

Razvoj aplikacija pomoću Oracle APEX alata

Buljan, Matea

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, University of Zagreb, Faculty of Humanities and Social Sciences / Sveučilište u Zagrebu, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:131:797470>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-04-17**



Repository / Repozitorij:

[ODRAZ - open repository of the University of Zagreb](#)
[Faculty of Humanities and Social Sciences](#)



SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE
ZNANOSTI
Ak. god. 2018./2019.

Matea Buljan

Razvoj aplikacija pomoću Oracle APEX alata

Završni rad

Mentor: dr. sc. Vedran Juričić

Zagreb, 2019.

Izjava o akademskoj čestitosti

Izjavljujem i svojim potpisom potvrđujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

(potpis)

Sadržaj

Sadržaj	iii
1. Uvod.....	4
2. Oracle APEX.....	5
2.1. Platforma.....	5
2.2. Mogućnosti.....	5
2.3. Arhitektura.....	6
2.3.1. REST arhitektura	6
2.3.2. SQL.....	11
2.4. Konferencije.....	15
2.5. Zadovoljni korisnici	16
3. Razvoj aplikacije pomoću Oracle Application Express alata	18
3.1. Arhitektura aplikacije	18
3.1.1. Alat za izgradnju aplikacija (engl. <i>App Builder</i>).....	18
3.1.2. SQL radionica (engl. <i>SQL Workshop</i>)	19
3.1.3. Razvoj tima (engl. <i>Team Development</i>).....	19
3.1.4. Zapakirane aplikacije (engl. <i>Packaged Apps</i>)	19
3.2. Izrada aplikacije	19
3.2.1. Shema baze podataka	19
3.2.2. Izrada aplikacije u Oracle APEX-u	20
4. O Oracle APEX-u nakon korištenja	27
4.1. Jednostavnost platforme	27
4.2. Brzina platforme	28
5. Zaključak.....	29
6. Literatura.....	30
Prilozi.....	33
Sažetak.....	36
Abstract	37

1. Uvod

U suvremenom društvu dolazi do stvaranja sve većeg broja informacija. Osim informacija postoje i podaci, ali njih međusobno treba razlikovati. Tkalcu u *Relacijskom modelu podataka* definira elementarnu informaciju kao opis jednog svojstva koje posjeduje određeni entitet. Ona ima materijalnu podlogu bilo izgovorenu, zapisanu na nekom mediju ili pohranjenu u glavi, materijalna podloga uvijek postoji. S druge strane, podatak definira kao skup znakova na memoriji koji prikazuje jedan ili više elemenata informacije.

Kako bi se što bolje snalazilo s podacima te imalo bolji uvid u njih, osmišljene su organizirane strukture koje dopuštaju njihovo pohranjivanje, odnosno baze podataka. „Baza podataka je kolekcija dosljednih podataka koji su korišteni aplikacijskim sustavima određenog poduzeća.“(C. J. Date, 2006). One imaju važnu ulogu u raznim područjima, široko su rasprostranjene i neupitno olakšavaju rad raznih firmi i društava. Neki od primjera njihove uporabe su prodajna mjesta, muzeji, arhivi, itd. Dobar primjer uporabe baze podataka su i knjižnice koje imaju svoj katalog. Katalog sadrži prikaz baze podataka s knjigama koje su dostupne za korištenje u knjižnici i za posuđivanje. Osim tih podataka, knjižnica također prilaže podatke o autoru i naslovu knjige, njenom žanru te još raznih podataka.

Za izradu baze podataka može se koristiti programski jezik ili odabrani alat. Postoje razni alati koji olakšavaju izradu, a neki od popularnijih su Microsoft Access, MySQL i Oracle Database. Oracle ima svoj alat imena Oracle Application Express (tj. Oracle APEX). U ovom radu fokus će biti na izradi aplikacije u tom alatu, a prije toga saznat će se nešto o samom Oracle APEX-u, njegovoj platformi, mogućnostima, arhitekturi, konferencijama i zadovoljnim korisnicima.

2. Oracle APEX

2.1. Platforma

Oracle je korporacija računalne tehnologije sa sjedištem u Americi. To je globalni opskrbljivač oblaka za računalne usluge tvrtke (Oracle, 2017). Prema podacima iz 2017. godine na Oracle brošuri, Oracleom se tada služilo 430 000 korisnika u 175 država. U sklopu Oraclea nastao je besplatni Oracle Application Express (tj. Oracle APEX). Oracle definira Oracle Application Express (APEX) kao nisko kodnu razvojnu platformu koja omogućuje izgradnju zadivljujućih, skalabilnih i sigurnih aplikacija s značajkama svjetske klase koje se mogu implementirati bilo gdje.

2.2. Mogućnosti

Oracle APEX samim tim što je nisko kodan (engl. *low code*), podrazumijeva da nije potrebno biti stručnjak u programskim jezicima da bi se napravila željena aplikacija. Razvijateljima aplikacije nudi se ono što im je potrebno za rješavanje poslovnih problema i to na što jednostavniji, praktičniji i učinkovitiji način. Mogućnosti koje Oracle APEX pruža svojim korisnicima su:

- pretvaranje podataka u informacije
- uređivanje podataka.
- stvaranje dijagrama
- ubacivanje podataka iz raznih izvora
- stvaranje obrazaca i automatska obrada upisanih podataka

Pretvaranjem podataka u informacije u ovom kontekstu smatra se uređivanje i primanje povratnih podataka na način koji najbolje zadovoljava potrebe korisnika. Podatke je lako pronaći zbog brze pretrage i lakog filtriranja. Također se mogu grupirati i raspoređivati po želji i prikazivati na nove načine, primjerice u obliku dijagrama. Uz to sve, svi filteri i grupiranja mogu se sačuvati do idućeg korištenja, i to u privatnom ili javnom obliku. Podaci se mogu praktično uređivati, nadopunjavati, kopirati i slično. Olakšano je i istovremeno uređivanje višestrukih podataka koji se nalaze u istim stupcima ili redovima. Oracle APEX omogućava vizualizaciju podataka dijagramima koje je lako uređivati, praktični su i brzi za napraviti te ih je moguće napraviti u raznim oblicima. Podržavaju ih svi moderni Internet pretraživači i

funkcioniraju bez obzira na platformu, veličinu ekrana ili karakteristike. Podatke koje će se koristiti u Oracle APEX aplikaciji moguće je ubaciti iz raznih izvora. Moguće ih je ubaciti iz lokalne baze podataka, Oracle Database ili bilo koje druge mrežne usluge. Osobe koje se koriste aplikacijom mogu u nju dodavati podatke ili ih uređivati putem obrazaca te će se iste automatski obraditi i sačuvati.

2.3. Arhitektura

U Oracle APEX-u zahtjevi su poslani putem pretraživača, putem mrežnog servera pa sve do baze podataka što omogućuje brzu dostupnost podataka i učinkovito izvođenje željenih radnji. U bazi podataka izvršeni su obrada, manipulacija podacima i poslovna logika. Cijeli proces izrade aplikacije koristi se trima komponentama koje zajedno čine Oracle RAD. To su Oracle REST Podatkovne Usluge (engl. Oracle REST *Data Services*, ORDS), Oracle APEX i Oracle Database. *Procedural Language* – PL i *Structured Query Language* – SQL su jezici na kojima se temelji direktna obrada podataka.

2.3.1. REST arhitektura

REST (*Representational State Transfer*) je arhitektura za postavljanje standarda između računalnih sistema na webu, olakšavajući sistemima da međusobno komuniciraju (Codecademy, 2019.). Web usluge koje koriste takvu arhitekturu zovu se RESTful Web usluge (engl. RESTful Web *services*, RWS), a nude suradnju između sistema na internetu. Web resursi su na World Wide Webu prvo bili opisani kao dokumenti ili datoteke identificirane svojim URL-ovima, međutim danas imaju više generičku i apstraktnu definiciju koja obuhvaća svaku stvar ili osobu koja se može identificirati, imenovati ili se može njome baratati na bilo koji način na webu. U RESTful web usluzi, zahtjevi prema izvornikovom URI-ju će izdati odgovor sa sadržajem (engl. *payload*) formatiranim u HTML-u, XML-u, JSON-u ili nekom drugom formatu. Odgovor sa sadržajem je dio prenesenih podataka koji ima namjeru poruke. Zaglavlje i metapodaci su slani samo da omogućuje izvršavanje odgovora sa sadržajem, to jest da podaci dođu na svoju destinaciju. Odgovor sa sadržajem može potvrditi da su se radile nekakve promjene u spremljenom resursu i odgovor može priložiti hipertekstualne linkove do ostalih povezanih resursa ili kolekcija resursa. Kada je HTTP korišten, a najčešći je, operacije koje su dostupne su GET, HEAD,

POST, PUT, PATCH, DELETE, CONNECT, OPTIONS i TRACE. Svi serveri općenite namjere moraju podržavati operacije GET i HEAD, dok su sve druge neobavezne.

REST arhitekturu je razvila grupa ljudi predvođena Roy Fieldingom paralelno s HTTP 1.1 od 1996. do 1999. bazirano na postojećem dizajnu HTTP-a 1.0 1996. Cilj im je bio stvoriti standard kojim će bilo koji server komunicirati s bilo kojim drugim serverom na svijetu (ReadMe, 2016.).

REST arhitektura ima svoja ograničenja, a to su : Klijent – Poslužitelj (engl. *Client - Server*), Neovisnost između zahtjeva (engl. *Stateless*), Priručna memorija (engl. *Cache*), Jedinstveno sučelje (engl. *Uniform interface*), Slojeviti sustav (engl. *Layered system*) i Kod na zahtjev (engl. *Code on demand*).

- Klijent i poslužitelj (*Client-Server*) razdvojeni su tako što svatko ima svoje zadatke. Prijenos podataka klijenta se poboljšava time što se pohranom podataka bavi poslužitelj, a ne klijent. Poslužitelji se ne brinu o korisničkom sučelju ili korisničkom stanju što ih čini jednostavnijima i skalabilnijima.
- Neovisnost između zahtjeva (*Stateless*) znači da primjerice svaki zahtjev od klijenta prema poslužitelju mora sadržavati sve informacije potrebne da bi se razumio zahtjev, bez korištenja prošlih zahtjeva ili konteksta za bolje razumijevanje. Time se potiče vidljivost, pouzdanost i skalabilnost.
- Priručna memorija (*Cache*) – to ograničenje zahtjeva da podaci unutar odgovora zahtjevu moraju biti implicitno ili eksplicitno označeni da se spremaju ili ne spremaju u priručnu memoriju. Ako se odgovori mogu spremati u priručnu memoriju, onda se isti mogu kasnije ponovno iskoristiti na zahtjev sličan tome. Time se poboljšava učinkovitost tako što se djelomično ili potpuno uklanja neke interakcije, što smanjuje prosječno vrijeme odaziva niza interakcija.
- Jedinstveno sučelje (*Uniform interface*) koje REST usluge imaju, razdvaja njihov stil arhitekture od ostalih mrežnih stilova. Ono pojednostavljuje i odvaja arhitekturu što omogućava da se svaki dio razvija samostalno i poboljšava vidljivost interakcija. Problem je u tome to jedinstveno sučelje smanjuje učinkovitost jer se podaci prenose u standardiziranom obliku, a ne u onom specifičnom za potrebe aplikacije. REST sučelje nije optimalno za druge oblike interakcije arhitekture za dobivanje jedinstvenog sučelja, stoga je potrebno

više ograničenja arhitekture kojima bi se vodile komponente. Definirana su četiri ograničenja sučelja: Identifikacija resursa, Manipulacija resursima kroz manipulaciju, Samoopisne poruke i Hipermedija kao motor aplikacija.

- Identifikacija resursa znači da su pojedini resursi identificirani u zahtjevima korištenjem URI-a u web baziranim REST sustavima. Resursi su konceptualno odvojeni od reprezentacija koje su vraćene klijentima. Na primjer, poslužitelj može slati podatke i svoje baze kao HTML, XML ili JSON.
 - Manipulacija resursima kroz manipulaciju znači da kada klijent drži reprezentaciju resursa uključujući metapodatke u prilogu, onda ima dovoljno informacija da izmjeni ili obriše resurse.
 - Samoopisne poruke znače da svaka poruka sadrži dovoljno informacija koje opisuju kako je obraditi.
 - Hipermedija kao motor aplikacija označava to da klijenti obavljaju stanje prijelaza samo kroz akcije koje su dinamički identificirane unutar hipermedije od strane poslužitelja. Osim jednostavnih fiksni ulaznih točaka aplikacije, klijent ne pretpostavlja da je bilo koja određena radnja dostupna za bilo koji resurs osim onih opisanih u prikazu koji je prethodno primljen od poslužitelja. Ne postoji univerzalno prihvaćeni format za predstavljanje veze između dva resursa.
- Slojeviti sustav (engl. *Layered system*) omogućava arhitekturi da bude sastavljena od hijerarhijskih slojeva tako što ograničava ponašanje komponente na način da svaka komponenta ne može vidjeti dalje od trenutnog sloja s kojim je u interakciji. Ograničavanjem poznavanja sustava na jedan sloj povezuje se ukupna složenost sustava i promovira se neovisnost podloge. Primarni nedostatak takvih sustava je što povećavaju vrijeme obrade podataka što smanjuje performansu.
 - Kod na zahtjev (engl. *Code on demand*) – REST omogućava proširenje klijentove funkcionalnosti preuzimanjem i izvršavanjem koda u obliku skripte ili appleta (Java applet, JavaScript). Tako se smanjuje broj značajki koje se mogu provesti. Dopuštanjem da se značajke preuzmu nakon implementacije dopušta proširenje sustava, ali i smanjuje vidljivost, no to je samo dodatno ograničenje unutar REST-a. Opcionalno ograničenje omogućava da se

dizajnira arhitektura koja podržava željeno ponašanje u općem slučaju, ali s razumijevanjem da se ono može onemogućiti u nekim kontekstima.

Prema Wikipedii, svojstva REST arhitekture su performansa, skalabilnost, jednostavnost, modifikacija, vidljivost, prenosivost i pouzdanost.

- Performansa – interakcije komponente mogu biti dominantan faktor u korisničkom doživljaju izvedbe i mrežnoj učinkovitosti. Način interakcije komponentata utječe na performansu.
- Skalabilnost – služi za potporu velikoj količini komponentata i interakciji između komponentata.
- Jednostavnost – u interakciji sučelja.
- Modifikacija – komponentata kako bi se zadovoljile potrebe za promjenom čak i u vrijeme rada programa.
- Vidljivost – prozirnost komunikacije između komponentata od strane predstavnika.
- Prenosivost – prenošenje komponentata pomicanjem programskog koda s podacima.
- Pouzdanost – otpornost na neuspjeh na sistemskoj razini prilikom nazočnosti kvarova unutar komponentata, konektora i podataka.

Interakcije koje su bazirane na REST-u događaju se koristeći konstrukcije koje su poznate svima tko je već koristio HTTP i to je prednost REST-a. Primjer toga je to što sve interakcije bazirane na REST-u komuniciraju svoj status koristeći HTTP statusne kodove. Primjerice, 404 znači da traženi resurs nije pronađen, 401 znači da zahtjev nije autoriziran, 200 znači da je sve u redu i 500 znači da se pojavila nepopravljiva greška aplikacije na serveru.

Detalji kao što su enkripcija i cijelost prijenos podataka ne rješavaju se dodavanjem novih tehnologija i okvira, nego oslanjajući se na *Secure Sockets Layer* (SSL) enkripciju i *Transport Layer Security* (TLS), Dakle, cijela REST arhitektura je osnovana na konceptima s kojima je većina razvijatelja već upućena.

REST je također jezično-ovisna arhitektura. Aplikacije osnovane na REST-u mogu biti pisane bilo kojim jezikom, bila to JAVA, Kotlin, .NET, AngularJS ili JavaScript. Dokle god taj jezik može raditi web-osnovane zahtjeve koristeći HTTP, moguće je da

se taj jezik koristi za dozivanje RESTful API ili web usluge. Također jer se web usluge mogu pisati bilo kojim jezikom, razvijateljii sa zadatkom implementiranja takvih usluga mogu birati tehnologije koje najbolje funkcioniraju za njihovu situaciju.

Prednost REST-a je i svakodnevnost. Na drugoj strani servera, tu je raznolikost okvira kojima je REST osnova za pomaganje razvijateljima kod kreiranja RESTful web usluge, uključujući RESTlet i Apache CXF. Sa strane klijenta, svi novi JavaScript okviri, kao što su JQuery, Node.js, Angular i EmberJS, imaju standardne knjižnice ugrađene u svoj API koje čine prizivanje RESTful web usluga i korištenje XML ili JSON bazirane podatke koje vrate, srazmjerno izravan poduhvat.

HTTP je REST-ova prednost, ali i mana. Primjerice, HTTP ne skladišti informacije između zahtjeva i odgovora koje su bazirane na stanju, što znači da se REST aplikacije moraju biti neovisne o zahtjevima i svi zadaci upravljanja stanjem se moraju izvršiti klijentom. Uz to, HTTP nema mehanizam slanja notifikacija od servera do klijentu i to otežava implementaciju bilo kakve usluge gdje server ažurira klijent bez korištenja klijentovog upita serveru za ažuriranja za određenih vremenskih intervala (*client pull*) (Bharathvaj Ganesan (Codeburst), 2018.) ili sličnog.

Također problem je u samoj definiciji „osnovanog na REST-u“. Neki razvijajući softvera netočno vjeruju da je sve što nije osnovano na SOAP-u REST. REST nema odlučan standard kojim se može potvrditi da je određeni dizajn REST (SearchAppArchitecture, 2019.).

Alternative REST-a su SOAP (engl. *Simple Object Access Protocol*), JSON (JavaScript Object Notation) i gRPC (*gRPC Remote Procedure Calls*). Razlika između REST-a i SOAP-a je ta što REST određuje položaj resursa na serveru, i osoba bira želi li ga promijeniti, obrisati ili izvući neke informacije o tome. Sa SOAP-om klijent neće direktno komunicirati s resursom, nego će pozvati zahtjev, a taj zahtjev će osloboditi pristup raznim objektima i resursima iza scene. SOAP također ima ugrađen veliki broj okvira i API-jeva uz HTTP, uključujući *Web Services Description Language* (WSDL) koji određuje strukturu podataka koji prolaze od klijenta do servera i obrnuto.

2.3.2. SQL

SQL (*Structured Query Language*) je strukturni upitni jezik, programski jezik visoke razine. Najpopularniji je računalni jezik za izradu, traženje, ažuriranje i brisanje podataka iz relacijskih baza podataka. Posebno je koristan u rukovanju strukturiranim podacima između čijih se varijabli nalaze veze.

SQL je primarno osnovan na relacijskoj algebri i računu orijentiranom na n-torke, stoga on sadrži velik broj izjava koje se neformalno mogu klasificirati kao podjezici, a to su: *Data Query Language* (DQL), *Data Definition Language* (DDL), *Data Control Language* (DCL), *Data Manipulation Language* (DML).

- *Data Query Language* (DQL) sastoji se od naredbi za pretraživanje i dohvaćanje podataka u bazi : npr. SELECT

Od velike pomoći tu je naredba JOIN koja spaja podatke više tablica, a da bi mogao funkcionirati, tablice moraju imati zajedničku kolonu. Oblici JOIN naredbi su:

- *Cartesian* JOIN koji spaja sve retke iz prve tablice sa svim recima iz druge tablice, a kao rezultat se dobije kombinacija svih zapisa u obje tablice
- *Natural/Equi* JOIN spaja zapise iz prve tablice s zapisima iz druge tablice ako je zadovoljen uvjet WHERE
- *Right Outer* JOIN spaja zapise iz prve tablice s drugom tablicom na temelju vrijednosti zadanih u JOIN uvjetu. Također i uključuje zapise iz druge tablice koji nemaju odgovarajući par u prvoj tablici što rezultira time da te kolone iz prve tablice budu NULL
- *Left Outer* JOIN funkcionira isto kao i *Right Outer* JOIN, ali obrnuto. Dakle, uključuje zapise iz prve tablice koje nemaju par u drugoj, i te kolone budu NULL
- *Full Outer* JOIN je kombinacija *Left Outer* JOIN-a i *Right Outer* JOIN-a. Prvo se izvlače svi zapisi iz prve tablice s odgovarajućim parom iz druge tablice, a onda isto tako obrnuto, preostali zapisi iz druge tablice koji nemaju par u prvoj tablici.

- *Data Definition Language* (DDL) sastoji se od naredbi za definiranje objekata baze podataka: npr. CREATE, DROP i ALTER

Naredbom CREATE kreiraju se objekti baze podataka – tablice, pogledi, indeksi. Najčešće se koristi pri opisivanju same strukture određenog objekta.

Naredbom DROP se brišu objekti baze podataka zajedno sa svim podacima koje je relacija sadržavala. Ta naredba ne smije se miješati s naredbom DELETE koja se koristi za brisanje podataka iz objekta, a nema samog objekta baze.

Naredbom ALTER mijenja se struktura tablice, odnosno modificira se ili nadopunjava već postojeća struktura.

- *Data Control Language* (DCL) sastoji se od naredbi za kontrolu podataka i njihovu sigurnost: npr. GRANT i REVOKE.

GRANT naredbom se ovlasti daju, a REVOKE-om se iste uklanjaju.

- *Data Manipulation Language* (DML) sastoji se od naredbi za manipuliranje podacima u bazi: npr. INSERT, DELETE, UPDATE.

INSERT naredbom se dodaju podaci u tablicu, DELETE naredba briše zapis iz baze, a UPDATE naredba se koristi za ažuriranje vrijednosti/podatka.

Iako je SQL često opisan kao deklarativni jezik, također uključuje i proceduralne elemente. SQL sadrži podatkovni upit, manipulaciju podacima (unos, izmjenu i brisanje), definiranje podataka (kreiranje sheme i modificiranje) i kontrolu pristupa podacima.

Dizajn SQL-a nije isti kao u svojoj teoretskoj osnovi, te su u njemu tablice i rezultati upita lista redova. Isti red može se pojaviti nekoliko puta.

U SQL-u tri temeljna tipa podataka su brojevi (engl. *numeric*), znakovni nizovi (engl. *character*), datum/vrijeme (engl. *date/time*). Brojevi mogu biti cjelobrojni (engl. *integer*) i decimalni (engl. *float*). Znakovni nizovi mogu biti fiksne duljine (engl. *character*) ili varijabilne duljine (engl. *varchar*). Datum i vrijeme mogu biti definirani pojedinačno ili kao integrali. Osim navedenih postoji i BLOB (*Binary Large Object*) odnosno RAW za binarne tipove podataka, CLOB (*Character Large Object*) i DBCLOB (*Double Byte Character Large Object*). BLOB služi za spremanje slika, zvuka i sličnih podataka u bazu maksimalne veličine do 2 GB. CLOB-om se u bazu

spremaju veći dokumenti do 2 GB, a DBCLOB se koristi za spremanje dokumenata napisanih korištenjem grafičkih stringova, isto do 2 GB veličine.

SQL sadrži nekoliko jezičnih elemenata uključujući: klauzule, izraze, predikate, upite i izjave.

- Klauzule (engl. *Clauses*) sastavni su dio izjava i upita (u nekim slučajevima su opcionalne).
- Izrazi (engl. *Expressions*) mogu proizvesti skalarne vrijednosti ili tablice koje se sastoje od stupaca i redova s podacima.
- Predikati (engl. *Predicates*) se koriste za ograničavanje efekata izjava i upita, ili za promjene smjera programa.
- Upiti (engl. *Queries*) vraćaju podatke na osnovu određenih kriterija i najvažniji su dio SQL-a.
- Izjave (engl. *Statements*) mogu kontrolirati transakcije, tok programa, veza i dijagnostiku.

Najvažniji dio SQL-a su upiti i vrše se naredbom SELECT koji zatim preuzima podatke iz jedne ili više tablica ili izraza. Naredba SELECT nema utjecaj na bazu podatka. Određene nestandardne implementacije naredbe SELECT mogu imati direktni utjecaj, kao što je SELECT INTO sintaksa koja postoji u nekim bazama podataka.

Upiti omogućavaju opisivanje željenih podataka, ostavljajući sistemima za upravljanje bazama podataka odgovornost za planiranje, optimizaciju, kao i za obavljanje operacija potrebnih da se prikažu rezultati zahtjeva. Upiti sadržavaju listu stupaca koje će biti uključene u konačan rezultat odmah nakon SELECT-a. SELECT je najsloženija izjava u SQL-u s opcionalnim ključnim riječima i klauzulama koje obuhvaćaju:

- FROM ukazuje na tablicu iz koje se preuzimaju podaci. Ona opcionalno može sadržavati i JOIN klauzulu koja određuje pravila za spajanje više tablica. (sh wiki)
- WHERE specificira koje redove je potrebno dohvatiti.
- GROUP BY grupira redove koji dijele svojstvo tako da agregatna funkcija može biti primijenjena svakoj grupi.

- HAVING bira između grupa koje su definirane GROUP BY klauzulom.
- ORDER BY specificira redoslijed kojim se vraćaju redovi.
- AS se može koristiti za privremeno preimenovanje tablica ili stupaca.

Bitan faktor u SQL-u su i operatori, a oni izvršavaju naredbe na zasebnim podacima i vraćaju rezultat. Podatkovni objekti se zovu operandi ili argumenti. Operatori su predstavljeni plusom (+), a operator koji provjerava NULL-ove su predstavljeni ključnim riječima IS NULL ili IS NOT NULL. Neke vrste SQL operatora su:

- Aritmetički operatori provode aritmetičke operacije ako su podaci brojevi. Ti operatori su zbrajanje(+), oduzimanje(-), množenje(*) i dijeljenje(/).
- Usporedni operatori je matematički simbol koji se koristi za usporedbu dviju vrijednosti. Rezultat usporedbe može biti TRUE(točno), FALSE(netočno) ili UNKNOWN(nepoznato).
- Logički operatori su oni koji su točni ili netočni. Oni vraćaju točne ili netočne vrijednosti kako bi kombinirali jednu ili više točnih ili netočnih vrijednosti. Logični operatori su AND(i), OR(ili), NOT(nije), IN(u), BETWEEN(između), ANY(bilo koji), ALL(svi), SOME(neki), EXISTS(postoji) i LIKE(kao). (geek.hr, 2003.)

IBM-ovi Donald D. Chamberlin i Raymond F. Boyce su razvili SQL nakon što su naučili o relacijskom modelu Tedd Codda ranih 1970.-ih. SQL se inicijalno zvao SEQUEL (*Structured English Query Language*) i bio je dizajniran za manipuliranje i dohvaćanje podataka spremljenih u IBM-ovoj bazi podataka. Prvotno su radili na Squareu, ali je bilo teško za koristiti i potom su započeli rad na SEQUEL-u. SEQUEL je kasnije postao SQL zbog toga što je SEQUEL bio zaštitni znak druge tvrtke. IBM je potom nastavio razvijati SQL, nakon što su se uvjerali u njegovu korisnost i praktičnost, a kasnije je razvijanje SQL-a započeo i sam Oracle.

U Oracleu se uz SQL koristi i PL (*Procedural Language*), a to je Oracleov proceduralni dodatak SQL-u. PL/SQL uključuje elemente proceduralnog jezika kao što su stanja i petlje, dopušta deklaraciju konstanti i varijabli, procedura i funkcija, vrsta i varijabli tih vrsta i okidača i može kontrolirati iznimke tj. greške. Tu su i nizovi podataka (engl. *array*), a to su objekti od kojih su svi iste veličine i vrste. Svaki taj

objekt zove se *array element*. Primjerice, može se imati *array integers*, *array znakova* ili bilo čega što ima definiranu vrstu podataka.

Može se stvoriti PL/SQL jedinice kao što su procedure, funkcije, paketi, vrste, okidači, koji su spremjeni u bazu podataka za ponovnu upotrebu sa strane aplikacija koje koriste bilo koji od Oracleovih programskih sučelja bazi podataka.

2.4. Konferencije

Konferencije su bitne za nove ideje, diskusiju i promoviranje proizvoda. 2018. godine Oracle APEX bio je sastavni dio sedam konferencija, a to su: ODTUG Kscope18, OUG Scotland 2018, SIOUG 2018, Oracle OpenWorld 2018, DOAG Conference & Exhibition, UKOUG Tech18 i HROUG 2018. ODTUG Kscope18 i Oracle OpenWorld 2018 konferencije su u Sjedinjenim Američkim Državama. Ostalih pet konferencija održava se u Europi, a jedna od tih je godišnja hrvatska konferencija HROUG koja se održava u Rovinju 16. – 19. listopada 2018. godine.

HROUG je Hrvatska udruga Oracle korisnika osnovana 1995. godine. Cilj udruge je razmjena znanja i iskustava o primjeni ORACLE tehnologije, unapređenje međusobnih odnosa i informiranja krajnjih korisnika kao i odnosa partnera koji koriste ORACLE tehnologiju, unapređenje odnosa korisnika s ORACLE korporacijom, te unapređenje organiziranosti informatičke struke i odnosa sudionika informatičke djelatnosti u zemlji i inozemstvu. Udruga nastoji povezati i unaprijediti rad fizičkih i pravnih osoba, tj. svih onih koji na području informatičke djelatnosti nalaze svoj profesionalni interes ili ostvaruju svoje ambicije i zadatke (HROUG, 2014.). HROUG naime više nije jedina APEX konferencija koja se održava na hrvatskim prostorima. Novonastala konferencija APEX-a, prvi put održana 2018. godine, zove se APEX Alpe Adria. APEX Alpe Adria je projekt između ljubitelja APEX-a iz triju zemalja: Slovenija, Hrvatska, Austrija. Zajedno smo vidjeli potrebu za APEX događajem u toj regiji. Organizatori te konferencije su Aljaž Mali (Slovenac), Dario Bilić (Hrvat) i Peter Raganitsch (Austrijanac). Prvu APEX Alpe Adria konferenciju posjetilo je 190 osoba te je bila održana u Austriji, dok je druga konferencija održana u Zagrebu.

2.5. Zadovoljni korisnici

Oracle APEX široko je rasprostranjen zbog njegovog uspjeha u rješavanju poslovnih problema što čini njegove korisnike zadovoljnima. Oracle je podijelio nekoliko priča zadovoljnih korisnika širom svijeta.

Dirk Böning-Cortier, vlasnik njemačke tvrtke Small Solutions izjavio je: „Small Solutions koristi agilni i korisnički usmjeren pristup u kombinaciji s tehnološkom platformom Oracle APEX kako bi pružili brza i ekonomična rješenja. Danas služimo gotovo sva odjeljenja Deutsche Bahn-a i možemo se osloniti na iskustvo preko 400 projekata s visokim zadovoljstvom kupaca.“

U Ujedinjenim Arapskim Emiratima, IT Manager Salim Hlayel Oracle APEX pohvalio je ovim riječima: „U Ankabut-u smo otkrili da Oracle Application Express igra izvrsnu ulogu u izgradnji poslovnih aplikacija. Danas imamo više od 16 aplikacija u produkciji, napravljenih na Oracle APEX tehnologiji. Koristi se u našim back-office sustavima za upravljanje prisutnošću svih naših članova, dnevnik aplikacijom, upravljanje projektima i puno obrazovnih i operacijskih aplikacija integrirano s informacijama o studentima i ERP sistemima.“

„Prebacujući se na Oracle Database Cloud Service, Goodus može brzo stvarati i modificirati poslovne aplikacije u stvarnom vremenu pomoću Oracle Application Express (Oracle APEX). To je također smanjilo cijenu održavanja, koristeći pretplatu za pristup raznim uslugama unutar Oracleovog oblak portfelja, uključujući Oracle Database Cloud Service, Oracle Database Backup Cloud Service, Oracle Storage Cloud Service i Oracle Compute Cloud Service. Odluka premještanja na Oracleove oblak usluge omogućila je razvojnim programerima da brzo razviju aplikacije bez nanošenja tereta troškova na hardver i osposobila je organizaciju da brže odgovara na razne poslovne zahtjeve kupaca.“ (Goodus, 2016.).

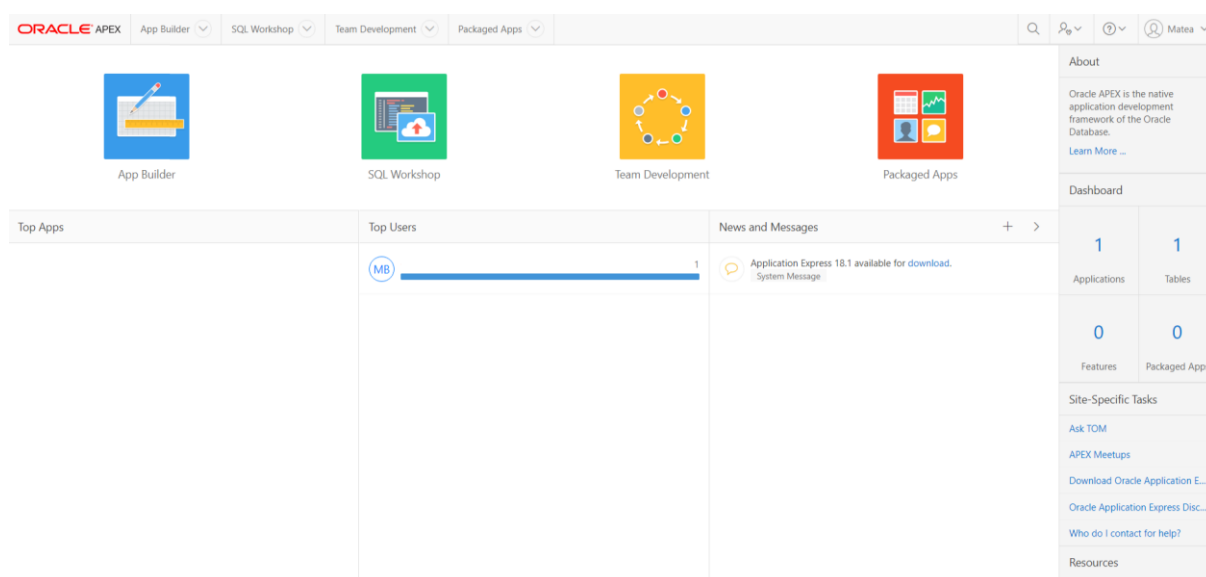
AIS Automation je tvrtka koja nudi automatizaciju korištenja strojeva, automatizaciju tvornica i slično, koja zahvaća prostore u Europi, Bliski Istok i Afriku, te izjavljuju: „Nakon mnogo godina dobrog iskustva s Oracle tehnologijom koju smo koristili na našim prostorima, Oracle je također bio prvi izbor za naš novi proizvod oblak. Oracle Application Express se dokazao idealnim razvojnim okvirom. Oracle Cloud donosi tehnologiju visokih performansi s najvećom mogućom fleksibilnošću. Također, naši

podaci---i naših korisnika---su pogodno zaštićeni. Sigurno ćemo birati Oracle za naš budući rast.“ (AIS Automation, 2018).

3. Razvoj aplikacije pomoću Oracle Application Express alata

3.1. Arhitektura aplikacije

Oracle APEX koristi se u internetskom pretraživaču i za njegovo korištenje treba napraviti račun. Nakon uspješnog otvaranja računa, može se otvoriti naslovna stranica (slika 1).



Slika 1. Naslovna stranica Oracle APEX-a (Izvor: Oracle APEX)

Na slici 1 se primjećuju četiri kvadrata raznih boja. Prvi plavi kvadrat je alat za izgradnju aplikacija (engl. *App Builder*). Iduća je SQL radionica (engl. *SQL Workshop*), zatim razvoj tima (engl. *Team Development*) i zapakirane aplikacije (engl. *Packaged Apps*).

3.1.1. Alat za izgradnju aplikacija (engl. *App Builder*)

U alatu za izgradnju aplikacija može se kreirati i urediti aplikacija, unijeti datoteke i vidjeti podatke o napravljenoj aplikaciji. Može se napraviti nova aplikacija, aplikacija iz datoteke s tabličnim prikazom podataka ili iskoristiti jednu od ranije spomenutih zapakiranih aplikacija. O tome kakve zapakiranih pakirane aplikacije postoje, saznat će se kasnije u radu.

3.1.2. SQL radionica (engl. *SQL Workshop*)

SQL radionica je, kako i sam naziv govori, mjesto gdje se izrađuje baza podataka i s kojom se komunicira pomoću SQL-a. Mogu se izraditi vlastite SQL naredbe koje se koriste kasnije u aplikaciji.

3.1.3. Razvoj tima (engl. *Team Development*)

Prostor u Oracle APEX-u posvećen razvoju tima služi za dodavanje i pregled ciljeva i radnji koje treba obaviti. Vrlo koristan je dodatak pregled kvarova gdje se mogu vidjeti trenutni kvarovi, zatvoreni kvarovi, njihova izvješća i podatke o osobi koja je radila na njihovom popravku. Razvoj tima također sadrži i pregled povratnih informacija korisnika koje se mogu filtrirati i grupirati.

3.1.4. Zapakirane aplikacije (engl. *Packaged Apps*)

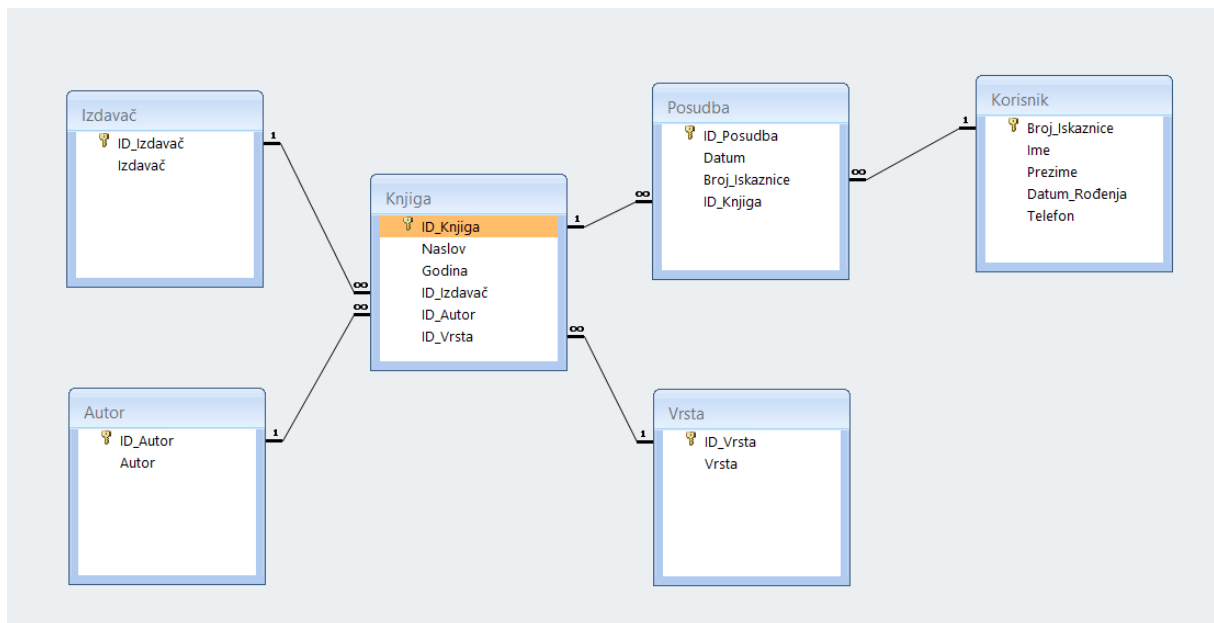
U zapakiranim aplikacijama mogu se pronaći dodatne aplikacije za korištenje u aplikaciji izrađenoj u Oracle APEX-u. Kod svake aplikacije je navedeno za što otprilike služi, kako bi se u moru ponuđenih aplikacija lakše snalazilo. Tako primjerice postoji aplikacija ankete uživo (engl. *Live Poll*) kod koje je navedeno da služi za marketing i praćenje. Klikom na aplikaciju, dolazi se na stranicu koja nudi instaliranje iste te njen pobliži opis i funkciju. Još neke od ponuđenih aplikacija su pratitelj greški (engl. *Bug Tracker*) i tablice uzoraka (engl. *Sample Charts*).

3.2. Izrada aplikacije

U ovom radu, pravit će se aplikacija „Knjižnica“. Ideja je da u aplikaciji budu stranice s popisima knjiga, učlanjenih korisnika i posuđenih knjiga. Osim toga bilo bi poželjno da se mogu unijeti nove knjige, korisnici ili posudbe. Na početku izrade potrebno je napraviti shemu baze podataka.

3.2.1. Shema baze podataka

Shema baze podataka sadrži imena tablica, njihove attribute i odnose među tablicama. Korisna je za lakše snalaženje i izradu tablica. Oracle APEX nema funkciju izrade sheme, stoga je u ovom radu korišten Microsoft Access alat. U tom alatu također su određene vrste veza među tablicama, primarni ključevi tablica i slično.



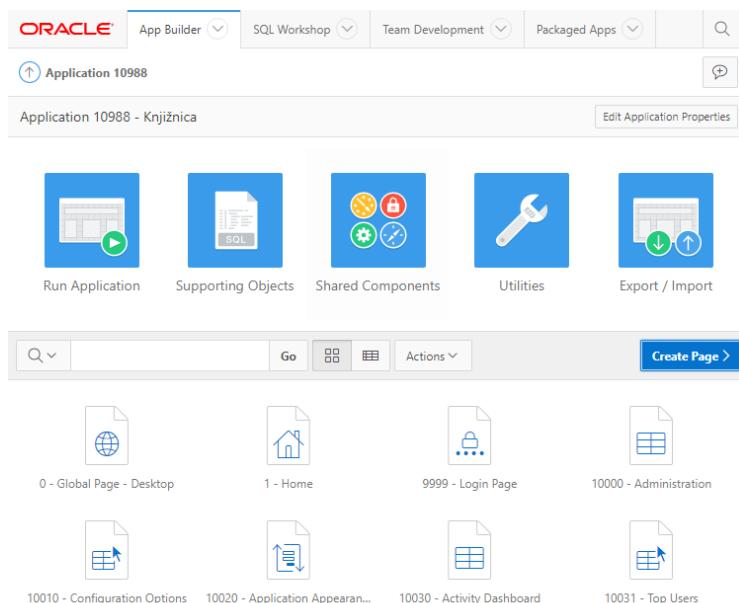
Slika 2. Shema baze podataka (Izrađeno u Microsoft Accessu)

Prvo su izrađene vanjske tablice *Autor*, *Izdavač*, *Vrsta*, i *Korisnik*. Zatim *Posudba* i naposljetku *Knjiga*. Svaka tablica ima primarni ključ *ID_“ime_tablice“*, te su među njima uspostavljene veze primarnih i stranih ključeva.

3.2.2. Izrada aplikacije u Oracle APEX-u

Nakon izrade sheme baze podataka, može se početi s izradom same aplikacije u Oracle APEX-u. Za to koristimo ranije spomenuti alat za izgradnju aplikacija (engl. *App Builder*). Klikom miša na taj alat dolazi se na stranicu s opcijama za izgradnju nove aplikacije, aplikacije iz tabličnog prikaza ili putem zapakirane aplikacije. Za ovaj rad, radit će se nova aplikacija.

Tijekom kreiranja aplikacije, potrebno je istoj dati ime. Aplikacija će se u ovom radu zvati *Knjižnica*. Moguće je i mijenjanje izgleda aplikacije, dodavanje stranica i Oracle APEX-ovih svojstava aplikaciji, primjerice dozvoljavanje korisnicima da daju povratne informacije, ili izvješća o aktivnostima korisnika ili o pogreškama. Naposljetku je prikazan ID broj aplikacije kojega joj dodjeljuje Oracle APEX, jezik, shema pod kojom je rađena aplikacija, tema i slično. Kreiranjem dolazi se na stranicu na kojoj se može pokrenuti aplikacija, izvoziti i uvoziti datoteke, dodavati stranice te uređivati aplikaciju. Tu su prikazane sve do sada napravljene stranice aplikacije.



Slika 3. Alat za izgradnju aplikacije (Izvor: Oracle APEX)

Nakon kreiranja aplikacije, potrebno je nekoliko tablica kako bi se dalje nastavilo s radom. Tablice se izrađuju u SQL radionici (engl. *SQL Workshop*) na vrhu stranice. Dolaskom na SQL radionicu, potrebno je otići na preglednik objekata (engl. *Object Browser*) gdje se nalaze sve napravljene tablice, pregledi (engl. *view*), indeksi i slično. Izrađivanje aplikacija funkcionirat će isto kao i shema baze podataka u Microsoft Accessu. Krenut će se od vanjskih tablica prema unutarnjima. Tablici daje ime, te se uređuju tablični stupci.

U tablici *Autor* postojat će dva atributa, *ID_AUTOR* i *AUTOR*. *ID_AUTOR* primarni je ključ ove tablice, a u koloni *AUTOR*, pisat će ime i prezime autora. Vrsta podataka *ID_AUTOR* je *NUMBER*, a *AUTOR* je *VARCHAR2*. Isto tako po shemi će se napraviti tablice *Izdavač*, *Vrsta* i *Korisnik*.

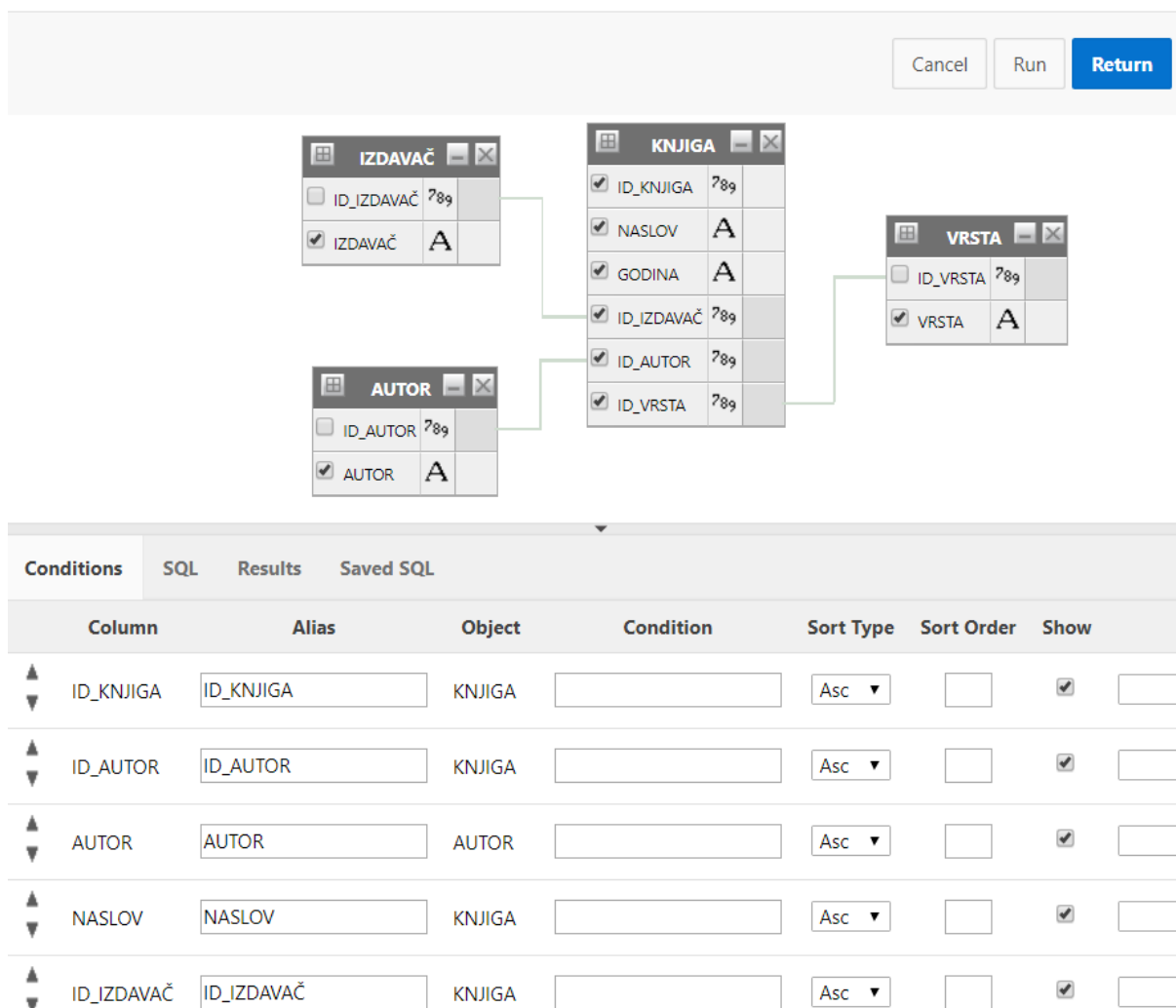
U tablici *Korisnik* primarni ključ je broj članske iskaznice, a drugi atributi su *IME*, *PREZIME*, *DATUM_ROĐENJA* i *TELEFON*, od kojih telefon može biti *NULL*. U cijeloj aplikaciji, to će biti jedino mjesto kojem je dopušteno da bude *NULL*. U SQL-u vrijednost koja nedostaje se označuje s *NULL* koji služi kao držač mjesta (engl. *placeholder*).

Još nenapravljene tablice su *Knjiga* i *Posudba*, te će se tim redoslijedom i napraviti iz razloga što je tablici *Posudba* potreban strani ključ *ID_KNJIGA*, koji se nalazi u tablici *Knjiga*.

Tijekom izrade tablice *Knjiga*, potrebno je odrediti strani ključ koji će se u toj tablici nalaziti. Strani ključ uspostavlja vezu između stupca ili stupaca u jednoj tablici i primarnog ili jedinstvenog ključa u drugoj tablici. U tablici *Knjiga* nalaze se tri strana ključa, ali je tijekom izrade moguće dodati samo jedan. U SQL radionici se kasnije može dodati još stranih ključeva. U tablicu *Posudba* dodan je strani ključ *ID_KNJIGA* tijekom izrade tablice.

Iznad tablica mnogo je različitih kartica koje otvaranjem daju određene podatke o tablici ili daju mogućnost uređivanja iste. Dodavanje ostatka ključeva može se napraviti klikom na karticu *Constraints* koja se nalazi među tim karticama. Dodavanje ključeva izgleda isto kao i na izradi tablice. Osim toga i kartice, koja prikazuje samu tablicu, tu su i podaci, indeks, model, statistike i tako dalje. Tablicu je moguće uređivati pomoću alata ispod tih kartica. Može se dodavati stupac, uređivati stupac, promijeniti njegovo ime, izbaciti stupac, promijeniti ime tablice, kopirati ju ili ju izbaciti.

Nakon tablica, potrebno je napraviti pregled. Pregled (engl. *view*) spaja podatke iz nekoliko tablica u jednu, a to će biti potrebno za aplikaciju rada. Trenutno je napravljeno šest tablica do kojih će podaci iz četiri od njih biti korištene u pregledu. Izrađuje ga se tamo gdje i tablice. Potrebno je dati mu naziv i zatim ga se može napraviti putem upita (engl. *query*) ili alata u kojemu se sastavlja upit (engl. *Query Builder*).



Slika 4. Izrada pregled *Knjige* u alatu u kojemu se sastavlja upit (Izvor: Oracle APEX)

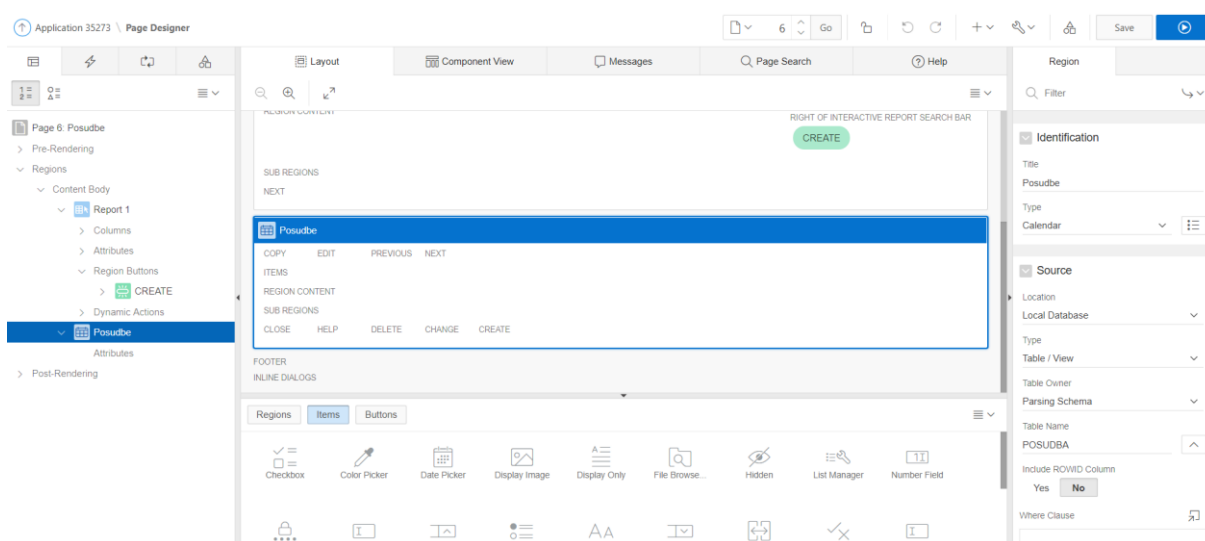
U alatu u kojemu se sastavlja upit označe se tablice koje želimo prikazati u pregled koji je nazvan *Knjige*. Tablice se prikazu i kvačicama se označe potrebni stupci. Za *Knjige* potrebni su *ID_KNJIGA*, *NASLOV*, *GODINA*, *ID_IZDAVAČ*, *IZDAVAČ*, *ID_AUTOR*, *AUTOR*, *GODINA*, *ID_VRSTA* i *VRSTA*. Iako svi ti stupci nisu potrebni za prikaz u izvješću, potrebni su za popunjavanje obrazaca za unos novih knjiga. Idući pregled koji je potrebno napraviti je *Posudba*. Njega se radi po istom principu, označe se potrebni stupci i povežu se strani ključevi s primarnima.

Nakon kreiranja tablica i pregleda, vrijeme je za izradu stranica aplikacije. Na stranici za uređivanje aplikacije, iznad popisa svih stranica koje aplikacija sadrži, nalazi se gumb *Napravi stranicu* (engl. *Create a Page*). Pritiskom na taj gumb otvara se dijaloški okvir na kojemu se može odabrati vrsta stranice koju želimo napraviti. Neke od mogućnosti su prazna stranica (engl. *blank page*), izvješće (engl. *report*), obrazac

(engl. *form*), grafikon (engl. *chart*) i kalendar (engl. *calendar*). Pritiskom na izvješće nude se razni oblici izvješća, a za ovaj rad je potrebna kombinacija izvješće-obrazac.

Izvješću je dodano ime *Knjige* i nalazi se na stranici 2, a obrascu *Unos nove knjige* i nalazi se na stranici 3. Za obrazac stavljeno je da bude dijaloški okvir. Zatim se može odrediti gdje će se izvješće i obrazac nalaziti u izborniku. Mogu biti nepovezani s izbornikom, povezati se s postojećom navigacijom ili napraviti novu te se povezati s njom. Nakon toga se dodaje tablica ili pregled koji želimo koristiti te njihovi stupci za koje želimo da budu u izvješću i oni koje želimo da se popunjavaju u formularu. Koristit će se pregled *Knjige*. U istoimenom izvješću *Knjige* vidljivi stupci su *ID_KNJIGA*, *NASLOV*, *AUTOR*, *IZDAVAČ*, *GODINA* i *VRSTA*, a u obrascu *Unos nove knjige* vidljivi su *ID_KNJIGA*, *NASLOV*, *ID_AUTOR*, *ID_IZDAVAČ*, *GODINA*, *ID_VRSTA*.

Iduća stranica koja se dodaje su *Posudbe*. Dodaje se istim koracima kao i stranica *Knjige*. Koristi se istoimeni pregled *Posudbe* i pravi se izvješće i obrazac. Na izvješću bit će vidljivi *ID_POSUDBA*, *ID_KNJIGA*, *NASLOV*, *BROJ_ISKAZNICE*, *IME*, *PREZIME* i *DATUM_POSUDBE*. U obrascu će se trebati popuniti *ID_POSUDBA*, *ID_KNJIGA*, *BROJ_ISKAZNICE*, *DATUM_POSUDBE*. Osim toga, na stranici *Posudbe* dodat će se kalendar na kojem će biti prikazani datumi posudbi knjiga. U Oracle APEX-u postoji prostor dizajnera stranice (engl. *page designer*) gdje se može uređivati dijelove određene stranice (slika 5).



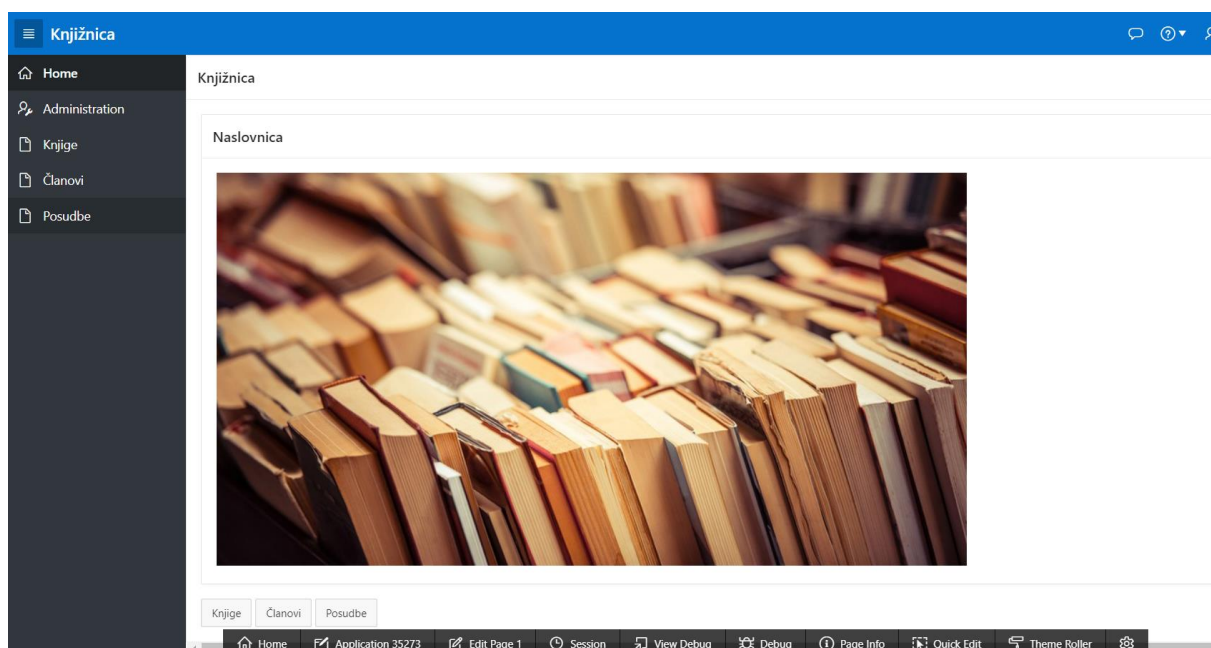
Slika 5. Dizajner stranice (Izvor: Oracle APEX)

Potrebno je dodati još stranicu *Članovi* s izvješćem o postojećim članovima te obrascem za unos novog člana. Nije bila potrebna izrada pregled nego samo korištenje već izrađene tablice *Korisnik*.

Tijekom izrade aplikacije, automatski se napravi naslovna stranica koja je prazna. Kako bi se upotpunila ta praznina, dodat će se gumbi koji vode do ostalih napravljenih stranica. Osim gumba dodat će se i slika koja se može staviti u dizajneru stranice. Željenu sliku prvo je potrebno uvesti s računala i zatim kopirati njen kod u *Region Image* željenog prostora, koji je u ovom slučaju naslovnica.

3.3. Konačan izgled aplikacije

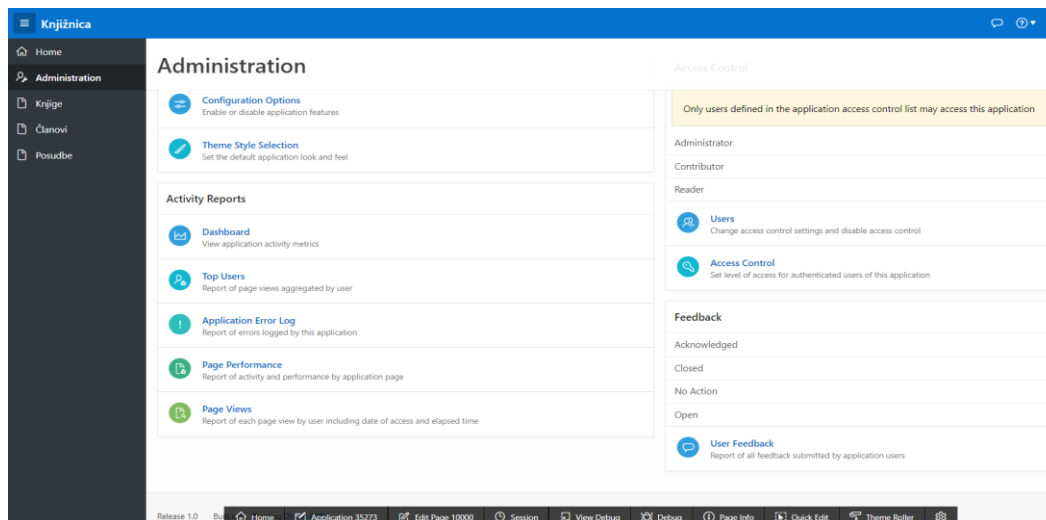
Korisnik aplikacije se u istu prvo mora prijaviti, a nakon toga dolazi na naslovnu stranicu aplikacije (slika 6).



Slika 6. Naslovna stranica aplikacije (Izrađeno u Oracle APEX-u)

Na naslovnoj stranici nalaze se ranije spomenuta slika, te tri gumba za odlazak na ostale stranice – Knjige, Članovi, Posudbe. S lijeve strane nalazi se izbornik za odlazak na Naslovnicu (engl. *Home*), Administraciju (engl. *Administration*) te Knjige, Članovi ili Posudbe.

Administracija (slika 7) je stranica dodana od strane Oracle APEX-a. Na njoj se mogu urediti tema i svojstva aplikacije, gledati aktivnosti korisnika aplikacije i slično.



Slika 7. Administracija (Izvor: Oracle APEX)

Na stranici Knjige vidljive su knjige koje knjižnica posjeduje te informacije o istima. Pritiskom na gumb *Stvori* (engl. *Create*), moguć je unos novih knjiga. Na stranici Članovi vidljivi su učlanjeni korisnici knjižnice te informacije o istima. Također, pritiskom na gumb *Stvori*, moguć je unos novih članova. Zadnja stranica ove aplikacije su Posudbe. U posudbama se vide podaci o posudbama knjiga i kalendar na kojemu se mogu očitati datumi posudbi i *ID_KNJIGA*. Tu se također može dodati novi redak s podacima o novoj posudbi pritiskom na gumb *Stvori*.

4. O Oracle APEX-u nakon korištenja

4.1. Jednostavnost platforme

U jednostavnost platforme ubraja se koliko je istu lako koristiti, u njoj se snalaziti, koliko je vizualno pregledna i praktična. Oracle APEX iako nudi velik broj opcija za izradu aplikacije, korisniku izlazi u susret time što on ne mora biti veliki poznavatelj SQL-a da bi napravio jednostavnu aplikaciju. Platforma pojednostavljuje stvaranje aplikacije te doista je nisko kodna jer se od korisnika nigdje ne traži da piše retke komandi i sve može napraviti pritiscima na miš. Naravno, ako korisnik preferira pisati SQL komande, on može tako raditi.

Vizualna preglednost aplikacije je zadovoljavajuća. Stranice na kojima se izrađuje aplikacija nisu pretrpane, iako nude mnogo opcija za izradu. Dizajn je jednostavan, stranice su većinom bijele, ali sadrže i obojane dijelove koji nisu pretjerani i pomažu oko snalaženja na stranicama. Uz to, korisnik može postaviti da platforma bude u tamnom načinu, ako mu je tako jednostavnije i ugodnije za raditi.

Iako Oracle APEX ide u susret korisniku pojednostavljivanjem cijelog procesa, ako se želi napraviti bolja aplikacija s više detalja u istoj, potrebno je uložiti vremena i kvalitetno se upoznati s platformom. Upravo zato što je tu puno opcija oko uređivanja stranica, nudi se širok spektar malih aplikacija koje se mogu koristiti unutar aplikacije koju korisnik radi.

Bitno je i poznavanje engleskog jezika i termina vezanih uz baze podataka i izrade aplikacija. Svaki termin nema svoju inačicu u hrvatskom jeziku, ali svakako je više bitno razumijevanje tih termina od samog prijevoda. Oracle APEX za dosta termina nudi objašnjenje. Pored radnji postoje upitnici u krugu iz kojih se, nakon pritiska mišem, pojavi oblak s objašnjenjem što je to. To je vrlo korisno i pomaže korisniku, ali se zna naći i dosta banalno napisanih objašnjenja koja korisniku nisu od pomoći.

Osim engleskog jezika, korisnik si za jezik platforme može postaviti i njemački, španjolski, francuski, talijanski, portugalski, japanski, korejski i kineski.

4.2. Brzina platforme

Brzina platforme označava koliko brzo platformi treba da provede zadani zadatak to jest koliko je ona responzivna. Osim toga označava i koliko se brzo i praktično može koristiti platformom.

U responzivnosti je Oracle APEX imao poteškoća. Naime, općenito je platforma dovoljno brza, ali kada treba odraditi neki proces, APEX je dosta puta to napravio sporo, ako uopće. S obzirom da se radi o platformi vezanoj uz baze podataka, bilo bi u redu da su ti procesi, s kojima je bilo problema, upravo vezani direktno za baze podataka, ali nisu bili. Problemi su se znali, između ostalog, pojavljivati tijekom prikaza samih tablica i njihovih atributa (bez njihovih vrijednosti). Naime, APEX ne bi izlistao traženu tablicu i trebalo bi se osvježiti stranicu u pregledniku, čime bi se mogao izgubiti napravljeni posao. Osim toga, tijekom kreiranja stranica aplikacije normalno je da platformi treba vremena da istu napravi, ali ponekad treba preko nekoliko minuta, a nema nikakav signal da se ona pravi. Bilo bi dobro uvesti nekakav vrteći znak za to, kao što primjerice internetski preglednici imaju kada se stranica osvježava ili samo učitava.

Osim sporosti, platforma je imala problem s točnosti u smislu da kada bi se nešto išlo namjerno obrisati, to bi i dalje bilo prisutno i ne bi se dalo otvoriti ni maknuti. Kod samog brisanja krivo napravljenih stavki, teško je naći kako to obrisati, što bi trebalo biti jasno naznačeno. Primjerice za stranicu koju se želi obrisati, umjesto da, primjerice, ima odmah na sebi znak za brisanje i uređivanje kada se preko nje prelazi mišem, korisnik mora pritisnuti desetak puta mišem, kako bi napravio jednostavnu radnju kao što je brisanje nečega. I to nije jasno naznačeno nego treba istražiti gdje se nalazi. Takve stvari inzistiraju da korisnik sve napravi bez greške iz prvog pokušaja, što je nerealno.

5. Zaključak

Oracle APEX široko je rasprostranjen alat koji se koristi za izgradnju aplikacija. Kroz izradu rada, pročitao se mnogo izjava zadovoljnih korisnika koji nisu imali ništa drugo za reći osim pohvala. Ipak, koristeći alat, spozna se više stvari u vezi alata, bile one pozitivne ili negativne. Izgled alata je vrlo suvremen, alat nudi mnogo opcija za uređivanje i uvoz već napravljenih dokumenata, tu su mnoge opcije za oblikovanje izgleda podataka u aplikaciji, osim toga svega, većina tih stvari već i je napravljena i samo ih treba oblikovati na željeni način. Alat je nisko kodan i nije potrebno pretjerano znanje programskih jezika, ali ako se želi napraviti vrlo dobra aplikacija, potrebno je poznavanje SQL-a i PL-a. Ipak, pronađeno je podosta mana gdje alat postaje vrlo spor, ponekad i ne reagira pa je potrebno potrošiti podosta vremena dok ponovno krene funkcionirati, zna odraditi krivu radnju i ponekad zahtjeva previše pritiska mišem kako bi se željena radnja odradila. Unatoč tim manama, alat nudi vrlo mnogo mogućnosti i preporuča ga se strpljivim osobama koje su dobro upoznate s PL/SQL-om i žele napraviti kvalitetnu aplikaciju.

6. Literatura

Popis literature:

Date, C. J. (2006). *An introduction to database systems*. Pearson Education India.

Hayes, R. M., Becker, J. (1970). *Handbook of data processing for libraries*. New York: Becker and Hayes, Inc.

Tkalac, S. (1988). *Relacijski model podataka*. Zagreb: Informator.

Wersig, G. (1971). *Information – Kommunikation – Dokumentation*. Berlin: Verlag Dokumentation.

Popis internetskih izvora:

Code Academy. (2019.) Preuzeto 13.08.2019. s <https://www.codecademy.com/articles/what-is-rest>

Easy Computer Academy. (2018). Preuzeto 18.09.2018. s <https://www.essentialsql.com/get-ready-to-learn-sql-server-what-is-a-null-value/>

Geek.hr. (2003.). Preuzeto 14.08.2019. s <https://geek.hr/webmajstori/uvod-u-baze-podataka-i-sql/>

Geek.hr. (2003.). Preuzeto 14.08.2019. s <https://geek.hr/webmajstori/sql-2-dio-tipovi-podataka/>

Geek.hr. (2003.). Preuzeto 14.08.2019. s <https://geek.hr/webmajstori/sql-3-dio-data-definition-language-ddl/>

Geek.hr. (2003.). Preuzeto 14.08.2019. s <https://geek.hr/webmajstori/sql-4-dio-data-query-language-dql/>

Geek.hr. (2003.). Preuzeto 14.08.2019. s <https://geek.hr/webmajstori/sql-5-dio-logicki-operatori-i-operatori-za-usporedbu/>

Geek.hr. (2003.). Preuzeto 14.08.2019. s <https://geek.hr/webmajstori/sql-6-dio-data-manipulation-language-dml/>

Hrvatska udruga Oracle korisnika. (2014). *Ciljevi i planovi*. Preuzeto 12.09.2018. s <http://www.hroug.hr/HrOUG-udruga/Ciljevi-i-planovi>

Kuserbajn, S. (2016.) Preuzeto 14.08.2019. s <https://zir.nsk.hr/islandora/object/vus:377/preview>

Medak, D. (2008.) Preuzeto 14.08.2019. s <http://www2.geof.unizg.hr/~dmedak/hr/baze01a.pdf>

Oracle i/ili njegovi suradnici. (2017). *Oracle Fact Sheet*. Preuzeto 12.09.2018. s <http://www.oracle.com/us/corporate/oracle-fact-sheet-079219.pdf>

Oracle. (bez dat.). Preuzeto 12.09.2018. s <https://apex.oracle.com/en/>

Oracle. (bez dat.). *Features*. Preuzeto 12.09.2018. s <https://apex.oracle.com/en/platform/features/>

Oracle. (bez dat.). *Architecture*. Preuzeto 12.09.2018. s <https://apex.oracle.com/en/platform/architecture/>

Oracle. (bez dat.). *Events*. Preuzeto 12.09.2018. s <https://apex.oracle.com/en/community/events/>

Oracle. (bez dat.). *Success Stories*. Preuzeto 12.09.2018. s <https://apex.oracle.com/en/solutions/success-stories/>

ReadMe. (2016.). *The History of REST APIs*. Preuzeto 12.8. 2019. s <https://blog.readme.io/the-history-of-rest-apis/>

SearchAppArchitecture. (2019.). *REST*. Preuzeto 13.08.2019. <https://searchapparchitecture.techtarget.com/definition/REST-REpresentational-State-Transfer>

webopedia. (2019.). *Array*. Preuzeto 13.08. 2019. s <https://www.webopedia.com/TERM/A/array.html>

Wikipedia. (2019.). Preuzeto 13.08.2019. s https://en.wikipedia.org/wiki/Representational_state_transfer

Wikipedia. (2019.). Preuzeto 13.08.2019. s <https://sh.wikipedia.org/wiki/SQL>

Wikipedia. (2019.). Preuzeto 13.08.2019. s <https://hr.wikipedia.org/wiki/SQL>

Wikipedia. (2019.). Preuzeto 13.08.2019. s <https://en.wikipedia.org/wiki/PL/SQL>

W3resource. (2019.). Preuzeto 14.08.2019. s
<https://www.w3resource.com/sql/operators.php>

Fotografija u aplikaciji (Slika 22.): Grigoryeva, Yulia. Shutterstock. Preuzeto
19.09.2018. s <https://theculturetrip.com/europe/russia/articles/the-10-best-books-by-leo-tolstoy-you-have-to-read/>

Prilozi

	Id Knjiga	Naslov	Autor	Izdavač	Godina	Vrsta
	3	Pobuna	Wladyslaw Stanislaw Reymont	Šareni dućan	1924	Roman
	4	Dužani cimetne boje	Bruno Schulz	Litteris	1934	Pripovijetka
	6	Ljubomora i medicina	Michał Choromański	Zora	1971	Roman
	7	Pravjek i ostala vremena	Olga Tokarczuk	NZMH	2001	Roman
	9	Panny z Wilka	Jarosław Iwaszkiewicz	Iskra	1984	Kratka priča
	11	Disneyland	Stanisław Dygat	Matica hrvatska	1968	Roman
	2	Medaljoni	Zofia Nałkowska	Nakladni zavod Hrvatske	1948	Novela
	8	Moralnošć pani Dulskiej	Gabriela Zapolska	Zakład Narodowy Ossolińskich	1966	Drama
	5	Radosć pisanja	Wisława Szymborska	Nova stvarnost	1997	Poezija
	10	Četvrta sestra	Janusz Głowacki	Gavella	1999	Drama
	1	Solaris	Stanisław Lem	AGM	1961	Novela
	12	Podróż z Herodotem	Ryszard Kapuściński	AGM	2008	Putopis

Prilog 1. Stranica Knjige (Izrađeno u Oracle APEX-u)

Unos nove knjige

Id Knjiga

Naslov

Id Autor

Id Izdavač

Godina

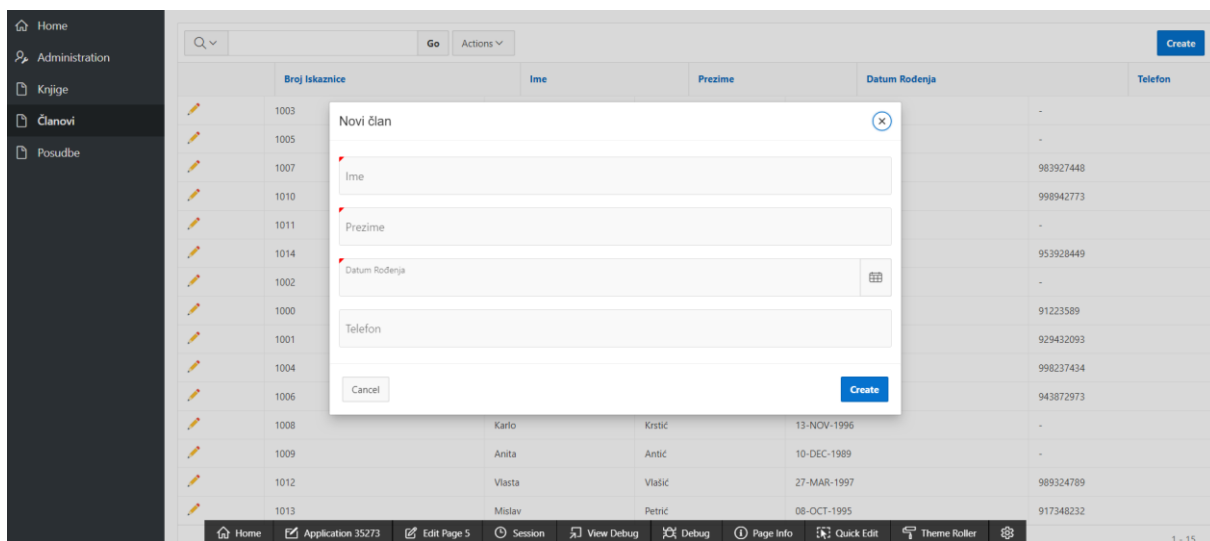
Id Vrsta

Cancel Create

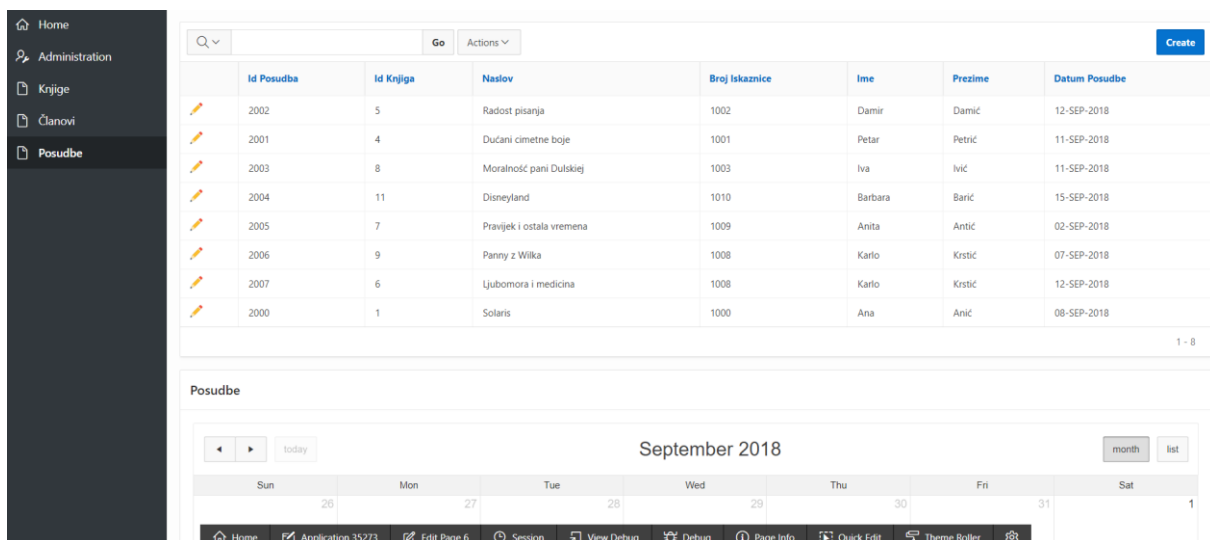
Prilog 2. Unos nove knjige (Izrađeno u Oracle APEX-u)

	Broj Iskaznice	Ime	Prezime	Datum Rođenja	Telefon
	1003	Iva	Ivić	24-FEB-2004	-
	1005	Kristina	Pavlić	12-APR-1989	-
	1007	Ana	Krstić	08-MAR-1992	983927448
	1010	Barbara	Barić	04-NOV-1996	998942773
	1011	Dominik	Domić	12-JAN-2003	-
	1014	Martina	Martić	10-OCT-1986	953928449
	1002	Damir	Damić	03-MAR-1973	-
	1000	Ana	Anić	24-SEP-1985	91223589
	1001	Petar	Petrić	01-FEB-1993	929432093
	1004	Antonio	Antić	17-MAY-1997	998237434
	1006	Ivana	Ivanić	16-JUN-1996	943872973
	1008	Karlo	Krstić	13-NOV-1996	-
	1009	Anita	Antić	10-DEC-1989	-
	1012	Vlasta	Vlašić	27-MAR-1997	989324789
	1013	Mislav	Petrić	08-OCT-1995	917348232

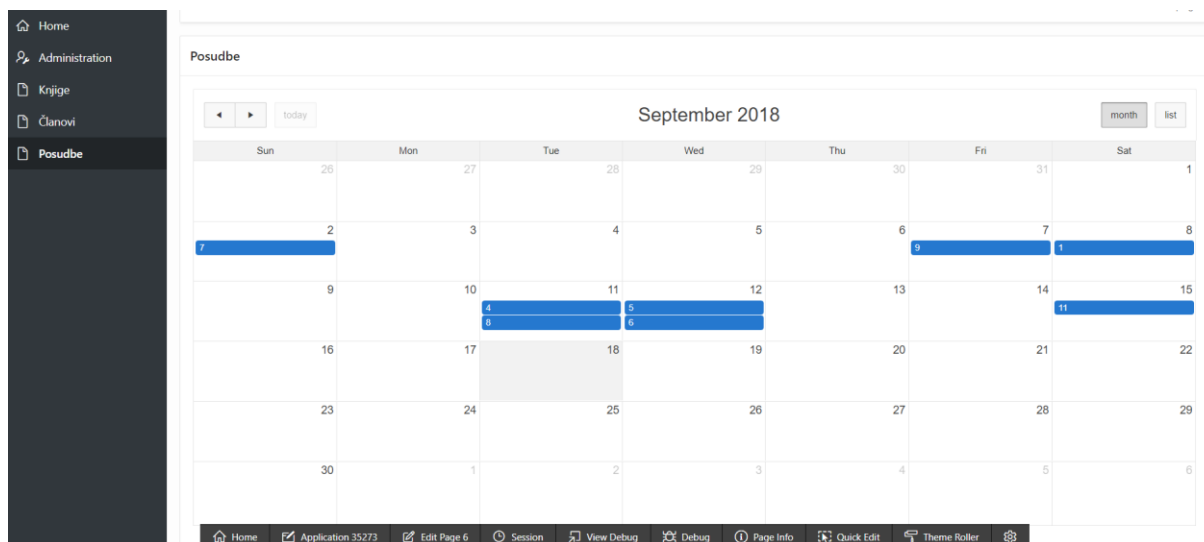
Prilog 3. Stranica Članovi (Izrađeno u Oracle APEX-u)



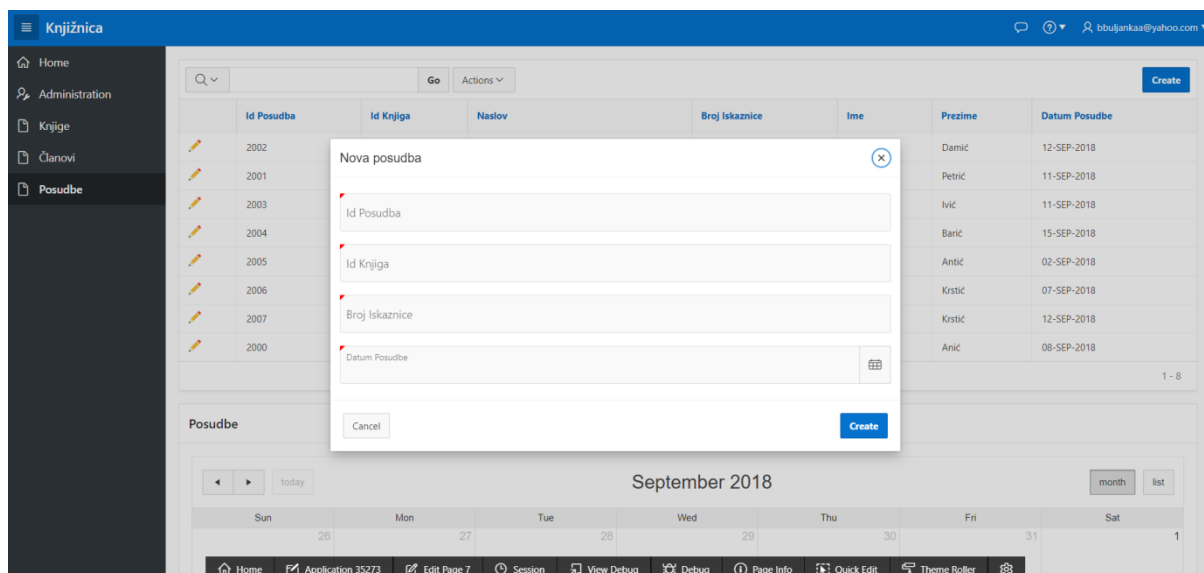
Prilog 4. Unos novog člana (Izrađeno u Oracle APEX-u)



Prilog 5. Stranica Posudbe (Izrađeno u Oracle APEX-u)



Prilog 6. Kalendar s datumima posudbi (Izrađeno u Oracle APEX-u)



Prilog 7. Unos nove posudbe (Izrađeno u Oracle APEX-u)

Razvoj aplikacija pomoću Oracle APEX alata

Sažetak

U ovome radu govori se o Oracle APEX-u kao platformi, o njegovim mogućnostima i arhitekturi. Poblje se objašnjavaju čimbenici njegove arhitekture (REST i SQL), a zatim se pregledava sučelje samog alata, što sve on nudi i kako. Za svaku aplikaciju bitni su i njeni korisnici, pa je stoga proučeno koje se to tvrtke i udruge koriste Oracle APEX-om, te zašto su odabrali baš taj alat. Osim toga, iza ovoga alata su i brojne konferencije, od kojih su neke i na hrvatskom području. Nakon teorijskog pregleda Oracle APEX-a, u njemu se kreira aplikacija „Knjižnica“ s bazom podataka u kojoj je moguće pratiti knjige koje knjižnica posjeduje, a uz to posudbe istih, kao i članove knjižnice. Naravno, te podatke je moguće dodatno upisivati u tu bazu podataka. Kroz teorijski i praktični dio ovoga alata, proučilo se kako on funkcionira, te je posljedično i prokomentiran rad samog alata, njegova praktičnost, jednostavnost i brzina.

Ključne riječi: Oracle APEX, REST, SQL, aplikacija

Application developing using Oracle APEX

Abstract

In this paper Oracle APEX as a platform is explained, its possibilities and architecture. Parts of its architecture (REST and SQL) are closely interpreted and then the tool's interface is examined, what does the tool offer and how does it offer. For every tool the users are important, so it is explored which companies and organizations use Oracle APEX and why. In addition, behind this tool, there are conferences some of which happen in Croatia. After theoretical review of the application, an application „Knjižnica“ is created in Oracle APEX. In the application it is possible to monitor books that the library contains, book loans and library's members. Except monitoring, a person can register new books, members and loans into the database. Through the theoretical and practical parts of the tool, it is investigated how it operates and its practicality, simplicity and speed are consequently commented on.

Keywords: Oracle APEX, REST, SQL, application