ручної роботи з даними із використанням SQL і JDBC. На відміну від інших persistence-рішень, Hibernate не приховує від вас можливість використання всіх можливостей SQL, і гарантує, що ваші доробки в реляційні технології та знання як і раніше мають силу.

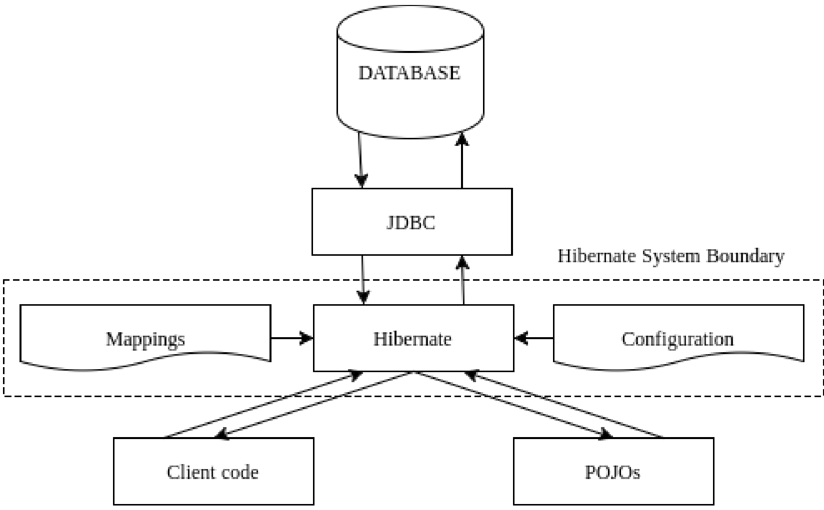

Рисунок 2 - Взаємодія з ORM

Переваги використання Hibernate:

- Hibernate забезпечує відображення Java класів в таблиці бази з використанням XML-файлів без написання коду;
- надає прості API-інтерфейси для зберігання та вилучення об'єктів Java безпосередньо з бази даних;
- при будь-якій зміні БД або у таблиці, потрібно лише змінити властивості XML-файлів;
- дозволяє абстрагуватися від незнайомих типів SQL і надає можливість працювати зі знайомими об'єктами Java.

Hibernate підтримує майже всі основні типи СУБД такі як DB2/NT, MySQL, PostgreSQL, FrontBase, Oracle, Microsoft SQL Server Database, Sybase SQL Server, Informix Dynamic Server.

Також Hibernate дозволяє використовувати безліч інших сучасних технологій, в тому числі такі:

- XDoclet Spring, який підтримує Attribute-орієнтоване програмування;

- J2EE, який надає API та виконавче середовище для розробки і виконання корпоративного програмного забезпечення, включаючи мережеві та веб - сервіси, та інші масштабовані, розподілені додатки.
- Maven, який використовується для управління (management) та збирання (build) програм.

Висновки

Таким чином, Hibernate може бути не при кращим рішенням для додатків, що зберігають всю свою бізнес-логіку у збережених процедурах, він скоріше підходить для об'єктно-орієнтованих моделей і логіки в середньому бізнес - шарі додатка, написаному мовою Java. Однак, Hibernate абсолютно точно може допомогти вам позбутися від інкапсуляції логіки специфічного SQL-коду, а також впоратися з повсякденними завданнями трансляції результатів ваших запитів з табличного представлення у граф об'єктів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Как работать с MySQL в Java - Wikipedia, the free encyclopedia [Електронний ресурс]. // - Как работать с MySQL в Java: http://devcolibri.com/1394, вільний, - Загол. з екрану.
2. Processing SQL Statements with JDBC - Wikipedia, the free encyclopedia [Електронний ресурс]. // - Processing SQL Statements with JDBC: https://docs.oracle.com/javase/tutorial/jdbc/basics/processingsqlstatements.html, вільний, - Загол. з екрану.
3. Х. М. Дейтел Програмування на Java / Х. М. Дейтел, П. Дж. Дейтел, С. И. Сантрі // М. : ДиаСофтЮП. - 2010. - С. 106.
4. Джеймс Р. Грофф SQL: полный справочник / Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель // М.: Вильямс, - 2011.
5. Александр Бондарь Microsoft SQL Server 2012 // Санкт- Петербург: БХВ-Петербург. - 2013.
6. И.И. Семенова Разработка клиент-серверных приложений в MicrosoftSQL Server 2005 // Омск: СибАДИ. - 2010.

Микитюк Максим Васильович, - ст. гр. 1КІ-13б факультету інформаційні технології та комп'ютерної інженерії. Вінницький національний технічний університет , м. Вінниця, maksymmikitiuk@gmail.com.
Науковий керівник: **Кисюк Дмитро Васильович -** асист. кафедри ОТ, Вінницький національний технічний університет, м. Вінниця, kneimad@gmail.com.

Maksym V. Mikitiuk - Department of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email : maksymmikitiuk@gmail.com.
Supervisor: **Dimitry V. Kisyuk**, assistant, Department of Information Technologies and Computer Engineering, Vinnytsia National Technical University.