

ELABORAREA UNEI APLICAȚII CU MANAGERUL PENTRU BAZE DE DATE H2

Veaceslav IURCO, IPLT M. Eminescu

<https://orcid.org/0000-0001-7595-3330>

Olga CERBU, State University of Moldova

<https://orcid.org/0000-0002-6278-7115>

Gabriel TUREȚCHI, Technical University of Moldova

<https://orcid.org/0000-0001-8123-6709>

Vadim GONȚA, CEITI

<https://orcid.org/0000-0002-7429-4262>

Rezumat. Această lucrare descrie sistemul de management al bazei de date H2 (Hypersonic 2). H2 folosește SQL pentru gestionarea DB. Este posibil să gestionați bazele de date existente scrise în diferite limbi prin intermediul consolei web H2 utilizând mai multe funcții compatibile cu bazele de date mai mari existente. Pentru aplicații mici, H2 poate fi avantajos, deoarece este pur și simplu inclus în aplicație.

Cuvinte cheie: H2 – Hypersonic 2; baze de date; operații CRUD; SQL.

DEVELOPING AN APPLICATION WITH THE DATABASE MANAGER H2

Abstract. This work describes the H2 (Hypersonic 2) database management system. H2 uses SQL for DB management. It is possible to manage existing databases written in various languages through the H2 web console using multiple functions compatible with existing larger DBs. For small applications, H2 can be advantageous because it is simply included in the application.

Keywords: H2 – Hypersonic 2; Data Bases; CRUD operations; SQL.

Introducere

H2 este un sistem de gestiune al bazelor de date relaționale (SGBD), opensource, independent de platformă, scris complet în limbajul Java. Autorul sistemului este Thomas Mueller, prima lansare a avut loc în decembrie 2005. În prezent codul sursă este accesibil programatorilor cointeresați în dezvoltarea proiectului, se perfecționează în continuu și se implementează componente noi care ajustează actualitatea utilizării acestui proiect în mai multe domenii, cu funcționalități mai vaste [1]. Importanța temei se evidențiază prin popularitatea dezvoltării și testării software în limbajul Java. Bazele de date elaborate în SGBD H2 sunt compatibile cu proiecte Java, fiind independente de platformă și simplu de dezvoltat. SGBD H2 este foarte folosit în procesul de dezvoltare și testare deoarece: permite setarea modului de lucru client-server sau inclus în aplicație, păstrarea datelor în memorie operativă sau sistemul de fișiere al calculatorului, posedă sintaxă pentru mărirea eficienței interogărilor SQL, posibilitate de integrare cu Hibernate, TopLink, Glassfish, Apache ActiveMQ, jOOQ și alte frameworkuri și tehnologii actuale. De asemenea sunt implementate funcții specifice pentru SGBD, inclusiv pentru citirea și scrierea fișierelor cu date separate prin virgulă (Comma Separated Value, CSV). Aplicația posedă volum mic al

distributivului, oferă consolă web cu interfața grafică. Prin intermediul consolei este posibilă gestionarea bazelor de date existente scrise în diverse limbaje. Astfel, acest sistem SGBD este foarte comod în implementare, dar și necesită să fie testat, studiat și să fie completate informațiile accesibile online. Precum și studierea în practică a funcționalităților specifice și oportunităților noi introduse în versiunile recente ale H2. Studierea modului de lucru al acestui sistem de gestiune al bazelor de date se va face prin studierea documentației oficiale prezentate de către dezvoltatorii proiectului, elaborarea exemplelor practice pe baza acestora. Se va efectua analiza specificului lucrului cu H2 reieșind din datele oficiale de testare prezentate în rapoarte comparativ cu alte sisteme de gestiune, precum și rezultatele obținute în elaborarea bazei de date și lucrul aplicației elaborate cu acest sistem de gestiune. După studierea bazelor teoretice de lucru cu H2 va fi elaborată baza de date pentru evidența datelor într-un centru lingvistic. Baza de date va conține evidența dicționarilor disponibile, persoanelor utilizatoare a aplicației, persoanelor care frecventează centrul și utilizează materialele lingvistice disponibile, mai multe tabele vor fi necesare pentru evidența detaliilor referitoare la cantitatea, disponibilitatea, descrierea materialelor didactice. Lucrul cu această bază de date se va produce cât prin intermediul consolei web prin intermediul browserului, atât și direct efectuând conexiunea și executarea instrucțiunilor SQL din aplicație. Se va efectua crearea bazei de date în modul inclus în aplicație, cu păstrarea datelor în memorie operativă în timpul scrierii aplicației, și se va transfera ulterior modul de păstrare a datelor în sistemul de fișiere. Se vor efectua funcții H2 de căutare după text întreg, lucru cu fișiere CSV, interogări SQL generale de creare, citire, reînnoire, ștergere (Create Read Update Delete, CRUD) și specifice bazelor de date elaborate cu H2. Ca valoare aplicativă, produsul software elaborat va putea fi utilizat într-un centru lingvistic de studiere a limbilor străine pentru evidența materialelor de studiu, dicționarilor, persoanelor care le împrumută, altor detalii necesare, dar, la necesitate va putea fi utilizată și pentru păstrarea evidenței ale altor date specifice. Datorită implementării în limbajul Java aplicația și baza de date H2 inclusiv va putea fi utilizată pe diverse platforme, inclusiv sisteme de operare precum Windows, Linux, MacOS. Ca scop este crearea unei aplicații de volum mic, astfel volumul codului sursă poate fi micșorat semnificativ utilizând Project Lombok, Spring Boot cu includerea Tomcat, Spring Data JPA pentru minimizarea codului necesar pentru crearea conexiunilor între baza de date și aplicație, Thymeleaf pentru reflectarea eficientă a datelor prelucrate în interfața utilizator. Se va implementa modul de accesare orientat pe obiect la conexiune cu baza de date, și se va implementa executarea unor interogări mai complexe prin executarea directă a interogărilor și scripturilor SQL din codul aplicației. Astfel, aplicația va putea fi folosită pentru comoditatea evidenței centrului de studiere al limbilor străine, la evidența datelor unei biblioteci. Aplicația poate fi utilizată local sau implementată varianta accesibilă web. Se va încerca să se identifice tehnologiile optime pentru comoditatea gestionării datelor prin intermediul pentru dezvoltare, utilizare împreună cu SGBD H2. Concluzionând ideile de mai sus, această lucrare are ca scop studiul

particularităților de lucru cu bazele de date H2, un SGBD popular în dezvoltare, implementarea bazelor de date, stabilirea conexiunii între aplicație web Java și bază de date, utilizarea instrumentelor actuale Java, reflectarea datelor bazei de date în interfața grafică utilizator și analiza rezultatelor din punct de vedere specificului H2, funcționarea și eficiența produsului elaborat, analiza avantajelor, dezavantajelor și oportunităților pentru elaborarea bazelor de date H2.

Date generale despre H2

H2 utilizează SQL pentru gestionarea BD. Este posibilă gestionarea bazelor de date existente scrise în diverse limbaje, prin intermediul consolei web H2, utilizând funcții multiple compatibile cu BD mai voluminoase existente. De asemenea SGBD H2 este foarte folosit în procesul de creare a produselor software, deoarece permite crearea testelor cu date în memorie operativă, dar și cu păstrarea pe disc. Există posibilități vaste ale interogărilor, funcțiile și sintaxa specializate permit efectuarea repetată a interogărilor SQL, definirea ciclurilor, funcții definite de utilizator. Pentru aplicații mici H2 poate fi folosit desinestătător deoarece se include simplu în aplicație. Pentru lucru cu codul Java se recomandă și este mai bine testată folosirea Java 8 și 11. Regimuri de lucru posibile sunt: regimul la nivel de server și regimul inclus în aplicație. Modul de păstrare a bazei de date în memorie implicit este astfel încât baza se creează și se inițializează atunci când este pornită aplicația, ulterior se distruge atunci când aplicația este oprită ori JVM se oprește. Toate datele cu păstrarea în memorie operativă se pierd la restartarea aplicației. Pentru a păstra datele la restartarea aplicației se poate de stocat datele în fișiere sau de adăugat parametrul `DB_CLOSE_DELAY` cu valoare -1 la adresa BD pentru ca datele să rămână în memoria operativă, ceea ce poate provoca excepții din cauza supraîncărcării memoriei. Trebuie de menționat că BD în memorie sunt localizate la JVM, deci este posibil de accesat URL al BD numai din același JVM și class loader environment.

Concluzionând cele de mai sus, sunt posibile următoare moduri de lucru după locul de păstrare al datelor:

- în memorie operativă – timp minimal de răspuns, nu e nevoie de accesat discul, datele se creează odată cu prima conexiune și se distruge odată cu închiderea ultimei conexiuni.
- persistent - pe disc în sistemul de fișiere al calculatorului.

După modul de lucru sunt susținute următoarele opțiuni:

- în mod inclus într-o aplicație Java, astfel se va putea de executat mai rapid operațiile decât în modul server;
- în modul server, deci baza de date va fi accesibilă de pe alt server.

Toate modurile de păstrare ale datelor pot fi implementate cu toate tipurile de lucru, deci pot exista modul: inclus și în memorie operativă, inclus și persistent, server și memorie operativă, server și persistent. În implementări specifice pot fi configurări care

implementează concomitent mai multe moduri din cele patru. Un exemplu al modul hibrid de lucru este cel care combină mod inclus și server. Prima aplicație conectează la BD în modul inclus. Concomitent este pornit serverul astfel încât alte aplicații, inclusiv din alte procese pot accesa concomitent BD. Conform setărilor implicite se va utiliza oricare port TCP liber. URL de accesare al bazei de date nu se modifică pentru diferite aplicații. De exemplu pot avea loc următoarele baze de date cu proprietăți definite după adresa URL. BD pe disc și inclusă:

- jdbc:h2:Users/user/directoriu/baza - baza cu păstrarea persistentă a informației (pe disc) se crează în directoriul “directoriu”;
- jdbc:h2:./baza - baza cu păstrarea persistentă a informației (pe disc) se crează în directoriul curent de lucru.

BD în memorie și inclusă:

- jdbc:h2:mem: - se crează baza fără denumire cu o conexiune posibilă
- jdbc:h2:mem:baza – datele se mențin în memorie operativă indicate de cuvântul cheie mem, baza se șterge după închiderea ultimei conexiuni, se admit mai multe conexiuni într-un proces
- jdbc:h2:mem:baza;DB_CLOSE_DELAY=-1 - datele se mențin în memorie operativă după închiderea tuturor conexiunilor. Datele se păstrează atât timp cât lucrează mașina virtuală dar se vor elimina la restartarea serverului.

Pe disc și Client/Server cu pornirea prealabilă a serverului în consolă se utilizează:

- jdbc:h2:tcp://localhost/~baza - baza se află în directoriul cu numele utilizatorului acestui calculator

În mod hibrid și fără pornirea prealabilă a serverului:

- jdbc:h2:/data/test;AUTO_SERVER=TRUE

Portul TCP se poate seta manual în felul următor: AUTO_SERVER_PORT = 8081. În memorie și Client/Server cu server TCP:

- jdbc:h2:tcp:<adresa ip>:<portul>/mem:baza

De exemplu, pentru baza în memorie se poate de pornit în aceeași aplicație server TCP și de accesat în felul următor BD: jdbc:h2:tcp://localhost:9092/mem:baza , și în modul următor pentru BD care se păstrează în fișier: jdbc:h2:tcp://localhost:9090/~baza.

Soluții în probleme de durabilitate

În documentație H2 este indicat că există probleme ACID referitoare la durabilitate, ce pot conduce la pierderea datelor tranzacțiilor terminate, la dispariția alimentării cu energie electrică. Sistemul nu folosește apeluri FileDescriptor.sync(), FileChannel.force() și fsync deoarece acestea încetează semnificativ lucrul sistemului, la utilizarea lor este posibil de efectuat numai 60 de operații pe secundă. De asemenea o altă cauză de a nu utiliza *fsync* constă în faptul că multe discuri se distribuie cu opțiunea setată de cash-are a inscrierii.

Pentru asigurarea durabilității în documentație se recomandă utilizarea SET WRITE_DELAY și CHECKPOINT SYNC, care sunt disponibile cu drepturi de administrator [2].

Restricții de volum

În prezent există unele restricții pentru volumul datelor [3]: Fișierul BD poate fi de maxim 4 TB, BD poate avea maxim 2^{64} de linii, 65535 tranzacții deschise, maxim 16384 coloane se pot afișa prin interogare de selectare. Totodată nu există limitări pentru o serie de parametri în afara de volum de memorie disponibil. Acestea sunt: număr maximal de tabele, indexuri, triggeruri, obiecte, număr de linii în interogare, lungimea liniei interogării select. De asemenea interogări asupra tabelor de metadate se pot efectua încet pentru mai mult de câteva sute meta tabele. În timp actual mărimea implicită a unei pagini este 2 KB, acest parametru poate fi setat manual. Trebuie de menționat că utilizarea unor cuvinte cheie precum cele prezentate în continuare ca denumiri de tabele, coloane și alți identificatori definiți de utilizator pot duce la erori deoarece termenii respectivi sunt cuvinte H2 rezervate de sistem: ROWNUM, SELECT, SYSDATE, HAVING, INNER, INTERSECT, IS, JOIN, LIKE, LIMIT, MINUS, NATURAL, NOT, NULL, ON, ORDER, PRIMARY, CURRENT_TIMESTAMP, CROSS, CURRENT_DATE, EXCEPT, EXISTS, FROM, SYSTIMESTAMP, TODAY, SYSTIME, TRUE, UNION, UNIQUE, WHERE, CURRENT_TIME, DISTINCT, FALSE, FOR, FULL, GROUP [4].

Concluzii

Deoarece bazele de date sunt componente indispensabile a majorității sistemelor informatice, dezvoltarea și testarea produselor de program devine cu mult mai simplă și efectivă la utilizarea unei SGBD precum H2, fiind opensource, poate fi instalată rapid, necesită volum minim de memorie, oferă interfață grafică intuitivă și ușor de utilizat pentru operații simple de gestionare a bazei de date, precum și documentație oficială detaliată pentru operații mai avansate.

Numărul mare de posibilități de lucru în diferite regimuri și scopuri al H2, împreună cu variate soluții software avansează dezvoltarea produselor de program, ușurează testarea și economisește semnificativ timpul și efortul depus al programatorilor, fiind totodată dezvoltat în scopuri necomerciale, astfel fiecare persoană interesată în implementarea funcționalităților noi și îmbunătățirii celor existente are posibilitatea să contribuie la dezvoltarea proiectului. Releas-urile regulate odată la 2-3 luni ale proiectului reflectă actualitatea și solicitarea semnificativă ale acestora.

SGBD H2 poate fi integrat și utilizat împreună cu majoritatea framework-urilor actuale pentru dezvoltarea aplicațiilor Java, susține comenzile SQL standard, dar și multiple comenzi și funcționalități adiționale. H2 susține modurile diferite de lucru: inclus și server, persistent și în memorie, ceea ce simplifică dezvoltarea diferitor produse de program.

Faptul că odată cu prestarea funcționalităților vaste nu se utilizează librării de la părți terțe are beneficii precum: viteza mai mare a operațiilor în SGBD, evitarea vulnerabilităților altor librării, dar și aspecte nefavorabile ca: lipsa rezultatelor de testare în volum necesar și necesitatea testării mai bune a funcționalităților noi.

Precum recunosc dezvoltatorii proiectului există probleme de durabilitate și datele bazelor de date se pot deteriora la stingerea curentului electric. Funcțiile librărilor terțe care ar putea spori durabilitatea nu se implementează pentru că vor încetini efectuarea operațiilor. Proiectul H2 se află în dezvoltare continuă și în fiecare release se fixează mai multe vulnerabilități. La etapa actuală de dezvoltare a proiectului nu este recomandată implementarea bazelor H2 în producție cu volum mare de date, fiind posibil ca unele funcționalități încă să nu fie bine testate. Pentru aplicații mici H2 poate fi folosit desinestătător deoarece se include simplu în aplicație și prezintă rezultate bune de lucru cu volum restrâns de date. De asemenea este foarte comod de folosit acest SGBD în scopuri de testare, cu volum restrâns de date care deseori nu trebuie să fie păstrat, dar poate fi inserat la utilizare în modul în memorie al bazei de date. Totodată, consola grafică a SGBD H2 poate fi folosită pentru lucru cu bazele de volum mare existente create cu alte dialecte SQL.

Bibliografie

1. <https://github.com/h2database/h2database> cod sursă al proiectului cu explicații.
2. <https://habr.com/ru/post/446380/>, „Evoluția H2”.
3. http://www.h2database.com/html/advanced.html#limits_limitations, „Limite H2”.
4. <http://www.h2database.com/html/advanced.html#compatibility>, „Cuvinte de Sistem Rezervate”.