**SVEUČILISTE U SPLITU**

**PRIRODOSLOVNO-MATEMATIČKI FAKULTET**

SEMINARSKI RAD

**INFORMATIČKI PROJEKT IZ BAZA PODATAKA**

**ZOOLOŠKI VRT**

Profesorica: Student:

Monika Mladenović Ivona Ercegovac

Split, 2022.

Sadržaj

[1 UVOD 1](#_Toc100341698)

[1.1 Općenito 1](#_Toc100341699)

[2 MODEL PODATAKA 2](#_Toc100341700)

[2.1 Opis projekta 2](#_Toc100341701)

[2.2 ENTITETI 3](#_Toc100341702)

[2.2.1 Zooloski 4](#_Toc100341703)

[2.2.2 Zaposlenik 4](#_Toc100341704)

[2.2.3 Posao 4](#_Toc100341705)

[2.2.4 Karta 4](#_Toc100341706)

[2.2.5 Vrsta\_Karte 4](#_Toc100341707)

[2.2.6 Mjesto 4](#_Toc100341708)

[2.2.7 Odjel 4](#_Toc100341709)

[2.2.8 Prostor 4](#_Toc100341710)

[2.2.9 Event 5](#_Toc100341711)

[2.2.10 Termin 5](#_Toc100341712)

[2.2.11 Organizacija 5](#_Toc100341713)

[2.2.12 Zivotinja 5](#_Toc100341714)

[2.2.13 Boraviste 5](#_Toc100341715)

[2.2.14 Vrsta 5](#_Toc100341716)

[2.2.15 Podvrsta 5](#_Toc100341717)

[2.2.16 Hrana 6](#_Toc100341718)

[2.2.17 Vrsta\_Hrane 6](#_Toc100341719)

[2.2.18 Hranjenje 6](#_Toc100341720)

[2.3 RELACIJE 7](#_Toc100341721)

[2.3.1 Relacija one-to-one 7](#_Toc100341722)

[2.3.2 Relacija one-to-many 7](#_Toc100341723)

[2.3.3 Relacija many-to-many 8](#_Toc100341724)

[2.4 KONCEPTUALNI MODEL 9](#_Toc100341725)

[2.5 LOGIČKI I RELACIJSKI MODELI 11](#_Toc100341726)

[2.6 DDL 14](#_Toc100341727)

[3 BAZA PODATAKA 17](#_Toc100341728)

[3.1 IZRADA KORISNIKA I DODJELA PRAVA 17](#_Toc100341729)

[3.2 IZRADA VEZE NA NOVOG KORISNIKA 18](#_Toc100341730)

[3.3 IZVRŠAVANJE DDL-A 19](#_Toc100341731)

[3.4 OSNOVNI UPITI: UNOS, AŽURIRANJE, BRISANJE PODATAKA 20](#_Toc100341732)

[3.4.1 UNOS PODATAKA (INSERT) 20](#_Toc100341733)

[3.4.2 AŽURIRANJE PODATAKA (UPDATE) 21](#_Toc100341734)

[3.4.3 BRISANJE PODATAKA (DELETE) 22](#_Toc100341735)

[3.5 SQL SELECT UPITI 23](#_Toc100341736)

[3.5.1 PRIMJER 1 23](#_Toc100341737)

[3.5.2 PRIMJER 2 24](#_Toc100341738)

[3.5.3 PRIMJER 3 25](#_Toc100341739)

[3.5.4 PRIMJER 4 26](#_Toc100341740)

[3.5.5 PRIMJER 5 27](#_Toc100341741)

# UVOD

## Općenito

Zoološki vrtovi su parkovi uređeni kako bi ljudi mogli promatrati životinje u donekle prirodnim uvjetima ne riskirajući pritom svoje živote. Sadržaj ovakvih parkova životinje su iz različitih krajeva svijeta, a uređenje je takvo da svaka životinja ima svoj prostor sličan onome u kojem je navikla živjeti. Svaki posjet zoološkom vrtu pruža nam zanimljivosti o životinjskom svijetu. Upravo zbog toga, najveći udio posjetitelja čine učenici i studenti, kojima su cijene ulaznica snižene. Često su organizirane atrakcije prilikom kojih gledatelji sudjeluju u svakodnevnici života određenih jedinki, primjerice javno hranjenje ili treniranje prilikom kojeg se životinje mogu vidjeti „u pokretu“.

# MODEL PODATAKA

## Opis projekta

Za ovaj projekt bilo je potrebno osmisliti i realizirati bazu podataka koja može poslužiti organizaciji zoološkog vrta. Baza treba sadržavati podatke o samom zoološkom vrtu (naziv), mjestu gdje se nalazi (grad, država, poštanski broj), te organizaciji čitavog parka. Dakle, baza podataka mora omogućiti vođenje evidencije o svim odjelima i ograđenim prostorima gdje životinje borave, koji su zabilježeni određenom oznakom. Potrebno je ostvariti evidenciju svih vrsta i podvrsta životinja, potom i samih životinja pojedinačno. Bitno je naglasiti da je sadržaj prostora promjenjiv, odnosno da životinje mogu mijenjati svoje boravište, prema tome u bazi podataka može doći do promjene određenih informacija. Također se treba voditi računa o hrani namijenjenoj tim životinjama (tko jede kakvu hranu, i u kojim količinama). Trebaju se spremati podatci o zaposlenicima zoološkog vrta (ime, prezime, datum rođenja, OIB, zvanje), te njihovom datumu zaposlenja i prestanka rada. Treba obratiti pažnju na promjenjivost ovih podatka, obzirom da jedan zaposlenik može raditi u više zooloških vrtova, te može biti trenutni ili bivši zaposlenik zabilježen u sustavu. Zaposlenici izdaju posjetiteljima karte čije cijene ovise o dobi ili statusu samoga posjetitelja. Prilikom posjeta zoološkom vrtu, gosta se informira o organiziranim atrakcijama koje se izvode na specifičnim prostorima u određeno vrijeme.

Do ovakve baze podataka, doći ćemo postepeno. Počevši od objašnjenja svih entiteta i relacija, sastavit ćemo konceptualni model koji će nam poslužiti pri konstruiranju prvo logičkog modela, a potom i relacijskog modela baze podataka.

## ENTITETI

Entitet je tablica svih podataka o nekom objektu kojeg želimo pohraniti u bazu podataka kako bi bila smislena i funkcionalna. Tijekom izrade baze podataka o zoološkim vrtovima, koristila sam ove entitete:

1. Zooloski
2. Zaposlenik
3. Posao
4. Karta
5. Vrsta\_Karte
6. Mjesto
7. Odjel
8. Prostor
9. Event
10. Termin
11. Organizacija
12. Zivotinja
13. Boraviste
14. Vrsta
15. Podvrsta
16. Hrana
17. Vrsta\_Hrane
18. Hranjenje

### Zooloski

Ovaj entitet namijenjen je za pohranu podataka o **nazivu** određenog zoološkog vrta, kako bi znali o kojem se parku radi obzirom da ćemo ih u ovoj bazi imati više.

### Zaposlenik

U ovom entitetu pohranjeni su podatci svih zaposlenika koji su radili ili rade u zoološkom vrtu. Dobivamo podatke o njihovim **imenima** i **prezimenima**, **datumu rođenja**, **OIB-u** te poslu za kojeg su zaduženi, odnosno njihovo **zvanje**.

### Posao

Pojedinci su zaposleni u zoološkom vrtu **od određenog datuma**, a ako nemaju više posao u navedenom parku, određen je i **datum prestanka rada**. Također je bitno napomenuti da se ista osoba može više puta zaposliti u istom ili različitim zoološkim vrtovima. Ovaj nam entitet omogućuje ažurno vođenje evidencije svih zaposlenika od nastanka zoološkog vrta do današnjeg datuma.

### Karta

Svaki posjetitelj dobije kartu koja ima ulogu računa prilikom kupnje. Prema tome, svaka kupljena karta je jedinstvena, što znači da ima svoju jedinstvenu **šifru**, a naravno i određenu **cijenu**.

### Vrsta\_Karte

**Vrsta** karte ovisi o dobi ili statusu posjetitelja, što je itekako bitan podatak jer različite vrste karti imaju različitu cijenu. Naprimjer, studenti plaćaju manji iznos od adolescenata koji ne studiraju.

### Mjesto

Ovaj entitet informira nas o lokaciji zoološkog vrta. Spremljeni su podatci **grada** i njegovog **poštanskog broja**. Također je pohranjen naziv **države** u kojoj se navedeni grad nalazi.

### Odjel

Svaki zoološki vrt podijeljen je na više odjela kako bi se ljudi u njemu lakše snalazili. Ti odjeli imaju svoj naziv, odnosno **oznaku** prema kojoj se prepoznaju.

### Prostor

Svaki odjel sadržava nekoliko ograđenih prostora rezerviranih za životinje. Kako bi se lako razlikovali, također su im pridodane **oznake**.

### Event

**Event** ili događaj, organizirana je atrakcija namijenjena za posjetitelje kako bi mogli gledati životinje „u pokretu“. Pohranjen je naziv raznih događaja koji se odnose na hranjenje, treniranje (obuku), higijensko održavanje životinja, te igranje s njima.

### Termin

Ovaj entitet pruža nam informacije o **vremenu** i **trajanju** organiziranog događaja.

### Organizacija

Entitet Organizacija povezuje podatke o prostoru, event-u i terminu, s ciljem da gost dobije povratnu informaciju sa svim potrebnim podatcima o događaju kojem želi prisustvovati.

### Zivotinja

U ovom entitetu navedeni su potrebni detalji o pojedinoj životinji. Dobivamo njen **naziv**, ali isto tako i **šifru** s obzirom da je svaka životinja jedinstvena. U zoološkom se vrtu primjerice mogu nalaziti dvije pande čiji je naziv isti, ali svaka ima svoju šifru.

### Boraviste

Svaka životinja nalazi se u svom prostoru, međutim kroz vrijeme može doći do izmjene boravišta što zahtjeva mogućnost ažuriranja tih podataka. Također je bitno napomenuti da se više životinja, čak različite vrste, može nalaziti u istom prostoru. Upravo zbog toga, potreban je entitet koji će sve te podatke povezati i voditi računa o **datumima** boravljenja pojedine životinje u određenom prostoru.

### Vrsta

Vrsta je skupina životinja koje imaju ista ili makar slična obilježja. To znači da najčešće imaju sličnu građu tijela, način razmnožavanja i hranjenja… općenito sličan način življenja. Poznate vrste su sisavci, ptice, ribe, rakovi, gmazovi, vodozemci, mekušci, paučnjaci i kukci. Dobivamo informaciju o **nazivu** vrste određene životinje.

### Podvrsta

Podvrsta je grupa istovrsnih životinja koje se međusobno razlikuju ali svejedno mogu boraviti zajedno, razmnožavati se, hraniti se istom hranom i slično. Ovim entitetom imamo zabilježen **naziv** podvrste. Naprimjer, divlje mačke su podvrsta sisavaca, a unutar te podvrste nalaze se pojedine životinje poput lavova, tigrova, snježnih leoparda itd.

### Hrana

Ovaj entitet vodi računa o **hrani** (namijenjenoj za životinje) koju određeni zoološki vrt posjeduje. U ovom slučaju, hrana obuhvaća okvirne primjere poput mesa, ribe, povrća, voća, orašastih plodova i slično.

### Vrsta\_Hrane

**Vrsta** hrane bitan je podatak zato što određenim životinjama koje jedu istu hranu odgovaraju različite namirnice te hrane. Naprimjer, tuljani i medvjedi jedu ribu, međutim tuljani se hrane tunom, a medvjedi lososom.

### Hranjenje

Ovim entitetom povezujemo sadržaj vrste hrane i podvrste životinje. Vodi se evidencija o hranjenju svih životinja – što koja podvrsta jede i koju **količinu** te hrane zahtjeva.

## RELACIJE

Relacija predstavlja odnos između dva entiteta. Postoje tri vrste relacija:

1. One-to-one (jedan na jedan) 1:1
2. One-to-many (jedan na više) 1:n
3. Many-to-many (više na više) m:n

### Relacija one-to-one

Svaki element jednog skupa može, ali i ne mora biti povezan samo s jednim elementom drugog skupa. Ovakvih relacija u ovoj bazi podataka nema.

### Relacija one-to-many

Relacija jedan na više funkcionira tako da jedan element može pripadati više drugih elemenata, ali ti drugi elementi mogu pripadati samo jednom elementu. Ova je relacija najčešća opcija u većini baza podataka, uključujući i ovu.

Jedno mjesto, odnosno grad, može sadržavati više zooloških vrtova. Međutim, jedan zoološki vrt može se nalaziti u samo jednom gradu.

Zaposlenici mogu dati posjetiteljima više karata, no jedna karta može biti dodijeljena od strane samo jednog zaposlenika. Ta karta funkcionira kao račun, tako da određena vrsta karte može se nalaziti na više računa, dok jedan račun, odnosno karta, može biti sklopljena na temelju samo jedne vrste karte. Drugim riječima, u posjet zoološkom vrtu može doći pet studenata, ili umirovljenika, koji će dobiti pet karata snižene cijene, jer je vrsta njihove karte studentska/umirovljenička. Međutim, zaposlenik ne može izdati kartu koja je i studentska i umirovljenička.

Jedan zoološki vrt može biti podijeljen na više odjela, a jedan od tih odjela nalazi se samo u navedenom zoološkom vrtu. Taj odjel može sadržavati nekolicinu ograđenih prostora u kojima životinje borave, ali taj prostor se nalazi u samo jednom odjelu.

Jedna vrsta životinje podijeljena je na brojne podvrste, no jedna od tih podvrsta pripada isključivo navedenoj vrsti. Ta podvrsta dijeli se na više različitih životinja, dok životinje mogu pripadati samo jednoj podvrsti.

Hrana se može podijeliti na više vrsta hrane. Uzmimo u primjer povrće kao hranu – dijeli se na salatu, rajčice, tikvice, krumpire i slično. S druge strane, u vrstu hrane ne spada sva preostala hrana, stoga imamo relaciju jedan na više.

### Relacija many-to-many

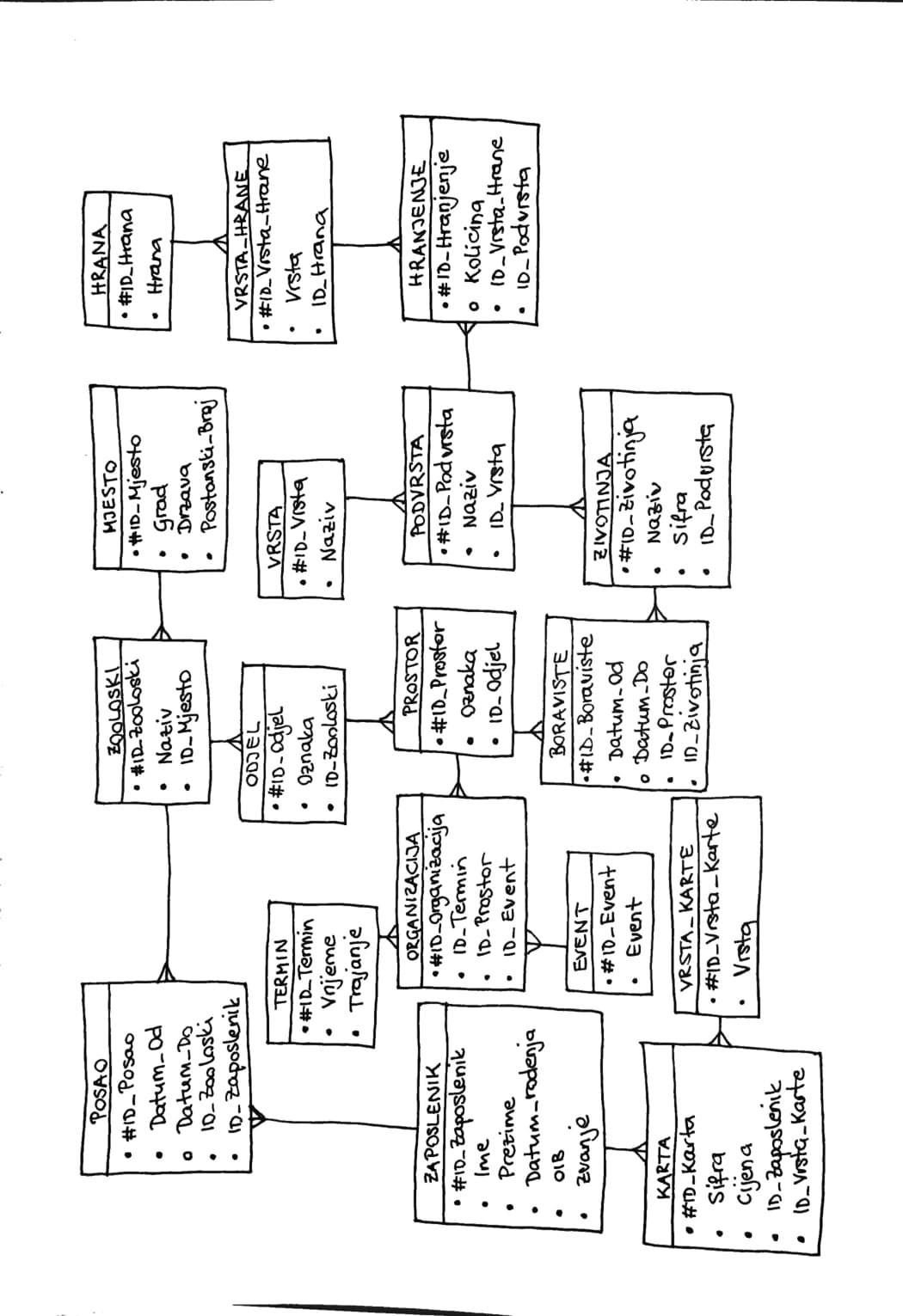
Ova relacija kaže da jedan entitet može imati više elemenata iz drugog entiteta, i suprotno. Takva veza između dva entiteta nije poželjna, pa se rješava na način da se uvodi novi entitet koji preuzima primarne ključeve tablica koje spaja.

U ovoj bazi podataka, relacija many-to-many pojavila se četiri puta:

1. Između entiteta „ZOOLOSKI“ i „ZAPOSLENIK“ postavila sam novi entitet „POSAO“ zato što jedan zaposlenik može kroz vrijeme raditi u više različitih zooloških vrtova, ili više puta u istom. Također, jedan zoološki vrt može imati više od samo jednog zaposlenika. Navedena tablica sadržava atribute početnog i završnog datuma zaposlenja, s tim da je završni datum neobavezan podatak jer ne mora biti određen u svakom trenutku. Ovom solucijom osigurala sam mogućnost ažurnih podataka o svim zaposlenicima počevši od nastanka samog zoološkog vrta.
2. Entitet „ORGANIZACIJA“ stavila sam između entiteta „PROSTOR“, „EVENT“ i „TERMIN“. Jedan događaj može se organizirati u više različitih prostora kroz duži period vremena, te se u jednom prostoru mogu prirediti različiti događaji. Bitno je i naglasiti da se događaji ne organiziraju uvijek u isto vrijeme s istim trajanjem, prema tome entitet „TERMIN“ je trebao također biti povezan s novim entitetom.
3. Između entiteta „ZIVOTINJA“ i „PROSTOR“ nalazi se entitet „BORAVISTE“ iz istih razloga kao u prvom primjeru s zaposlenicima. Dakle, jedna životinja s vremenom može mijenjati prostore u kojima boravi, dok je jedan prostor gotovo uvijek namijenjen za boravak više životinja zajedno. Ovaj entitet nam također pruža uvid u atribute početnih i završnih datuma useljenja životinje u određeni prostor.
4. „HRANJENJE“ je entitet koji sam postavila između entiteta „PODVRSTA“ i „VRSTA\_HRANE“ zato što jedna podvrsta životinje može jesti više različitih vrsta hrane, te jedna vrsta hrane može biti pojedena od strane više različitih vrsta životinja. Sadržaj ovog entiteta je atribut koji nas informira o **količini** hrane koja se izdvaja za određenu podvrstu životinje – neobavezan podatak zbog fleksibilne vrijednosti koja nije u svakom trenutku zabilježena.

## KONCEPTUALNI MODEL

Konceptualni model predstavlja nam početnu ideju kako bi baza podataka trebala izgledati. Takav opis strukture omogućuje nam olakšanu simulaciju teme koju sam model predstavlja. Na sljedećoj slici imamo shemu entiteta međusobno povezanih relacijama, uključujući u opis sve ključeve i atribute, te njihovu obaveznost. Sadržaj koji je neobavezan ne mora, ali i može, u svakom mogućem trenutku biti unesen u bazu podataka, za razliku od obaveznih podataka koji uvijek moraju biti uneseni kako bi baza podataka ispravno funkcionirala.

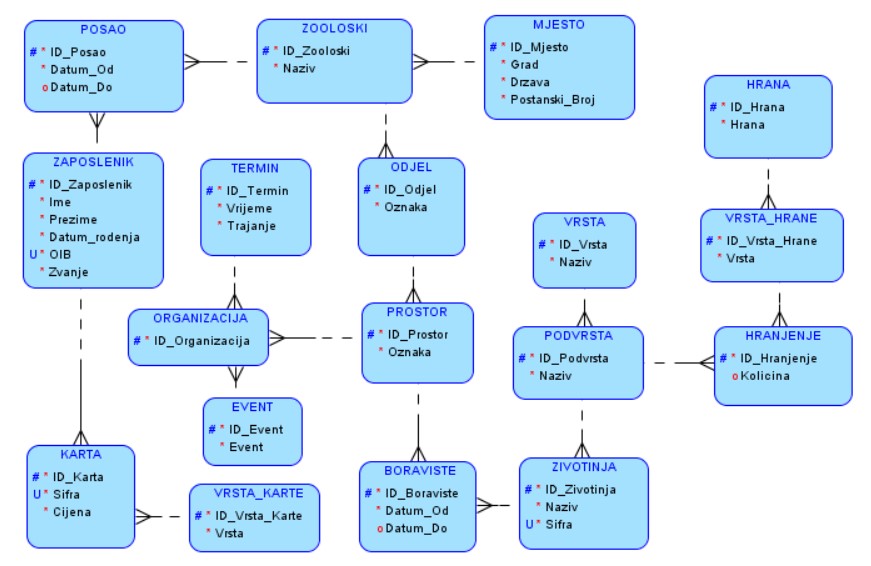


Slika 1: Konceptualni model

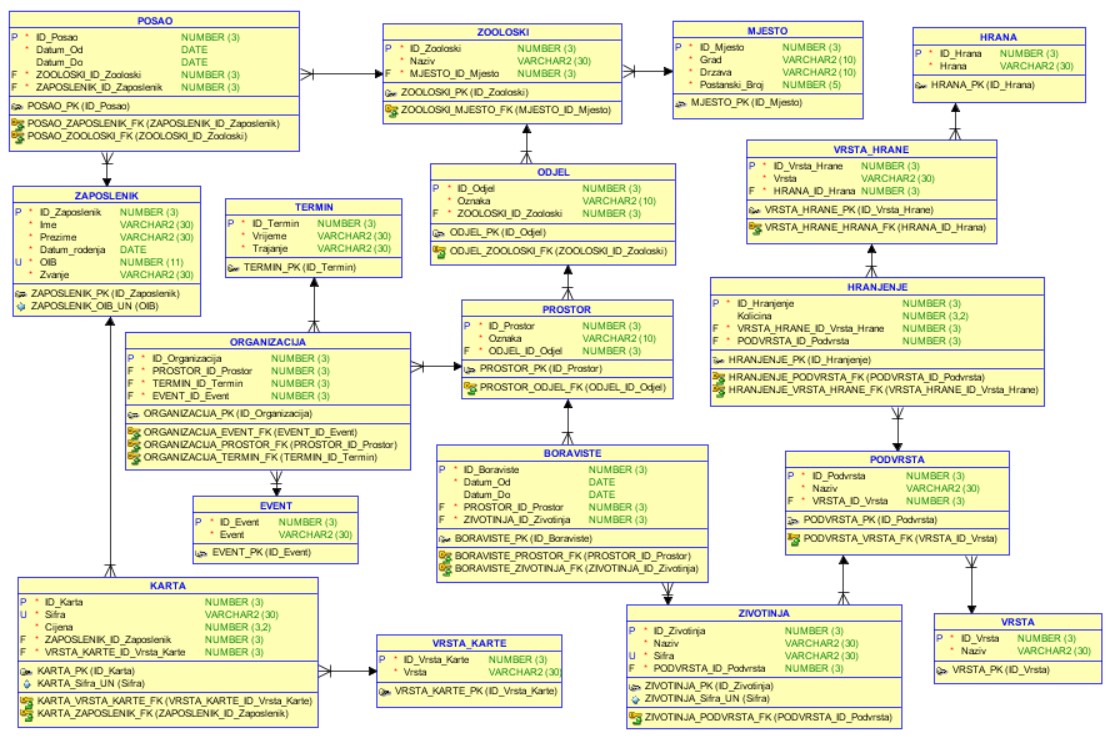
## LOGIČKI I RELACIJSKI MODELI

Logički model baze podataka je preslik konceptualnog modela u Datamodeler-u, ali bez naznačenih stranih ključeva. Dakle, predstavlja nam sve entitete međusobno povezane određenim vrstama relacija. Navedeni entiteti sadržavaju svoje primarne ključeve, te potrebne atribute čija je obaveznost istaknuta, kao i jedinstvenost podatka.

Nakon završetka logičkog modela generira se relacijski model baze podataka. Iz ovog modela osim entiteta, njihovih atributa te relacija koje ih međusobno povezuju, dobivamo uvid u još više informacija kao što su strani ključevi, te ograničenja i tipovi podataka svih atributa.



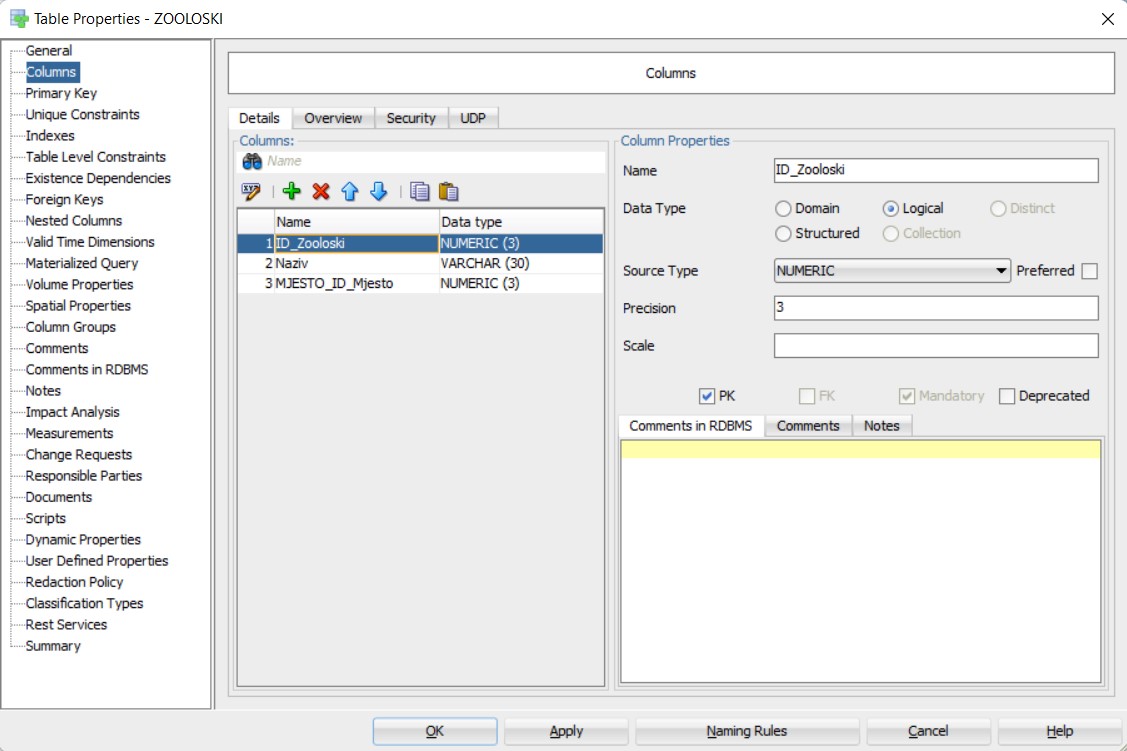
Slika 2: Logički model



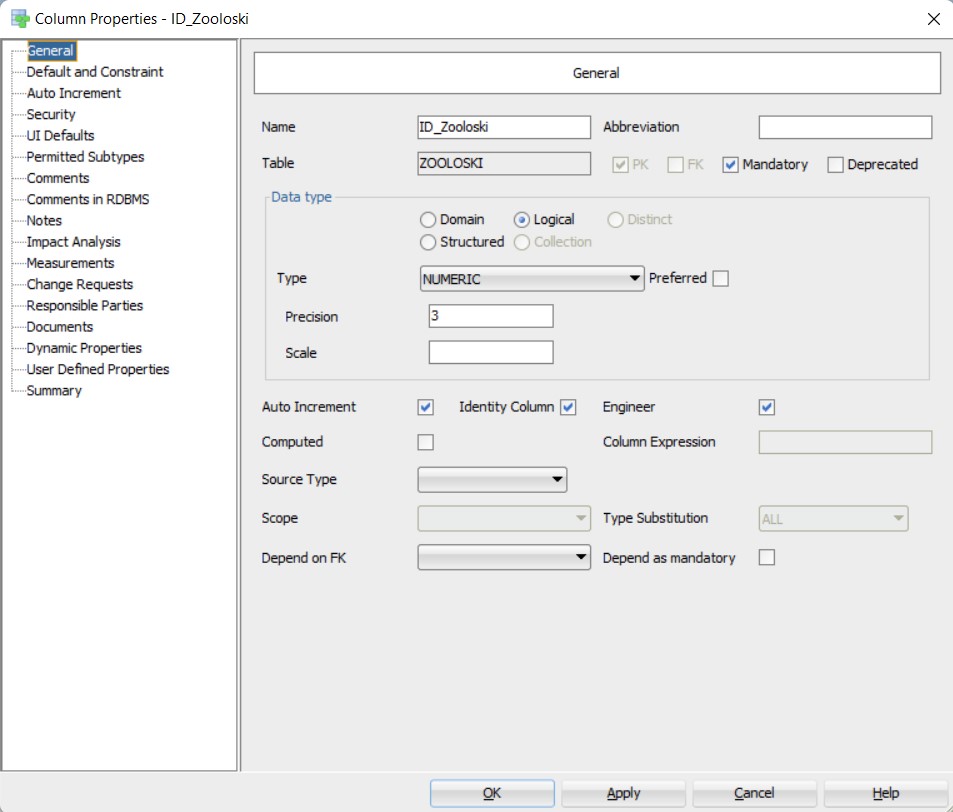
Slika 3: Relacijski model

## DDL

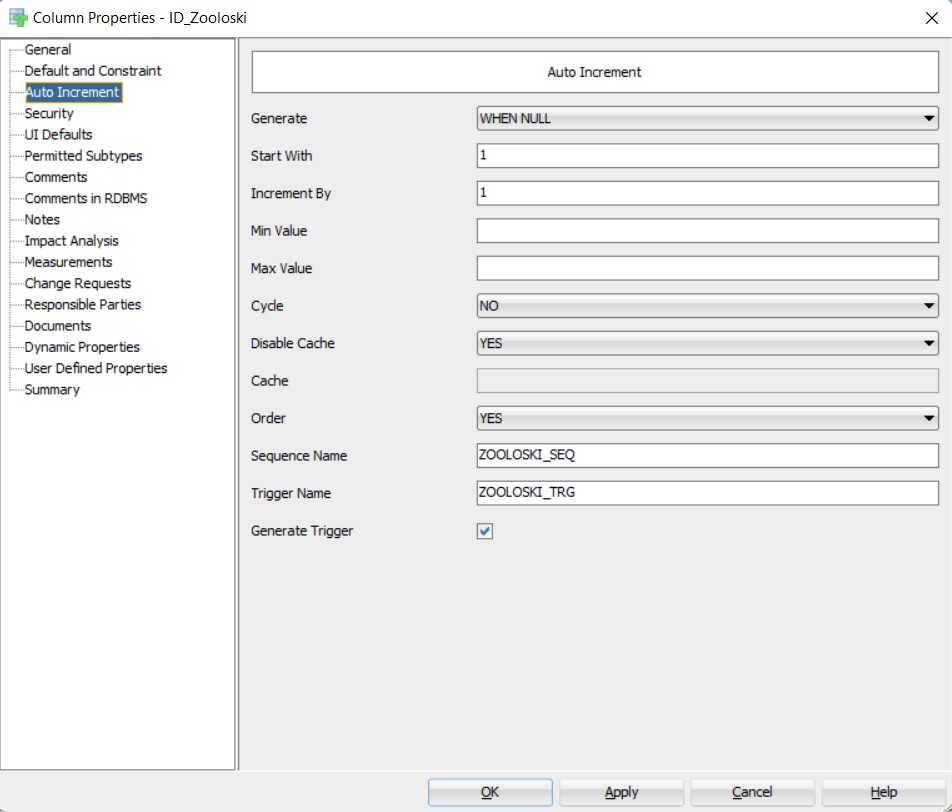
Nakon kreiranja logičkog i relacijskog modela baze podataka potrebno je izraditi DDL kojeg ćemo izvršiti u SQL Developer-u kako bismo tamo mogli unositi, ažurirati ili uklanjati podatke, te izvršavati razne upite. Obzirom da u Oracle-u ne postoji ugrađeni tip podataka za primarne ključeve, prije izrade DDL-a potrebno je svim primarnim ključevima napraviti sekvencu (*sequence*) i okidač (*trigger*).



Slika 4: Dvoklik na primarni ključ entiteta

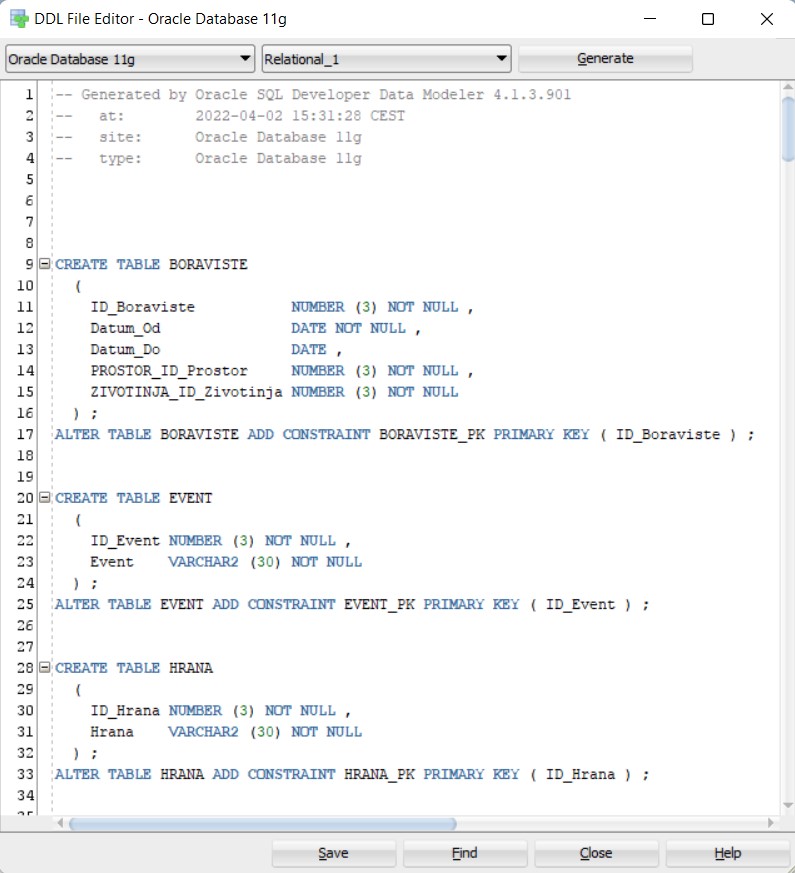


Slika 5: Odabiremo "Auto Increment" i "Identity Column"



Slika 6: Izrada sekvence i okidača

Nakon što smo napravili sve sekvence i okidače u relacijskom modelu, možemo izraditi DDL. Klikom na ikonu *Generate DDL* u SQL Modeler-u ostvarujemo DDL koji nam predstavlja kod svih kreiranih entiteta, ključeva i atributa, te sekvenci i okidača baze podataka.



Slika 7: Prikaz DDL-a

# BAZA PODATAKA

Baza podataka je skup međusobno povezanih podataka pohranjenih u memoriji računala. Nakon sustavnog prikupljanja tih podataka, baza nam omogućuje jednostavno upravljanje njima. Upravljanje podrazumijeva unos, ažuriranje, te brisanje navedenog sadržaja.

## IZRADA KORISNIKA I DODJELA PRAVA

Izrada DDL-a omogućuje nam nastavak rada s bazom u Oracle Developer-u, međutim kako bismo mogli pokrenuti kod, moramo prvo uspostaviti novog korisnika te mu pridodati prava. Za početak otvorimo Oracle developer i napravimo navedenog korisnika unutar konekcije „SYSTEM“ koju smo ostvarili prilikom instalacije samoga programa. Naredbe ćemo pisati u *Worksheet*-u već postojećeg korisnika.

U prvom koraku pridodali smo korisniku naziv (baze podataka), te ga identificirali lozinkom koju je iz očitih razloga najbolje postaviti istu kao i naziv. Drugim korakom pridodali smo korisniku određena prava koja će nam kasnije biti potrebna. Naposlijetku označimo naredbu koju želimo ostvariti i klikom na botun koji prikazuje zelenu strelicu, pokrenemo. Prikazat će se obavijesti da je korisnik kreiran te da su mu prava uspješno pridodana.

Slika na kojoj se prikazuje tekst

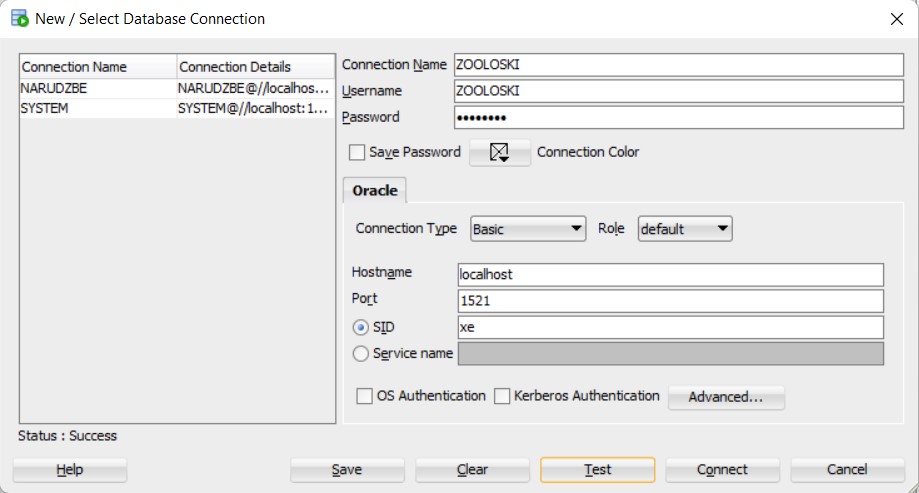
Opis je automatski generiran

Slika 8: Izrada korisnika i dodjela prava

## IZRADA VEZE NA NOVOG KORISNIKA

Nakon izrade korisnika možemo napraviti novu konekciju na njega. Klikom na ikonu zelenog plusa pojavljuje se prozor u kojem postavljamo sljedeće podatke.

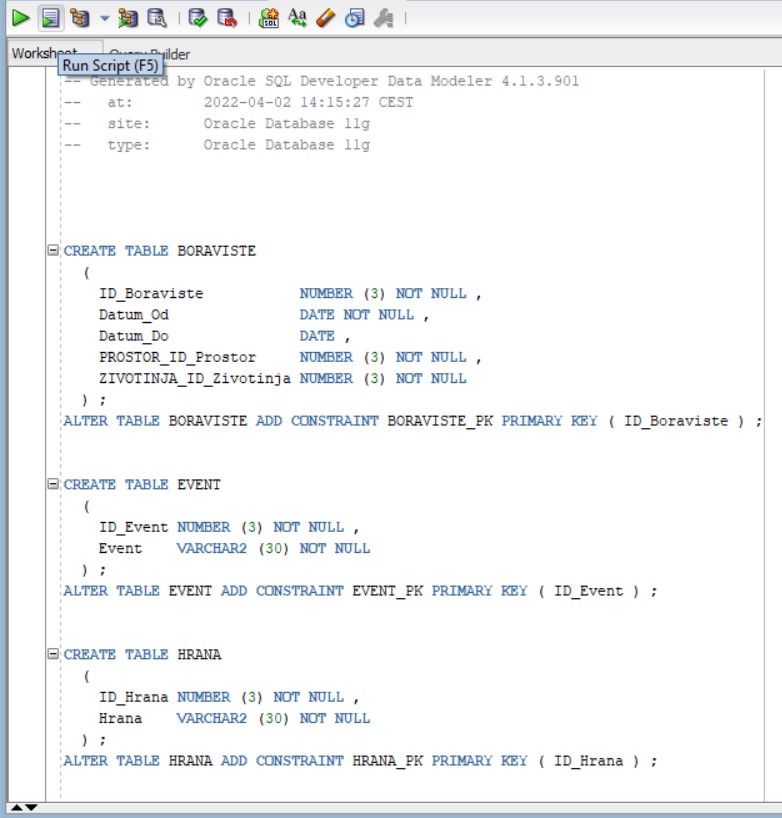
*Username* i *Password* moraju biti onakvi kakve smo ih postavili prilikom izrade korisnika, a *Connection* *Name* možemo staviti po volji, međutim preporučeno je odabrati naziv isti kao i naziv korisnika. Ako smo sve podatke unijeli ispravno, testiranje će proći uspješno te možemo spremiti vezu.



Slika 9: Izrada veze na novog korisnika

## IZVRŠAVANJE DDL-A

Sada će nam poslužiti .ddl datoteka koju smo generirali u DataModeler-u. Kopirat ćemo sav tekst iz te datoteke i zalijepiti ga u *Worksheet* unutar nove veze koju smo kreirali. Zatim pokrenemo čitav kod klikom na *Run Script* ili F5 na tipkovnici.



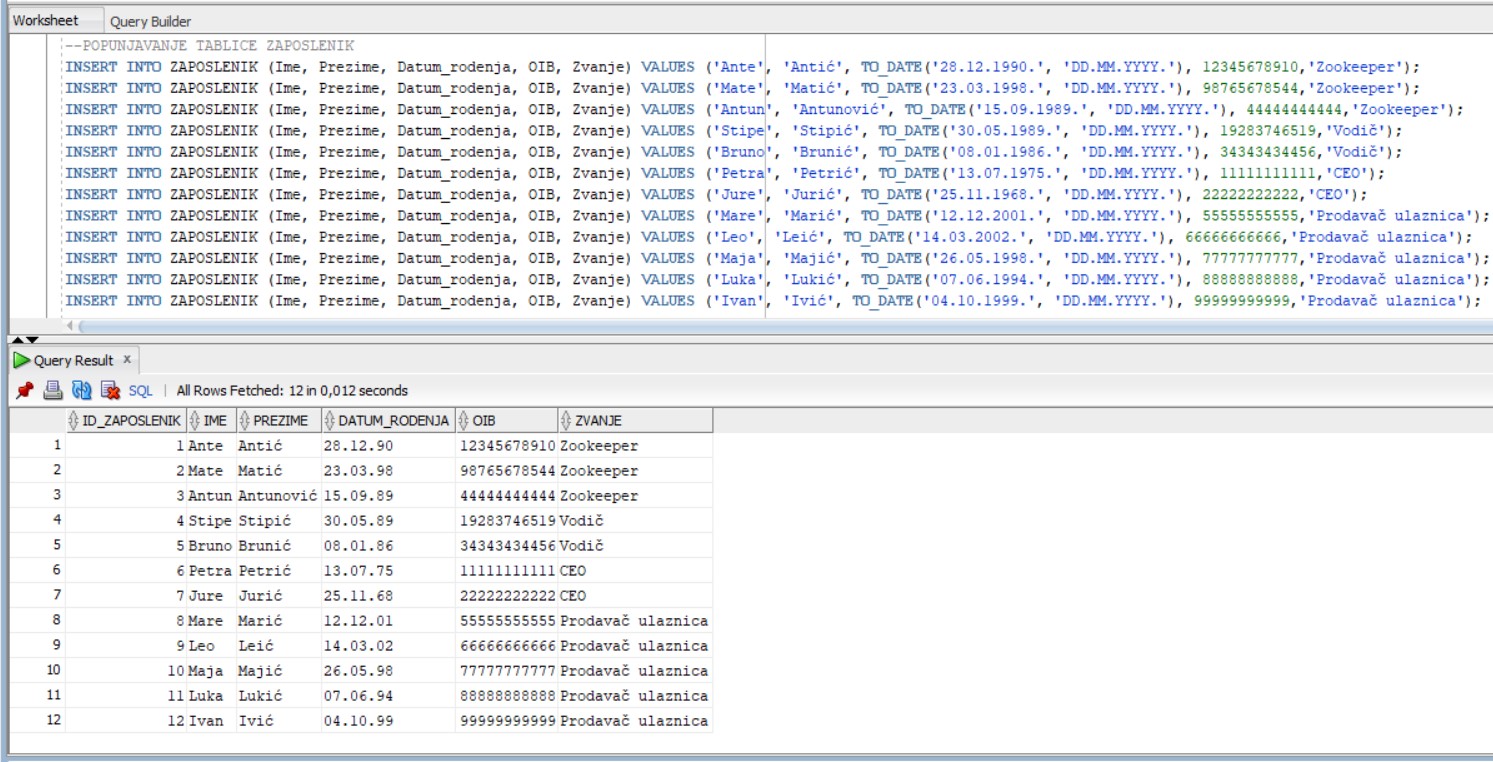
Slika : klik na Run Script nakon postavljanja koda u Worksheet

## OSNOVNI UPITI: UNOS, AŽURIRANJE, BRISANJE PODATAKA

### UNOS PODATAKA (INSERT)

„INSERT“ je naredba koja nam služi za unos podataka u određenu tablicu. Prilikom korištenja ove naredbe, potrebno je navesti ključnu riječ *into* kako bi naglasili gdje želimo unijeti novi podatak. Nakon toga ispisujemo i ključnu riječ *values* pomoću koje navedenom atributu pridodajemo određene vrijednosti.

Potrebno je krenuti od onih tablica koje ne sadržavaju nikakve strane ključeve. U tablicama koje ipak sadržavaju strane ključeve, njih unosimo kao broj bez navodnika.



Slika 11: Popunjavanje tablice ZAPOSLENIK

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Slika 12: Popunjavanje tablice ZOOLOSKI sa stranim ključem koje određuje mjesto zoološkog vrta

### AŽURIRANJE PODATAKA (UPDATE)

Naredbu „UPDATE“ primjenjujemo kako bismo izmijenili već postojeći podatak, odnosno da bismo ga ažurirali. S ovom naredbom dolazi uporaba ključnih riječi *set* i *where* koje specificiraju podatak kojeg izmjenjujemo.

Uzmimo za primjer zaposlenika kojem smo krivo unijeli ime prilikom unošenja podataka u tablicu. Uneseno je ime *Ivan* umjesto *Marko*.

Slika na kojoj se prikazuje stol

Opis je automatski generiran

Slika : Tablica ZAPOSLENIK prije ažuriranja krivo unesenog podatka

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Slika : Tablica ZAPOSLENIK nakon ažuriranja krivo unesenog podatka

### BRISANJE PODATAKA (DELETE)

Kako bi uklonili neželjeni podatak iz baze podataka, ispisujemo naredbu „DELETE“. Kao i kod unosa, moramo naglasiti o kojoj se tablici radi, međutim u ovom slučaju koristimo ključnu riječ *from*. Osim toga, ključnom riječi *where* treba navesti sadržaj kojeg namjeravamo izbrisati.

U bazu podataka dosada smo unijeli osam vrsta životinja od kojih koristimo samo šest. Kako bi obrisali nepotrebne podatke, primijenit ćemo navedenu naredbu.

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Slika 15: Brisanje podataka "Riba" i "Kukac" iz tablice VRSTA

U stupcu „ID\_VRSTA“ uočavamo da nedostaju brojevi tri i osam. To su upravo brojevi koji su prethodno označavali dva izbrisana podatka.

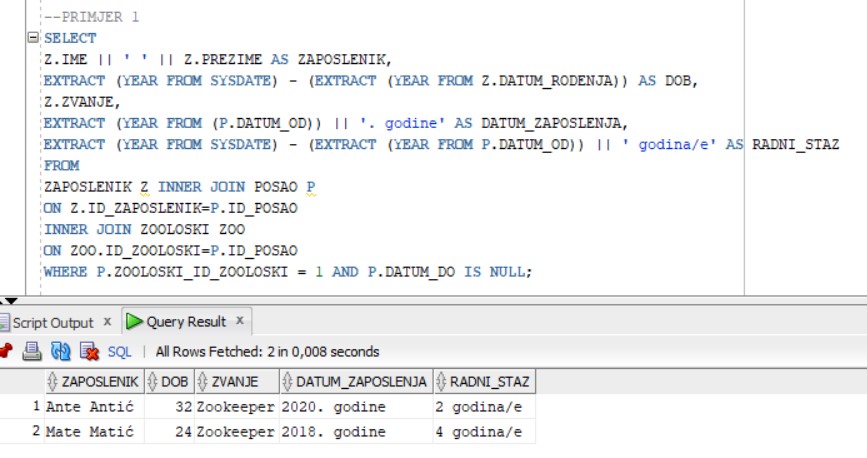
## SQL SELECT UPITI

Osim upravljanja podatcima, baza podataka omogućuje nam i korištenje SQL složenih Select upita. Uporaba „SELECT-a“ korisniku daje uvid u smisleno povezane podatke iz više tablica istovremeno. Dakle, to je mogućnost prikaza određenog skupa tablica sa specifično određenim podatcima koji se mogu prilagoditi po potrebi.

Naredbu započinjemo ključnom riječi *select*, zatim zapisujemo ono što želimo prikazati (ako želimo prikazati sve dostupne podatke, koristimo znak zvjezdice), te onda pomoću *from* navodimo tablice čiji nas podatci zanimaju. Kako bi povezali dvije ili više tablica koristimo *join*, te nakon toga izjednačavamo primarne sa stranim ključevima pomoću ključne riječi *on*.

### PRIMJER 1

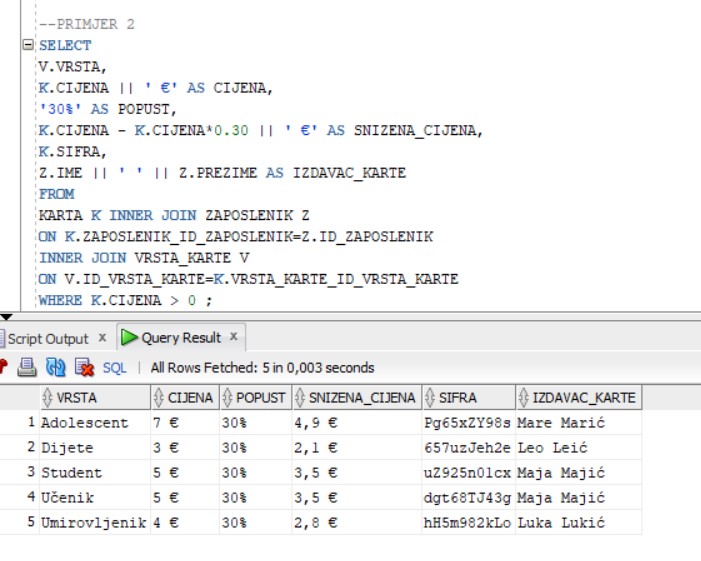
Potrebno je prikazati sve trenutne zaposlenike zoološkog vrta u Zagrebu. Treba se ispisati njihovo **ime i prezime, dob**, **zvanje**, **datum zaposlenja**, te **radni staž**. Nazivi stupaca moraju biti smisleni, kao i informacije u njima.



Slika : Primjer 1

### PRIMJER 2

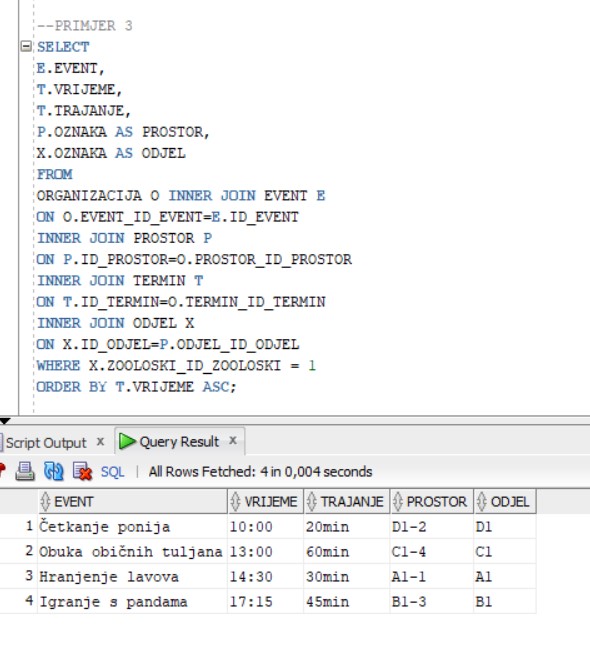
Treba napraviti upit kojim ćemo dobiti uvid u sve dosada kupljene karte, osim besplatnih. Svim kartama cijena treba biti snižena za 30%. Potrebno je ispisati **vrstu** karte, originalnu **cijenu**, **popust**, **sniženu cijenu**, jedinstvenu **šifru** karte, te **ime i prezime** zaposlenika koji ju je izdao. Stupce treba smisleno nazvati, te je potrebno dodati za valutu cijena znak eura.



Slika : Primjer 2

### PRIMJER 3

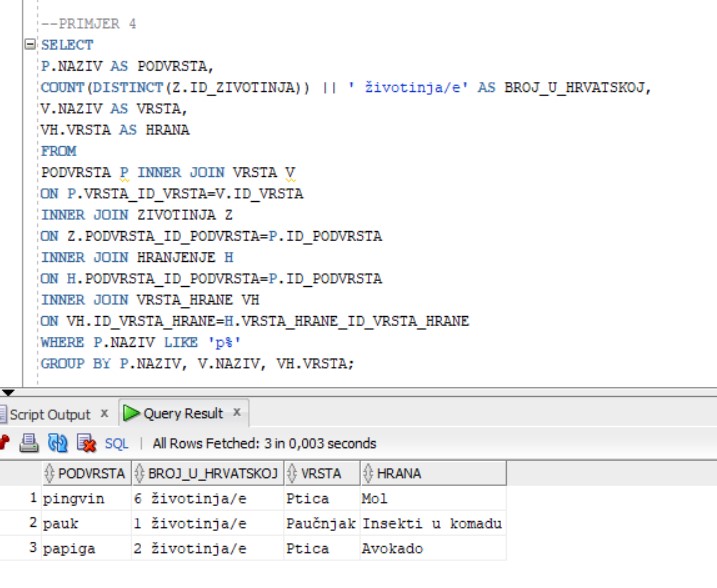
Ovim upitom izradit ćemo upute za sve event-ove dana zoološkog vrta u Zagrebu. Tablica se treba sastojati od **naziva događaja**, te **vremena** njegovog početka, **trajanja** i **prostora** u kojem se izvodi. Također je potrebno naglasiti u kojem se **odjelu** nalazi taj prostor radi lakšeg snalaženja. Event-ove treba ispisati po redu kojim se i događaju tijekom dana.



Slika : Primjer 3

### PRIMJER 4

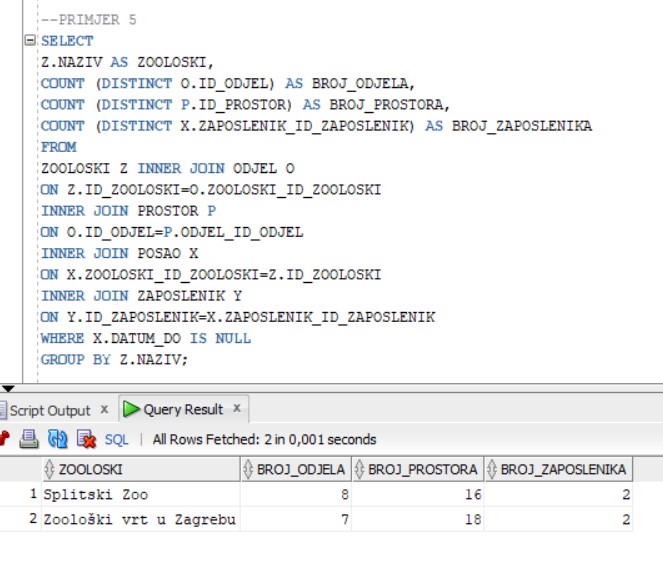
Potrebno je pronaći sve **podvrste** životinja čiji naziv počinje slovom P. Treba izračunati **koliko ima životinja** (navedenih podvrsta) zabilježenih u zoološkim vrtovima **u Hrvatskoj**. Ispisati još i **vrstu** kojoj pripadaju te **hranu** koju jedu.



Slika : Primjer 4

### PRIMJER 5

U ovom primjeru treba se izraditi „evidencija“ trenutno dostupne opreme **zooloških vrtova**. Dakle, treba izračunati **koliko** se **odjela** i **prostora** nalazi u svakom od njih, te **koliko** ima **zaposlenika koji su trenutno zaposleni.**



Slika : Primjer 5