

PROJEKAT

Predmet: Baze podataka

Tema: IT akademija

Student: Jovana Ostojić

Profesor: dr Danijela Tešendić

Novi Sad, 2023

Sadržaj

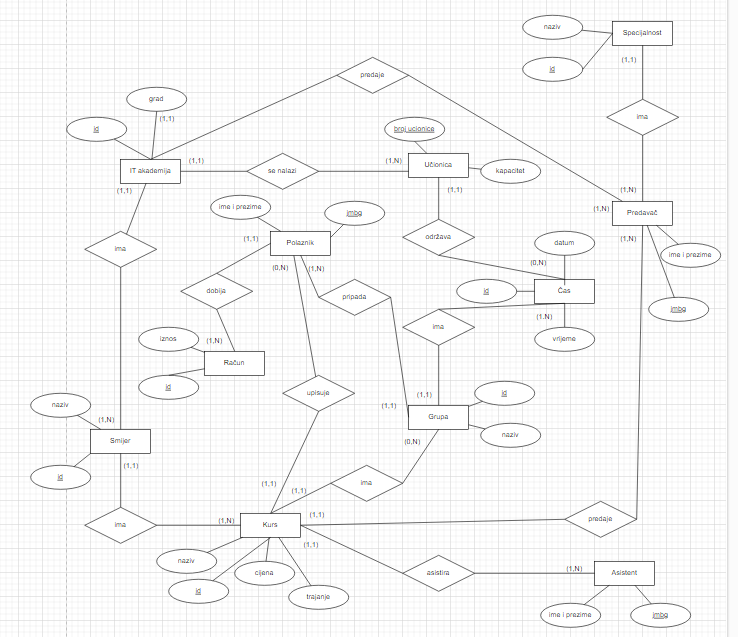
1. Opis informacionog sistema
2. Projektovanje baze podataka
   1. ER model
   2. Relacioni model
3. Naredbe za kreiranje šeme baze podataka
4. Unos podataka
5. Upiti nad podacima
6. Opis informacionog sistema

IT\_akademija

Ovaj informacioni sistem nam omogućava da pratimo rad jedne akademije u ovom slučaju se radi o IT akademiji koju prepoznajemo po njenom id i gradu u kome se nalazi.

Ta akademija ima svoje smijerove koje razlikujemo po nazivu i id. Svaki smijer ima svoje kurseve koje takođe razlikujemo po nazivu i id ali sada imamo cijenu i trajanje jer polaznik bira kurs koji je u sklopu nekog smijera. Polaznik ima svoje ime i prezime kao i jedinstveni matični broj po kome ga prepoznajemo u bazi. On pri upisu kursa dobija račun na kome je iznos kao i id. Na It akademiji kao i na određenom kursu imamo predavača koji predaje. Njega prepoznajemo po imenu i prezimenu kao i jedinstvenom matičnom broju. Svaki predavač ima svoju specijalnost. Na kursu se takođe nalaze asistenti sa istim atributima. Na kursu se nalazi grupa polaznika kojoj se dodjeli ime. Ta grupa pohađa neke časove u određeno vrijeme i datum. Časovi se održavaju u nekoj učionici.

1. Projektovanje baze podataka
   1. ER model



It akademija može imati najmanje jedan smijer ali može i više njih, smijer se može nalaziti na jednoj It akademiji ali isti smijer može biti i na nekoj drugoj akademiji. (1:N)

It akademija može imati jednu učionicu ali može i više njih, ta učionica se može nalaziti na jednoj It akademiji. (1:N)

It akademija može imati jednog predavača a može i više njih dok jedan predavač može predavati samo na jednoj akademiji. (1:N)

Smijer može imati jedan kurs a može i više njih ali kurs se može nalaziti samo na jednom smijeru. (1:N)

Kurs ne mora da upiše niti jedan polaznik a može da upiše i više njih, polaznik može da upiše samo jedan kurs. (1:N)

Kurs ne mora da ima niti jednu grupu ukoliko nema polaznika a može imati i više grupa ukoliko je mnogo polaznika. Određena grupa može da bude samo na jednom kursu jer polaznici mogu upisati samo jedan kurs. (1:N)

Na kursu može da predaje jedan predavač a može i više njih ali jedan predavač može predavati na jednom kursu. (1:N)

Na kursu mora da asistira bar jedan asistent a može i više njih dok asistent asistira samo na jednom kursu. (1:N)

Grupa može biti na više časova ali na jednom času može biti samo jedna grupa. (1:N)

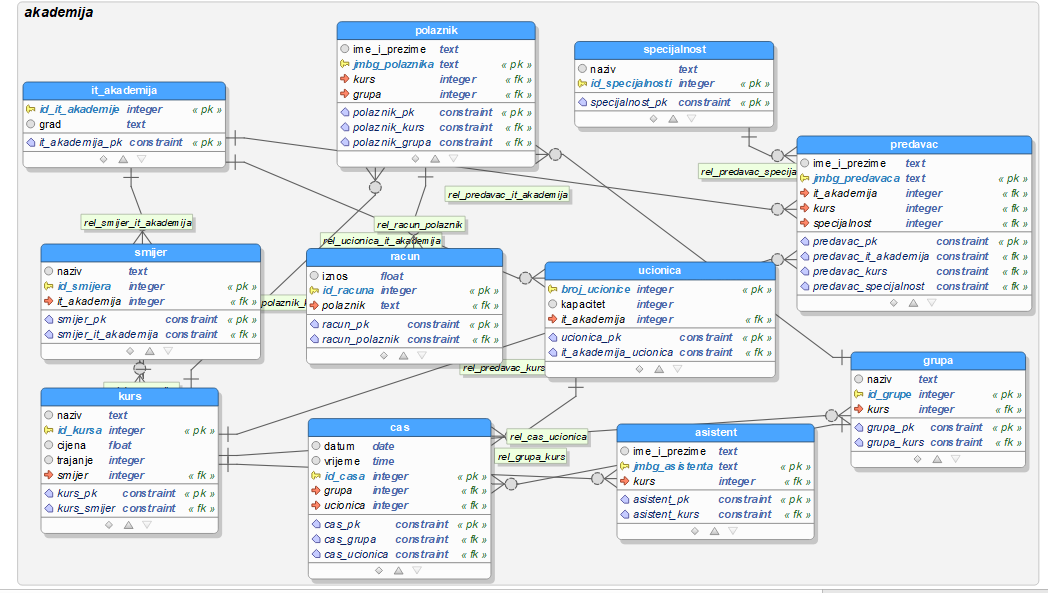
Grupi može da pripada jedan polaznik a može i više njih, dok polaznik može pripadati samo jednoj grupi. (1:N)

Jedan čas se može održavati u jednoj učionici ali u jednoj učionici ne mora da bude niti jedan čas održavan a može i više njih. (1:N)

Predavač ima jednu specijalnost za koju je specijalizovan. Specijalnost pripada jednom predavaču a može istu specijalnost da ima i više predavača. (1:N)

Polaznik može dobiti jedan račun a može i više njih ukoliko upiše više kurseva. Jedan račun može dobiti jedan polaznik. (1:N)

* 1. Relacioni model



It akademija ima id što predstavlja njen PK zatim smijer ima takođe id it akadamije što predstavlja FK a ima i svoj id\_smijera što predstavlja PK. Kurs ima svoj id što predstavlja PK i ima id smijera što je FK. Čas ima id i to mu je PK a ima i id grupe što je FK kao i broj učionice što je FK. Učionica ima svoj broj što je PK i ima id it akademije što je FK. Račun ima svoj id i to mu je PK takođe ima jmbg polaznika što je FK. Polaznik ima svoj jmbg kao PK i ima id kursa i id grupe kao FK. Predavač ima svoj jmbg kao id i ima id it akademije kao FK, zatim id kursa i id specijalnosti kao FK. Specijalnost ima svoj id kao PK. Asistent ima svoj jmbg kao PK i ima id kursa kao FK.

1. Naredbe za kreiranje šeme baze podataka

-- Database generated with pgModeler (PostgreSQL Database Modeler).

-- pgModeler version: 0.9.2

-- PostgreSQL version: 12.0

-- Project Site: pgmodeler.io

-- Model Author: ---

-- Database creation must be done outside a multicommand file.

-- These commands were put in this file only as a convenience.

-- -- object: new\_database | type: DATABASE --

-- -- DROP DATABASE IF EXISTS new\_database;

-- CREATE DATABASE new\_database;

-- -- ddl-end --

--

-- object: it\_akademija | type: SCHEMA --

-- DROP SCHEMA IF EXISTS it\_akademija CASCADE;

CREATE SCHEMA it\_akademija;

-- ddl-end --

-- object: akademija | type: SCHEMA --

-- DROP SCHEMA IF EXISTS akademija CASCADE;

CREATE SCHEMA akademija;

-- ddl-end --

SET search\_path TO pg\_catalog,public,it\_akademija,akademija;

-- ddl-end --

-- object: akademija.it\_akademija | type: TABLE --

-- DROP TABLE IF EXISTS akademija.it\_akademija CASCADE;

CREATE TABLE akademija.it\_akademija (

id\_it\_akademije integer NOT NULL,

grad text,

CONSTRAINT it\_akademija\_pk PRIMARY KEY (id\_it\_akademije)

);

-- ddl-end --

-- object: akademija.ucionica | type: TABLE --

-- DROP TABLE IF EXISTS akademija.ucionica CASCADE;

CREATE TABLE akademija.ucionica (

broj\_ucionice integer NOT NULL,

kapacitet integer,

it\_akademija integer,

CONSTRAINT ucionica\_pk PRIMARY KEY (broj\_ucionice)

);

-- ddl-end --

-- object: akademija.smijer | type: TABLE --

-- DROP TABLE IF EXISTS akademija.smijer CASCADE;

CREATE TABLE akademija.smijer (

naziv text,

id\_smijera integer NOT NULL,

it\_akademija integer,

CONSTRAINT smijer\_pk PRIMARY KEY (id\_smijera)

);

-- ddl-end --

-- object: akademija.kurs | type: TABLE --

-- DROP TABLE IF EXISTS akademija.kurs CASCADE;

CREATE TABLE akademija.kurs (

naziv text,

id\_kursa integer NOT NULL,

cijena float,

trajanje integer,

smijer integer,

CONSTRAINT kurs\_pk PRIMARY KEY (id\_kursa)

);

-- ddl-end --

-- object: akademija.asistent | type: TABLE --

-- DROP TABLE IF EXISTS akademija.asistent CASCADE;

CREATE TABLE akademija.asistent (

ime\_i\_prezime text,

jmbg\_asistenta text NOT NULL,

kurs integer,

CONSTRAINT asistent\_pk PRIMARY KEY (jmbg\_asistenta)

);

-- ddl-end --

-- object: akademija.grupa | type: TABLE --

-- DROP TABLE IF EXISTS akademija.grupa CASCADE;

CREATE TABLE akademija.grupa (

naziv text,

id\_grupe integer NOT NULL,

kurs integer,

CONSTRAINT grupa\_pk PRIMARY KEY (id\_grupe)

);

-- ddl-end --

-- object: akademija.racun | type: TABLE --

-- DROP TABLE IF EXISTS akademija.racun CASCADE;

CREATE TABLE akademija.racun (

iznos float,

id\_racuna integer NOT NULL,

polaznik text,

CONSTRAINT racun\_pk PRIMARY KEY (id\_racuna)

);

-- ddl-end --

-- object: akademija.polaznik | type: TABLE --

-- DROP TABLE IF EXISTS akademija.polaznik CASCADE;

CREATE TABLE akademija.polaznik (

ime\_i\_prezime text,

jmbg\_polaznika text NOT NULL,

kurs integer,

grupa integer,

CONSTRAINT polaznik\_pk PRIMARY KEY (jmbg\_polaznika)

);

-- ddl-end --

-- object: akademija.cas | type: TABLE --

-- DROP TABLE IF EXISTS akademija.cas CASCADE;

CREATE TABLE akademija.cas (

datum date,

vrijeme time,

id\_casa integer NOT NULL,

grupa integer,

ucionica integer,

CONSTRAINT cas\_pk PRIMARY KEY (id\_casa)

);

-- ddl-end --

-- object: akademija.predavac | type: TABLE --

-- DROP TABLE IF EXISTS akademija.predavac CASCADE;

CREATE TABLE akademija.predavac (

ime\_i\_prezime text,

jmbg\_predavaca text NOT NULL,

it\_akademija integer,

kurs integer,

specijalnost integer,

CONSTRAINT predavac\_pk PRIMARY KEY (jmbg\_predavaca)

);

-- ddl-end --

-- object: akademija.specijalnost | type: TABLE --

-- DROP TABLE IF EXISTS akademija.specijalnost CASCADE;

CREATE TABLE akademija.specijalnost (

naziv text,

id\_specijalnosti integer NOT NULL,

CONSTRAINT specijalnost\_pk PRIMARY KEY (id\_specijalnosti)

);

-- ddl-end --

-- object: it\_akademija\_ucionica | type: CONSTRAINT --

-- ALTER TABLE akademija.ucionica DROP CONSTRAINT IF EXISTS it\_akademija\_ucionica CASCADE;

ALTER TABLE akademija.ucionica ADD CONSTRAINT it\_akademija\_ucionica FOREIGN KEY (it\_akademija)

REFERENCES akademija.it\_akademija (id\_it\_akademije) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: smijer\_it\_akademija | type: CONSTRAINT --

-- ALTER TABLE akademija.smijer DROP CONSTRAINT IF EXISTS smijer\_it\_akademija CASCADE;

ALTER TABLE akademija.smijer ADD CONSTRAINT smijer\_it\_akademija FOREIGN KEY (it\_akademija)

REFERENCES akademija.it\_akademija (id\_it\_akademije) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: kurs\_smijer | type: CONSTRAINT --

-- ALTER TABLE akademija.kurs DROP CONSTRAINT IF EXISTS kurs\_smijer CASCADE;

ALTER TABLE akademija.kurs ADD CONSTRAINT kurs\_smijer FOREIGN KEY (smijer)

REFERENCES akademija.smijer (id\_smijera) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: asistent\_kurs | type: CONSTRAINT --

-- ALTER TABLE akademija.asistent DROP CONSTRAINT IF EXISTS asistent\_kurs CASCADE;

ALTER TABLE akademija.asistent ADD CONSTRAINT asistent\_kurs FOREIGN KEY (kurs)

REFERENCES akademija.kurs (id\_kursa) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: grupa\_kurs | type: CONSTRAINT --

-- ALTER TABLE akademija.grupa DROP CONSTRAINT IF EXISTS grupa\_kurs CASCADE;

ALTER TABLE akademija.grupa ADD CONSTRAINT grupa\_kurs FOREIGN KEY (kurs)

REFERENCES akademija.kurs (id\_kursa) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: racun\_polaznik | type: CONSTRAINT --

-- ALTER TABLE akademija.racun DROP CONSTRAINT IF EXISTS racun\_polaznik CASCADE;

ALTER TABLE akademija.racun ADD CONSTRAINT racun\_polaznik FOREIGN KEY (polaznik)

REFERENCES akademija.polaznik (jmbg\_polaznika) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: polaznik\_kurs | type: CONSTRAINT --

-- ALTER TABLE akademija.polaznik DROP CONSTRAINT IF EXISTS polaznik\_kurs CASCADE;

ALTER TABLE akademija.polaznik ADD CONSTRAINT polaznik\_kurs FOREIGN KEY (kurs)

REFERENCES akademija.kurs (id\_kursa) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: polaznik\_grupa | type: CONSTRAINT --

-- ALTER TABLE akademija.polaznik DROP CONSTRAINT IF EXISTS polaznik\_grupa CASCADE;

ALTER TABLE akademija.polaznik ADD CONSTRAINT polaznik\_grupa FOREIGN KEY (grupa)

REFERENCES akademija.grupa (id\_grupe) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: cas\_grupa | type: CONSTRAINT --

-- ALTER TABLE akademija.cas DROP CONSTRAINT IF EXISTS cas\_grupa CASCADE;

ALTER TABLE akademija.cas ADD CONSTRAINT cas\_grupa FOREIGN KEY (grupa)

REFERENCES akademija.grupa (id\_grupe) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: cas\_ucionica | type: CONSTRAINT --

-- ALTER TABLE akademija.cas DROP CONSTRAINT IF EXISTS cas\_ucionica CASCADE;

ALTER TABLE akademija.cas ADD CONSTRAINT cas\_ucionica FOREIGN KEY (ucionica)

REFERENCES akademija.ucionica (broj\_ucionice) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: predavac\_it\_akademija | type: CONSTRAINT --

-- ALTER TABLE akademija.predavac DROP CONSTRAINT IF EXISTS predavac\_it\_akademija CASCADE;

ALTER TABLE akademija.predavac ADD CONSTRAINT predavac\_it\_akademija FOREIGN KEY (it\_akademija)

REFERENCES akademija.it\_akademija (id\_it\_akademije) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: predavac\_kurs | type: CONSTRAINT --

-- ALTER TABLE akademija.predavac DROP CONSTRAINT IF EXISTS predavac\_kurs CASCADE;

ALTER TABLE akademija.predavac ADD CONSTRAINT predavac\_kurs FOREIGN KEY (kurs)

REFERENCES akademija.kurs (id\_kursa) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

-- object: predavac\_specijalnost | type: CONSTRAINT --

-- ALTER TABLE akademija.predavac DROP CONSTRAINT IF EXISTS predavac\_specijalnost CASCADE;

ALTER TABLE akademija.predavac ADD CONSTRAINT predavac\_specijalnost FOREIGN KEY (specijalnost)

REFERENCES akademija.specijalnost (id\_specijalnosti) MATCH FULL

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- ddl-end --

1. Unos podataka

INSERT INTO akademija.it\_akademija (id\_it\_akademije, grad)

VALUES ('1', 'Novi Sad');

INSERT INTO akademija.it\_akademija (id\_it\_akademije, grad)

VALUES ('2', 'Beograd');

INSERT INTO akademija.smijer (naziv, id\_smijera, it\_akademija)

VALUES ('Web Design', '1', '1');

INSERT INTO akademija.smijer (naziv, id\_smijera, it\_akademija)

VALUES ('Programiranje', '2', '2');

INSERT INTO akademija.kurs (naziv, id\_kursa, cijena, trajanje, smijer)

VALUES ('WordPress', '1', '199.99', '3', '1');

INSERT INTO akademija.kurs (naziv, id\_kursa, cijena, trajanje, smijer)

VALUES ('OOP', '2', '189.99', '3', '2');

INSERT INTO akademija.grupa (naziv, id\_grupe, kurs)

VALUES ('WP1', '1', '1');

INSERT INTO akademija.grupa (naziv, id\_grupe, kurs)

VALUES ('OOP1', '2', '2');

INSERT INTO akademija.ucionica (broj\_ucionice, kapacitet, it\_akademija)

VALUES ('1', '30', '1');

INSERT INTO akademija.ucionica (broj\_ucionice, kapacitet, it\_akademija)

VALUES ('2', '20', '2');

INSERT INTO akademija.cas (datum, vrijeme, id\_casa, grupa, ucionica)

VALUES ('06.06.2023.', '15:30', '1', '1', '1');

INSERT INTO akademija.cas (datum, vrijeme, id\_casa, grupa, ucionica)

VALUES ('07.07.2023.', '14:30', '2', '2', '2');

INSERT INTO akademija.polaznik (ime\_i\_prezime, jmbg\_polaznika, kurs, grupa)

VALUES ('Petar Petrovic', '2345678912345', '1', '1');

INSERT INTO akademija.polaznik (ime\_i\_prezime, jmbg\_polaznika, kurs, grupa)

VALUES ('Pero Peric', '5678912345678', '2', '2');

INSERT INTO akademija.racun (iznos, id\_racuna, polaznik)

VALUES ('199.99', '123', '2345678912345');

INSERT INTO akademija.racun (iznos, id\_racuna, polaznik)

VALUES ('189.99', '1234', '5678912345678');

INSERT INTO akademija.predavac (ime\_i\_prezime, jmbg\_predavaca, it\_akademija, kurs)

VALUES ('Zeljko Zeljkic', '3456789123456', '1', '1');

INSERT INTO akademija.predavac (ime\_i\_prezime, jmbg\_predavaca, it\_akademija, kurs)

VALUES ('Dragan Dragic', '7891234567891', '2', '2');

INSERT INTO akademija.specijalnost (naziv, id\_specijalnosti)

VALUES ('web dizajn', '1');

INSERT INTO akademija.specijalnost (naziv, id\_specijalnosti)

VALUES ('java', '2');

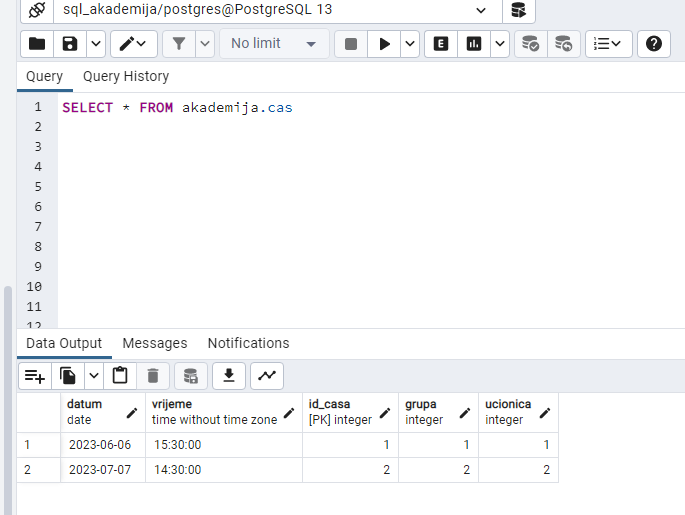
INSERT INTO akademija.asistent (ime\_i\_prezime, jmbg\_asistenta, kurs)

VALUES ('Milos Milosevic', '1234567891234', '1');

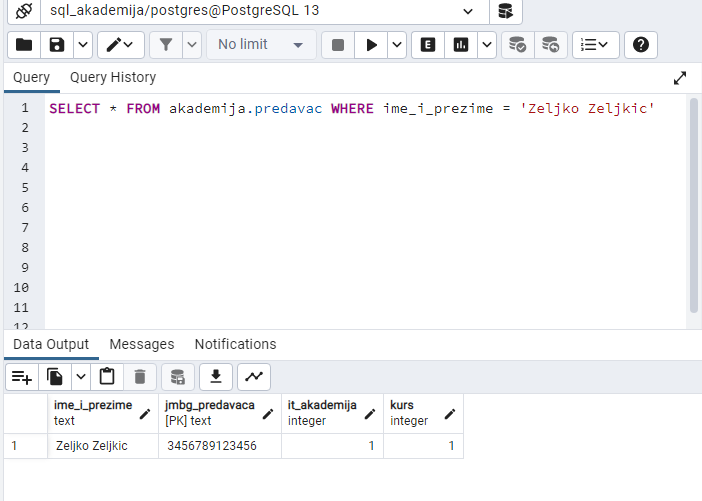
INSERT INTO akademija.asistent (ime\_i\_prezime, jmbg\_asistenta, kurs)

VALUES ('Miko Mikic', '4567891234567', '2');

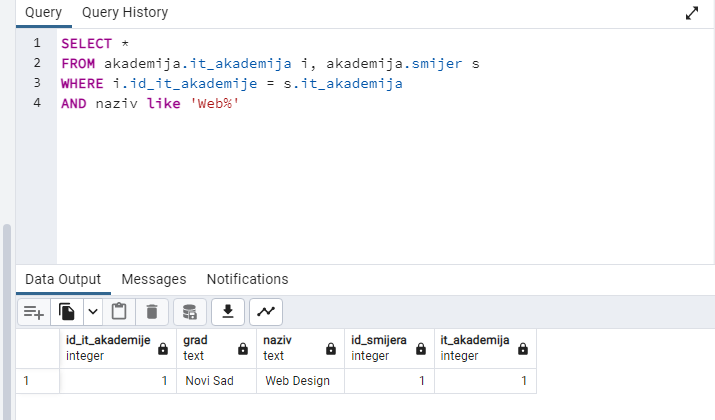
1. Upiti nad podacima
2. Prikaz svih podataka iz tabele



1. Filtriranje podataka iz jedne tabele na osnovu nekog kriterijuma



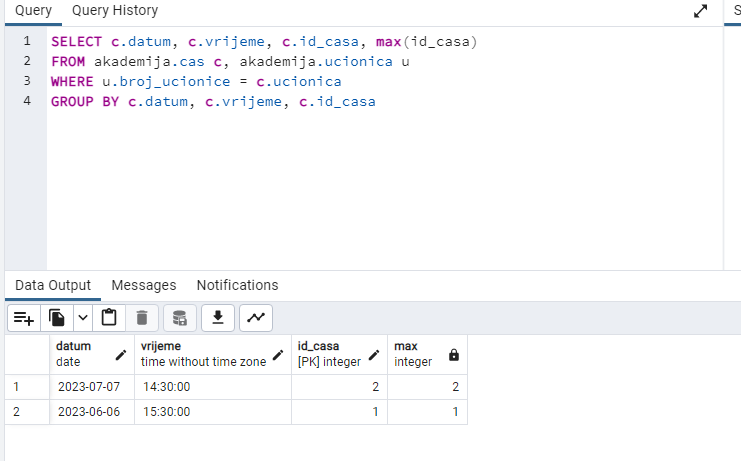
1. Upit sa spajanjem dve tabele i uslovom



1. Upit sa spajanjem tri tabele i uslovom



1. Group by upit sa spajanjem dve tabele i primenom skupovne funkcije



1. Ugnježdeni upit

