Usporedba SQL server i Oracle sustava za upravljanje relacijskim bazama podataka

Varešković, Dražena

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, University of Split, Faculty of science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:166:766508

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-30**

Repository / Repozitorij:

Repository of Faculty of Science

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

ZAVRŠNI RAD

USPOREDBA SQL SERVER I ORACLE SUSTAVA ZA UPRAVLJANJE RELACIJSKIM BAZAMA PODATAKA

Dražena Varešković

Split, rujan 2022.

Temeljna dokumentacijska kartica

Završni rad

Sveučilište u Splitu
Prirodoslovno-matematički fakultet
Odjel za informatiku
Ruđera Boškovića 33, 21000 Split, Hrvatska

USPOREDBA SQL SERVER I ORACLE SUSTAVA ZA UPRAVLJANJE RELACIJSKIM BAZAMA PODATAKA

Dražena Varešković

SAŽETAK

Cilj završnog rada bio je napraviti usporedbu dva najkorištenija sustava za relacijske baze podataka, SQL Server i Oracle. Usporedba je osmišljena na način da se korisnik može lakše odlučiti za vrstu sustava pri izradi vlastite baze podataka. Mnogo je temeljnih razlika između Oracle i SQL Servera, a jedna od razlika je jezik koji koriste. Ovim radom bit će prikazane usporedbe SQL Servera i Oracle SQL Developera vezano za modeliranje i implementaciju baze podatka. Nakon što smo se upoznali s osnovnom teorijom vezanom uz područje relacijskih baza podataka prikladno je da ju pretočimo u praktični dio. Za potrebe ovog projekta osmišljena je i realizirana baza podataka koja može učinkovito poslužiti za potrebe srednje škole. Nakon kreiranja logičkog i relacijskog modela baze podataka potrebno je izraditi DDL kojeg ćemo izvršiti u Oracle SQL Developeru kako bismo relacijski model baze podataka pretočili u relacijsku bazu podataka. Nakon što smo u Data Modeleru kreirali DDL, slijedi nekoliko usporedba vezanih za SQL Server i Oracle kao što su spajanje na bazu, usporedba tipova podataka, razlike u transakcijskom dijelu, usporedba strukture pohrane te razlike u cijeni.

Ključne riječi: relacijska baza podataka

Rad sadrži: 41 stranica, 25 grafička prikaza, 3 tablice i 7 literaturnih navoda.

Izvornik je na hrvatskom jeziku.

Mentor: **Doc. Dr. sc. Monika Mladenović**, docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Ocjenjivači: **Doc. Dr. sc. Monika Mladenović**, docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Doc. Dr. sc. Goran Zaharija, docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Doc. Dr. sc. Divna Krpan, docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad prihvaćen: **rujan 2022.**

Basic documentation card

Thesis

University of Split
Faculty of Science
Department of informatics
Ruđera Boškovića 33, 21000 Split, Croatia

COMPARASION OF SQL SERVER AND ORACLE RELATIONAL DATABASE MANAGEMENT SYSTEMS

Dražena Varešković

ABSTRACT

The goal of the final work was to make a comparison of the two most used systems for relational databases, SQL Server and Oracle. The comparison is designed in such a way that the user can more easily decide on the type of system when creating his own database. There are many fundamental differences between Oracle and SQL Server, and one of the differences is the language they use. This paper will present comparisons between SQL Server and Oracle SQL Developer related to database modeling and implementation. After we have familiarized ourselves with the basic theory related to the field of relational databases, it is appropriate to translate it into a practical part. For the purposes of this project, a database was designed and implemented that can effectively serve the needs of the high school. After creating the logical and relational database model, it is necessary to create DDL, which we will execute in Oracle SQL Developer in order to convert the relational database model into a relational database. After we have created the DDL in Data Modeler, there are several comparisons related to SQL Server and Oracle, such as connection to the database, comparison of data types, differences in the transactional part, comparison of the storage structure, and price differences.

Keywords: relational database

Thesis consists of: 41 pages, 25 figures, 3 tables and 7 references **.** Original language: Coratian

Mentor: **Monika Mladenović, Ph.D.** Assistant Professor of Faculty of Science, University of Split

Reviewers: **Monika Mladenović, Ph.D.** Assistant Professor of Faculty of Science, University of Split

Goran Zaharija, Ph.D. Assistant Professor of Faculty of Science, University of Split

Divna Krpan, Ph.D. Assistant Professor of Faculty of Science, University of Split

Thesis accepted: **September 2022.**

ZAHVALA

Zahvaljujem se svojoj mentorici Doc. Dr. sc. Moniki Mladenović na brojnim savjetima i velikoj pomoći pri izradi ovog završnog rada.

Veliko hvala i svim mojim prijateljima i kolegama bez kojih ovaj studij ne bi prošao tako zanimljivo.

Hvala cijeloj mojoj obitelji koja je na bilo koji način pomogla prilikom mog studiranja.

Najveće hvala mojim roditeljima i sestrama na velikoj podršci, strpljenju, povjerenju te ogromnom razumijevanju i bezgraničnoj ljubavi.

Sadržaj:

Uvod 1
1. Razvojna tehnologija 2
1.1. Relacijske baze podataka 2
1.2. Sustavi za upravljanje relacijskim bazama podataka 4
1.3. Oracle 6
1.4. SQL Server 7
1.5. Usporedba SQL Server-a i Oracle-a 8
1.6. Modeliranje podataka 11
1.6.1. Entiteti i atributi 12
1.6.2. Relacije 12
1.6.3. Data Modeler 13
2. Razvoj modela baze podataka 14
2.1. Relacijski model 14
2.2. Izrada konceptualnog modela 15
2.3. Logički i relacijski model 16
2.4. Podešavanje u Data Modeleru 19
2.5. Izrada DDL-a prema modelu 21
2.6. Usporedba izrade baze podataka na SQL Server i Oracle 23
2.6.1. Izrada baze podataka na Oracle-u 23
2.6.2. Izrada baze podataka na SQL Server 26
2.6.3. Tipovi podataka SQL Server u odnosu na Oracle 29
2.6.4. Razlika u transakcijskom dijelu SQL Server i Oracle 31
2.6.5. Usporedba strukture pohrane 33
2.6.6. Razlike u cijeni i popularnosti 33
Zaključak 37
Literatura 38
Pregled slika 39
Skraćenice 41

Uvod

Motivacija za odabir teme završnog rada bila je slušanje premeta na drugoj godini studija „Baze podataka“ i izrade vlastitog projekta iz predmeta „Informatički projekt iz baza podataka“. Ono što me potaklo na pisanje ove teme jest usporedba dva najzastupljenija sustava za upravljanje bazom podataka. Ovim radom bit će prikazane usporedbe SQL Servera i Oracle SQL Developera vezano za modeliranje i implementaciju baze podatka.

Cilj rada je upoznavanje s osnovnim funkcionalnostima, te prikazati korištenje istih na konkretnom primjeru.

Rad je podijeljen u dva poglavlja. Prvo poglavlje se odnosi na razvojnu tehnologiju te sadrži šest podpoglavlja. U ovih šest podpoglavlja se spominju relacijske baze podataka, konekcija (spajanje) baze podataka SQL Servera i Oraclea te njihova sigurnost i općenito modeliranje podataka. U predzadnjem podpoglavlja slijedi usporedba SQL Servera i Oraclea na čemu je rad i osnovan.

Drugo poglavlje je razvoj modela baze podataka.

Zatim slijedi Zaključak kojim se rezimira cijeli rad.

1. Razvojna tehnologija

Među mnogim sustavima za upravljanje relacijskim bazama podataka (SUBP, engl. database management system – DBMS), dva su najpopularnija i široko korištena Oracle i SQL Server. Mnogo je temeljnih razlika između Oracle i SQL Servera, a jedna od razlika je jezik koji koriste. Oracle je sustav relacijske baze podataka koji podržava mnogo platformi, kao što su Windows, Linux, Solaris, OS X itd. Dostupan je u različitim inačicama, od verzija za osobnu uporabu do verzija poslovne klase. Kao i Oracle, SQL Server je i sustav relacijskih baza podataka. Jezik koji koristi SQL Server je T-SQL, tj. Transact-SQL. SQL Server podržava samo Windows. SQL Server ne dijeli bazu podataka s korisnikom. Ako korisnik želi pristupit bazi podataka, tada je korisniku dodijeljen ID ključ prijave. Iako su i Oracle i SQL Server RDBMS, oni imaju razlike. Oracle ima snažnije uslužne programe za upravljanje bazom podataka od SQL Servera. Ako uzmemo u obzir velike tablice i indekse, Oracle omogućava podjelu velikih tablica na raspon particija , dok SQL Server podjelu ne nudi.

1.1. Relacijske baze podataka

Relacijske baze podataka razvio je engleski matematičar Edgar Frank Codd-a. Prezentirao ga u svom članku „*A Relational Model of Data for Large Shared Dana Banks*“1970. Radio je u to vrijeme u IBM-u. Ono što je mislio jest da će oni u zajednici prvi implementirati relacijski model spremanja podataka, ali to se nije dogodilo zbog ustrajnosti IBM-a na korištenju već provjerenog, ali koji je već zastario hijerarhijski model sustava IMS. Oracle je prva tvrtka koja je implementirala relacijski model podataka. I danas je sinonim za baze podataka. Codd je dugo vremena proveo u borbi sa raznim proizvođačima SUBP-ova jer su oni ti koji su krivo ili djelomično implementirali relacijski podatkovni model, a nazivali su ga relacijskim. On je kao rezultat ove duge borbe objavio pravila koja SUBP mora poštivati kako bi njihov model bio relacijski:

1. Predstavljanje informacija – podaci se reprezentiraju kao vrijednosti u relacijama, tj. tablicama.

2. Pravilo pristupa – preko kombinacije imena tablica, vrijednosti primarnog ključa i imena atributa svaki podatak u tablici mora biti dostupan.
3. Tretiranje nepoznatih vrijednosti – nepoznata vrijednost neovisno o tipu je vrijednost *NULL*. Ona nije isto što i prazan znak ili broj 0 (nula) ili varijabla.
4. Dinamički online katalog – Rječnik baze podataka mora biti spremljen kao i svi ostali podaci u bazi kako bi nad tim podacima autorizirani korisnici mogli postavljati upite koristeći se (SQL) upitnim jezikom.
5. Pravilo sveobuhvatnog jezika – mora postojati jezik za komunikaciju sa bazom podataka koji podržava relacijske operatore (modifikaciju podataka, definiciju podataka i administraciju).
6. Pravilo pogleda – u ovom pravilu se definiraju tzv. Tablice pogleda sa *SELECT* naredbom koja dohvaća podatke iz jedne ili više tablica. Svi se pogledi moraju moći ažurirati.
7. Pravila ažuriranja skupova – podaci iz relacijske baze podataka mogu biti preuzeti u skupovima podataka iz jedne ili više tablica. Operacije umetanja, ažuriranja i brisanja moraju biti podržane za skupove podataka, a ne samo za jedan redak jedne tablice.
8. Nezavisnost fizičkih podataka – aplikacije koje pristupaju podacima u relacijskoj bazi podataka ne smiju biti ovisne o promjenama u fizičkom načinju spremanja podataka.
9. Nezavisnost logičkih podataka – promjena strukture baze podataka ne smije uzrokovati ponovnu izradu baze podataka ili aplikacije.
10. Nezavisnost integriteta podataka – sustav se mora brinuti o integritetu baze podataka, a ne aplikacije izvana.
11. Distribuirana nezavisnost – kada se uvede distribuirana verzija SUBP-a , aplikacija mora nastaviti operativno raditi.
12. Pravilno o nenarušavanju – integritet podataka ne smije biti narušen.
13. Nulto pravilo – da bi sustav bio relacijski mora koristiti isključivo relacijske mogućnosti baze podatka kod upravljanja.

SUBP se smatra relacijskim ako poštuje šest pravila od navedenih trinaest. Relacijski model je najsporiji i najfleksibilniji. (Tonči Carić, 2015)

Najpoznatije implementacije sustava za upravljanje relacijskim bazama su :

- SQL Server,
- Oracle
- Microsoft Access

- PostgreSQL,
- MYSQL i dr.

Najpoznatiji sustavi za upravljanje relacijskim bazama podataka su svakako SQL Server o Oracle čije će se značajke i funkcionalnosti usporediti u ovom radu. U sljedeće poglavlju opisat će se svojstva sustava za upravljanje relacijskim bazama podataka.

1.2. Sustavi za upravljanje relacijskim bazama podataka

Sustav upravljanja bazom podataka je softver za pohranu i dohvaćanje korisničkih podataka uz razmatranje odgovarajućih sigurnosnih mjera. Sastoji se od grupe programa koji manipuliraju bazom podataka. Skraćenica za sustav upravljanja bazom podataka SUBP, dok je na engleskom jeziku skraćenica DBMS što je akronim od Database Management System. Na primjer: MySQL, Oracle itd. su vrlo popularna komercijalna baza podataka koja se koristi u različitim aplikacijama. Database Management System (DBMS) pruža sučelje za izvođenje različitih operacija kao što su stvaranje baze podataka, pohranjivanje podataka u nju, ažuriranje podataka, stvaranje tablice u bazi podataka i još mnogo toga. Pruža zaštitu i sigurnost bazi podataka. U slučaju više korisnika, ona također održava dosljednost podataka. (Peterson, 2021)

Karakteristike DBMS-a :

- Koristi digitalni repozitorij uspostavljanja na serveru za pohranu i upravljanje informacijama.
- Može pružiti jasan i logičan prikaz procesa koji manipulira podacima.
- DBMS sadrži automatske postupke sigurnosnog kopiranja i oporavka.
- Sadrži atomarnost, konzistetnost, izolaciju, durabilnost što je skraćenica na engleskom jeziku ACID (engl. Atomicity, Consistency, Isolation, Durability), svojstva koja održavaju podatke u dobrom stanju u slučaju kvara.
- Može smanjiti složen odnos između podataka.
- Koristi se za podršku manipulaciji i obradi podataka.
- Koristi se za pružanje sigurnosti podataka.
- Bazu podataka može pregledavati s različitih gledišta u skladu sa zahtjevima korisnika.

Vrste DBMS-a su :

- Hijerarhijska baza podataka
- Mrežna baza podataka
- Relacijska baza podataka
- Objektno orijentirana baza podataka

Slika 1.1 Vrste DBMS-a

U hijerarhijskoj bazi podataka podaci su organizirani poput stabla, gdje čvorovi stabla predstavljaju zapise, a grane stabla polja, tj. podaci se pohranjuju hijerarhijski.

Mrežna baza podataka podržava mnoge do mnoge relacije gdje se može povezati više korisničkih zapisa.

Relacijska baza podataka ima strukturu koja omogućuje korisnicima identificiranje i pristup podacima u odnosu na drugi podataka u bazi podataka. Podaci se u ovoj strukturi pohranjuju u obliku tablica.

Objektno orijentirana baza podataka koristi mali pojedinačni softver koji se naziva objektima. Ovdje svaki objekt sadrži dio podataka i upute za radnje s podacima.

DBMS se primjenjuje u različitim popularnim aplikacijama sustava kao što su bankarstvo, zrakoplovne kompanije, financije, prodaja i proizvodnja, te sveučilišta. DBMS služi kao

učinkovit rukovoditelj kako bi uravnotežio potrebe više aplikacija koristeći iste podatke. (Peterson, 2021)

1.3. Oracle

Oracle baza podataka je relacijska baza podataka i njena skraćenica na engleskom jeziku je RDMS što je akronim od Relational Database Management System, sustav tvrtke *Oracle Corporation*. Softver je izrađen oko okvira relacijske baze podataka. Omogućuje korisnicima pristup podatkovnim objektima pomoću SQL jezika. Oracle se široko koristi u cijelom svijetu. Ako pogledamo sa strane administracije korisnika, kod Oraclea se korisnici spajaju na bazu pomoću korisničkih računa. Za svaki su korisnički račun definirane privilegije i prava nad objektima u bazi. Budući da je mnogo različitih privilegija mogu se grupirati u role, a role se onda mogu dodijeliti korisniku. Za svaki se korisnički račun može definirati zasebni profil. Profili omogućuju i administraciju zaporki. Kako bi korisnik mogao koristiti bazu moraju mu se dodijeliti određene privilegije. Sistematske privilegije omogućuju korisnicima spajanje na bazu, kreiranje objekata u bazi, dodavanje novih korisnika i drugo. Izvođenje upita i mijenjanje podataka u tablicama omogućuju prava nad objektima. Novi korisnici ne mogu raditi apsolutno ništa – ne mogu se niti spojiti na bazu dok im se ne dodijeli sistemska *CREATE SESSION* privilegija. Kako bi korisnik ima pravo na pregled ili kreiranje tablica, administrator mu mora dodijeliti *SELECT TABLE* i *CREATE TABLE* privilegije. U nekom trenutku bazu mogu koristiti stotine korisnika. Postoji vjerojatnost da će više korisnika istovremeno pristupati istim podacima i pokušavati ih modificirati. Stoga, svaka baza podataka mora sadržavati ACID koncept. Oracle koristi mehanizme zaključavanja tablica i redaka za vrijeme izvođenja transakcija, kako bi osigurao konzistentnost, izolaciju i trajnost podataka u bazi. (Vrbanić, 2009)

Značajke Oraclea:

1. Jednostavnost oporavka podataka u usporedbi s bazama podataka.
2. RDMS sustav može lako podnijeti velike količine podataka
3. Omogućuje promjenu platformi u bilo kojem trenutku
4. Daje mogućnost za strategije skaliranja
5. Baza podataka omogućuje ponovno pokretanje stvarnih radnih opterećenja proizvodnje, uključujući internetska korisnička i skupna radna opterećenja, u testnim okruženjima
6. Podrška za tehnologije virtualizacije specifične za hardver i OS

7. Pruža VMWare podršku za testna i produktivna SAP okruženja
8. Ako primarna baza podataka postane nedostupna, baza podataka stanja čekanja može djelovati kao primarna baza podataka
9. Može se koristiti za čitanje, izvješćivanje, testiranje ili sigurnosne kopije, smanjujući opterećenje primarne baze podataka
10. Neprekinuta obrada za korisnike koja eliminira potrebu za ručnim oporavkom

1.4. SQL Server

Kao i Oracle, SQL Server je i sustav za upravljanje relacijskim bazama podataka. Jezik koji koristi SQL Server je T-SQL, tj. Transact-SQL. SQL Server sadrži naredbe za potpun rad s relacijskom bazom podataka. SQL sadrži grupe naredbi za:

- Kreiranje objekata (eng. *Data Definition Language* – DDL)
- Manipuliranje podacima (eng. *Data Manipulation Language* – DML)
- Kontrola podataka (eng. *Data Control Language* – DCL)

CREATE, ALTER i *DROP* su naredbe koje spadaju u DDL i one služe za kreiranje i ažuriranje objekata, a objekti mogu biti tablice, pogledi i dr. Kako bi unosili, mijenjali i brisali određene podatke koristimo naredbe za manipuliranje podacima, a to su *INSERT*, *UPDATE* i *DELETE*.

Zaštita sustava SQL Server može se promatrati kao niz koraka koji uključuju četiri područja: platformu, provjeru autentičnosti, objekte i aplikacije koje pristupaju sustavu. Platforma za SQL Server uključuje fizički hardver i mrežne sustave koji povezuju klijente s poslužiteljima baze podataka i binarne datoteke koje se koriste za obradu zahtjeva baze podataka. Fizička sigurnost se odnosi na ograničavanje neovlaštenog pristupa podatkovnim centrima ili drugim komponentama poslužitelja. Za stvaranje , izmjenu ili odustajanje pravila potrebna je dozvola PROMIJENI SIGURNOSNA PRAVILA. Za stvaranje ili odbacivanje sigurnosnih pravila potrebna je dozvola *ALTER*. Sigurnosna pravila primjenjuju se na sve korisnike, uključujući dbo korisnike u bazi podataka. Ako je sigurnosno pravilo stvoreno pomoću programa, korisnici moraju imati dozvolu *SELECT* ili *EXECUTE* za predikatnu funkciju i sve dodatne tablice, prikaze ili funkcije koje se koriste unutar predikatne funkcije. Za povezivanje SQL Servera pomoću SSMS, najprije pokrenemo *Microsoft SQL Server Management Studio*. Zatim se pojavi prozor s određenim podacima kao što su naziv poslužitelja, provjera autentičnosti i lozinka za korisnika. Lozinka se unosi ona koja je bila postavljena pri instalaciji SQL Servera. (Kellyn Gorman, 2019)

Neke od značajki SQL Servera :

- Alati za podršku *SQL Server Profiler, BI alati, SQL Server Managment Studio* .
- Nudi online podršku i dokumentaciju te podršku za proizvode uživo.
- Pruža naprednu mogućnost prilagodbe za mapiranja vrsta podataka te brisanje i preimenovanje objekata.
- Prikazuje poruke o pogreškama i upozorenjima o migraciji u prozoru tijeka.
- Jedinstveno, integrirano okruženje za upravljanje i autoritizaciju modela baze podataka sustava SQL Server.
- Dijalozi za promjenu pozamašne promjene omogućuju pristup više alata kada je dijalog otvoren.
- Značajka monitora aktivnosti s filtriranjem i automatskim osvježavanjem.
- Uvoz i izvoz iz sql servera managment studija.

1.5. Usporedba SQL Server-a i Oracle-a

SQL Server i Oracle su izvrsne RDBMS opcije i mogu se koristiti na sličan način. Oba nude rješenja na razini poduzeća, a oba olakšavaju oporavak podataka od ostatka konkurencije. *Microsoftovu* je možda lakše naučiti, ali ta jednostavnost dolazi s nekim nedostatcima. Iako *Microsoft* nudi alate kao što su *SQL Server Profiler* i *BI Tools* (engl. *Business Intelligence Tools*), možda vam neće trebati za učinkovito pokretanje podataka. Ono u čemu je *Microsoft* neusporediv je razina podrške, budući da je njegova dokumentacija zvjezdana i pruža podršku za proizvode uživo, ali Oracle je bolji u rukovanju većim količinama podataka. Oracle instancom naziva ono što SQL Server bazom podataka. Ono što se događa u pozadini ova dva sustava, također je jedna od razlika. Tamo gdje Oracle ima Oracle izvršnu datoteku, SQL Server ima jednu za svoju bazu podataka. Oracle također ima i brojne pozadinske procese koji se izvode za svaku bazu podataka. Svaki od ovih procesa obavlja posao ili sigurnosno kopira drugi proces kao dio sekundarne odgovornosti. Za razliku od SQL Servera, Oracle ima svaki proces jasno imenovan na razini OS-a. Također, ova dva sustava se razlikuju i po strukturi pohrane koju možemo vidjeti u tablici ispod. Oracle server je „connection-based“ (temelji se na vezi) i nudi višestruke rezultate upita po vezi, potrebna je samo jedna veza, višestruke sesije (engl. sessions) po vezi, višestruke transakcije po vezi i pristup raspodijeljenim bazama podataka preko linkova baze podataka. SQL Server je „stream-based“ i nudi samo jedan aktivni rezultat upita po vezi i uobičajeno se koristi nekoliko veza. Razlika u transakcijskom dijelu je početak i kraj transakcije. Kao kraj transakcije, SQL Server koristi *auto.commit mod* po „defaultu“ dok Oracle zahtjeva *COMMIT* izjavu po „defaultu“. S obzirom na to da u

Oracleu ne postoji ugrađeni tip podataka za primarne ključeve, prije kreiranja DDL-a potrebno je svim primarnim ključevima napraviti sekvencu i okidač. Primjeri razlika između SQL Server i Oracle su platforme, čitanje podataka, naredbe, jezici implementacije, podržani programski jezici i sl. koje su dodatno opisane u tablici ispod. (dbaklevar, 2019)

Tablica 1.1 Usporedba SQL Servera i Oraclea

SQL SERVER	ORACLE
Može se instalirati samo na nekoliko platformi, Windows i Linux	Radi na širokom rasponu platformi (Linux, Windows, Solaris, OS X, AIX, HP-UX, z/OS)
Jedinica za pohranu: -Page (8 KB fiksna) -Za alokaciju prostora za tablicu i indekse: Extent (64 KB fiksna)	Jedinica za pohranu: -Block (logički blok se sastoji od jednog ili više blokova OS-a, veličina bloka je definirana inicijalizacijskim parametrima)
Koristi kopiju zapisa tako da se tijekom izmjene omogućuje čitanje izvornih podataka	Koristi metodu blokiranja redaka ili stranica. Nikad ne dopušta čitanje dok je stranica blokirana
Svaku naredbu izvršava zasebno, tako da će biti teško napraviti promjene ako se tijekom procesa naiđe na bilo kakva greške	Svaku novu vezu baze podataka tretira kao novu transakciju
Podržani programski jezici: C#, C++, Go, Java, JavaScript, PHP, Python, R, Visual Basic itd.	Podržani programski jezici : C, C#, C++, Erlang, Fortran, Haskell, Java, JavaScript, Lisp, PHP, Python, R , Scala, Tcl, Visual Basic, Ruby, Perl, OCaml, Objective C, itd.
Podaci o skripti na strani poslužitelja: Transact SQL, .NET languages, R, Python and (s SQL Server 2019) Java	Podaci o skripti na strani poslužitelja: PL/SQL

Ako pogledamo usporedbu prema tipu podataka SQL Servera i Oracle , ima dosta sličnosti ali i razlika. Jedna od razlika u SQL Server i Oracle su rezervirane riječi. Mnoge su Oracleove rezervirane riječi valjan objekt ili imena stupca u SQL Serveru. Na primjer, *DATE* je rezervirana riječ u Oracleu, ali nije rezervirana riječ u SQL Serveru. Stoga nijedan stupac ne smije imati naziv *DATE* u Oracleu, ali se stupac može nazvati *DATE* u SQL Serveru. U tom slučaju treba odabrati ime objekta koji nije rezervirana riječ.

Fizičke i logičke metode pohrane podataka *IMAGE* i *TEXT* podaci razlikuju se od Oraclea do SQL Servera. U SQL Serveru pokazivač na *IMAGE* ili *TEXT* podatke pohranjen je s redcima u tablici, dok su *IMAGE* i *TEXT* podaci pohranjeni posebno. Ovaj raspored dopušta stupac *IMAGE* ili *TEXT* podataka po tablici. U Oracleu se podaci *IMAGE* mogu pohraniti u *BLOB* polje tipa, a *TEXT* podaci se mogu pohraniti u *CLOB* polje tipa. Oracle dopušta više stupaca *BLOB* ili *CLOB* stupaca po tablici. *BLOB* i *CLOB* mogu, ali i ne moraju biti pohranjeni u redu, ovisno o njihovoj veličini. Kod SQL Servera, T-SQL omogućuje korisnicima definiranje i imenovanje vlastitih tipova podataka koji nadopunjuju sistemske tipove podataka. Korisnički definirana vrsta podataka može se koristiti kao tip podataka za bilo koji stupac u bazi podataka. Zadane postavke i pravila mogu se vezati za ove korisnički definirane vrste podataka, koje se automatski primjenjuju na pojedinačne stupce ovih korisnički definiranih vrsta podataka. Prilikom migracije na Oracle PL/SQL, radna ploča za migraciju određuje osnovni tip podataka za svaki korisnički definirani tip podataka i pronalazi ekvivalentnu vrstu podataka PL/SQL. (Oracle, 2021)

Tablica 1.2 Tipovi podataka

SQL Server	Oracle
INTEGER	NUMBER (10)
DECIMAL (p, [q])	NUMBER (p, [q])
NUMERIC (p, [q])	NUMBER (p, [q])
FLOAT [(p)]	FLOAT [(p)]
CHAR (n)	CHAR (n)
DATETIME	DATE
TIMESTAMP	NUMBER

MONEY	NUMBER (19,14)
VARCHAR (n)	VARCHAR2 (n)
NCHAR (n)	CHAR (n)
NVRCHAR (n)	VARCHAR2 (n*2)
TEXT	CLOB
IMAGE	BLOB

Razlika u numeričkim tipovima podataka, gdje na Oracle imamo samo *number* dok na SQL Server imamo više njih. Kada gledamo tekstualne tipove podatka, kod Oracle imamo *varchar2* kao i kod SQL Server *varchar* koji koristimo za većinu teksta dok je *char* tekstualni tip podatka češće korišten na SQL Server nego Oracle.

1.6. Modeliranje podataka

Modeliranja podataka su skup pravila koja određuju kako može izgledati logička struktura baze. To je osnova za koncipiranje, projektiranje i implementiranje baze. Podaci su logički organizirani po nekom modelu. Model podataka je reprezentacija elemenata (entiteta ili objekata) ključnih za sustav te kako su elementi međusobno povezani. Izrađuje se i modificira dok dovoljno dobro ne predstavlja sustav. Modeliranje se najčešće (preporučljivo) radi na početku projekta prilikom prikupljanja zahtjeva. Model je veza između korisnika i dizajnera sustava, jedinstveni jezik koji bi trebali razumjeti oba „svijeta“. Vrste oblikovanja su konceptualno, logičko i fizičko. Konceptualno oblikovanje je shema cijele baze, sastavljena od entiteta, atributa i veza. Logičko oblikovanje je logička shema, koja je u slučaju relacijskog modela sastavljena od relacija (tablica). Sastavni dio logičkog oblikovanja je i takozvana normalizacija. Fizičko oblikovanje jest fizička shema cijele baze, dakle opis njezine fizičke građe. Osnovni koncepti ER modela su entiteti, atributi i relacije, koji su detaljno opisani dalje u tekstu.

1.6.1. Entiteti i atributi

Entitet (engl. *Entity*) je sve ono što možemo identificirati iz stvarnog svijeta. To može biti biće, stvar, pojava, objekt, koncept itd. svaki entitet predstavlja skup onog što opisuje. Tako npr. škola jest entitet zbog toga što škole možemo razlikovati po lokaciji, godini izgradnje, vrsti i sl.

Atribut (engl. *Attribute*) predstavlja određeno svojstvo ili obilježje nekog entiteta. U kontekstu baza podataka atributi su sve ono što pobliže opisuje entitet. Tako su navedeni pojmovi (lokacija, dogina izgradnje, vrsta …) atributi entiteta škola.

Kako bismo svaki entitet mogli identificirati, potrebno ga je na neki način označiti kao jedinstvenog. To se postiže upotrebom primarnog ključa (engl. *Primary key*) koji može biti jedan od osnovnih atributa entiteta kao npr. OIB osobe ili to može biti nekakav broj koji kada se jednom upotrijebi nema mogućnost da bude dodijeljen nekom drugom entitetu. Primarni ključ se također može kreirati i korištenjem više od jednog atributa i tada se n naziva kompozitnim primarnim ključem (engl. *Composite primary key*).

1.6.2. Relacije

Entiteti se uvijek nalaze u nekakvim odnosima koji su u bazi podataka predstavljeni vezama ili relacijama (engl. *Realtion*). Postoji više vrsta veza između entiteta, no najčešće korištene su binarne veze (engl. *Binary Relationship*) koje imaju više funkcionalnosti:

- 1:1 (jedan na jedan) -> Svaki entitet jednog skupa može, ali i ne mora biti povezan samo s jednim entitetom drugog skupa.
- 1:N (jedan naprema više) -> Relacija jedan na više kaže da jedan element može pripadati više drugih elemenata, ali ti drugi elementi mogu pripadati samo jednom elementu. Ova relacija je najkorištenija u većini baza podataka pa tako i u ovoj.
- M:N (više naprema više) -> Jedan entitet može imati više elemenata iz drugog i obrnuto. Ovakva relacija nije poželjna u bazama podataka, a s obzirom na to da se pojavljuje dosta često onda se rješava tako što se uvede novi entitet čiji će primarni ključ u većini slučajeva biti sastavljen od primarnih ključeva osnovnih entiteta. Takav primarni ključ tada postaje kompozitni primarni ključ.

1.6.3. Data Modeler

Pri izradi modela podataka pomoću Data Modelera, preporučeno je da nazivi tablica i stupaca započnu slovom, ne smiju se koristiti rezervirane riječi. Nazivi tablica moraju biti jedinstveni unutar iste sheme, te također i nazivi stupaca moraju biti jedinstveni u jednoj tablici. U Data Modeleru se izrađuje i DDL koji će nam poslužiti pri izradi naše baze podataka u oba sustava. U drugom poglavlju ćemo moći vidjeti na točnom primjeru kako se podešava Data Modeler.

2. Razvoj modela baze podataka

Nakon što smo se upoznali s osnovnom teorijom vezanom uz područje relacijskih baza podataka prikladno je da ju pretočimo u praktični dio. Za potrebe ovog projekta osmišljena je i realizirana baza podataka koja može učinkovito poslužiti za potrebe srednje škole. Srednja škola je ustanova za učenike koji su završili prvih 9 razreda osnovne škole, te se odlučili na sljedeća četiri ili tri razreda (ovisno o vrsti koju su upisali). Ova baza će omogućiti korisniku uvid u osnovne podatke kao što su osobine učenika, predmeta, ocjena koje su učenici postigli, njihovih nastavnika te razreda koje su završili ili ne.

Baza podataka treba imati ažurnu evidenciju učenika (ime i prezime, OIB, ime oca, adresa, …), razreda (godina, broj, razrednik, …), nastavnika, predmeta i učeničkih ocjena. Osnovni zahtjevi korisnika su sljedeći:

- imati uvid u učenikove ocjene,
- imati neke statističke informacije (broj učenika koji su uspješno (ili ne) završili određeni razred, prosjek ocjena za razred i učenike, uspješni predmet, …).

Važno je spomenuti da jedan nastavnik može raditi u više škola, a također i u jednoj školi može raditi više nastavnika. Isto tako jedan nastavnik može predavati više predmeta, dok jedan predmet može predavati više nastavnika. Učenik može imati više predmeta, ali jedan predmet može imati i više učenika. Također, u jednoj školi može biti više učenika, ali učenik ne može ići u više škola. U razred je dopušteno da ide više učenika, ali učenik ne može ići u više razreda. Škola može sadržavati više razreda, ali jedan određen razred ne može biti u više škola. Županija može imati više mjesta dok mjesto ne može biti u više županija istovremeno. Pa tako i jedna škola ne može biti u više mjesta, ali mjesto može sadržavati više škola a i ne mora sadržavati niti jednu.

2.1. Relacijski model

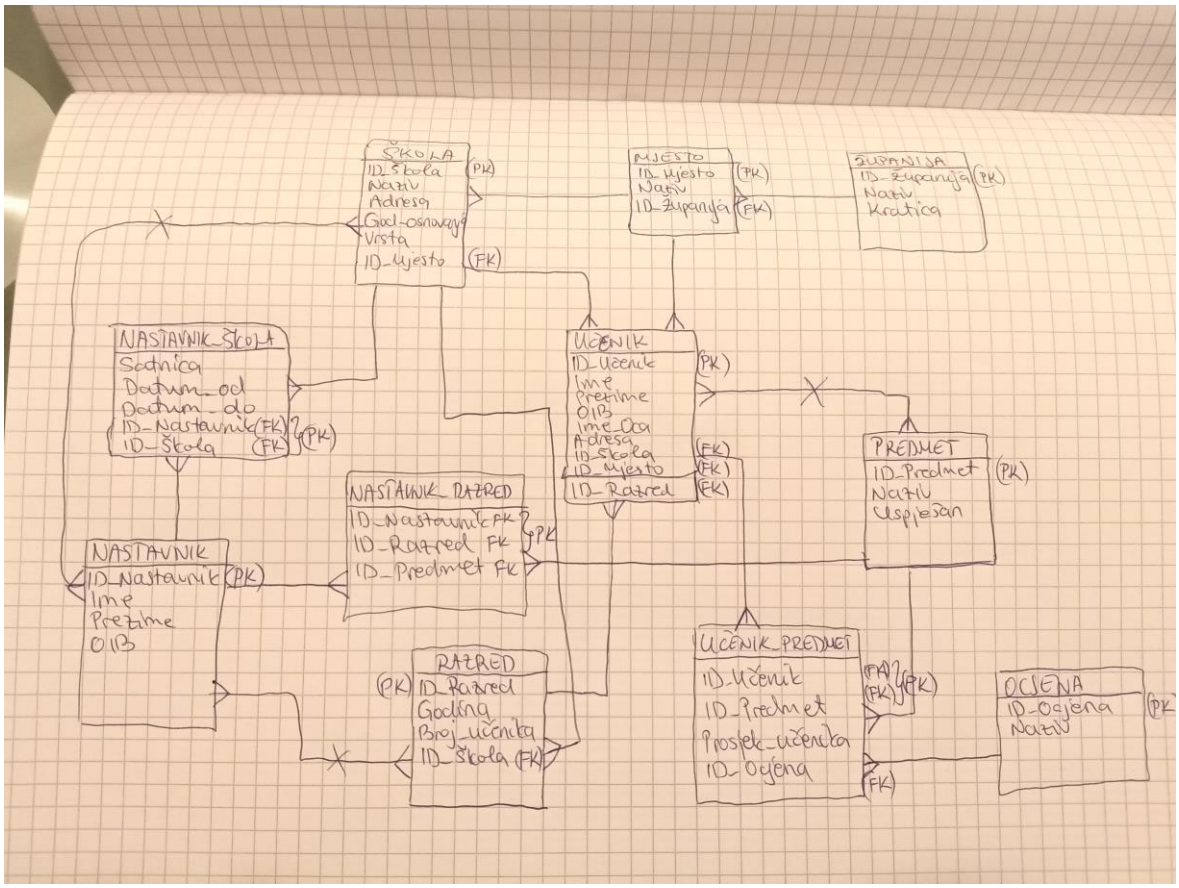
Kako bi zamišljena baza podataka postala nešto što nije samo zamisao, nego proizvod koji se može koristiti potrebno je proći kroz određene faze razvoj baza podataka. Razvoj baze tj. modeliranje podataka započinje konceptualnim modeliranjem (koje rezultira konceptualnim modelom) čija je svrha izgradnja osnovne sheme baze podataka.

Konceptualni model se reprezentira, u ovom slučaju kao skica na papiru, kao shema koja sadrži sve potrebne entitete, atribute te veze između entiteta. Razvoj konceptualnog modela u suštini započinje korištenjem papira i olovke. U ovoj početnoj fazi osmišljavanja baze potrebno je prikupiti sve potrebne informacije koje želimo da sadrži naša baza podataka.

Iz drugog poglavlja, nakon kratkog opisa funkcionalnosti koje će nuditi baza podataka potrebno je taj opis predočiti u konceptualno shemu. Pri tome je potrebno znati značenje određenih pojmova poput entiteta, relacija između entiteta, atributa, primarnog ključa itd. , koje možemo naći čitajući dalje kroz rad.

2.2. Izrada konceptualnog modela

Nakon upoznavanja s nekim od osnovnih pojmova relacijskih baza podataka, njihovog značenja, kao i njihove primjene u bazi podataka za srednju školu konačno možemo opisati zamišljenu bazu te ju skicirati i vizualno predočiti.



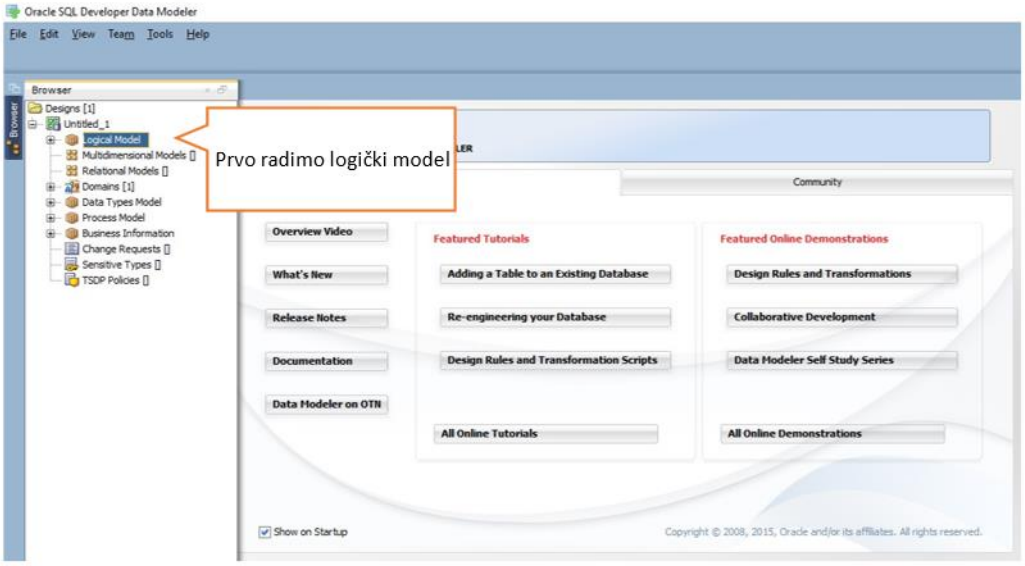
Slika 2.1 Konceptualni model

Iz slike možemo vidjeti da se osmišljena baza sastoji od 11 entiteta s njihovim atributima, primarni (otisnuti podebljano) i strani (otisnuti koso) ključeva:

a. ŠKOLA (**id_škola**, naziv, adresa, god_osnovanja, vrsta, *id_mjesto*)
b. MJESTO (**id_mjesto**, naziv, *id_županija*)
c. ŽUPANIJA (**id_županija**, naziv, kratica)
d. NASTAVNIK_ŠKOLA (satnica, datum_od, datum_do, **id_nastavnik**, **id_škola**)
e. UČENIK (**id_učenik**, ime, prezime, oib, ime_oca, adresa, *id_škola*, *id_mjesto*, *id_razred*)
f. PREDMET (**id_predmet**, naziv, uspješan)
g. NASTAVNIK_RAZRED (**id_nastavnik**, **id_razred**, *id_predmet*)
h. NASTAVNIK (**id_nastavnik**, ime, prezime, oib)
i. RAZRED (**id_razred**, godina, broj_učenika, *id_škola*)
j. UČENIK_PREDMET (**id_učenik**, **id_predmet**, prosjek_učenika, *id_ocjena*)
k. OCJENA (**id_ocjena**, naziv)

2.3. Logički i relacijski model

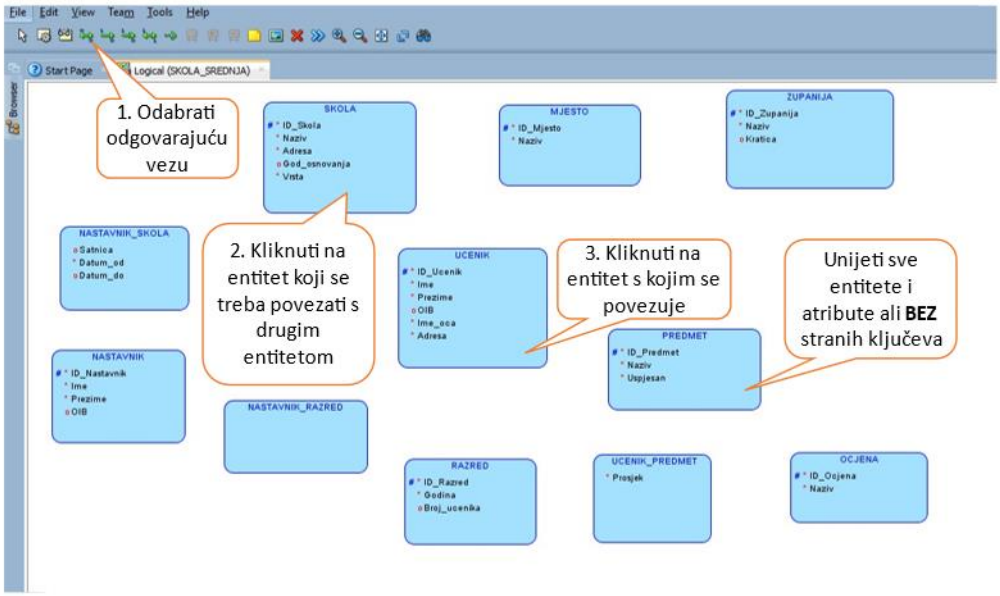
Nakon što smo osmislili bazu, kreirali njenu početnu shemu, na red dolazi druga faza razvoja baze podataka odnosno logičkog modeliranja podataka. Za izradu logičkog modela korišten je Oracle SQL Developer Data Modeler (u nastavku Data Modeler). (Slika 2.2)



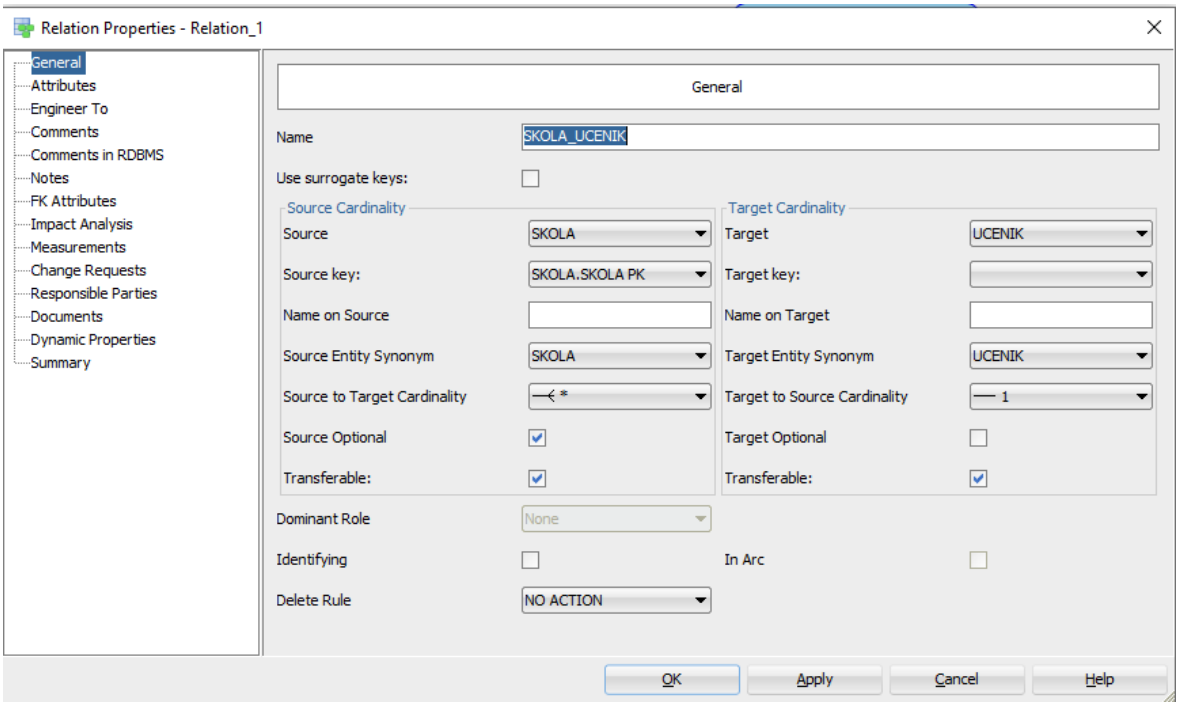
Slika 2.2 Izrada logičkog modela u Data Modeleru

U ovom dijelu razvoja baze podataka stvaramo entitete te njihove pripadne atribute (sve osim stranih ključeva) i njihova svojstva. Isto tako, kreiramo veze između entiteta i određujemo njihova svojstva. Kako bi došli do logičkog modela moramo proći kroz nekoliko koraka.

Kada stvorimo sve potrebne entitete i njihove atribute, potrebno ih je povezati odgovarajućim relacijama kao i postaviti svojstva svake veze.

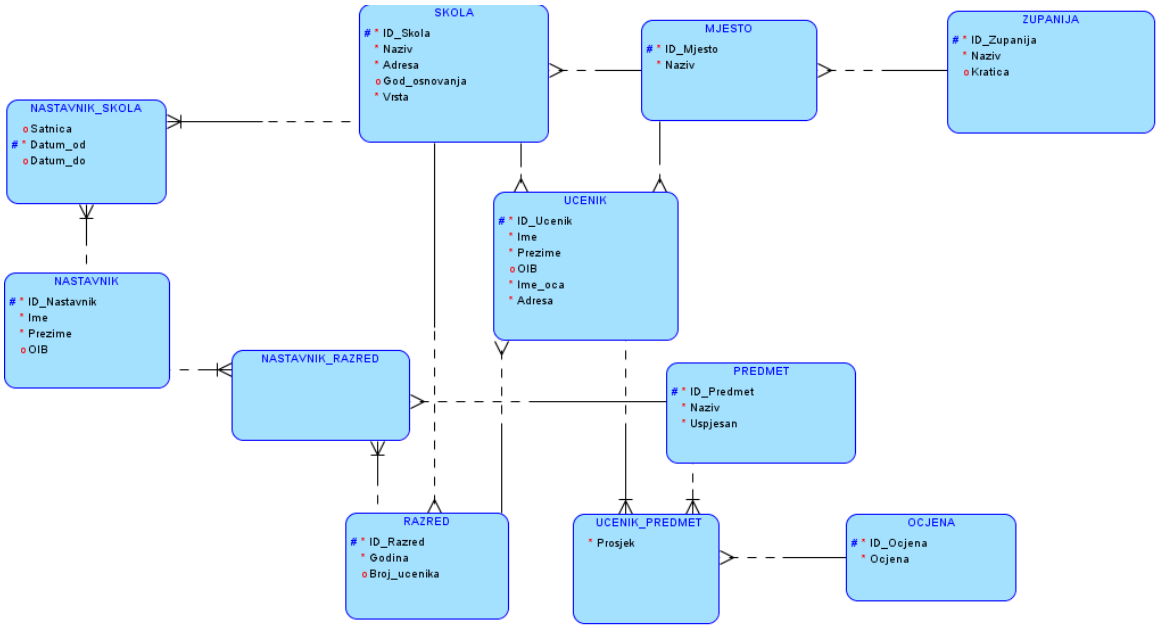


Slika 2.3 Dodavanje veze između entiteta



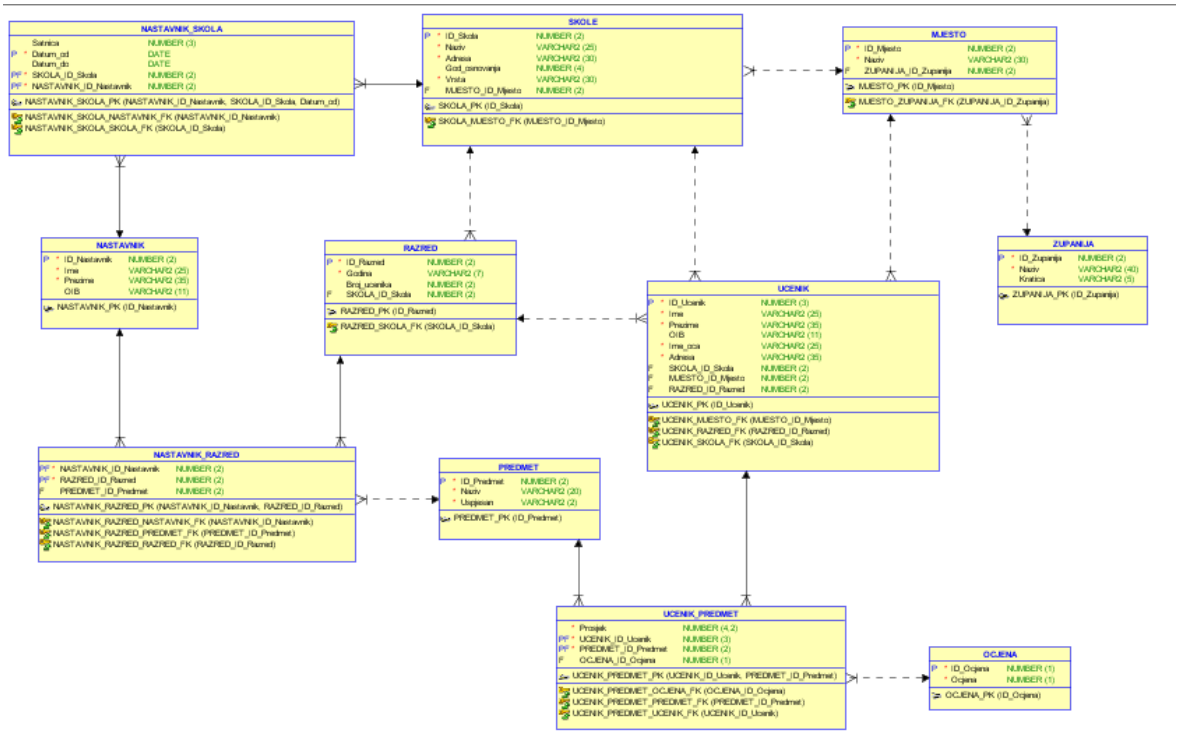
Slika 2.4 Postavljanje svojstava dodane veze

Kreiranjem svih entiteta, atributa te njihovih veza dobijemo logički model u kojem možemo vidjeti sve spomenute pojmove kao i primarne i kompozitne ključeve.



Slika 2.5 Logički model

Sljedeći korak jest kreiranje relacijskog modela baze podataka. To ćemo postići tako što ćemo pronaći ikonu „>>“ te klikom na nju kreirati relacijski model. U prikazanom relacijskom modelu osim entiteta, relacija i atributa, možemo vidjeti i tipove podataka kao i strane ključeve u svim tablicama te vidimo i ograničenja svih atributa.

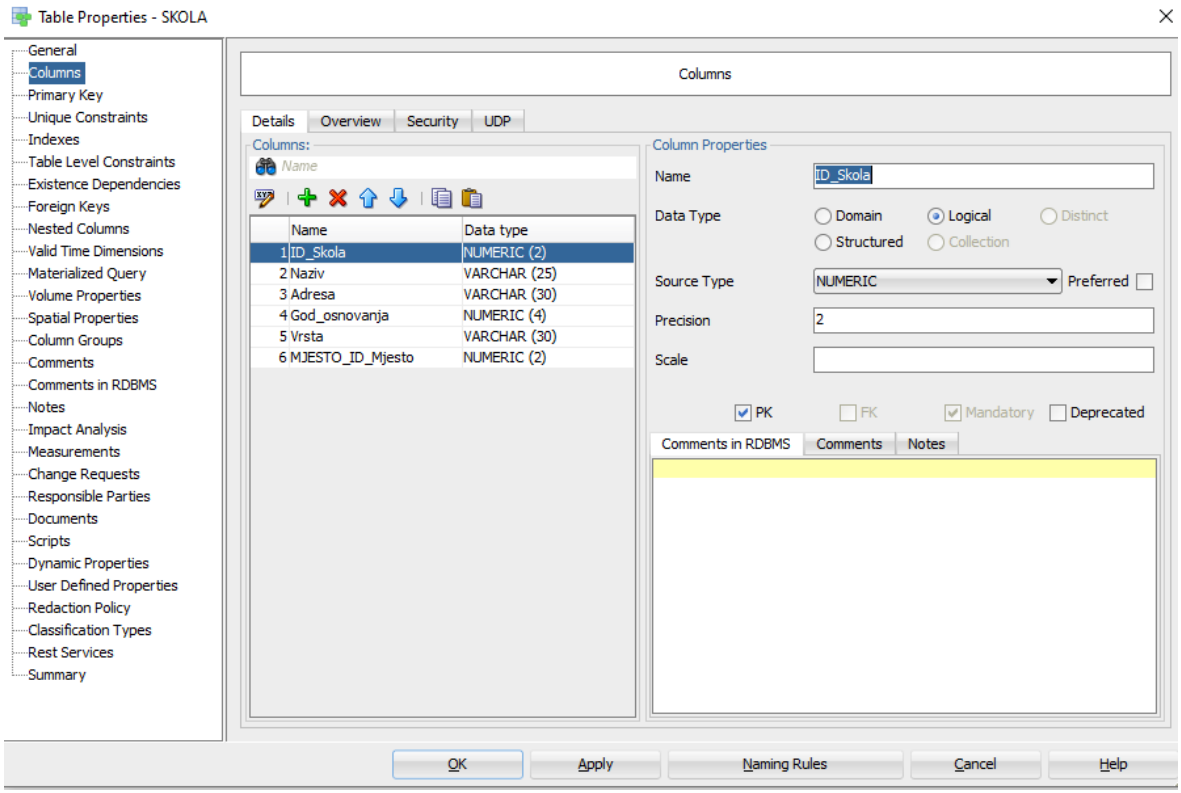


Slika 2.6 Relacijski model

2.4. Podešavanje u Data Modeleru

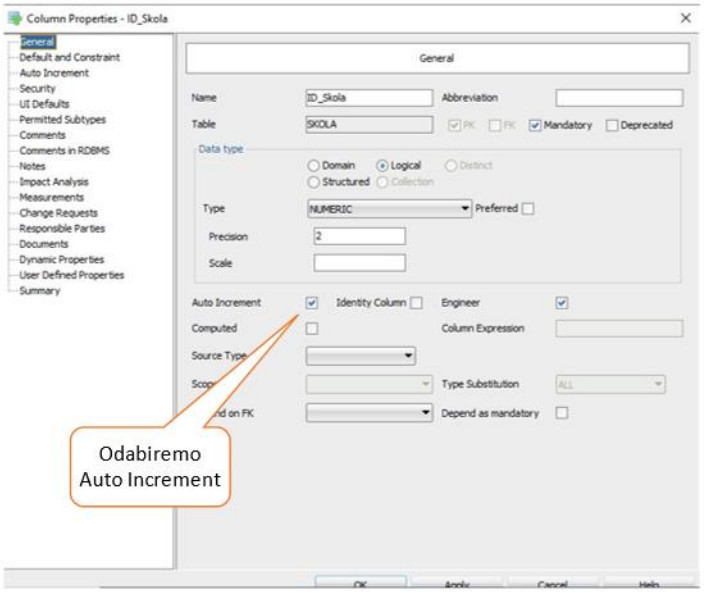
Nakon kreiranja logičkog i relacijskog modela baze podataka potrebno je izraditi DDL kojeg ćemo izvršiti u Oracle SQL Developeru (u nastavku Developer) kako bismo relacijski model baze podataka pretočili u relacijsku bazu podataka.

S obzirom na to da u Oracle-u ne postoji ugrađeni tip podataka za primarne ključeve, prije kreiranja DDL-a potrebno je svim primarnim ključevima napraviti sekvencu i okidač. Odaberemo atribut za koji želimo napraviti sekvencu i okidač.



Slika 2.7 Odabir atributa za sekvencu i okidač

Svakom primarnom ključu, također je potrebno označiti i auto inkrement.

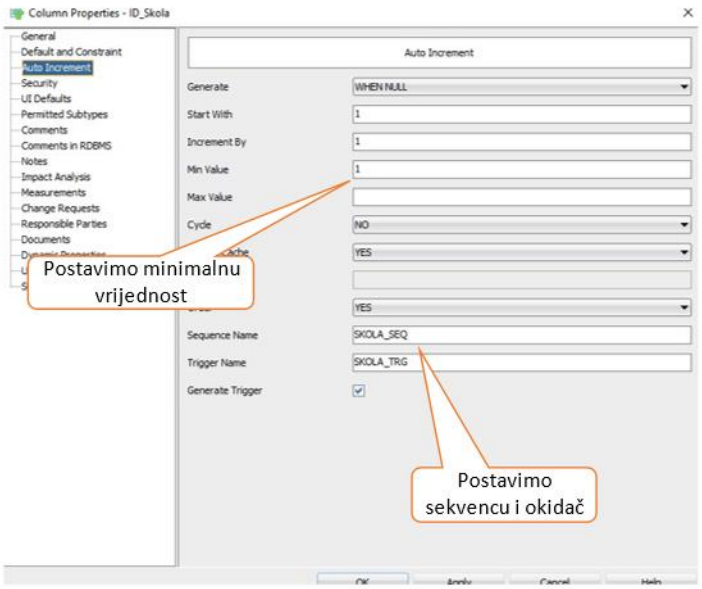


Slika 2.8 Auto Increment

Naravno, bitno je i postaviti svojstva auto incrementa.

Sekvenca (engl. *sequence*) predstavlja shemu objekta koja generira niz numeričkih vrijednosti. Koristi se vrlo često u bazama podataka (kao npr. u ovom slučaju) zato što svaki atribut u tablici koji predstavlja primarni ključ zahtjeva jedinstvenu vrijednost (u ovom slučaju numeričku vrijednost).

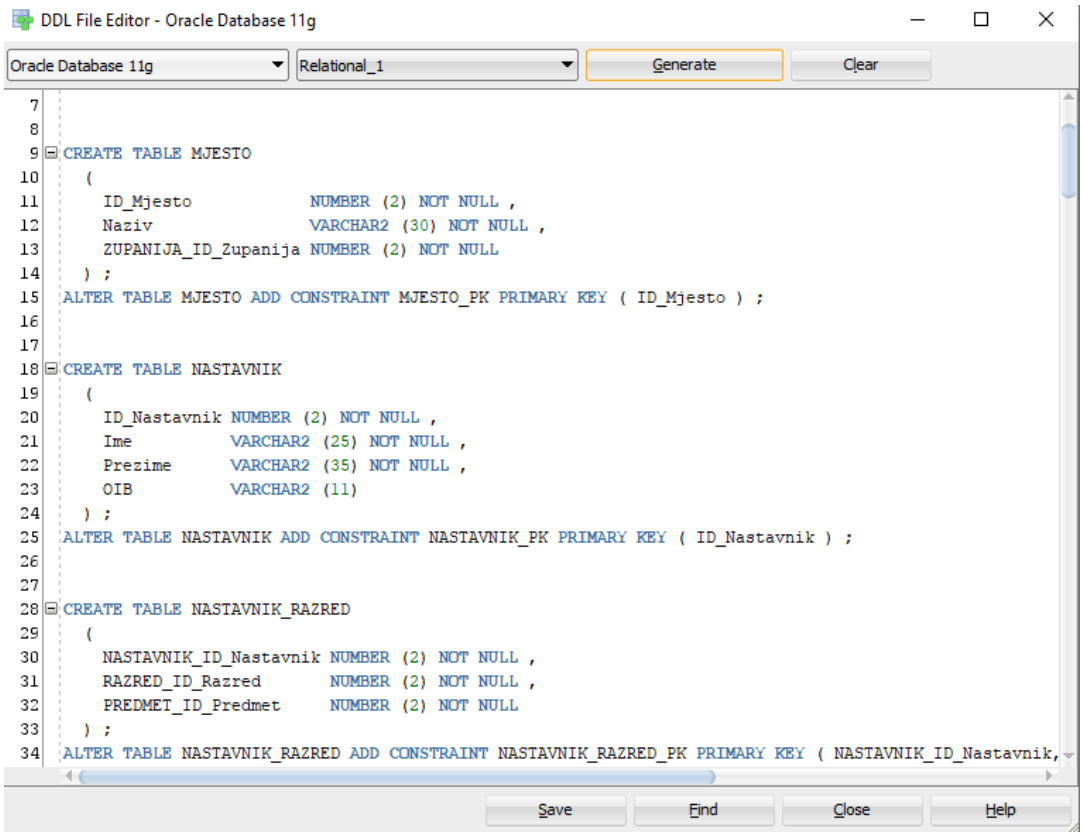
Okidač (engl. *trigger*) je procedura koja je pohranjena u bazi podataka. Ona se automatski poziva svaki put kada se dogodi neki poseban događaj u bazi kao npr. umetanje novog reda u određenu tablicu.



Slika 2.9 Postavljanje svojstava auto incrementa

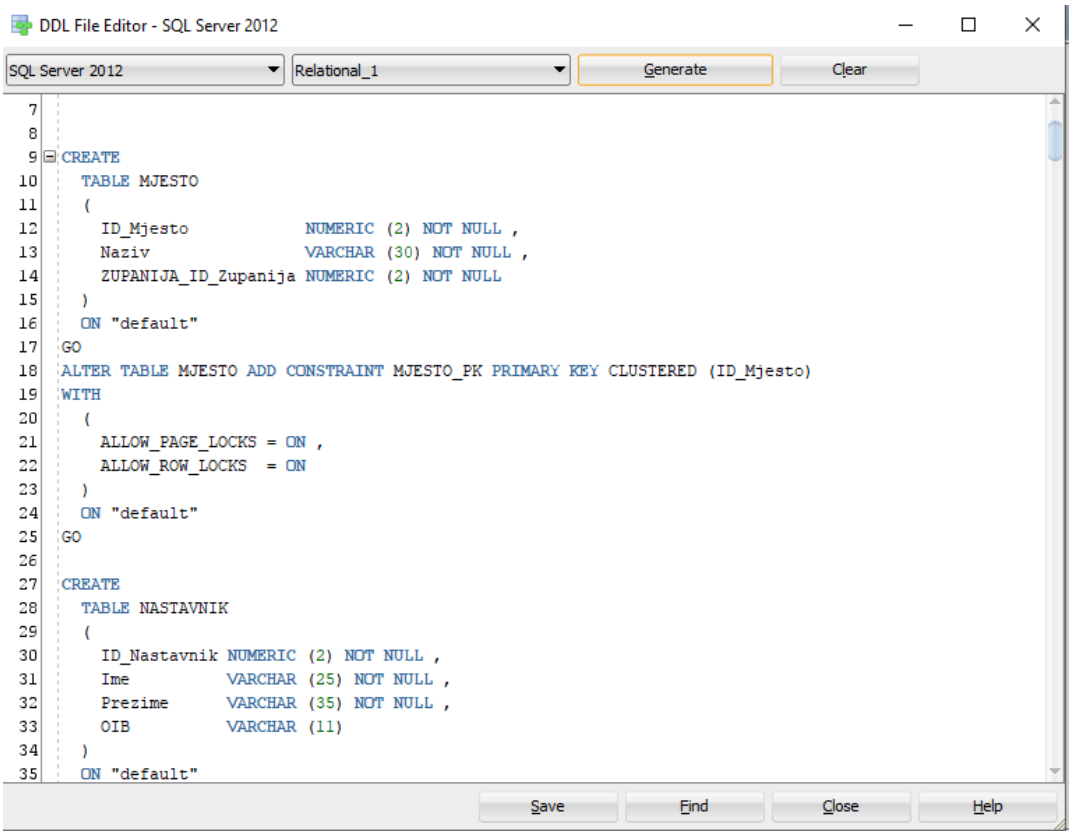
2.5. Izrada DDL-a prema modelu

Sljedeći korak je generiranje DDL-a. klikom na ikonu *Generate DDL* u Data Modeleru (), generiramo traženi DDL za Oracle () i SQL Server (**Pogreška! Izvor reference nije pronađen.**) koji predstavlja kod za kreiranje svih entiteta, atributa, sekvenci i okidača naše baze podataka.



Slika 2.10 DDL za Oracle

Važno je spomenuti da pri izradi DDL možemo odabrati i bazu za koju želimo izraditi željeni DDL. DDL nam nudi neke od takvih baza kao što su baza za Oracle Database 12c, Oracle Database 11g, Oracle Database 10g, Oracle 9i, SQL Server 2012, 2008, 2005, 2000 i druge. Kako DDL ne nudi za SQL Server godinu mlađu od 2012, korištena je upravo ova zadnja godina. Stoga je potrebno odabrati u ovom slučaju bazu za Oracle i SQL Server te ih spremiti.



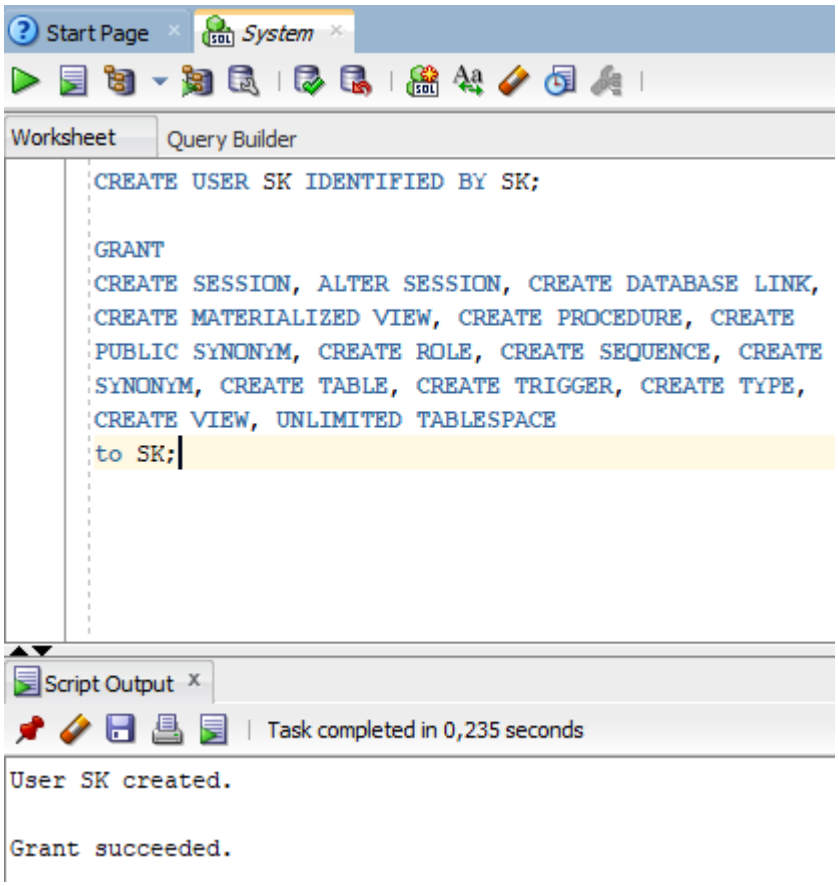
Slika 2.11 DDL za SQL Server

2.6. Usporedba izrade baze podataka na SQL Server i Oracle

Nakon što smo u Data Modeleru kreirali DDL, slijedi nekoliko usporedba vezanih za SQL Server i Oracle kao što su spajanje na bazu, usporedba tipova podataka, razlike u transakcijskom dijelu, usporedba strukture pohrane te razlike u cijeni.

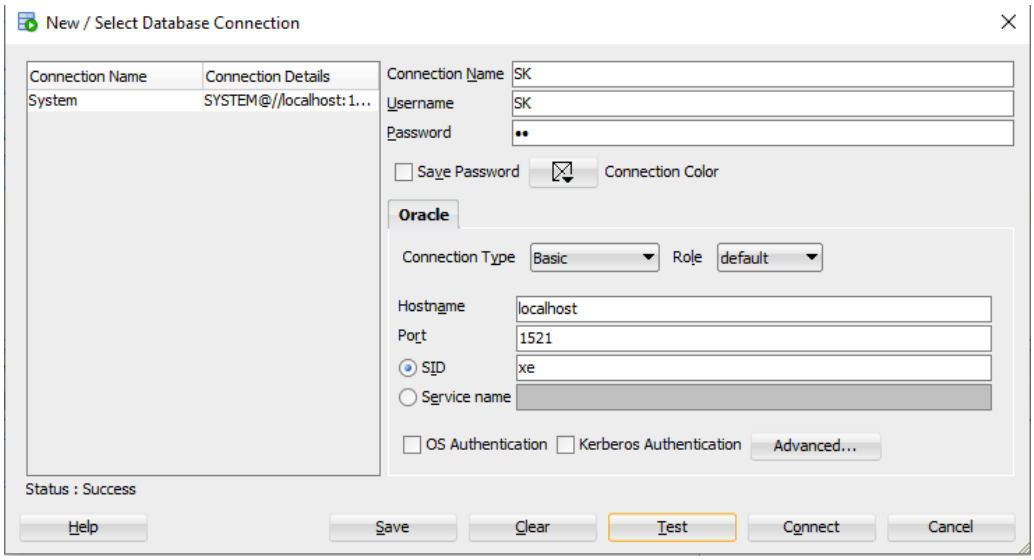
2.6.1. Izrada baze podataka na Oracle-u

Kod Oracle-a je potrebno otvoriti SQL Developer te u njemu kreirati novog korisnika jer ono što SQL Server naziva bazom, Oracle to naziva korisnikom, dati mu određene ovlasti koje mu dopuštaju upravljanje bazom podataka.



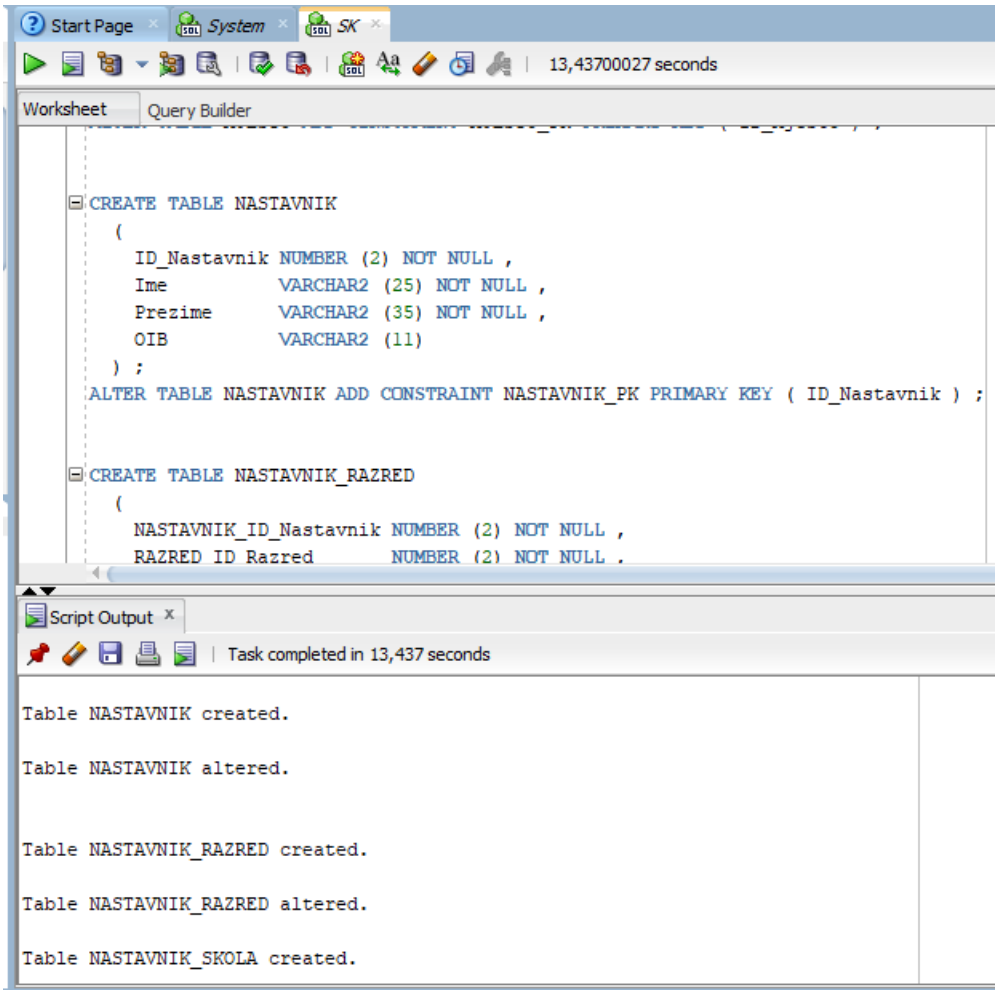
Slika 2.12 Kreiranje korisnika i davanje ovlasti korisniku

Nakon dodijele određenih ovlasti, potrebno je zbog sigurnosti napraviti prijavu na novog korisnika.



Slika 2.13 Prijava na novog korisnika Oracle

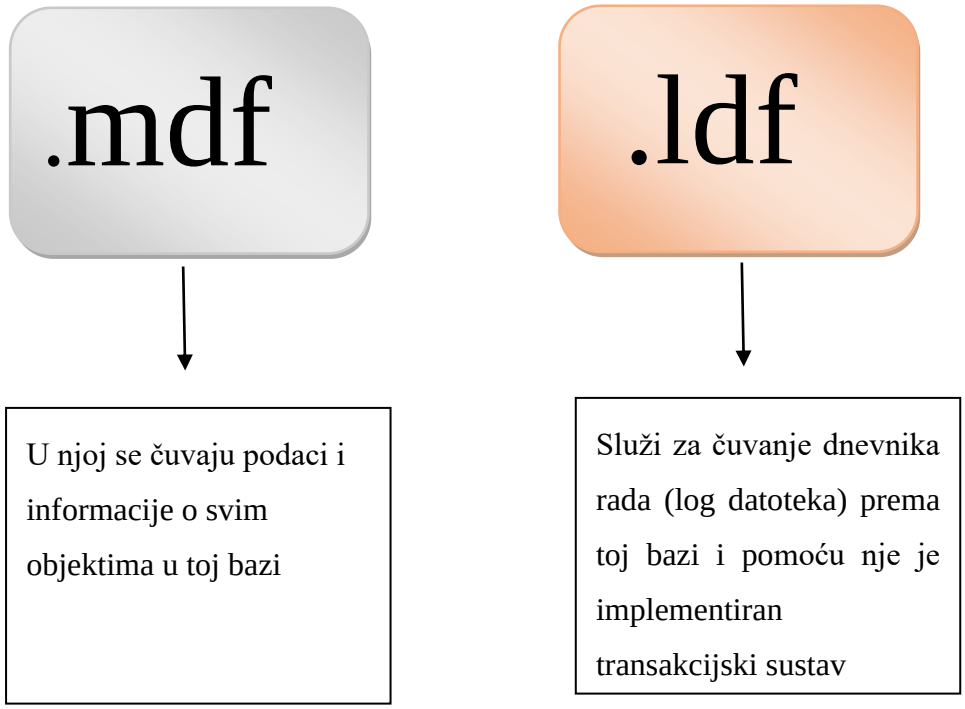
Kako bi smo mogli dalje nastaviti s radom naše baze, potrebno je prethodni generirani DDL izvršiti u konekciji kreiranog novog korisnika.



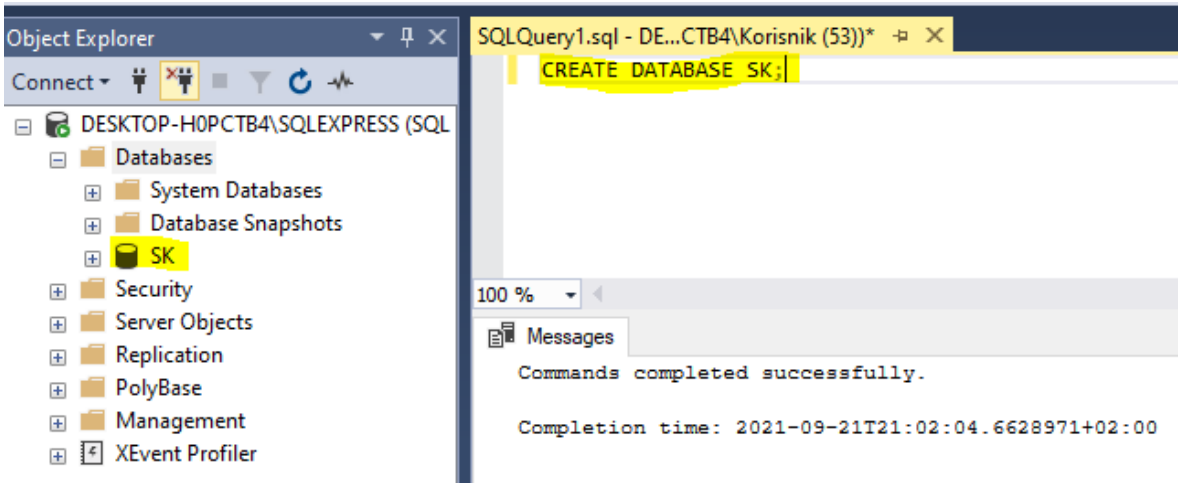
Slika 2.14 Izvršavanje DDL-a na konekciji korisnika kod Oracle

2.6.2. Izrada baze podataka na SQL Server

Kod SQL Servera potrebno je otvoriti Microsoft SQL Server Management Studio 18 te napraviti novu bazu podataka. Jednu bazu podataka čine dvije fizičke datoteke:

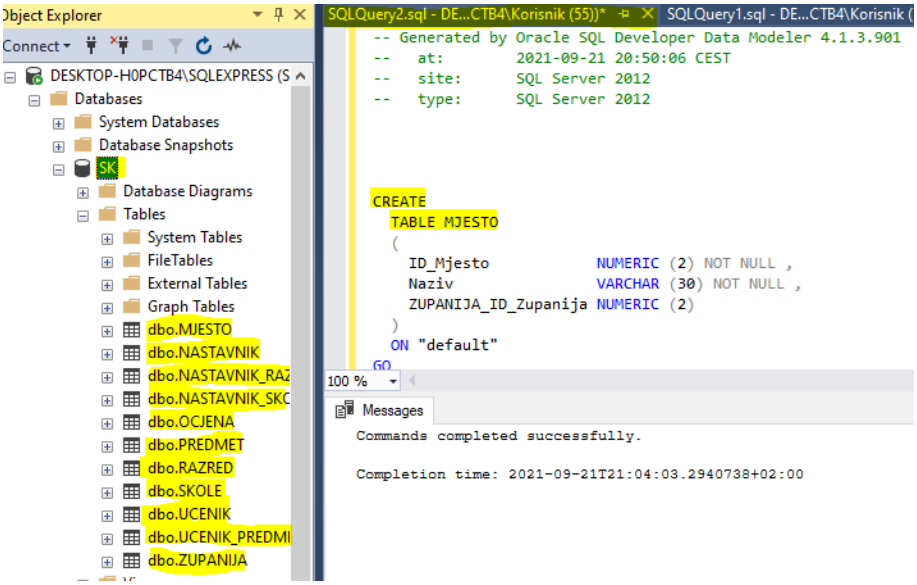


Kreiranje baze je vrlo jednostavno. To radimo pomoću grupe naredbi ***CREATE DATABASE*** *NazivBaze* i naša baza je već stvorena.



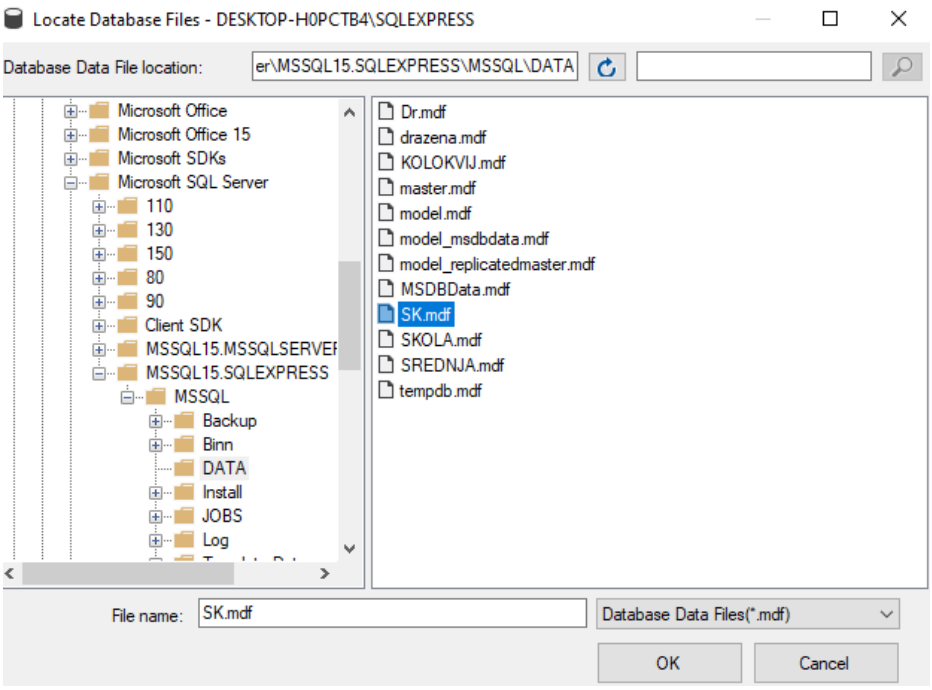
Slika 2.15 Kreiranje baze

Dalje se izrađuju sve tablice i tu ćemo koristiti već gotovu DDL datoteku spomenutu u podpoglavlju prije.



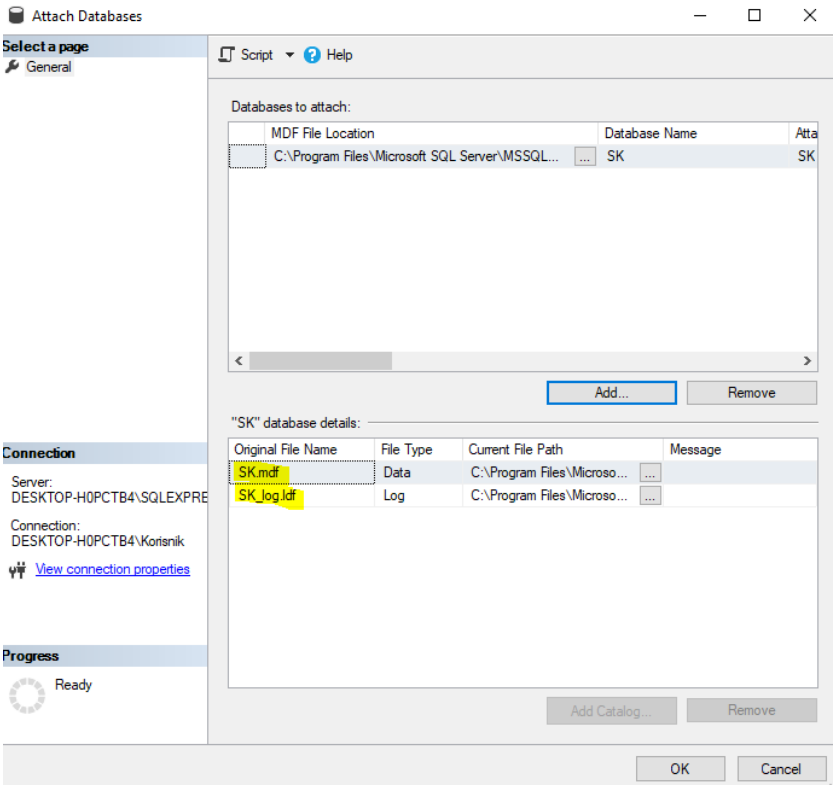
Slika 2.16 Izvršavanje DDL-a i kreiranje tablica

Dakle, u SQL Serveru nema nikakvih dodjeljivanja prava i sl. zbog toga je jednostavniji i lakši za naučiti rukovati s njim, ali ne i sigurniji. Za spajanje na već postojeću bazu potrebno je ići na *Database* u SQL Exploreru kliknuti na *Attach*, te učitati gotovu .mdf datoteku.



Slika 2.17 Učitavanje gotove baze (.mdf datoteka)

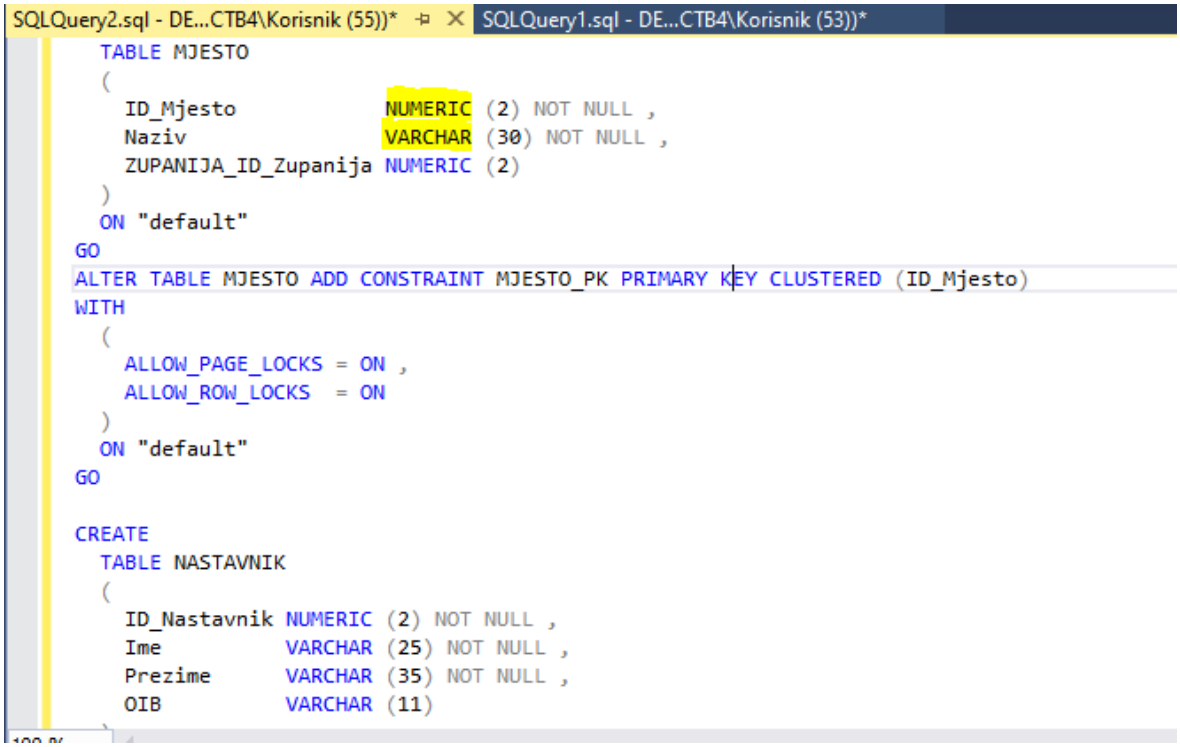
Dakle, potrebno je učitati samo gotovu .mdf datoteku i nakon učitavanja dobijemo automatski i našu ldf datoteku.



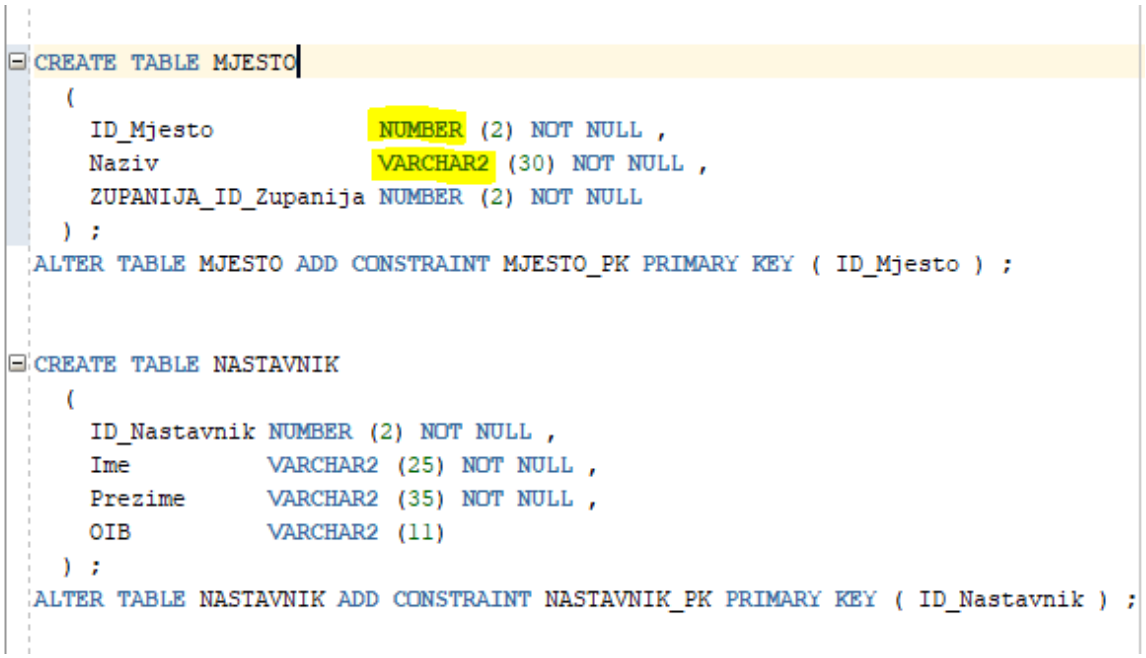
Slika 2.18 Učitavanjem .mdf datoteke se učita i .ldf datoteka

2.6.3. Tipovi podataka SQL Server u odnosu na Oracle

Jos jedna razlika je u tipovima podataka. Kao što možemo vidjeti na slici, tipovi podataka u SQL Serveru su NUMERIC, VARCHAR, a kod Oraclea NUMBER (s ograničenjem), VARCHAR2.



Slika 2.19 Tipovi podataka kod SQL Servera



```
CREATE TABLE MJESTO
  (
    ID_Mjesto            NUMBER (2) NOT NULL ,
    Naziv                VARCHAR2 (30) NOT NULL ,
    ZUPANIJA_ID_Zupanija NUMBER (2) NOT NULL
  ) ;
ALTER TABLE MJESTO ADD CONSTRAINT MJESTO_PK PRIMARY KEY ( ID_Mjesto ) ;


CREATE TABLE NASTAVNIK
  (
    ID_Nastavnik NUMBER (2) NOT NULL ,
    Ime          VARCHAR2 (25) NOT NULL ,
    Prezime      VARCHAR2 (35) NOT NULL ,
    OIB          VARCHAR2 (11)
  ) ;
ALTER TABLE NASTAVNIK ADD CONSTRAINT NASTAVNIK_PK PRIMARY KEY ( ID_Nastavnik ) ;
```

Slika 2.20 Tipovi podataka kod Oraclea

Sljedeće što možemo primijetiti jest da se kod SQL Servera iza primarnih ključeva ne generiraju sekvence i okidači kao kod Oraclea. Sekvence i okidači , kao što je prije spomenuto u tekstu , služe da bi svaki primarni ključ bio jedinstven. Oni kod SQL Server postoje ali ih nije potrebno koristiti za generiranje primarnih ključeva jer postoji *autonumber*.

```
ALTER TABLE UCENIK
ADD CONSTRAINT UCENIK_SKOLA_FK FOREIGN KEY
(
SKOLA_ID_Skola
)
REFERENCES SKOLE
(
ID_Skola
)
ON
DELETE
  NO ACTION ON
UPDATE NO ACTION
GO


-- Oracle SQL Developer Data Modeler Summary Report:
--
```

Slika 2.21 Generiranje sekvence i okidača kod SQL Servera nije potrebno

Sekvence i okidači , kao što je prije spomenuto u tekstu , služe da bi svaki primarni ključ bio jedinstven. Oni kod SQL Server postoje ali ih nije potrebno koristiti za generiranje primarnih ključeva jer postoji *autonumber*.

```
CREATE SEQUENCE MJESTO_SEQ START WITH 1 MINVALUE 1 NOCACHE ORDER ;
CREATE OR REPLACE TRIGGER MJESTO_TRG BEFORE
  INSERT ON MJESTO FOR EACH ROW WHEN (NEW.ID_Mjesto IS NULL) BEGIN :NEW.ID_Mjesto := MJESTO_SEQ.NEXTVAL;
END;
/

CREATE SEQUENCE NASTAVNIK_SEQ START WITH 1 MINVALUE 1 NOCACHE ORDER ;
CREATE OR REPLACE TRIGGER NASTAVNIK_TRG BEFORE
  INSERT ON NASTAVNIK FOR EACH ROW WHEN (NEW.ID_Nastavnik IS NULL) BEGIN :NEW.ID_Nastavnik := NASTAVNIK_SEQ.NEXTVAL;
END;
/
```

Slika 2.22 Generiranje sekvence i okidača kod Oracle-a

2.6.4. Razlika u transakcijskom dijelu SQL Server i Oracle

Kao što je ranije rečeno, jezik koji koriste SQL Server i Oracle za dohvaćanje podataka je različit. SQL Server koristi T-SQL (eng. Transact-SQL), dok Oracle koristi PL/SQL (eng. Procedural SQL).

Početak transakcije je eksplicitno kod SQL Server, a implicitno kod Oracle-a dok za kraj transakcije SQL Server koristi *auto-commit mod* po „defaultu“ , a Oracle zahtjeva *COMMIT* izjavu po „defaulutu“.

```
INSERT INTO MJESTO
    VALUES(6, 'Zadar')
INSERT INTO ZUPANIJA
    VALUES('ZDŽ','Zadarska Županija',6)
```

Slika 2.23 Kraj transakcije – SQL Server

Na primjer, ako radimo *UPDATE* u SQL Serveru, ako nešto slučajno pogriješimo, povratka nema, ali kod Oracle to nije isto. Prije nego napravimo *UPDATE* na Oracle bazi

mi možemo izvršiti provjeru te izmijeniti ako bi bile neke greške prije nego se pošalje zahtjev *COMMIT*.

```
INSERT INTO MJESTO
  VALUES(6, 'Zadar');
COMMIT;
INSERT INTO ZUPANIJA
  VALUES('ZDŽ','Zadarska Županija',6);
COMMIT;
```

Slika 2.24 Kraj transakcije – Oracle

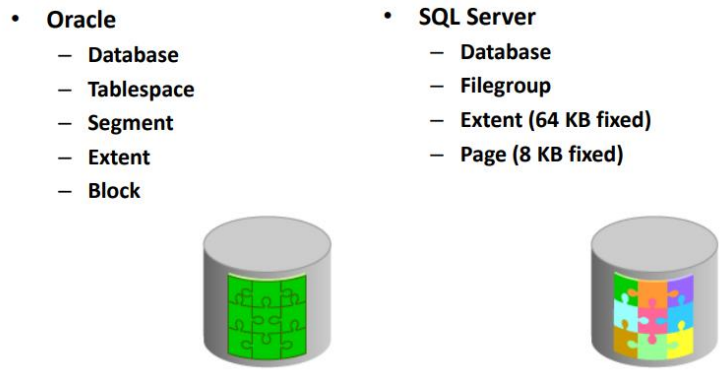
Transakcija baze podataka simbolizira jedinicu rada koja se izvodi u sustavu upravljanja bazom podataka (ili sličnom sustavu) protiv baze podataka i tretira se na koherentan i pouzdan način neovisno o drugim transakcijama. Transakcija obično predstavlja bilo kakvu promjenu u bazi podataka. Transakcije u okolini baze podataka imaju dvije glavne svrhe:

1. pružiti pouzdane jedinice rada koje omogućuju točan oporavak od kvarova i održavaju bazu podataka dosljednom čak i u slučajevima kvara sustava, kada se izvršavanje zaustavlja (potpuno ili djelomično) i mnoge operacije nad baza podataka ostaje nedovršena, s nejasnim statusom.
2. Da bi se osigurala izolacija između programa koji istovremeno pristupaju bazi podataka. Ako se ova izolacija ne pruži, ishodi programa mogu biti pogrešni.

Transakcija baze podataka, po definiciji, mora biti atomska (mora ili dovršiti u cijelosti ili uopće ne imati učinak), dosljedna (mora odgovarati postojećim ograničenja u bazi podataka), izolirani (ne smije utjecati na druge transakcije) i trajni (mora se zapisati u trajnu pohranu). Transakcija se obično započinje pomoću naredbe *BEGIN*. Kada sustav obradi izraz *COMMIT*, transakcija završava uspješnim dovršenjem. Izraz *ROLLBACK* također može završiti transakciju poništavajući bilo koji posao izveden od *BEGIN* .

2.6.5. Usporedba strukture pohrane

Osnovna jedinica za pohranu na SQL Server je *Page* (8 KB fiksno) dok je za alokaciju prostora za tablice i indekse *Extent* (64 KB fiksno). Oracle ima osnovnu jedinicu za pohranu *Block* i taj se jedan logički blok sastoji od jednog ili više blokova OS-a. Veličina logičkog bloka je definirana inicijalizacijskim parametrima.



Slika 2.25 Usporedba strukture pohrane

2.6.6. Razlike u cijeni i popularnosti

Neke od DBMS su izdvojene u tablici ispod, gdje možemo vidjeti da vodeće mjesto zauzima naravno Oracle, a na trećem je mjestu SQL Server. Vidimo da se to ne mijenjam baš kroz godine, da je Oracle svakako najpopularniji i najkorišteniji. Ono što je bitno jest cijena. Oracle je dosta skuplji od SQL Servera, kao što ćemo i tečajeve od Oraclea naći po znatno većoj cijeni od SQL Servera. Ono što svakako možemo vidjeti u tablici jest da je Elasticsearch ranije bio na osmom mjestu, te se pomaknuo u zadnje vrijeme na sedmo mjesto, a IBM Db2 odgurao na osmo mjesto. (DB-engines, 2022)

Tablica 2.1 Neke od popularnosti DBMS

Rang			DBMS	Model baze podataka	Rezultat		
Rujan 2022	Kolovoz 2022	Rujan 2021			Rujan 2022	Kolovoz 2022	Rujan 2021
1.	1.	1.	Oracle	Relacijski	1238.25	-22.54	-33.29
2.	2.	2.	MySQL	Relacijski	1212.47	+9.61	-0.06
3.	3.	3.	Microsoft SQL Server	Relacijski	926.30	-18.66	-44.55
4.	4.	4.	PostgreSQL	Relacijski	620.46	+2.46	+42.95
5.	5.	5.	MongoDB	Dokument	489.64	+11.97	-6.87
6.	6.	6.	Redis	Vrijednost ključa	181.47	+5.08	+9.53
7.	8.	8.	Elasticsearch	Pretraživač	151.44	-3.64	-8.80
8.	7.	7.	IBM Db2	Relacijski	151.39	-5.83	-15.16
9.	9.	11.	Microsfot Access	Relacijski	140.03	-6.47	+23.09
10.	10.	9.	SQLite	Relacijski	138.82	-0.05	+10.17

Oracle je u vlasništvu Oracle Corporation i može raditi na velikom broju platformi kao što su Windows, Linux, Solaris, HP-UX i OS-X. SQL Server je u vlasništvu Microsofta i može se koristiti samo na Windows platformu. Nedavno se verzija SQL Servera iz 2017. može pokrenuti i na Linux platformama.

Sve glavne SQL naredbe u Oracleu kao što su *INSERT*, *UPDATE*, *DELETE* i *MERGE* izvode se paralelno. Oracle redovito održava diferencijalne, potpune i inkrementalne sigurnosne kopije svojih podataka na razini datoteke. Glavne SQL naredbe u SQL Serveru kao što su *INSERT*, *UPDATE*, *DELETE* i *MERGE* izvode se serijski. SQL Server redovito održava potpune, djelomične i inkrementalne sigurnosne kopije svojih podataka.

Služba za brigu o kupcima, rješavanje problema i podrška tvrtke Oracle dostupni su 24×7 i pravodobno odgovaraju na vaše upite, ali će vam biti naplaćen svaki poziv podrške ovisno

o slučaju podrške. Rukovanje pogreškama također se učinkovito upravlja jer Oracle svaku novu transakciju baze podataka tretira kao novu vezu. SQL Server, s druge strane, pruža tehničke napomene, opise grešaka, skripte, zakrpe i ažuriranja bez ikakvih dodatnih troškova. Međutim, mogućnosti rukovanja pogreškama u SQL Serveru nisu vrlo razrađene budući da SQL Server izvršava svaku naredbu zasebno pa je prilično teško ispraviti bilo koju pogrešku uočenu u procesu.

Model cijena za Oracle osmišljen je tako da odgovara proračunu i ciljevima tvrtke. Oracle nudi 4 izdanja za svoje klijente - Oracle Database Standard Edition 2, Oracle Database Enterprise Edition, Oracle Database Personal Edition i Oracle Database Express Edition. Express Edition je besplatan za studente i prenosiv je. Mnogo toga o SQL Serveru u odnosu na Oraclea stvar je mišljenja, ali ono o čemu se najviše raspravlja je usporedba cijena. Licence za SQL Server znatno su jeftinije od Oracle. Prema jednoj procjeni, poslužitelj s 4 CPU-a i 4 jezgre po CPU-u koštao bi 380.000 USD s Oracleom, ali samo 114.000 USD s SQL Serverom. Troškovi rastu još više s dodatnim značajkama. Iako cijena naljepnice može biti niža na SQL Serveru, trebali bismo također uzeti u obzir ukupne troškove vlasništva tijekom vijeka trajanja baze podataka. To uključuje razmatranja kao što su podrška, održavanje i produktivnost tijekom svakodnevne upotrebe. U tom smislu, čini se da Oracle pobjeđuje, prema jednoj studiji Oracle Database 11g i Microsoft SQL Server 2015. Kvalificirani administratori baze podataka mogu obavljati tipične funkcije 41 posto brže u Oracleu nego u SQL Serveru. Ove uštede vremena se zbrajaju tijekom vremena. Ista studija procjenjuje da Oracle-ova poboljšanja produktivnosti DBA mogu uštedjeti tvrtkama do 33.000 USD po DBA godišnje.

Oracle je bolji izbor ako vaša tvrtka treba imati bržu procesorsku snagu podržanu jakim softverom. Oracle ima jedinstvene značajke kao što su paralelna obrada upita, usluge pravovremene korisničke podrške i mogućnosti rukovanja pogreškama 24×7. SQL Server je dobra opcija ako vam brzina i procesorska snaga nisu glavna briga. Čak i ne tehnički korisnici mogu ga naučiti brže od Oraclea i ima većinu značajki drugih tradicionalnih RDBMS-ova. U svakom slučaju, obje baze podataka su sredstva za vaše poslovanje i pomoći će u upravljanju vašim klijentima i zaposlenicima.

Kontrola transakcija još je jedna važna polazna točka između Oraclea i SQL Servera. U relacijskoj bazi podataka, transakcija je jedna jedinica rada koja može sadržavati više operacija ili zadataka. Kako ne bi došlo do odstupanja u podacima, baza podataka mora primijeniti ili vratiti sve operacije u transakciji. Jedna neuspjela operacija uzrokovat će

neuspjeh cijele transakcije. Na primjer, kada klijenti prenose novac između svojih tekućih i štednih računa, prijenos bi trebao biti jedna transakcija. Ova se transakcija sastoji od tri operacije:

1. Skidanje novca s tekućeg računa
2. Dodavanje novca na štedni račun
3. Evidentiranje prijenosa u bankovnoj knjizi

Ako bilo koja od ovih operacija ne uspije, baza podataka mora vratiti cijelu transakciju kako bi stanje na računu klijenta bilo ispravno. SQL Server izvršava operacije u transakciji sekvencijalno i pojedinačno. Zbog toga je teško ili čak nemoguće vratiti transakciju unatrag ako se pogreška dogodi na pola puta. Naredba *BEGIN TRANSACTION* deklarira početak transakcije, a naredba *COMMIT* deklarira kraj. Vraćanje nije moguće nakon najnovije izjave *COMMIT*. Međutim, u Oracleu je svaka nova veza s bazom podataka nova transakcija. Za većinu slučajeva upotrebe, baza podataka radi promjene u memoriji i predaje ih samo eksplicitnom naredbom *COMMIT*. To je fleksibilnije i omogućuje veću ručnu kontrolu pogrešaka.

Zaključak

Proces razvoja relacijske baze podataka za škole kao i usporedba različitih stavki u SQL Serveru i Oracleu pokazao se vrlo zahtjevnim, ali i zanimljivim. Kako se niti jedan posao ne može ni približno kvalitetno odraditi bez temeljite pripreme isti princip važi i za izradu baze podataka te usporedbu u dva različita sustava. Svakako, jedan o najzanimljivijih dijelova ovog projekta bio je onaj u kojem se uz pomoć DDL-a mogle izraditi dvije baze u ovom slučaju za SQL Server i Oracle te nismo morali posebno praviti dvije strane. Na početku je bilo potrebno dobro razmisliti i osmisliti što sve opisuje neku školu te s kojim pojmovima je ona usko vezana. Tako je pri usporedbi dvaju spomenuta sustava lakše bilo usporediti neke određene pojmove i doći do zaključka.

Uzevši u obzir sve navedeno dolazimo do zaključka da je područje informatike koje obuhvaća baze podataka jedno od obećavajućih područja jer se konstantno unapređuje i svoje primjene nalazi u svim sferama ljudskog života. Preporučujem za manje baze koristiti SQL Server, a za veću količinu podataka Oracle.

Literatura

[1] M. B. Tonči Carić, Uvod u relacijske baze, Zagreb: Creative Commons Imenovanje–Nekomercijalno–Bez prerada, 2015.

[2] R. Peterson, »Guru99,« 30 8 2021. [Mrežno]. Available: https://www.guru99.com/what-is-dbms.html.

[3] V. Vrbanić, »sigurnost.zemris.fer.hr,« 2 2009. [Mrežno]. Available: http://sigurnost.zemris.fer.hr/ostalo/2009_vrbanic/Sigurnost_podataka_u_Oracle_bazi.pdf.

[4] A. H. D. N. J. R.-J. A. S. D. R. B. W. Kellyn Gorman, Introducing Microsoft SQL Server 2019, Birmingham: Packt, 2019.

[5] dbaklevar, »DBAKevlar,« 20 3 2019. [Mrežno]. Available: https://dbakevlar.com/2019/03/oracle-vs-sql-server-architecture/.

[6] Oracle, »Oracle Help Centar,« 2021. [Mrežno]. Available: https://docs.oracle.com/cd/E10405_01/appdev.120/e10379/ss_oracle_compared.htm.

[7] »DB-engines,« 2022. [Mrežno]. Available: https://db-engines.com/en/ranking.

Pregled slika

Slika 1.1 Vrste DBMS-a 5
Slika 2.1 Konceptualni model 15
Slika 2.2 Izrada logičkog modela u Data Modeleru 16
Slika 2.3 Dodavanje veze između entiteta 17
Slika 2.4 Postavljanje svojstava dodane veze 18
Slika 2.5 Logički model 18
Slika 2.6 Relacijski model 19
Slika 2.7 Odabir atributa za sekvencu i okidač 20
Slika 2.8 Auto Increment 20
Slika 2.9 Postavljanje svojstava auto incrementa 21
Slika 2.10 DDL za Oracle 22
Slika 2.11 DDL za SQL Server 23
Slika 2.12 Kreiranje korisnika i davanje ovlasti korisniku 24
Slika 2.13 Prijava na novog korisnika Oracle 24
Slika 2.14 Izvršavanje DDL-a na konekciji korisnika kod Oracle 25
Slika 2.15 Kreiranje baze 26
Slika 2.16 Izvršavanje DDL-a i kreiranje tablica 27
Slika 2.17 Učitavanje gotove baze (.mdf datoteka) 28
Slika 2.18 Učitavanjem .mdf datoteke se učita i .ldf datoteka 28
Slika 2.19 Tipovi podataka kod SQL Servera 29
Slika 2.20 Tipovi podataka kod Oraclea 30
Slika 2.21 Generiranje sekvence i okidača kod SQL Servera nije potrebno 30
Slika 2.22 Generiranje sekvence i okidača kod Oracle-a 31

Slika 2.23 Kraj transakcije – SQL Server 31

Slika 2.24 Kraj transakcije – Oracle 32

Slika 2.25 Usporedba strukture pohrane 33

Skraćenice

RDBMS	*Relational Database Management System*	sustav za upravljanje relacijskom bazom podataka
IBM	*International Business Machines*	američka tvrtka
SUBP	*Sustav za upravljanje bazama podataka*	
IMS	*Integrated Management System Integrirani*	sustav upravljanja
DBMS	*Database Management System*	sustav upravljanja bazom podataka
ACID	*atomicity, consistency, isolation, durability*	atomarnost, konzistetnost, izolacija, durabilnost
OS	*Operating system*	operacijski sustav
SAP	*Systems Applications and Products*	sistemske aplikacije i proizvodi