

Sveučilište Sjever

**Fakultet organizacije i informatike**

Varaždin

Projekt iz Baza Podataka 2

Prikaz baze podataka na temi *gacha* igrice

Sandra Sačarić,

14.01.2024., u Varaždinu

## Sadržaj

Uvod.....	3
Opis aplikacijske domene .....	4
Prikaz ERA dijagrama.....	5
Opis korištenog alata za modeliranje baze i administraciju baze.....	6
Opis korištenih tehnologija .....	7
Prikaz funkcionalnosti razvijenog sučelja.....	8
Prikaz aplikacije.....	12
Zaključak.....	14
Literatura.....	15

## Uvod

Projekt koji će za cilj imati stvaranje sveobuhvatnog ERA modela za inovativnu gacha igru, usmjerenu na igrače koji tragaju za raznovrsnim dostignućima, likovima, oružjima i materijalima. Razmotreni su različiti aspekti igre kako bi se igračima omogućilo jedinstveno iskustvo vezano za evidentiranje svog napretka te je pažljivo osmišljena aplikacijska domena.

Aplikacijska domena ovog projekta započinje s pojedinačnim igračem, koji kroz igru ima priliku ostvarivati različite dobitke putem banneri, određenih vremenskih perioda tijekom kojih se mogu dobiti određeni likovi ili oružja. Svaki korisnik ima priliku pratiti i bilježiti svoje postignuća unutar igre, prateći datum stjecanja dobivenih predmeta.

Im toga, projekt obuhvaća različite aspekte igre, uključujući područja, materijale i misije, gdje svaki igrač može sakupljati i dovršavati ih prema vlastitim preferencijama.

Posebna pažnja posvećena je i provjeri kompatibilnosti između oružja i likova, gdje se koristi ternarna veza kako bi se omogućila prilagodljivost izbora kombinacija za svakog igrača, uz istovremeno održavanje konzistentnosti vrsta oružja.

Implementacija baze podataka odvijala se uz korištenje Oracle APEX platforme, pružajući brz i učinkovit razvoj aplikacije. Alat drawio koristio se za grafički prikaz ERA modela, dok je Oracle SQL Developer bio ključan za pisanje SQL upita i skripti.

Ukupno, projekt predstavlja temelj za kompleksan i interaktivan svijet gacha igre, pružajući mogućnosti za daljnje proširenje i unapređenje, te ostavlja prostor za dodavanje novih funkcionalnosti kako bi se zadovoljile rastuće potrebe zajednice igrača.

## Opis aplikacijske domene

Ideja iza cijelog sustava kreće od samog igrača. Svaki pojedinačni igrač ima više različitih dostignuća ili nagrada (u formi oružja, likova, ili materijala) koje može bilježiti. S obzirom da se radi o *gacha* igrici, u njoj su često definirani *banneri* koji služe kao određeni vremenski period unutar kojeg se može dobiti određeni lik ili oružje – ukratko, neki *item*. Svaki igrač može preko određenog banneru osvojiti item.

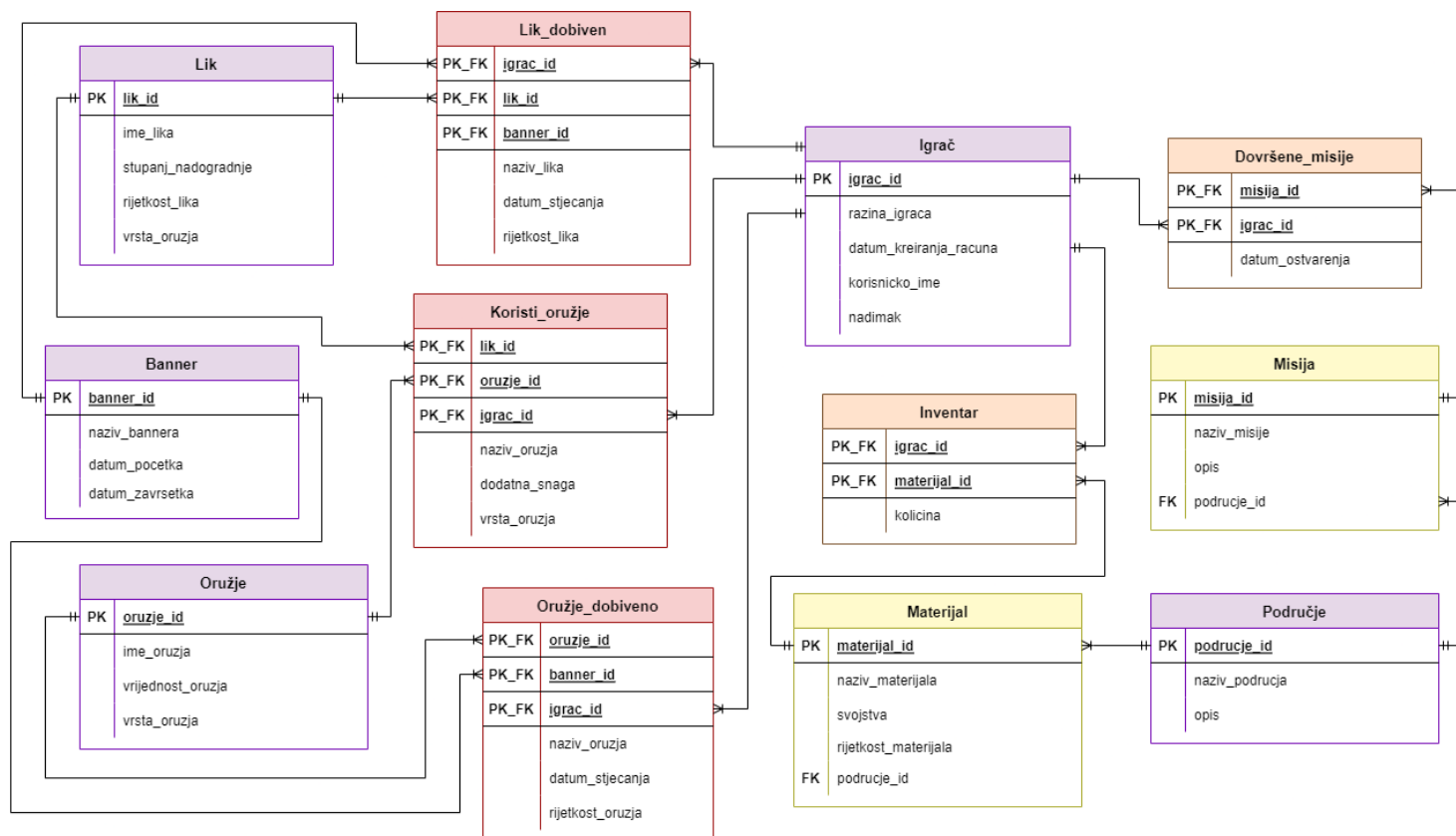
Iz tog razloga su definirane dvije ternarne veze preko kojih se provjerava je li dobitak itema valjan ili ne, na način da se u svakoj tablici koja nastane kao rezultat ternarnih veza prilikom pokušaja unosa definira datum stjecanja itema, koji bi se onda uspoređivao s datumom trajanja banneru koji je odabran od strane korisnika kao banner na kojem je item dobiven. Te bi se funkcionalnosti kasnije omogućile preko funkcija i okidača.

Nadalje, definirana su razna područja unutar igre, od kojih svako područje uz sebe može imati vezane materijale i misije, koje zatim svaki igrač može pronaći, odnosno riješiti. Dok svaki materijal može biti skladišten u igračevom inventaru, svaka misija može biti dovršena – što stvara potrebu za kreiranjem dviju novih slabih entiteta, s obzirom da se radi o vezama više-više, gdje jedan igrač može imati više sakupljenih materijala, a jedan materijal može biti sakupljen od više igrača, dok misiju može dovršiti više igrača, a igrač naravno može dovršiti više misija.

Osim toga, postoji i entitet koji je osmišljen za ostvarenje potrebe provjere kompatibilnosti nekog oružja i lika. I oružje i lik bi morali imati definiranu vrstu oružja u formi atributa – ideja je da svaki lik može koristiti samo jedno oružje, a oružje može biti samo jedne vrste. Stvorila bi se treća tablica koja bi povezivala igrača, lika, i oružje, te bi se provjeravale vrijednosti vrsta oružja u dvjema tablicama (lik i oružje) prema određenom ID-ju. U slučaju da se preklapaju, unos bi bio omogućen, a ukoliko bi vrste oružja bile različite, unos ili promjena podataka bili bi onemogućeni preko funkcije i okidača. Prikaz te opis navedenih funkcija i okidača kasnije će se spominjati. Inicijalno je to bilo odrađeno u formi binarne veze, ali je uočen problem.

Naime, razlog zašto zadnje spomenuti fenomen **ne bi** mogao u ovom slučaju biti riješen preko binarne veze više-više i stvaranja trećeg slabog entiteta između lika i oružja je činjenica da određena kombinacija oružja i lika **ne mora** biti ista za svakog igrača. Ideja je da svaki igrač može odabrati kako bi pospojio lika i oružje na svoj način, doklegod se vrste poklapaju. Konkretno, ako bismo postavili da lik A koristi oružje B, mogli bismo također staviti da taj isti lik A koristi i oružje C, ali se ne bi znalo kod kojeg je igrača postavljeno koje oružje. Iz tog sam razloga riješila problem preko ternarne veze, tako da se za svaku kombinaciju oružja i lika nadovezujemo na ID igrača. Tada, više igrača mogu koristiti iste kombinacije, ali ne moraju, i u tom se slučaju točno zna tko koristi koju.

## Prikaz ERA dijagrama



Slika 1 ERA model (samostalna izrada)

Prilikom početka rada na projektu, ERA model se sastojao od 15-ak entiteta s isključivo binarnim vezama, ali kroz rad na aplikaciji, unatoč tome što je sve radilo ispravno, nisu se mogle postići stvari na način koji su točno isprva zamišljene.

Iz tog je razloga smanjen broj entiteta kroz unos par ternarnih veza. Do ideje o samim entitetima i kako bi bili povezani došlo je kroz razmišljanje o funkcijama i okidačima koji bi mogli biti implementirani, što je ujedno i potaknulo izmjenu ERA modela. Jednom kad su se uvele ternarne veze, moglo se lakše osigurati da se unosi i promjene podataka izvršavaju samo u određenim uvjetima.

S obzirom na to, trenutni broj entiteta svodi se na 12 - tri ternarne veze, dok su sve ostale su binarne. U jednom trenu je postojala ideja provjere kompatibilnosti oružja i lika preko jedne unarne veze, ali je trenutni način lakši i efikasniji.

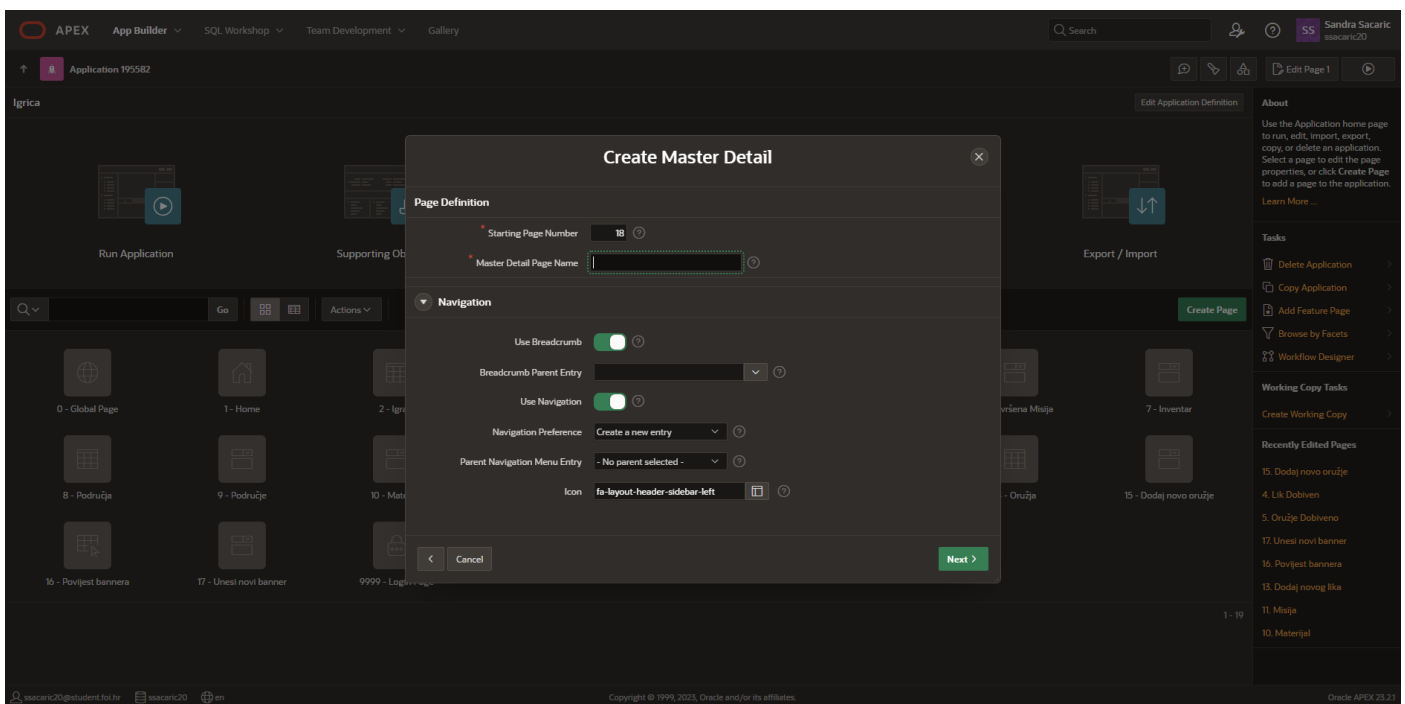
## Opis korištenog alata za modeliranje baze i administraciju baze

Za modeliranje i administraciju baze podataka korišten je Oracle APEX, snažan alat za razvoj web aplikacija koji se integrira s Oracle bazom podataka. Oracle APEX pruža intuitivan grafički korisnički sučelja za brzo i efikasno stvaranje ERA modela. Kroz APEX su implementirane master-detail forme koje uključuju klasične izvještaje i interaktivne gridove za fleksibilno prikazivanje podataka.

Kreirano je više od 10 tablica, koristeći APEX-ov vizualni dizajner za postavljanje atributa i veza između tablica.

Dodatno, implementirane su tri funkcije i tri okidača kako bi se automatizirale određene akcije u bazi podataka, poboljšavajući njenu učinkovitost i održavanje.

Sve ove operacije, uključujući modeliranje tablica, definiranje veza, izrada funkcija i okidača, izvedene su putem vizualnog sučelja Oracle APEX-a, što je rezultiralo brzim i efikasnim razvojem ERA modela. Oracle APEX je dokazao svoju vrijednost kao alat koji omogućuje integriranje kompleksnih funkcionalnosti baze podataka u jednostavne i pristupačne web aplikacije.



Slika 2 Prikaz alata (Oracle APEX)

## Opis korištenih tehnologija

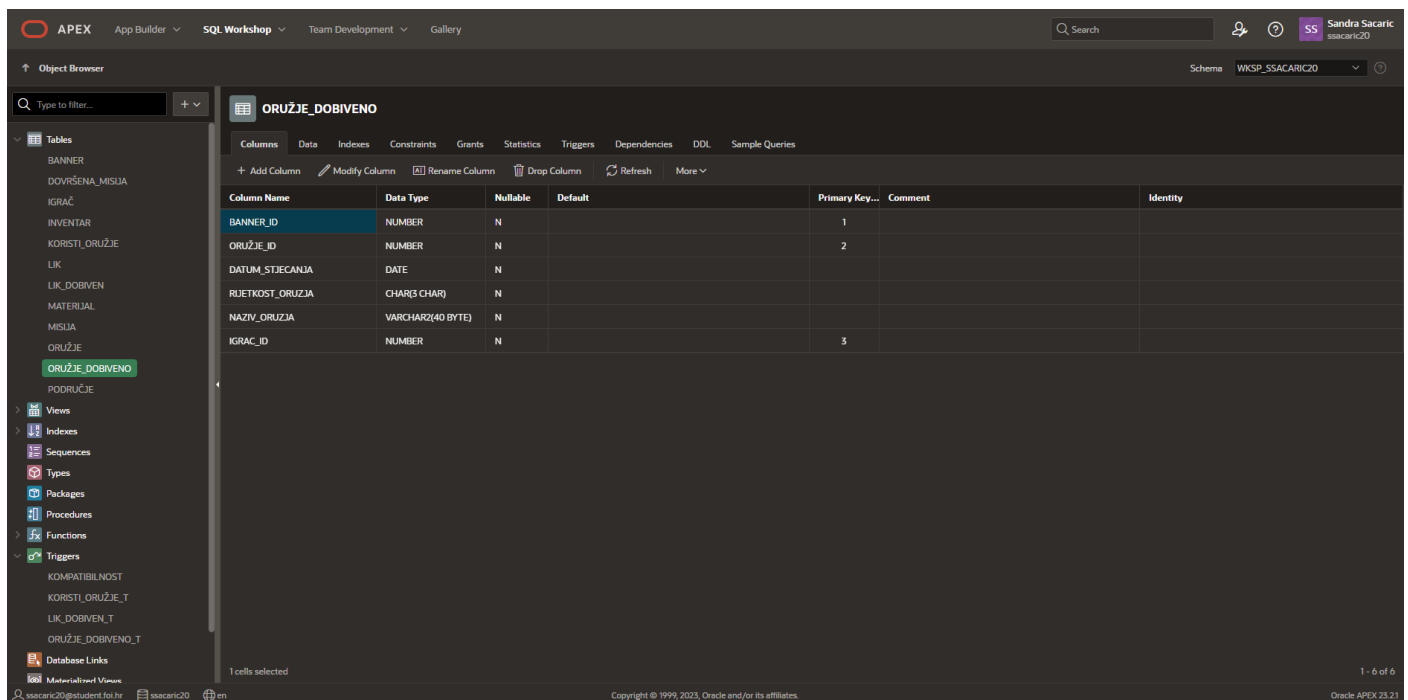
Za izradu projekta koristila se verzija 23.2.

Kako bi se kreirao grafički prikaz ERA modela, korišten je alat drawio, s obzirom da je dosta jednostavan i intuitivan za korištenje. Tijekom modeliranja baze podataka, stvorene su različite tablice, sveukupno njih 12, implementirani su različiti tipovi veza, uglavnom binarne s pokojom ternarnom vezom, te su dodani okidači za automatizaciju određenih akcija.

Kako se u Oracle APEX-u okidač može stvoriti samo u odnosu na jednu tablicu, a budući da su potrebe ovog projekta zahtjevale referenciranje na druge tablice unutar baze, bilo je potrebno stvoriti funkcije koje bi omogućile pravilno izvođenje okidača.

Konkretno, nastojala se ostvariti ideja da se unos novog ili promjena postojećeg dobivenog lika omogućiti samo kada je datum stjecanja dobivenog lika unutar trajanja određenog banneru. Drugim riječima, da se svakom igraču datum stjecanja nekog lika uspoređuje s datumom početka i datumom završetka banneru na kojem je dobiven, budući da su likovi dostupni samo u određenim razdobljima, na određenim bannerima, i zato što je svaki banner različitog trajanja pa je potrebno razlikovati svaki.

Razvojno okruženje uključivalo je Oracle SQL Developer za pisanje SQL upita i praćenje izvođenja skripti. Programski jezik korišten za dodatne skripte bio je SQL.



The screenshot displays the Oracle APEX SQL Workshop interface. On the left, the 'Object Browser' shows a tree of database objects, with 'ORUŽJE\_DOBIVENO' selected under the 'Tables' category. The main panel shows the table's structure with columns: BANNER\_ID, ORUŽJE\_ID, DATUM\_STJECANJA, RIJETKOST\_ORUŽJA, NAZIV\_ORUŽJA, and IGRAC\_ID. The table is highlighted in the Object Browser, and the main panel shows its structure.

Column Name	Data Type	Nullable	Default	Primary Key...	Comment	Identity
BANNER_ID	NUMBER	N		1		
ORUŽJE_ID	NUMBER	N		2		
DATUM_STJECANJA	DATE	N				
RIJETKOST_ORUŽJA	CHAR(3 CHAR)	N				
NAZIV_ORUŽJA	VARCHAR(240 BYTE)	N				
IGRAC_ID	NUMBER	N		3		

Slika 3 Prikaz tablica, okidača (samostalna izrada)

## Prikaz funkcionalnosti razvijenog sučelja

Prikaz radnje prilikom pokušaja unosa koji krši uvjet vezan za atribut „Rijetkost lika“, gdje unos mora poprimiti jednu od triju vrijednosti (R, SR, ili SSR) i jednog jednostavnog upita.

The screenshot shows the 'Dodaj novog lika' (Add new character) form in Oracle APEX. The form has the following fields: 'Ime lika' (Name of character) with the value 'Pukomir', 'Stupanj nadogradnje' (Upgrade level) with the value '2', 'Rijetkost lika (R, SR, SSR)' (Rarity of character) with the value 'SSG', and 'Vrsta oružja' (Weapon type) with the value 'Katalizator'. An error message is displayed: '1 error has occurred' with the details 'ORA-02290: check constraint (WKSP\_SSACARIC20.LIK\_CON) violated'. The form has 'Poništi' (Cancel) and 'Kreiraj' (Create) buttons.

Slika 6 Unos (Oracle APEX)

```
SQL

alter table "WKSP_SSACARIC20"."LIK" add constraint
"LIK_CON" check ( "RIJETKOST_LIKA" IN ('R', 'SR', 'SSR') );
```

Slika 5 SQL constraint (samotalna izrada)

Conditions

SQL

Results

Saved SQL

```
select MISIJA.MISIJA_ID as MISIJA_ID,
       MISIJA.NAZIV_MISIJE as NAZIV_MISIJE,
       "IGRAČ".IGRAC_ID as IGRAC_ID,
       "IGRAČ".KORISNICKO_IME as KORISNICKO_IME,
       "DOVRŠENA_MISIJA".DATUM_OSTVARENJA as DATUM_OSTVARENJA
from "IGRAČ" "IGRAČ",
     "DOVRŠENA_MISIJA" "DOVRŠENA_MISIJA",
     MISIJA MISIJA
where "IGRAČ".IGRAC_ID="DOVRŠENA_MISIJA".IGRAC_ID
      and MISIJA.MISIJA_ID="DOVRŠENA_MISIJA".MISIJA_ID
```

Conditions

SQL

Results

Saved SQL

MISIJA_ID	NAZIV_MISIJE	IGRAC_ID	KORISNICKO_IME	DATUM_OSTVARENJA
1	hvtanje lopova	1	Bababuj	01/01/2023

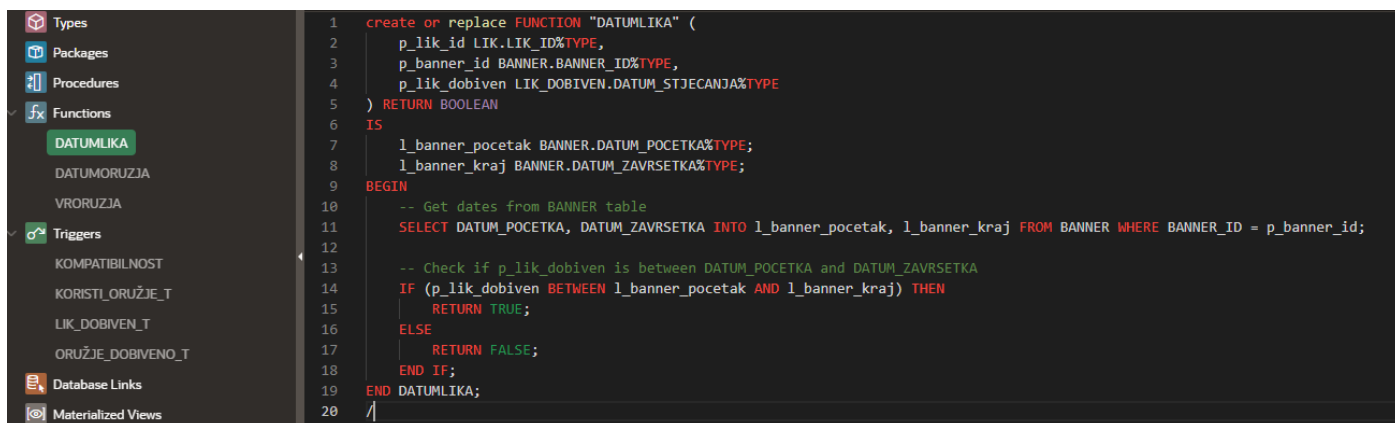
row(s) 1 - 1 of 1

Download

Slika 4 Prikaz jednostavnog upita (samotalna izrada)



Prikaz funkcije koja dohvaća vrijednosti datuma početka i završetka iz tablice BANNER te datum stjecanja lika iz tablice LIK\_DOBIVEN, koja ih potom uspoređuje prema ID-ju bannera (tamo gdje se vrijednosti iz obje tablice preklapaju).



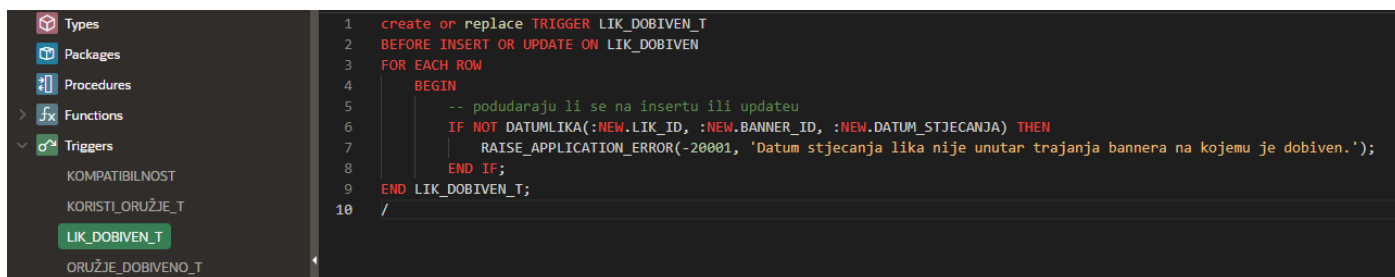
```

1 create or replace FUNCTION "DATUMLIKA" (
2     p_lik_id LIK.LIK_ID%TYPE,
3     p_banner_id BANNER.BANNER_ID%TYPE,
4     p_lik_dobiven LIK_DOBIVEN.DATUM_STJECANJA%TYPE
5 ) RETURN BOOLEAN
6 IS
7     l_banner_pocetak BANNER.DATUM_POCETKA%TYPE;
8     l_banner_kraj BANNER.DATUM_ZAVRSETKA%TYPE;
9 BEGIN
10    -- Get dates from BANNER table
11    SELECT DATUM_POCETKA, DATUM_ZAVRSETKA INTO l_banner_pocetak, l_banner_kraj FROM BANNER WHERE BANNER_ID = p_banner_id;
12
13    -- Check if p_lik_dobiven is between DATUM_POCETKA and DATUM_ZAVRSETKA
14    IF (p_lik_dobiven BETWEEN l_banner_pocetak AND l_banner_kraj) THEN
15        RETURN TRUE;
16    ELSE
17        RETURN FALSE;
18    END IF;
19 END DATUMLIKA;
20 /

```

Slika 7 Funkcija (samostalna izrada)

Prikaz okidača koji provjerava vrijednost koju vraća funkcija DATUMLIKA, svakog puta prilikom inserta ili updatea novog ili postojećeg reda tablice LIK\_DOBIVEN. Konkretno, ako vraća vrijednost false, u tom slučaju se poziva greška i prikazuje se prikladna poruka.

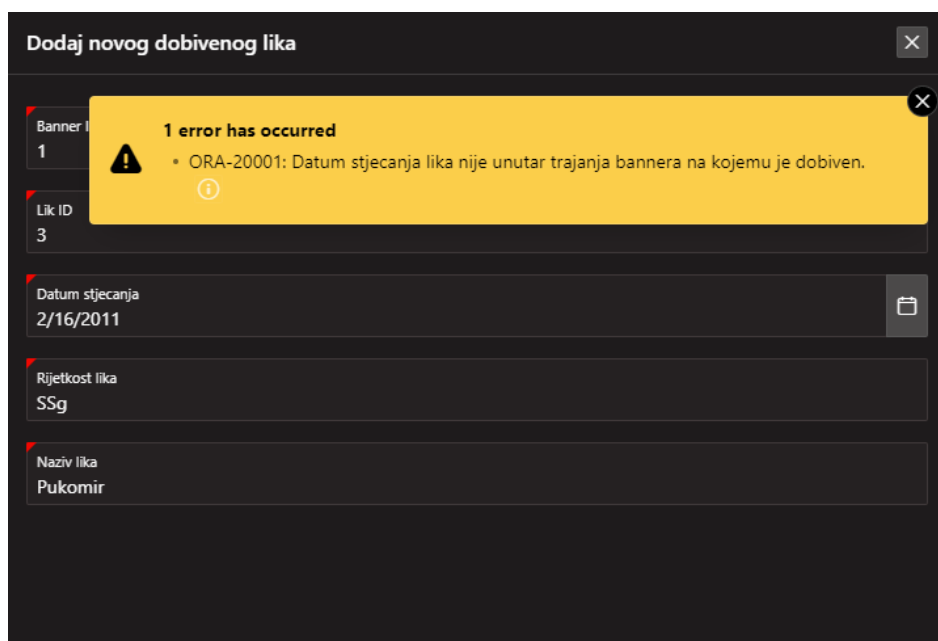


```

1 create or replace TRIGGER LIK_DOBIVEN_T
2 BEFORE INSERT OR UPDATE ON LIK_DOBIVEN
3 FOR EACH ROW
4 BEGIN
5     -- podudaraju li se na insertu ili updateu
6     IF NOT DATUMLIKA(:NEW.LIK_ID, :NEW.BANNER_ID, :NEW.DATUM_STJECANJA) THEN
7         RAISE_APPLICATION_ERROR(-20001, 'Datum stjecanja lika nije unutar trajanja bannera na kojemu je dobiven.');
```

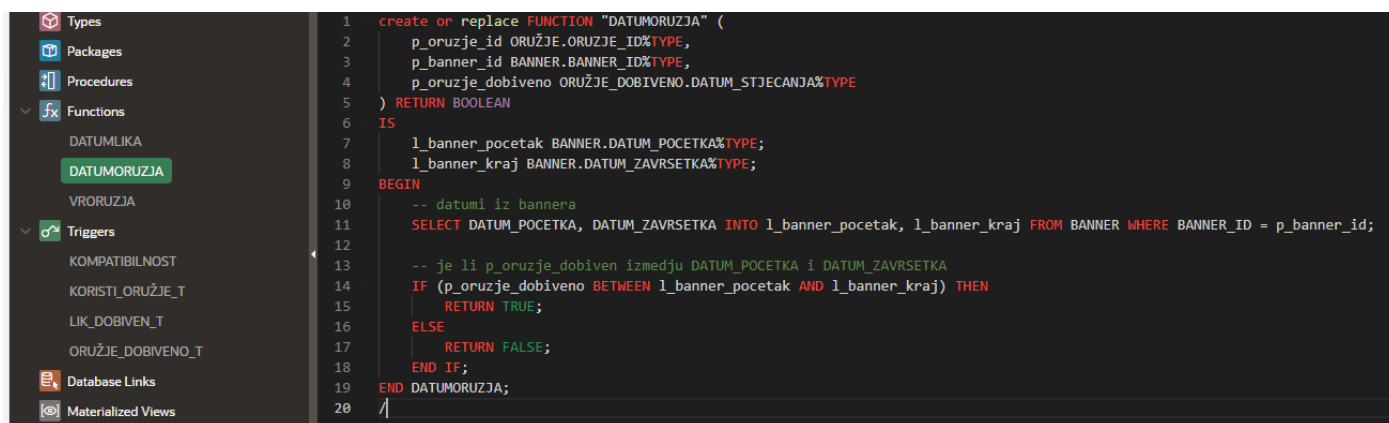
Slika 8 Okidač (samostalna izrada)

Ovako izgleda izbacivanje greške prilikom pokušaja unosa neispravnog datuma, odnosno unosa datuma stjecanja nekog lika van opsega trajanja bannera.



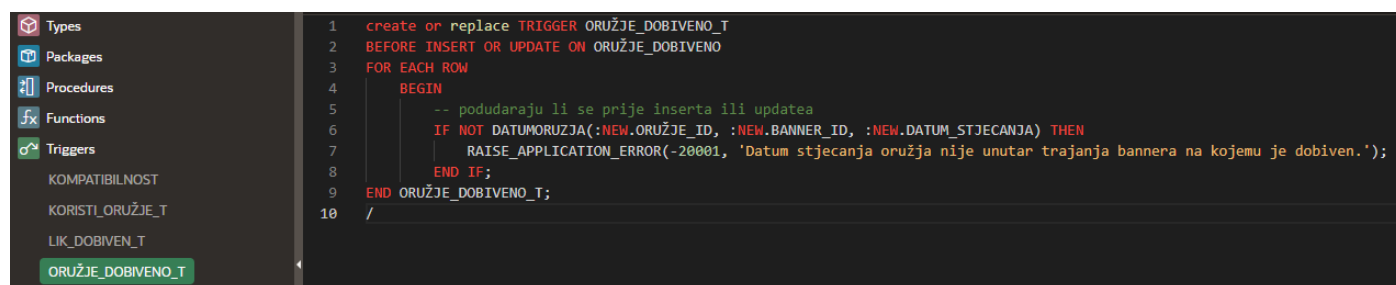
Slika 9 Prikaz greške (Oracle APEX)

Gotovo identičan kod je korišten za izradu funkcije i okidača koji služe za provjeru unosa ili promjena nad tablicom ORUŽJE\_DOBIVENO. Ime funkcije je DATUMORUZJA, a ime okidača je ORUŽJE\_DOBIVENO\_T.



```
1 create or replace FUNCTION "DATUMORUZJA" (  
2   p_oruzje_id ORUŽJE.ORUŽJE_ID%TYPE,  
3   p_banner_id BANNER.BANNER_ID%TYPE,  
4   p_oruzje_dobiveno ORUŽJE_DOBIVENO.DATUM_STJECANJA%TYPE  
5 ) RETURN BOOLEAN  
6 IS  
7   l_banner_pocetak BANNER.DATUM_POCETKA%TYPE;  
8   l_banner_kraj BANNER.DATUM_ZAVRSETKA%TYPE;  
9 BEGIN  
10  -- datumi iz bannera  
11  SELECT DATUM_POCETKA, DATUM_ZAVRSETKA INTO l_banner_pocetak, l_banner_kraj FROM BANNER WHERE BANNER_ID = p_banner_id;  
12  
13  -- je li p_oruzje_dobiven izmedju DATUM_POCETKA i DATUM_ZAVRSETKA  
14  IF (p_oruzje_dobiveno BETWEEN l_banner_pocetak AND l_banner_kraj) THEN  
15    RETURN TRUE;  
16  ELSE  
17    RETURN FALSE;  
18  END IF;  
19 END DATUMORUZJA;  
20 /
```

Slika 10 Funkcija (samostalna izrada)

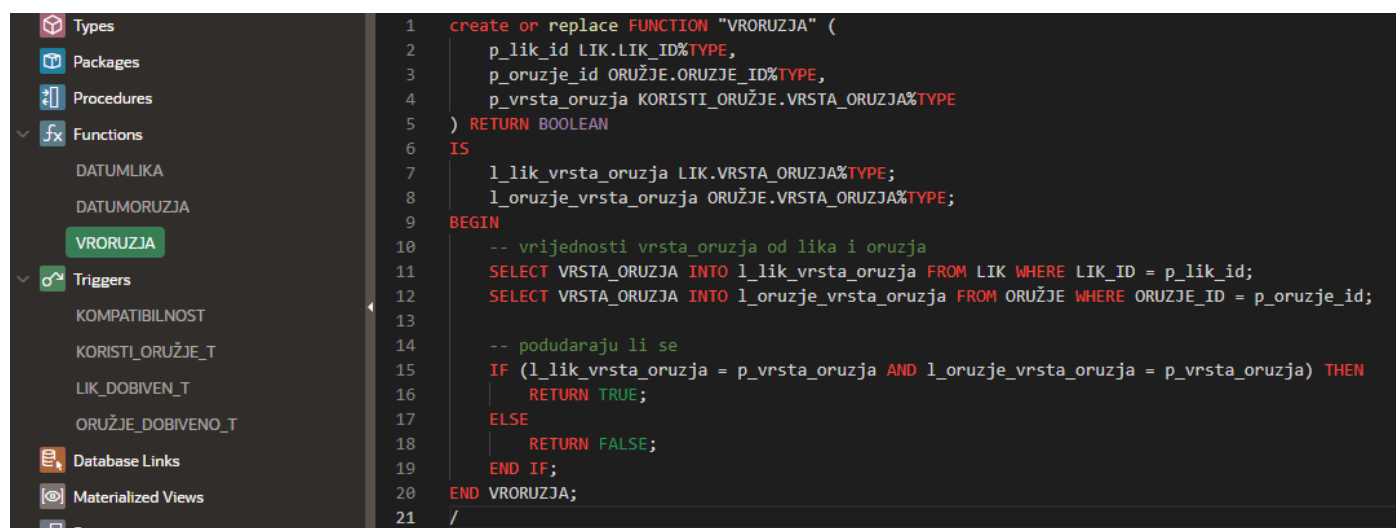


```
1 create or replace TRIGGER ORUŽJE_DOBIVENO_T  
2 BEFORE INSERT OR UPDATE ON ORUŽJE_DOBIVENO  
3 FOR EACH ROW  
4 BEGIN  
5   -- podudaraju li se prije inserta ili updatea  
6   IF NOT DATUMORUZJA(:NEW.ORUŽJE_ID, :NEW.BANNER_ID, :NEW.DATUM_STJECANJA) THEN  
7     RAISE_APPLICATION_ERROR(-20001, 'Datum stjecanja oružja nije unutar trajanja bannera na kojemu je dobiven.');
```

Slika 11 Okidač

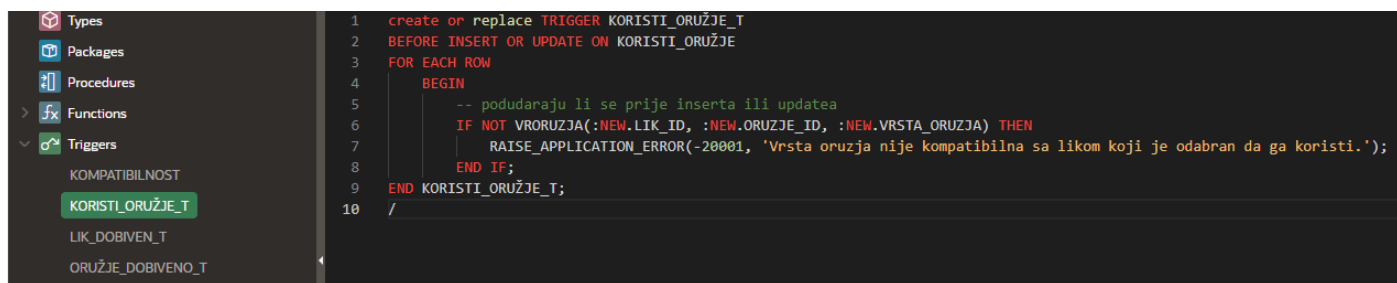
Nadalje, ideja iza tablice KORISTI\_ORUŽJE je bila da se koristi kao provjera kompatibilnosti oružja i lika, s obzirom da svako oružje mora imati vrstu kao atribut, a svaki lik može koristiti samo jedno određeno oružje.

Kako je već bilo prije spomenuto, u Oracle APEX-u nije omogućeno da se referenciraju tablice van one za koju se stvara okidač pa je iz tog razloga stvorena funkcija koja dohvaća VRSTA\_ORUZJA iz triju tablica te ih potom uspoređuje tamo gdje se ID-jevi lika i oružja podudaraju. Ako funkcija vraća vrijednost false, izbacuje se greška.



```
1 create or replace FUNCTION "VRORUZJA" (  
2   p_lik_id LIK.LIK_ID%TYPE,  
3   p_oruzje_id ORUŽJE.ORUŽJE_ID%TYPE,  
4   p_vrsta_oruzja KORISTI_ORUŽJE.VRSTA_ORUZJA%TYPE  
5 ) RETURN BOOLEAN  
6 IS  
7   l_lik_vrsta_oruzja LIK.VRSTA_ORUZJA%TYPE;  
8   l_oruzje_vrsta_oruzja ORUŽJE.VRSTA_ORUZJA%TYPE;  
9 BEGIN  
10  -- vrijednosti vrsta_oruzja od lika i oruzja  
11  SELECT VRSTA_ORUZJA INTO l_lik_vrsta_oruzja FROM LIK WHERE LIK_ID = p_lik_id;  
12  SELECT VRSTA_ORUZJA INTO l_oruzje_vrsta_oruzja FROM ORUŽJE WHERE ORUŽJE_ID = p_oruzje_id;  
13  
14  -- podudaraju li se  
15  IF (l_lik_vrsta_oruzja = p_vrsta_oruzja AND l_oruzje_vrsta_oruzja = p_vrsta_oruzja) THEN  
16    RETURN TRUE;  
17  ELSE  
18    RETURN FALSE;  
19  END IF;  
20 END VRORUZJA;  
21 /
```

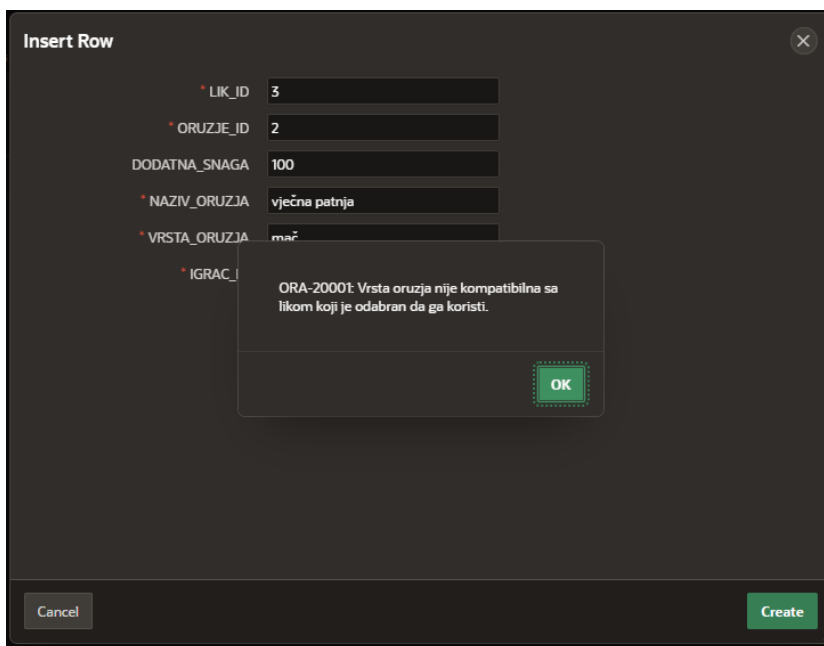
Slika 12 Funkcija (samostalna izrada)



```
1 create or replace TRIGGER KORISTI_ORUŽJE_T
2 BEFORE INSERT OR UPDATE ON KORISTI_ORUŽJE
3 FOR EACH ROW
4 BEGIN
5     -- podudaraju li se prije inserta ili updatea
6     IF NOT VRORUŽJA(:NEW.LIK_ID, :NEW.ORUŽJE_ID, :NEW.VRSTA_ORUŽJA) THEN
7         RAISE_APPLICATION_ERROR(-20001, 'Vrsta oruzja nije kompatibilna sa likom koji je odabran da ga koristi.');
```

Slika 13 Okidač (samostalna izrada)

Konkretno, u ovom primjeru se pokušava provjeriti kompatibilnost slučaja gdje je za oružje ID-ja 2 postavljena vrsta „mač“, dok je za vrstu oružja koje koristi lik ID-ja 3 postavljena vrijednost „katalizator“. Iz tog razloga se izbacuje greška prilikom unosa. Jednom kad se odabere ili oružje kojem je vrsta oružja postavljena na katalizator ili lik koji koristi mač, unos bi bio moguć.



Insert Row

\* LIK\_ID 3

\* ORUŽJE\_ID 2

DODATNA\_SNAGA 100

\* NAZIV\_ORUŽJA vječna patnja

\* VRSTA\_ORUŽJA mač

\* IGRAC\_

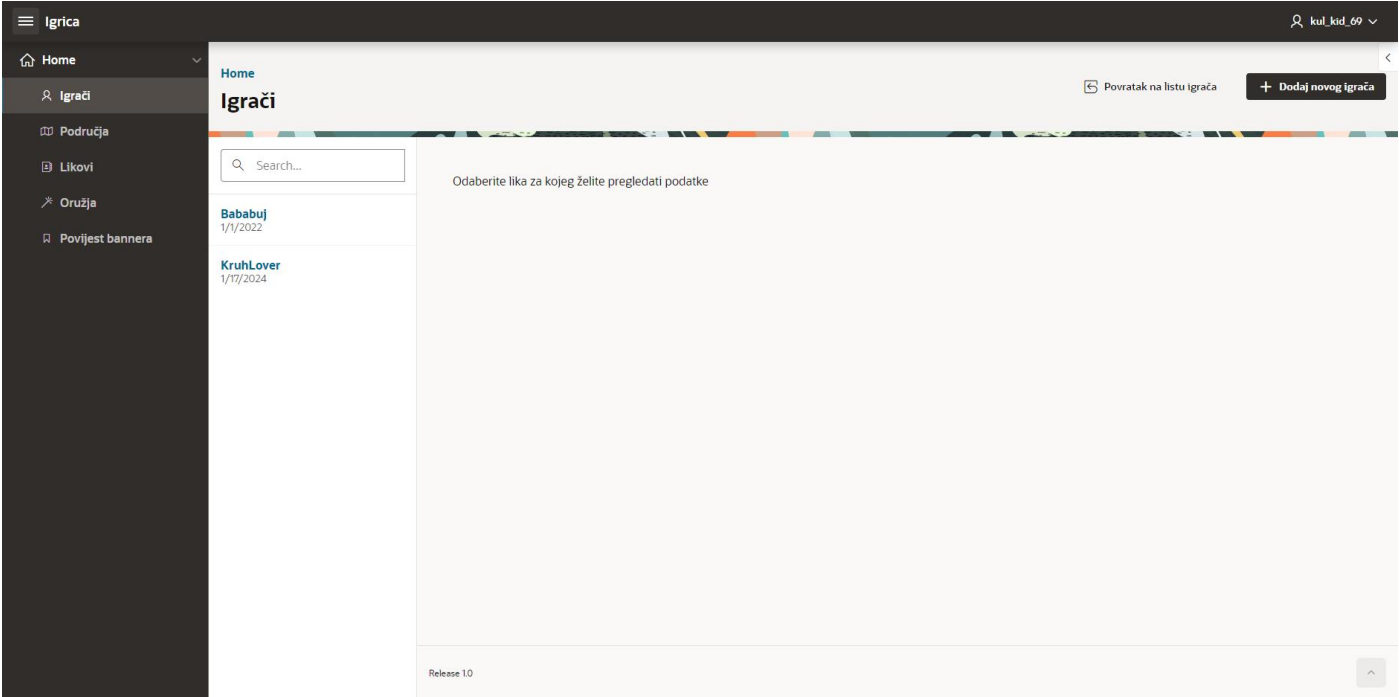
ORA-20001: Vrsta oruzja nije kompatibilna sa likom koji je odabran da ga koristi.

OK

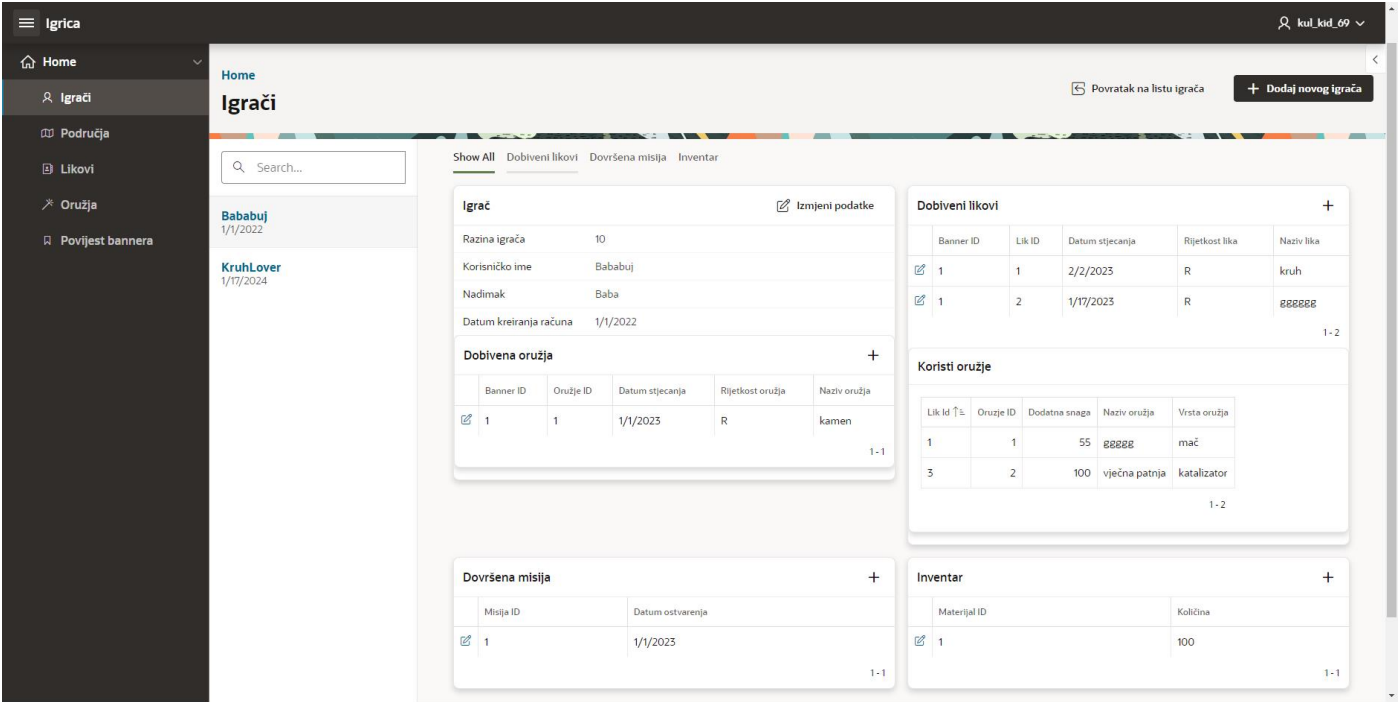
Cancel Create

Slika 14 Prikaz neispravnog unosa (Oracle APEX)

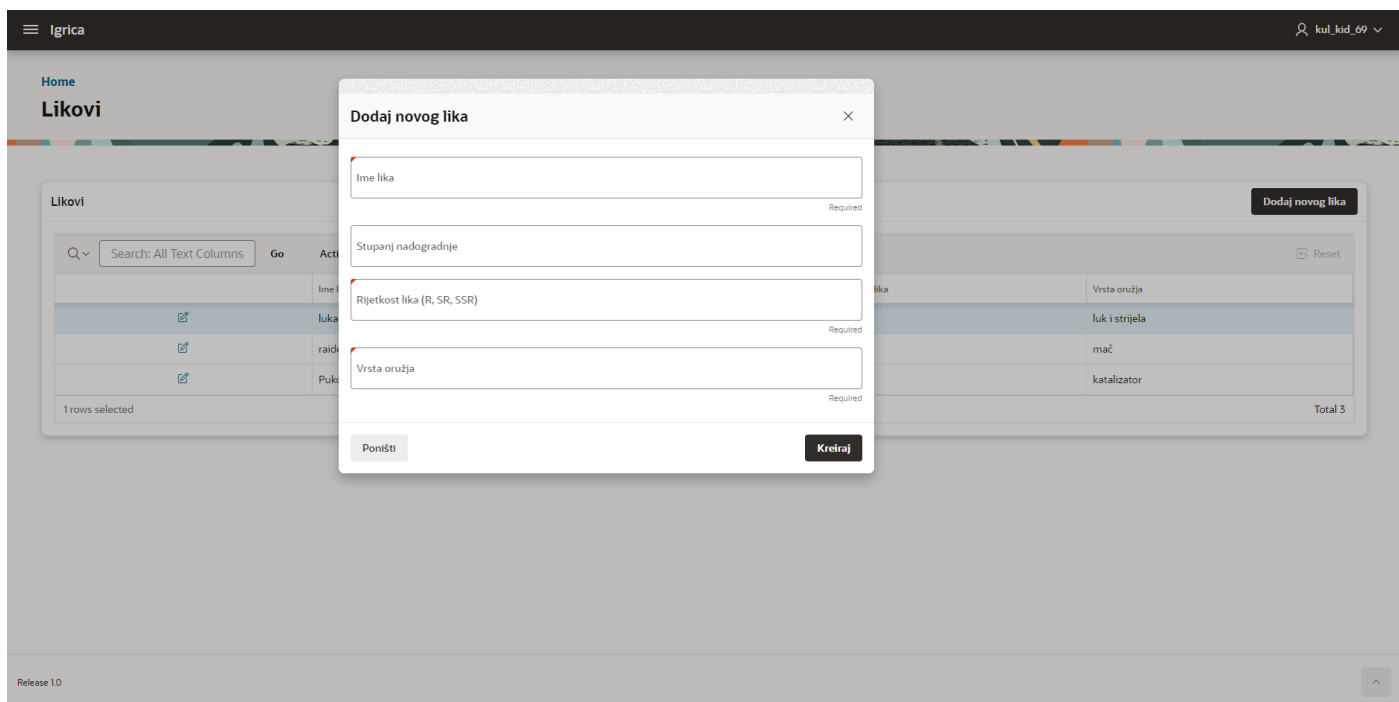
# Prikaz aplikacije



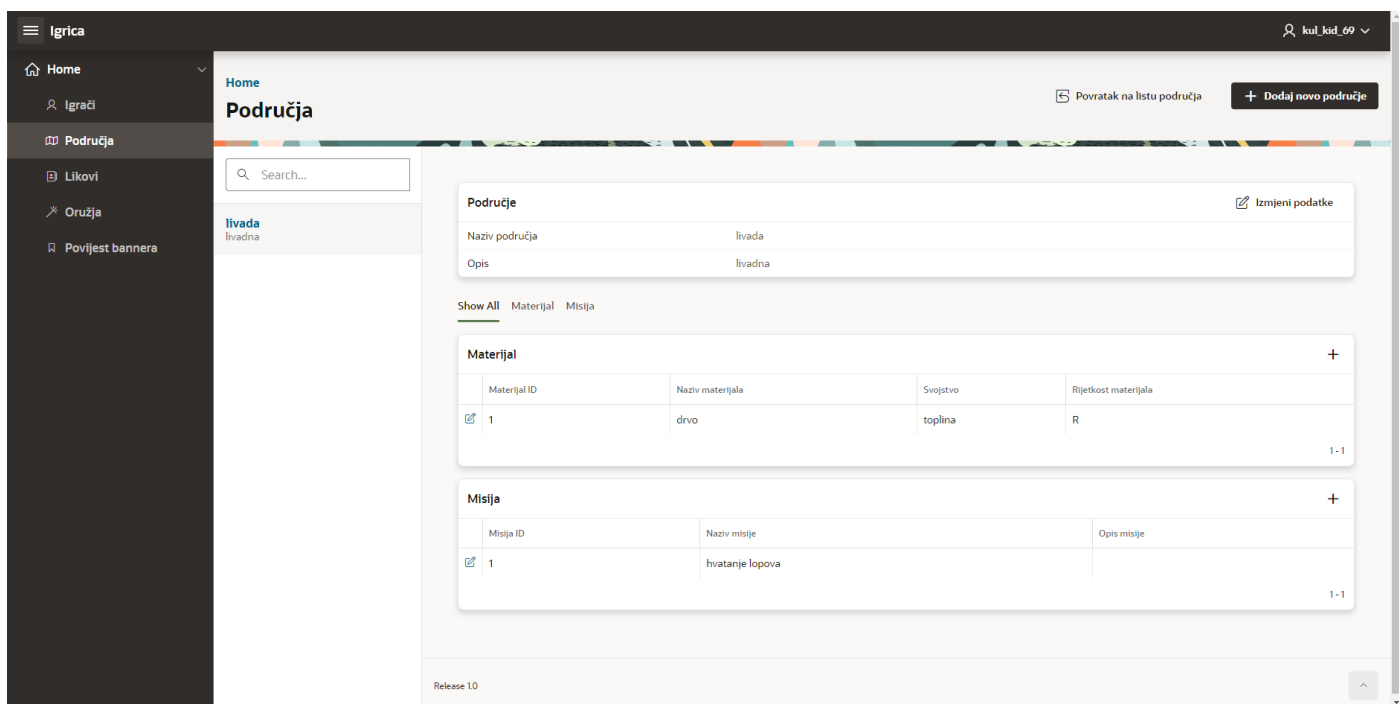
Slika 15 Prikaz liste igrača (samostalna izrada)



Slika 16 Prikaz podataka o odabranom igraču (samostalna izrada)



Slika 17 Prikaz interaktivnog grida i unosa (samostalna izrada)



Slika 18 Prikaz područja i detalja (samostalna izrada)

## Zaključak

U ovom projektu, fokus je bio na izradi ERA modela za sustav temeljen na konceptu gacha igre. Svaki igrač ima mogućnost ostvarivanja različitih dostignuća ili nagrada putem različitih banner-a, definiranih vremenskih perioda unutar kojih se mogu dobiti određeni likovi, oružja ili materijali. ERA model je oblikovan tako da podržava praćenje i provjeru valjanosti ovih dobitaka kroz ternarne veze i tablice koje bilježe datum stjecanja i trajanje banner-a.

Dodatno, sustav uključuje definiranje područja, materijala i misija, prateći kako svaki igrač sakuplja materijale i dovršava misije. Ovo je implementirano putem slabih veza više-više između igrača, materijala i misija, stvarajući kompleksan sustav prikaza i praćenja aktivnosti igrača unutar igre.

Također, projekt uključuje i provjeru kompatibilnosti između oružja i likova, gdje se koristi ternarna veza kako bi se omogućila fleksibilnost odabira kombinacija za svakog igrača, a istovremeno se osigurava da vrste oružja odgovaraju.

U procesu modeliranja korišten je alat drawio za grafički prikaz ERA modela. Za implementaciju baze podataka korištena je Oracle APEX, koji je omogućio brz i efikasan razvoj aplikacije. SQL Developer je korišten za pisanje SQL upita i skripti, dok su funkcije i okidači bili ključni za automatizaciju određenih akcija unutar baze podataka.

Kroz ovaj projekt, ostvarena je kompleksna i funkcionalna baza podataka koja podržava raznolike aktivnosti igrača unutar gacha igre, prateći njihove dobitke, sakupljanje materijala i dovršavanje misija. Sustav pruža temelj za daljnje proširenje i razvoj, nudeći mogućnosti za dodavanje novih funkcionalnosti u skladu s potrebama igre.

## Literatura

1. Oracle APEX help center (managing pages), [link](#), datum pistupanja: 14.01.2024.
2. Oracle APEX forums (trigger on a multiple tables), [link](#), datum pistupanja: 15.01.2024.
3. Oracle APEX help center (creating triggers), [link](#), datum pistupanja: 15.01.2024.
4. Oracle APEX forums (function inside a trigger), [link](#), 15.01.2024.