

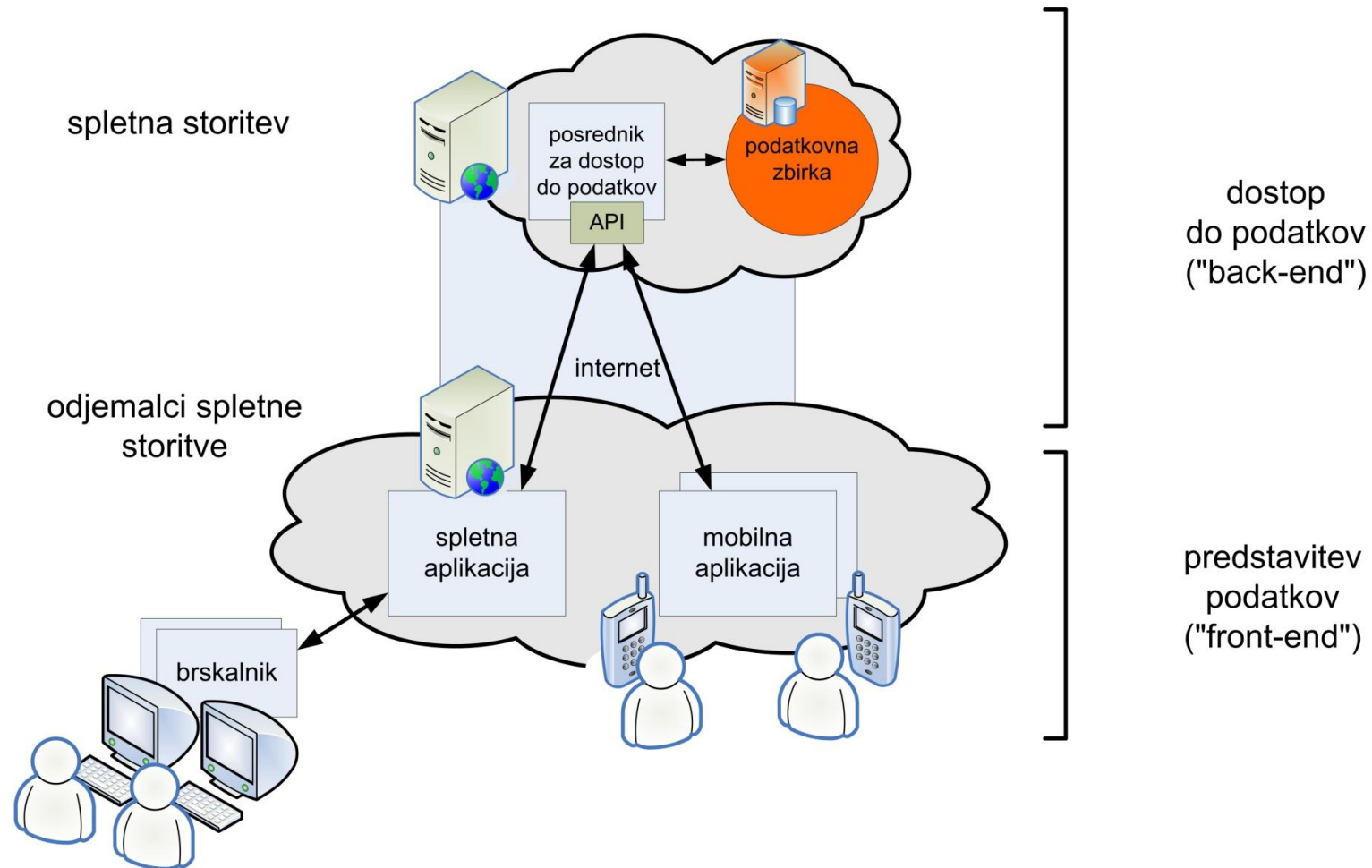


Podatki aplikacije

Danes: podatkovni model aplikacije

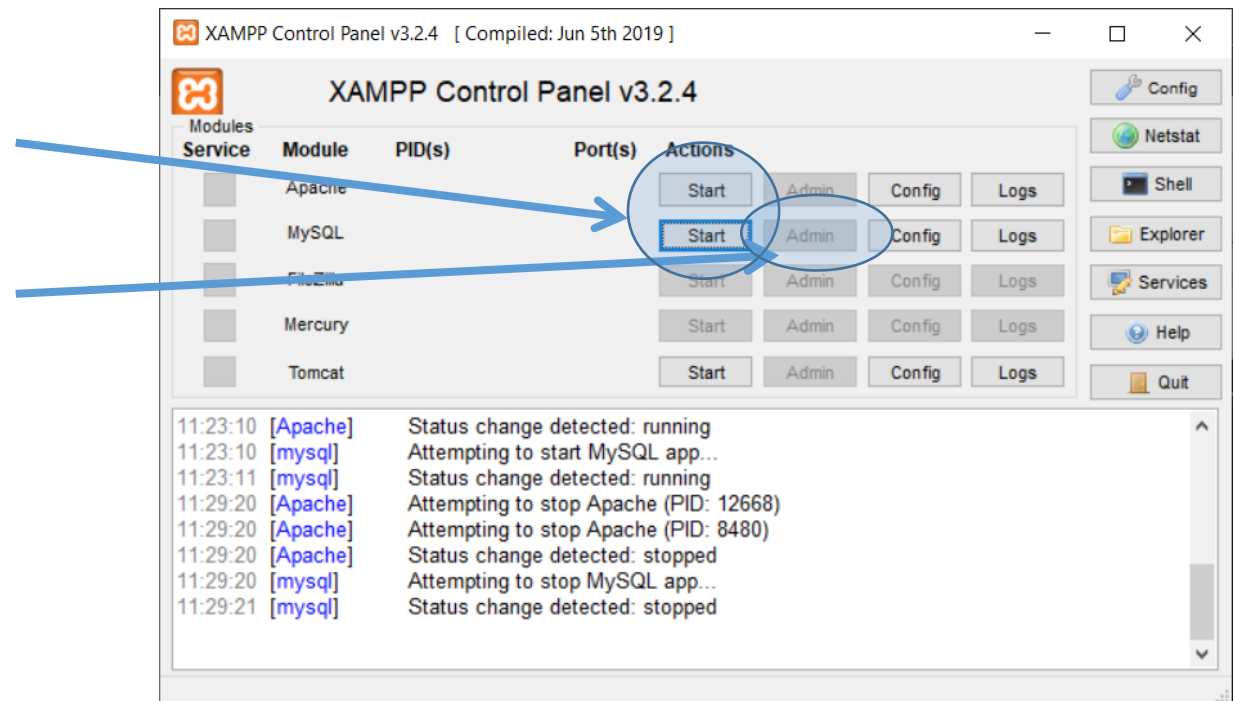
- Pripravili bomo vse potrebno za shranjevanje podatkov
 1. določitev podatkovnega modela
 2. izdelava zbirke podatkov
 3. pridobivanje in manipulacija s podatki s pomočjo poizvedbenega jezika SQL
- Naslednjič bomo dostopali do podatkov iz danes narejene zbirke prek spletnega strežnika

Podatki v arhitekturi sistema



Podatkovni strežnik

- Uporabili bomo podatkovni strežnik MySQL (MariaDB), ki je del paketa XAMPP
- [Namestitev paketa XAMPP](#)
- Zaženite nadzorno ploščo XAMPP-a, v njej pa še spletni strežnik Apache in podatkovni strežnik MySQL
- Poženite administratorski vmesnik podatkovnega strežnika MySQL



Izdelava zbirke podatkov

- V zavihku *Databases/Zbirke podatkov* ustvarite novo zbirko in jo poimenujte “igra”

Strežnik: 127.0.0.1

Zbirke podatkov SQL Stanje Uporabniški računi

Zbirke podatkov

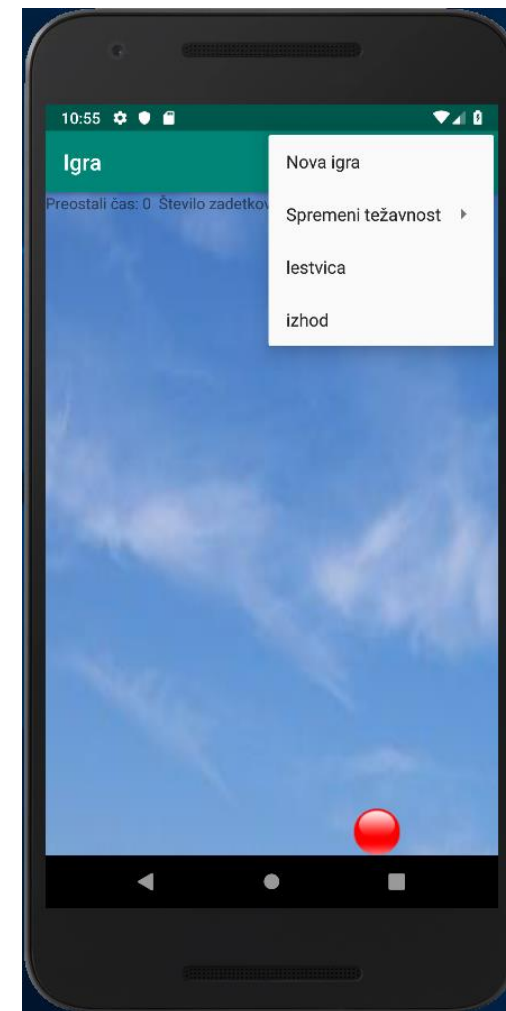
Ustvari zbirko podatkov ?

igra utf16_general_ci Ustvari

Zbirka podatkov	Pravilo za razvrščanje znakov	Dejanje
<input type="checkbox"/> test	latin1_swedish_ci	Preveri privilegije
<input type="checkbox"/> ohomvadmin	utf8_bin	Preveri privilegije

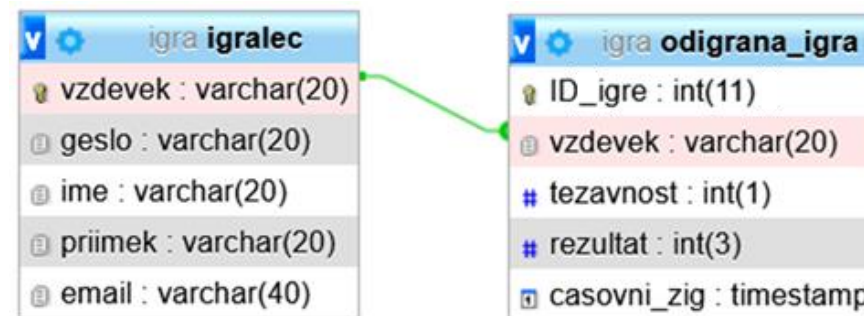
Podatkovni model

- Radi bi izdelali informacijski sistem za vodenje rezultatov mobilne igre, ki jo bomo sicer izdelali na zadnjih vajah
- Po zaslonu se z različnimi hitrostmi premika »balonček«.
- Naloga igralca je, da ga s prstom »poči«, s tem pa dobi točko.
- Igra ima omejen čas in težavnost (hitrost premikanja balončka)
- Podatkovni model naj ima dve entiteti: igralca in odigrano igro
- Vsak registriran igralec naj ima
 - svoj **vzdevek** (oz. uporabniško ime) in **geslo** za prijavo v igro
 - poleg tega pa še **ime**, **priimek** in **email** naslov
- Odigrana igra vsebuje podatke
 - **rezultat** igre - ovrednoten s številom točk, ki jih igralec v igri doseže
 - **težavnostno** stopnjo (1, 2, 3)
 - čas, ko je bila igra odigrana (**časovni žig**)



Udejanjanje podatkovnega modela

- V relacijskih podatkovnih zbirkah udejanjimo podatkovni model s pomočjo *tabel*
- Podatki v tabelah so zbrani v *stolpcih* (ang. fields) in *vrsticah* (ang. rows)
- Tabela ima natančno določeno število stolpcev in poljubno število vrstic
- Stolpci
 - predstavljajo lastnosti (atribute) entitete, predstavljene v tabeli
 - imajo določen podatkovni tip in omejen obseg vrednosti (lahko pa je vnaprej določen nabor možnih vrednosti)
- Vrstice v tabeli predstavljajo posamezne zapise, pri čemer je zapis sestavljen iz vrednosti lastnosti, ki so definirane v stolpcih
- Za udejanjanje našega podatkovnega modela bomo torej potrebovali dve tabeli s pripadajočimi stolpci:



Izdelava tabel

- Izberite pravkar ustvarjeno zbirko podatkov s seznama zbirk
- Ustvarite tabeli *igralec* in *odigrana_igra* z ustreznim številom stolpcev
- V naslednjem koraku stolpce poimenujte v skladu s podatkovnim modelom ter jim določite vrsto in dolžino

Server: 127.0.0.1 » Database: igra

Structure SQL Search Query Export

Create table

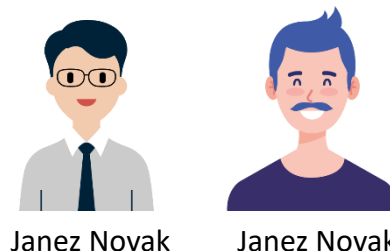
Name: igralec Number of columns: 5

Ime tabele: igralec Dodaj Ime tabele: odigrana_igra Dodaj 1 stolpec(-ce)

Ime	Vrsta	Dolžina/Vrednosti	Ime	Vrsta	Dolžina/Vrednosti	Privzeto
vzdevek	VARCHAR	20	ID_igre	INT		Brez
Izberite iz osrednjih stolpcev			Izberite iz osrednjih stolpcev			
geslo	VARCHAR	20	vzdevek	VARCHAR	20	Brez
Izberite iz osrednjih stolpcev			Izberite iz osrednjih stolpcev			
ime	VARCHAR	20	tezavnost	ENUM	'1','2','3'	Brez
Izberite iz osrednjih stolpcev			Izberite iz osrednjih stolpcev		Uredi vrednosti ENUM/SET	
priimek	VARCHAR	20	rezultat	INT	3	Brez
Izberite iz osrednjih stolpcev			Izberite iz osrednjih stolpcev			
email	VARCHAR	40	casovni_zig	TIMESTAMP		CURRENT_TIMESTAMP
Izberite iz osrednjih stolpcev			Izberite iz osrednjih stolpcev			

Zagotavljanje enoličnosti zapisov v tabeli

- Zgodi se lahko, da je vsebina v različnih vrsticah tabele identična, zato posameznih vrstic ne moremo ločiti med seboj, čeprav bi to bilo nujno potrebno
- Primera
 - zagotoviti moramo enolično identiteto igralcev (preprečiti podvojene vzdevke)
 - ločiti moramo med posameznimi odigranimi igrami istega igralca na isti težavnosti, ki so se končale z istim rezultatom
- Zapise (vrstice) med seboj razločujemo s pomočjo posebnega atributa, ki mu pravimo *enolični identifikator* (id) ali *primarni ključ* (ang. *primary key*)
 - vrednosti primarnega ključa se med seboj vedno razlikujejo, zato ta stolpec enolično razločuje med posameznimi zapisi (vrsticami) v tabeli



Primarni ključ

- Nalogo zagotavljanja enoličnosti lahko zaupamo sistemu za upravljanje z zbirkami podatkov
- V MySQL nastavimo stolpcu vlogo primarnega ključa (id-ja), tako da mu nastavimo lastnost *Index* na vrednost *PRIMARY*
- V tabeli *igralec* posebnega stolpca za id ne potrebujemo, saj lahko kot primarni ključ služi kar vzdevek
- V tabeli *odigrana_igra* naj bo primarni ključ poseben stolpec *IDigre*
- Sistemu lahko prepustimo tudi skrb za izbiro vrednosti primarnega ključa
 - ob vsakem novem zapisu vpiše v polje s primarnim ključem vrednost, ki je za 1 večja od vrednosti v predhodnem zapisu
 - v tabeli *odigrana_igra* pri *IDigre* izberite možnost *A_I* (AUTO INCREMENT)

Name	Type ?	Length/Values ?	Index	A_I
IDigre	INT		PRIMARY	<input checked="" type="checkbox"/>
vzdevek	VARCHAR	20	--	<input type="checkbox"/>
tezavnost	INT	1	--	<input type="checkbox"/>
rezultat	INT	3	--	<input type="checkbox"/>

Povezave med tabelami (tuji ključi)

- Podatki v različnih tabelah so navadno med seboj na nek način povezani (imamo relacijsko zbirko!)
- Odigrana igra, na primer, pripada določenemu igralcu, ta pa je definiran v tabeli *igralci*
 - tabeli *odigrana_igra* in *igralec* torej nista popolnoma neodvisni, ampak sta medsebojno povezani prek lastnosti (oz. stolpca) *vzdevek*
 - *vzdevek* je primarni ključ v tabeli *igralec* in *tuji ključ* (ang. *foreign key*) v tabeli *odigrana_igra*



Povezave med tabelami (tuji ključi)

- Izberemo tabelo *odigrana_igra*, zavihek *Structure/Struktura* in povezavo *Relation view/Pogled relacij*
- Stolpec *vzdevek* povežemo z istoimenskim stolpcem iz tabele *igralec* in izberemo možnost *RESTRICT*, ki vsiljuje ohranjanje strukturne integritete
 - npr. preprečuje dodajanje nove igre z vzdevkom igralca, ki v tabeli *igralci* ne obstaja
 - preprečuje brisanje (ali spreminjanje) vzdevka igralca iz tabele *igralci*, če ima kakšno odigrano igro

Strežnik: 127.0.0.1 » Zbirka podatkov: test » Tabela: odigrana_igra

Prebrskaj Struktura SQL Izkazje Vstavi Izvozi Uvozi Privilegiji Operacije Sledenje Sprožilci

Zgradi novo tabelo Pogled relacij

Omejitve tujih ključev

Dejanja	Lastnosti omejitev	Stolpec	Omejitev tujih ključev (INNODB)		
			Zbirka podatkov	Tabela	Stolpec
Zavrzi	odigrana_igra_ibfk_1 ON DELETE RESTRICT ON UPDATE RESTRICT	vzdevek	test	igralec	vzdevek

+ Dodaj stolpec



Naloga: testiranje

- Izberite zavihek *Insert/Vstavi* in v obe tabeli dodajte nekaj zapisov
- Preizkusite, če zagotavljanje enoličnosti zapisov in njihove strukturne integritete deluje

Izvoz podatkovne zbirke

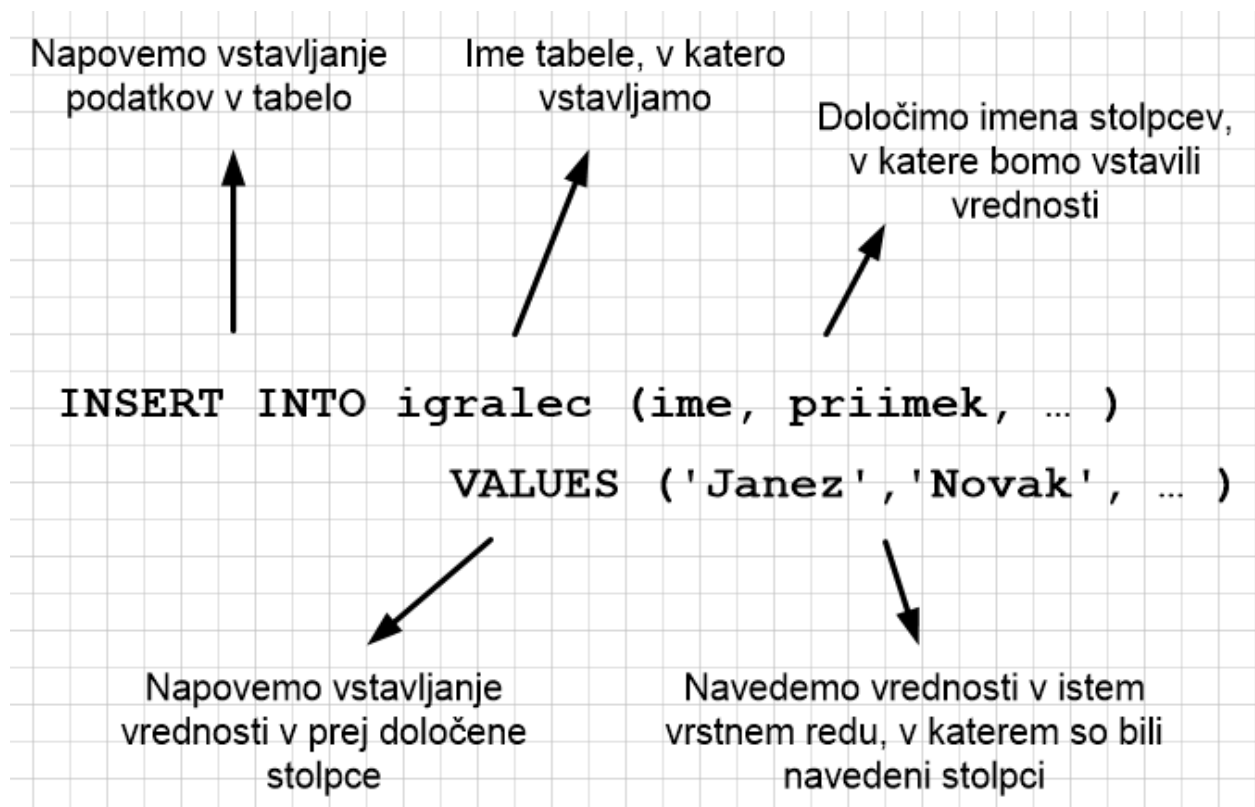
- Podatkovno zbirko (strukturo in podatke) lahko izvozimo in preselimo v drug sistem upravljanja s podatkovnimi zbirkami
 - izberite želeno podatkovno zbirko
 - izberite zavihek *Export/Izvozi*
 - izberite način izvoza *Po meri*
 - izberite obliko "SQL"
 - pri možnostih ustvarjanja objektov izberite *Dodaj izjavo CREATE DATABASE / USE*
 - shranite izdelano datoteko SQL na želeno mesto
- Datoteko SQL lahko uvozite v sistem upravljanja s podatkovnimi zbirkami s pomočjo zavihka *Import/Vstavi*

Delo s podatki

Jezik SQL

- Jezik SQL (Structured Query Language) omogoča pridobivanje podatkov iz podatkovne zbirke in vstavljanje novih podatkov vanjo
- Najpogostejši ukazi s podatki v zbirki
 - SELECT – pridobivanje podatkov
 - UPDATE – posodabljanje podatkov
 - DELETE – brisanje podatkov
 - INSERT INTO – vstavljanje podatkov

Vpis v podatkovno zbirko



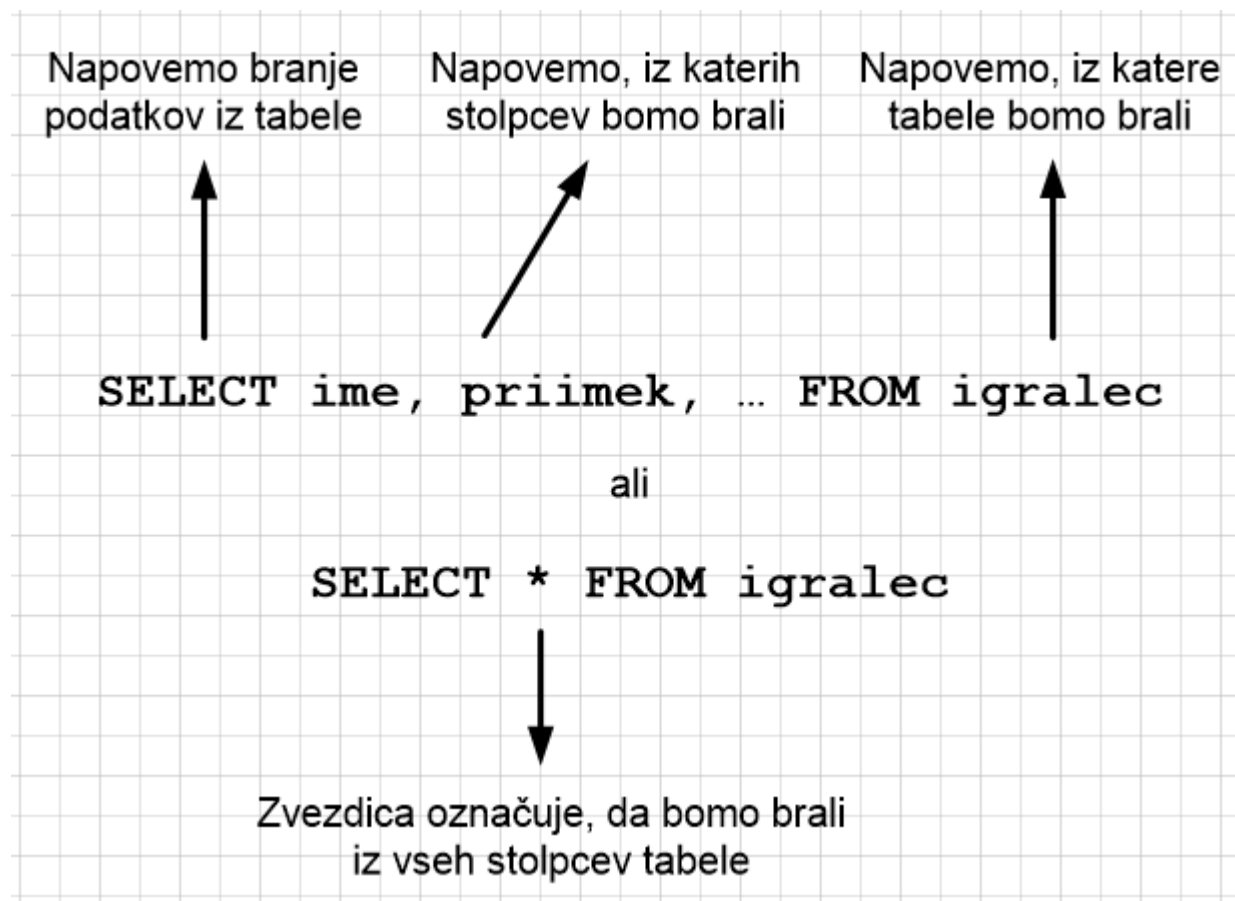
Znakovne nize v SQL poizvedbi zapisujemo v enojnih narekovajih!



Zapišite poizvedbi, ki v zbirko dodata novega igralca in odigrano igro. Poizvedbi preizkusite v zavihku *SQL*.



Branje iz podatkovne zbirke



Zapišite in preizkusite poizvedbo, ki iz zbirke pridobi vse podatke o vseh registriranih igralcih (razen gesla)

Filtriranje rezultatov

```
SELECT stolpec1, stolpec2, ... FROM ime_tabele WHERE pogoj;
```

- Primeri pogojev
 - *vzdevek*='maja'
 - *tezavnost*=2
 - *vzdevek* LIKE 'ma%'
- Uporaba operatorjev =, <, >, <=, >=, <>, [BETWEEN](#), [LIKE](#), [IN](#)



Zapišite in preizkusite poizvedbe, ki iz zbirke pridobijo

- vse podatke o izbranem igralcu (razen gesla)
- podatke o težavnosti, rezultatu in času vseh odigranih iger izbranega igralca
- lestvico najboljših na izbrani težavnosti

Posodabljanje in brisanje podatkov

```
UPDATE ime_tabele  
SET stolpec1 = vrednost1, stolpec2 = vrednost2, ...  
WHERE pogoj;
```

```
DELETE FROM ime_tabele WHERE pogoj;
```



Zapišite in preizkusite poizvedbo, ki

- posodobi vse podatke o izbranem igralcu (razen njegovega vzdevka)
- izbriše izbranega igralca iz zbirke

Domača naloga

- Premislite, kakšno aplikacijo boste izdelali za izpit
- Preučite, katere podatke bo aplikacija potrebovala, in načrtujte podatkovni model
- Izvedite podatkovni model v podatkovni zbirki
- Namestite [Node.js](https://nodejs.org/)



Dodatek

Namestitev paketa XAMPP

- S spletnega mesta <http://www.apachefriends.org> prenesite zadnjo različico kompleta XAMPP za operacijski sistem, ki ga uporabljate
- Poleg spletnega strežnika Apache izberite tudi namestitev podatkovnega strežnika MySQL
- Na operacijskem sistemu Windows namestite paket XAMPP v mapo **C:/xampp** (oziroma nekam izven mape Program Files)

