Зборник радова Факултет техничких наука, Нови Сад

UDK: 004.9
DOI: https://doi.org/10.24867/17BE19Djurdjevic

ПРИМЕНА *ETL* ПРОЦЕСА У ПОСТУПКУ ТРАНСФОРМАЦИЈЕ РЕЛАЦИОНЕ БАЗЕ ПОДАТАКА У ГРАФОВСКИ ОРИЈЕНТИСАНУ БАЗУ ПОДАТАКА

APPLICATION OF ETL PROCESS IN THE TRANSFORMATION PROCEDURE OF A RELATIONAL DATABASE INTO GRAPH-ORIENTED DATABASE

Добривоје Ђурђевић, *Факултет техничких наука, Нови Сад*

Област – ПРИМЕЊЕНЕ РАЧУНАРСКЕ НАУКЕ И ИНФОРМАТИКА

Кратак садржај – *У овом раду презентовано је коришћење ETL (Extract Transform Load) процеса у циљу аквизиције података за апликацију музичке енциклопедије коју је могуће претраживати постављањем упита на природном језику. Ови процеси представљају унапређење система који је у ту сврху у својој првој верзији користио web crawler. За пружање одговора на постављено питање систем користи технике машинског учења sequence to sequence, ради превођења питања са природног на упитни језик графовске базе и пружање одговора. За имплементацију мапирања базе података коришћена су два ETL процеса. Први ETL процес мапирање из релационе у релациону базу података и реализован је у ODI (Oracle Data Integrator) алату. Док је други ETL процес мапирања из релационе у графовски оријентисану базу података, реализован коришћењем Neo4j ETL алата.*

Кључне речи: *релациона база података, графовски оријентисана база података, ETL процес, трансформација, моделовање*

Abstract – *This paper will present the use of the ETL (Extract Transform Load) process, used in order to acquire data for the application of the music encyclopedia that can be searched by querying in natural language. These processes represent an improvement on a system that used a web crawler for data acquiring in its first version. To provide an answer to the question, the system uses sequence to sequence machine learning techniques, in order to translate the question from natural to query language of the graph database and answer provisioning. Two ETL processes were used to implement database translation from relational to graph model. The first is mapping from a relational to a relational database and is implemented in the ODI (Oracle Data Integrator) tool. While the second ETL is a mapping process that maps a relational to a graph-oriented database, using the Neo4j ETL tool.*

Keywords: *Relational database, Graph-Oriented database, ETL process, transformation, modeling*

1. УВОД

Информационе технологије чине неизоставни део модерног друштва, могу се пронаћи у практично сваком аспекту свакодневног живота, било да представљају примарни или алтернативни начин за извршавање неког задатка. Позитивни ефекти које носе са собом су многи, олакшавају велики број активности и задатака који су пред нама, пружају различите начине за комуникацију, аутоматизују послове и производњу, убрзавају банкарске услуге и пословање, олакшавају едукацију и обезбеђују доступност информација на длану. У основи скоро сваког информационог система постоји и база података задужена да одржи интегритет и доступност података који су од интереса.

Различити начини коришћења и конзумације информација довели су до експанзије и креирања нових типова база података. Поред класичних и свакако најраспрострањенијих релационих база података, које већ дуго представљају стандард у индустрији, све су више заступљене *NoSQL, Document, Key-Value*, хијерархијске, графовски оријентисане и друге базе података, доступне за интеграцију у *DW (data warehouse)* и *cloud* оријентисаним системима. Поред тога у последњих двадесет година настали су и концепти попут *Big Data* и *Blockchain* технологија који се заснивају на информацијама које анализирају, структуирају и чине их доступним у новом облику.

Основа за писање овог дипломског рада представља наставак развоја дигиталне музичке енциклопедије, односно концепта који је замишљен и започет као тема за бечелор дипломски рад аутора [1]. У тренутном облику ова музичка енциклопедија садржи малу количину информација из домена, приближно 5000 чворова и 5000 веза смештених у графовски оријентисану базу података. Подаци су прикупљени уз помоћ *web scraping* техника. Како би се ова апликација од концепта претворила у корисну енциклопедију која садржи велику количину података и која може да пружи одговор на бројна питања из домена музике, потребно је обезбедити те податке и структуирати их. Управо то представља мотивацију за писање овог мастер рада, овакав задатак пред себе поставља могућност истраживања и коришћења различитих технологија за аквизицију података.

Поменути систем који се ослања на *web scraping* технике како би обезбедио податке за енциклопедију ограничен је и тежак за одржавање, стога главни циљ овог пројекта је заменити овај систем ефикаснијим,

НАПОМЕНА:
Овај рад проистекао је из мастер рада чији је ментор био др Милан Челиковић, доцент.

робуснијим и системом који ће моћи да обезбеди велику количину података. Циљ овог рада је истраживање и имплементација *ETL [2]* процеса. Многи информациони системи користе *ETL* процесе приликом реорганизације података из различитих разлога, попут миграције базе података у неки нови облик, било да то диктира модернизација система, потреба за променом основног концепта, скалирање система, интеграција различитих извора података у један или подела једног система на више мањих уже специјализованих. Циљ пројекта представља преузимање података из бесплатне релационе базе података и њихова трансформација у графовски оријентисану базу података. Реализација овог задатка заснива се на коришћењу *ETL* процеса при интеграцији података кроз *ODI [3]* платформу и *Neo4j ETL* алата.

2. ПРЕГЛЕД ТРЕНУТНОГ СТАЊА У ОБЛАСТИ

У оквиру овог поглавља описан је концепт *ETL* процеса (због своје основне намене да се подаци из једног облика и извора складиштења трансформишу и пребаце у други облик и у други тип складишта) током времена и према потреби корисника, креирани су различити алати и приступи за моделовање и реализацију оваквих процеса. Данас их има много и тешко их је категорисати и приказати све. У овом поглављу ће бити приказан основни ток једног *ETL* процеса, могућа варијација основном концепту *ETL* процеса, са прегледом најпопуларнијих алата који се користе. Поред овога користе се различити приступи при моделовању и оптимизацији поменутих процеса, а они место проналазе у различитим окружењима и технологијама попут нпр. у анализи *Blockchain*-а [4]. *DW* системи се заснивају на прикупљању, управљању и анализи великих сетова података. Као ефикасан начин за прикупљање, трансформацију и континуалну интеграцију различитих извора *ETL* процеси често чине саставни део *DW* система [5].

2.1 *ETL* Процес

ETL процес [6] представља процедуру која за циљ има копирање података из једног или више извора у одредишни систем који репрезентује податке у другачијем контексту у односу на почетни систем. Овај процес укључује преузимање *extract* података из хомогених или хетерогених извора, трансформацију *transform* података према потребама система који их преузима у циљу складиштења, анализе и претраге. Последњи корак у овом процесу представља перзистенција података *load* у жељену базу података.

Extract као почетни корак представља увод у овај процес, како би подаци били даље процесирани, битно је да буду преузети успешно и да се при томе очува њихов интегритет, ово може посебно да буде изазовно уколико се преузимање врши из више различитих извора. Такође у овом кораку често се ради са различитим моделима, сваки извор података практично може бити имплементиран са другом технологијом и према другом формату података, које је потребно ускладити. У склопу преузимања података потребно је извршити и валидацију, тај корак може додатно да филтрира и обезбеди тачније податке у циљаној бази података.

Transform - трансформација података као следећи корак, према дефинисаном сету правила припрема податке за циљани модел и њихово уписивање у циљану базу података. Још један циљ овог корака је *data cleansing* – чишћење података које додатно обезбеђује да само жељени подаци буду прослеђени финалном кораку и буду сачувани у новој бази, ово такође представља изазов када је више извора из којих се подаци преузимају.

Load – уписивање података у *flat file*, базу података релационе или било које друге парадигме или *DW,* са инкременталним или *overwrite* приступом, ово варира од конкретне примене, али неопходно је да се трансформисани подаци сачувају и да буду доступни у новом облику за даље коришћење.

3. ОПИС СИСТЕМА

Основна идеја коју има имплементирана апликација, односно музичка енциклопедија, је да пружи брз одговор кориснику на различита питања из домена музике и да то оствари пруживши кориснику једноставан начин за интеракцију са системом. Замисао је да се питања попут: које инструменте одређени извођач свира, из које државе или града он потиче, да ли је члан неке групе, ко су остали чланови те групе, листа свих нумера које изводи група или албуми које је издала, одакле потиче одређени инструмент и како је настао и сл., постављају на једноставан начин како би се добио тражени одговор. Добро познати системи који су послужили као инспирација и који пружају овакве функционалности су персонални асистенти базирани на техникама вештачке интелигенције, међу којима су вероватно најпознатији „Google Assistant“, „Alexa“ и „Siri“. Овакви системи пружају велику базу знања и одговор на скоро свако питање са једноставним и интуитивним начином интеракције. Управо те функционалности захтевају коришћење различитих техника и решавање разноврсних проблема, што њихову имплементацију чини веома комплексном и изазовном. Циљ овог пројекта је био имплементирање једног система који извршава сличан задатак и истраживање техника које би могле у ту сврху да послуже. Као домен је изабрана музика, јер је са овим ограничењем лакше обезбедити базу знања коју систем може да претражује.

3.1. Архитектура система

Основна архитектура система приказана је на слици 1.

Систем се састоји од три компоненте од којих је прва софтверски агент са дуалном улогом – пружа начин интеракције корисника са системом и омогућава му претрагу базе података. Као лак и интуитиван начин интеракције софтверски агент кориснику омогућава да постави систему питање на природном језику, односно на енглеском језику. Затим се постављено питање преводи на упитни језик графовски оријентисане базе, односно у овом случају *Cypher* језик. Овако преведено питање систем користи за претраживање базе података, при чему се добија повратна информација као одговор.

Слика 1: *Архитектура предложеног система*

Друга компонента система је графовски оријентисана база података. Овакав модел је изабран због могућности једноставног претраживања веза између типова ентитета, без потребе прорачунавања веза приликом извршавања упита и интуитивности коју пружа при анализи ових веза. Такође због својих карактеристика постоји могућност једноставних измена и додавања нових веза.

Трећа компонента је систем за аквизицију података, а њен задатак је прикупљање и упис података од интереса у базу података. Систем се ослања на претраживање отворене и бесплатне базе података *MusicBrainz*.

3.2. Унапређење система

Идеја за унапређење система у овој фази је прибављање што већег броја података. Како је *MusicBrainz* база података отвореног типа, постоји могућност да се креира локална копија ове базе. Локална копија пружа одличне могућности за преузимање свих или само неких података који су од интереса. Сама база садржи велики број података који је већ прикупљен и структуиран, што у великој мери олакшава задатак.
С обзиром да музичка енциклопедија користи графовски оријентисану базу података, а да је *MusicBrainz* база релационог типа, ипак је потребно извршити одређена прилагођавања како би било могуће преузимање података. Унапређење система обухвата замену подсистема за аквизицију података базираног на *web scraping* техникама са *ETL* процесима у циљу прикупљања и трансформације података у жељени облик.

4. СИСТЕМ ЗА АКВИЗИЦИЈУ ПОДАТАКА

У овом поглављу описан је систем за аквизицију података, ова подкомпонента система представљена је у првој верзији овог пројекта, као основни начин за аквизицију података.
Систем за аквизицију података је замишљен као аутоматизована софтверска компонента, што значи да након покретања од стране корисника све своје функционалности обавља самостално. За циљ има претрагу и прикупљање података од значаја за систем и њихово уписивање у базу података. Систем је имплементиран као w*eb scraper* коришћењем *.Net framework-a, C#* програмског језика и *Abot Web Crawler framework-a.*

5. ПРЕВОЂЕЊЕ РЕЛАЦИОНЕ У ГРАФОВСКИ ОРИЈЕНТИСАНУ БАЗУ ПОДАТАКА

Релационе и графовски оријентисане базе се у великој мери концептуално разликују. Ипак основна намена им је у суштини иста – да обезбеде перзистенцију података и релација међу њима и да обезбеде интегритет и доступност података структуираних према шеми базе.

Једна од очигледних разлика је начин на који су подаци повезани. Основна компонента графа је веза између два чвора, која има подједнаку важност као и сами подаци које повезује. Ово граф чини интуитивним, лаким за интерпретацију и анализу веза између података. Везе као и чворови у графу могу имати атрибуте који их додатно описују. За разлику од графа у релационом моделу, везе између ентитета се реализују референцирањем на страни кључ, због чега су везе у релационом моделу мање очигледне, теже уочљиве поготово код комплексних модела.

Повезивање ентитета се прорачунава приликом извршавања *SQL (Structured Query Language)* упита коришћењем *JOIN* клаузуле и ослања се на подударање примарног и страног кључа. Ове операције могу бити веома захтевне за системске ресурсе у зависности од упита и количине података која се претражује и повезује. Приликом повезивања ентитета са кардиналитетом *N:N* (*many-to-many)* неопходно је увести ***JOIN*** табелу која садржи стране кључеве оба типа ентитета и тиме даље повећава захтеве за системским ресурсима.

Све ово релациони модел чини мање пожељним ако је анализа веза један од примарних захтева система, као што је у овом случају пружање одговора на питања попут „*Who are members of <Artist group>?*“.
Основне компоненте које чине релациони модел података присутне су и у графу као моделу. У циљу лакшег превођења релационог у графовски модел [7], дате су следеће смернице:

1. Табела у релационом моделу еквивалентна је лабели у графовском моделу.
2. Сваки ред у табели релалционог модела представља један чвор у графу.
3. Колона у табели релационог модела представља атрибут чвора у графу.
4. Страни кључ у релационом моделу замењује се везом између чворова у графу и није га потребно после чувати као атрибут чвора.
5. *JOIN* табела релационог модела трансформише се у везу између чворова, а колоне ове табеле постају атрибути те везе.

5.3 *SQL TO SQL*, први *ETL* процес

Први процес за основни циљ има преузимање података од интереса из релационе базе података *MusicBrainz* и њихову трансформацију у нови релациони модел који се затим уписује у *Oracle DW* базу података. Поред тога у овом кораку се врши припрема за наредни *ETL* процес који представља трансформацију новог релационог модела у модел графовски оријентисане базе. Мапирање се врши уз помоћ *ODI* алата, који на лак начин може да обезбеди интеграцију више различитих извора у одредишну базу података, па стога он практично заузима централно место приликом трансформације података.

5.4. *SQL TO GRAPH*, други *ETL* процес

Neo4j desktop поред *DBMS (Database Management System)* за графовски оријентисане базе података садржи и скуп алата који омогућавају различите операције. Један од тих алата је *Neo4j ETL* алат, који долази са графичким интерфејсом али и као *CLI (Command-line interface)* алат. За овај алат се

дефинише изворна база података, у овом случају то је *Oracle DW* из претходног *ETL* процеса, затим се преузимају потребне информације и креира се графовски модел. При реализацији овог процеса коришћен је алат са графичким интерфејсом, јер су биле потребне само мање измене, приликом ревидирања графа попут кориговања назива повезника између чворова.

6. ЗАКЉУЧАК

Унапређење пројекта завршено је успешно, систем за аквизицију реализован као *web crawler* замењен је *ETL* процесом који омогућава да подаци буду преузети директно из изворне *MuiscBrainz* базе података, при чему је графовска база података проширена са новим лабелама, везама и великом количином чворова које садржи.

Систем за аквизицију података имплементиран као *web crawler* представљао је добру почетну тачку при развоју концепта ове апликације. Овакав приступ омогућава имплементацију једноставног система који прикупља само основне информације и чини почетни сет података. Он такође пружа могућност да се креира потпуно аутоматизован систем, који би прикупљао податке и инкрементално их додавао у базу података у сваком пролазу, који је могуће периодично пуштати или према заказаном распореду, и да при томе води рачуна о интегритету података. Ипак имплементација оваквог система доста зависи и од циљаног извора који се претражује и парсира, односно његове комплексности, али и учесталости и типа измена које се дешавају над њим. Ови аспекти могу драстично да повећају комплексност приликом имплементације и одржавања *web crawler*-а и парсирања информација од значаја. Уколико се неки делови или комплетно све *html* странице динамички генеришу, парсирање података са таквих страница може да постане практично немогуће или јако тешко изводљиво.

Када је реч о реализацији *ETL* процеса приликом имплелементације овог пројекта у почетку је потребно доста труда док се конфигуришу сви изабрани алати, базе података, виртуелне машине, односно сав потребан софтвер. Након овог корака сам процес је доста једноставан, пружа већу контролу над подацима али може да буде временски захтеван. Уколико би циљ био само да се подаци пребаце из изворне *MusicBrainz* базе у одредишну графовски оријентисану базу, у овом случају цео први *ETL* процес је могао бити изостављен, уз директне измене на изворној бази и покретање *SQL to Graph ETL* процеса. Ипак овакав приступ би ограничио евентуално проширивање и могућност интеграције са другим базама података, па је из тог разлога је коришћено релационо мапирање и *ODI*. Такође не би било могуће итеративно ажурирање базе података које *ODI* пружа као могућност приликом креирања мапирања. Поред тога локална копија изворне базе података била би измењена и тиме би било онемогућено конфигурисати репликацију између локалне копије базе и *MusicBrainz* сервера. Ова функционалност и тренутна архитектура целог система омогућава и потпуну аутоматизацију овог процеса. Имплементација ове функционалности била би одлично унапређење овог система. Реализација би захтевала конфигурисање репликације, екстерне окидаче или скрипт за покретање мапирања која чине први *ETL* процес у *ODI* алату, коришћење *Neo4j ETL CLI* алата за аутоматизацију другог *ETL* процеса и скрипт који дефинише кораке за овај процес. На овај начин било би могуће преузимати све измене и нове податке унете у *MusicBrainz* базу потпуно аутоматски. Још једно од могућих унапређења било би интегрисање више различитих извора података. На овај начин било би могуће још више проширити скуп података са додатним информацијама. Поред овога, како би се даље проширио скуп података, могуће је имплементирати и засебан систем који корисницима дозвољава да уносе нове податке, који нису прикупљени интеграцијом из других извора података. Имајући у виду све до сада наведено, у ситуацији када је цела база података доступна, нема пуно смисла користити *web crawler, ETL* приступ је много боља опција за аквизицију података у сваком аспекту.

7. ЛИТЕРАТУРА

[1] Ђурђевић, Д., 2019. *Развој музичке енциклопедије, засноване на графовски оријентисаној бази података и софтверским агентима*, Нови Сад: Факултет Техничких Наука.

[2] Simitsis, A. Vassiliadis, P. & Sellis, T., 2005. *Extraction-Transformation-Loading Processes*. [Online] Research Gate Available at: https://www.researchgate.net/publication/239638567_Extraction-Transformation-Loading_Processes [Accessed 29 September 2021].

[3] Oracle, 2021. *Oracle Data Integrator 12.2.1.4.0*. [Online] Oracle Available at: https://docs.oracle.com/en/middleware/fusion-middleware/data-integrator/12.2.1.4/ [Accessed 9 August 2021].

[4] Galici, R. Ordile, L. Marchesi, M. Pinna, A. & Tonelli, R., 2020. *Applying the ETL Process to Blockchain Data. Prospect and Findings.* [Online] MDPI Available at: https://doi.org/10.3390/info11040204 [Accessed 10 August 2021].

[5] Santos, R. & Bernardino, J., 2008. *Real-time data warehouse loading methodology*. [Online] Research Gate Available at: https://www.researchgate.net/publication/221524861_Real-time_data_warehouse_loading_methodology [Accessed 20 August 2021].

[6] Wikipedia, 2021. *Extract, transform, load*. [Online] Available at: https://en.wikipedia.org/wiki/Extract,_transform,_load [Accessed 21 August 2021].

[7] Neo4j, 2021. *Model: Relational to Graph* [Online] Available at: https://neo4j.com/developer/relational-to-graph-modeling/ [Accessed 23 August 2021].

Кратка биографија:

Добривоје Ђурђевић рођен је 1990. год. у Београду. Бечелор рад из области Рачунарских наука и информатике одбранио је 2019. године. Исте године уписује мастер студије на Факултету техничких наука.