

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET PRIMIJENJENE MATEMATIKE I INFORMATIKE



MODERNI SUSTAVI BAZA PODATAKA
SEMINARSKI RAD

BAZA PODATAKA ZA FITNES

LARA SLIŠKOVIĆ

Osijek, 2024.

SADRŽAJ

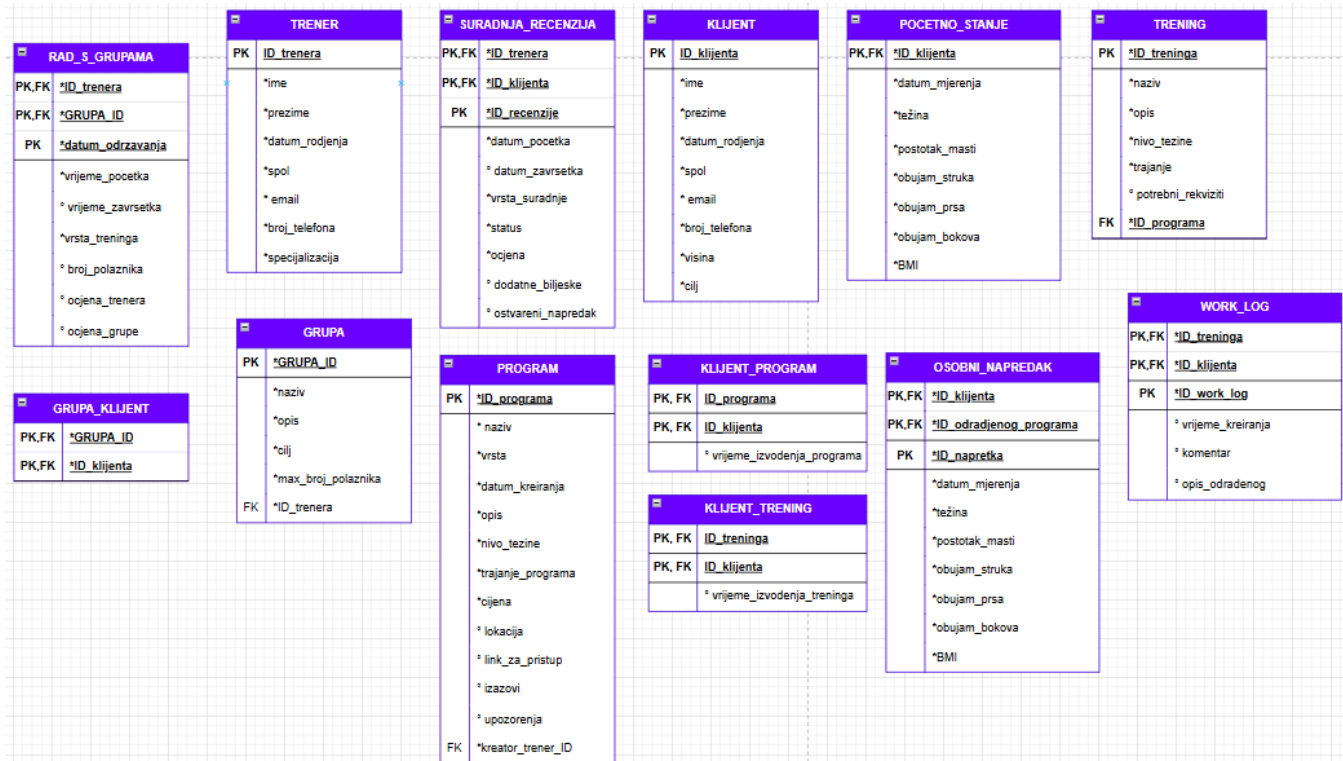
1. UVOD	3
2. MEV I RELACIJSKI MODEL.....	4
3. KREIRANJE TABLICA I UNOSI	7
4. JEDNOSTAVNI UPITI	10
5. UPITI NAD VIŠE TABLICA	13
6. UPITI S AGREGIRAJUĆIM FUNKCIJAMA.....	16
7. PODUPITI I OSTALI UPITI	19
8. ZADANE VRIJEDNOSTI, UVJETI I KOMENTARI.....	24
9. INDEKSI.....	26
10. PROCEDURE	27
11. OKIDAČI.....	29
12. ZAKLJUČAK	31

1. UVOD

Svakim danom sve više se susrećemo sa pojmom fitnesa, bilo kao dijelom zdravog života, ispuhivanjem viška energije, održavanjem tijela u formi ili ispunjavanjem slobodnog vremena.. Fitnes kao usluga odnosi se na centre koji pružaju mogućnost provođenja tjelesne aktivnosti. Takvih centara je veliki broj i bilo koji centar u kojem se provodi tjelesna aktivnost može se registrirati kao fitnes centar. Upravo radi velikog broja fitnes centara, ali i njihovih korisnika te onih koji to tek planiraju postati, učinilo se zgodno napraviti bazu podataka za jednu fitnes zajednicu. Obzirom da se i sama bavim raznim aktivnostima i pratim rad pojedinih fitnes trenera, konkretan primjer koji će biti predstavljen u slijedećim stranicama ovog seminara jedinstven je produkt moje osobne mašte i uvjerenja. Njegov cilj je predstaviti model koji može pratiti razne aktivnosti i stvoriti najbolje moguće programe koji se baziraju na “feedback” svakog klijenta. Sve dokumentirano prati skriptu koja je priložena zajedno uz ovaj dokument.

2. MEV I RELACIJSKI MODEL

Model entiteta i veza (Slika 1.) sastoji se od 13 entiteta. Veze između entiteta uglavnom su 1:1, 1:N te jedna N:N. Kao osobe bez kojih ništa od ovoga ne bi ni postojalo, izdvajaju se treneri. Oni su ti koji su uzor i koji vode fitnes “seanse” te kreiraju programe koje klijenti prate. Svaki trener može, ali i ne mora biti autor nekog od postojećih programa te može imati suradnje s pojedincima ili grupama, ali i ne mora. Nadalje, krećući od klijenta, osobe koja je također ključna u stvaranju ove zajednice, vidljivo je da se cijeli model bazira na njegovoj volji pri izboru aktivnosti. Klijent, ukoliko želi krenuti pohađati aktivnosti unutar zajednice, ima 2 opcije; može rezervirati svoje mjesto za privatne treninge s trenerom ili može pohađati grupe koje također vode treneri iz centra. Svakoj grupi pripada jedan trener koji ju vodi, a detalji oko specifične grupe su formirani unutar entiteta za rad s grupama. Tamo se nalaze i specifični atributi; ocjena grupe te ocjena trenera koje se formiraju prilikom svakog novog sastanka s grupama te održane sesije, te broj polaznika koji statistički prati iste kako se ne bi dogodilo da polaznika bude više nego što je predviđeno za svaku grupu. Taj dio je reguliran uvjetom, a sve u svrhu maksimalnog posvećenja svim pripadnicima grupe. Za klijente koji ne preferiraju rad unutar grupa, osim fizičkog dolaska u centar na programe, postoji opcija i “online” kupnje programa. Svaki program sastoji se od različitih treninga namijenjenih za taj program. Podrazumijeva se da obje vrste privatnih klijenata trebaju odraditi svaki trening koji je namijenjen pojedinom programu radi ostvarivanja maksimalnih rezultata, ali to ponovno nije nužno radi pronalaska optimalnog tempa svakom pojedincu. Klijenti koji prate treninge iz “online” programa imaju mogućnost davanja “feedbacka” koji bi dosta značio u daljnjem napredovanju cijele zajednice. Prilikom kreiranja te povratne informacije u vidu “workloga” okidač provjerava kojem tipu programa trening pripada. Svi klijenti nakon odrađenog cjelovitog programa treninga imaju mogućnost pohraniti informacije o svome osobnom napretku kako bi se mogle pratiti promjene u odnosu na njihovo početno stanje što je također jedan od entiteta. Treba napomenuti da je klijent slobodan u svakom trenutku izabrati svoj način treniranja kroz različite programe, no trenutna vrsta tipa klijenta se ne prati jer ne pruža nikakve značajne informacije budući da je cijeli sustav dosta fleksibilan. Omogućeno je također zabilježiti recenziju o suradnji s pojedinim trenerom.



Slika 2.

3. KREIRANJE TABLICA I UNOSI

Kreiranje tablica u bazi provodi se u potpunosti nakon dobivenog relacijskog modela, a zatim se redom unose tablice na način da se prvo unesu tablice koje ne sadrže strane ključeve, a nakon toga one čiji se ključevi odnose na primarne ključeve, sada već kreiranih tablica. Na isti način se unose i podaci. U prilogu iz SQL skripte vidljivo je kreiranje najbitnijih tablica i popunjavanje istih. Prije samog kreiranja prvo se uklone sve tablice koje već imaju imena koja će se koristiti.

Postupak kreiranja određenih tablica kroz priloge:

```
DROP TABLE IF EXISTS trener CASCADE;
DROP TABLE IF EXISTS klijent CASCADE;
DROP TABLE IF EXISTS grupa CASCADE;
DROP TABLE IF EXISTS grupa_klijent CASCADE;
DROP TABLE IF EXISTS rad_s_grupama CASCADE;
DROP TABLE IF EXISTS suradnja_recenzija CASCADE;
DROP TABLE IF EXISTS proram CASCADE;
DROP TABLE IF EXISTS trening CASCADE;
DROP TABLE IF EXISTS klijent_program CASCADE;
DROP TABLE IF EXISTS klijent_trening CASCADE;
DROP TABLE IF EXISTS pocetno_stanje CASCADE;
DROP TABLE IF EXISTS osobni_napredak CASCADE;
DROP TABLE IF EXISTS work_log CASCADE;
```

Slika 3.

```
-- Table: trener
CREATE TABLE trener (
  ID_trenera SERIAL PRIMARY KEY,
  ime VARCHAR(255) NOT NULL,
  prezime VARCHAR(255) NOT NULL,
  datum_rodjenja DATE NOT NULL,
  email VARCHAR(255) NOT NULL UNIQUE,
  broj_telefona VARCHAR(20),
  specijalizacija VARCHAR(255)
);

-- Table: klijent
CREATE TABLE klijent (
  ID_klijenta SERIAL PRIMARY KEY,
  ime VARCHAR(255) NOT NULL,
  prezime VARCHAR(255) NOT NULL,
  datum_rodjenja DATE NOT NULL,
  spol CHAR(1) CHECK (spol IN ('M', 'F')),
  email VARCHAR(255) NOT NULL UNIQUE,
  broj_telefona VARCHAR(20),
  visina FLOAT NOT NULL CHECK (visina > 0),
  cilj VARCHAR(100)
);
```

Slika 4.

```
-- Table: osobni_napredak
CREATE TABLE osobni_napredak (
    klijent_id INT NOT NULL,
    ID_odradjenog_programa INT NOT NULL,
    ID_napretka SERIAL NOT NULL,
    CONSTRAINT NAPREDAK PRIMARY KEY (ID_odradjenog_programa, klijent_id, ID_napretka),
    datum_mjerenja DATE NOT NULL,
    tezina NUMERIC(5, 2) CHECK (tezina > 0),
    postotak_masti NUMERIC(5, 2) CHECK (postotak_masti >= 0 AND postotak_masti <= 100),
    obujam_struka NUMERIC(5, 2) CHECK (obujam_struka > 0),
    obujam_prsa NUMERIC(5, 2) CHECK (obujam_prsa > 0),
    obujam_bokova NUMERIC(5, 2) CHECK (obujam_bokova > 0),
    BMI NUMERIC(5, 2) CHECK (BMI > 0)
);

-- Table: work_log
CREATE TABLE work_log (
    trening_id INT NOT NULL,
    klijent_id INT NOT NULL,
    ID_work_log SERIAL NOT NULL,
    CONSTRAINT LOG PRIMARY KEY (trening_id, klijent_id, ID_work_log),
    vrijeme_kreiranja DATE,
    komentar TEXT,
    opis_odradenog TEXT,
    FOREIGN KEY (trening_id) REFERENCES trening(ID_treninga),
    FOREIGN KEY (klijent_id) REFERENCES klijent(ID_klijenta)
);
```

Slika 5.

Unosi u priložene tablice:

```
-- Insert data into trener
INSERT INTO trener VALUES
(1, 'Ivan', 'Ivić', '1980-01-01', 'ivan.ivic@example.com', '0912345678', 'Fitness'),
(2, 'Ana', 'Anić', '1985-02-02', 'ana.anic@example.com', '0923456789', 'Yoga'),
(3, 'Marko', 'Markić', '1990-03-03', 'marko.markic@example.com', '0934567890', 'Bodybuilding');

-- Insert data into klijent
INSERT INTO klijent (ID_klijenta, ime, prezime, datum_rodjenja, spol, email, broj_telefona, visina, cilj) VALUES
(1, 'Petar', 'Petrović', '1995-05-05', 'M', 'petar.p@example.com', '0956781234', 180, 'Lose Weight'),
(2, 'Karla', 'Karla', '1993-06-06', 'F', 'karla.k@example.com', '0945671234', 165, 'Gain Muscle'),
(3, 'Luka', 'Lukić', '1991-07-07', 'M', 'luka.l@example.com', '0934561234', 175, 'Improve Endurance'),
(4, 'Ema', 'Emić', '1988-08-08', 'F', 'ema.e@example.com', '0923451234', 170, 'Maintain Fitness'),
(5, 'Tomislav', 'Tomić', '1992-09-09', 'M', 'tomislav.t@example.com', '0912341234', 185, 'Lose Weight'),
(6, 'Ivana', 'Ivić', '1994-10-10', 'F', 'ivana.i@example.com', '0919876543', 160, 'Improve Flexibility'),
(7, 'Sara', 'Sarić', '1996-11-11', 'F', 'sara.s@example.com', '0928765432', 168, 'Gain Muscle'),
(8, 'Ante', 'Antić', '1989-12-12', 'M', 'ante.a@example.com', '0937654321', 178, 'Improve Strength'),
(9, 'Nina', 'Ninić', '1990-01-01', 'F', 'nina.n@example.com', '0946543210', 167, 'Lose Weight'),
(10, 'Filip', 'Filipović', '1987-02-02', 'M', 'filip.f@example.com', '0955432109', 182, 'Maintain Fitness'),
(11, 'Marta', 'Martić', '1986-03-03', 'F', 'marta.m@example.com', '0964321098', 164, 'Gain Muscle'),
(12, 'Domagoj', 'Domagojić', '1985-04-04', 'M', 'domagoj.d@example.com', '0973210987', 179, 'Improve Endurance'),
(13, 'Lucija', 'Lucić', '1984-05-05', 'F', 'lucija.l@example.com', '0982109876', 161, 'Maintain Fitness'),
(14, 'Kristijan', 'Kristić', '1983-06-06', 'M', 'kristijan.k@example.com', '0991098765', 180, 'Gain Muscle'),
(15, 'Tina', 'Tinić', '1992-07-07', 'F', 'tina.t@example.com', '0910987654', 172, 'Lose Weight');
```

Slika 6.


```

-- Insert data into osobni_napredak
INSERT INTO osobni_napredak (kljent_id, ID_odradjenog_programa, datum_mjerenja, tezina,
                             postotak_masti, obujam_struka, obujam_prsa, obujam_bokova, BMI) VALUES
(1, 1, '2024-03-31', 85.0, 25.0, 90.0, 100.0, 95.0, 27.5),
(2, 2, '2024-06-27', 70.0, 20.0, 75.0, 85.0, 80.0, 23.4),
(3, 3, '2024-04-01', 75.0, 18.0, 80.0, 90.0, 85.0, 24.5),
(4, 1, '2024-06-15', 95.0, 30.0, 100.0, 110.0, 105.0, 29.5);

-- Insert data into work_log for clients attending online trainings
INSERT INTO work_log (trening_id, kljent_id, vrijeme_kreiranja, komentar, opis_odradenog) VALUES
(21, 8, '2024-06-11', 'Impressed with the instructor!', 'Online yoga session at home'),
(21, 9, '2024-06-12', 'Effective stretches!', 'Online yoga session at home'),
(21, 10, '2024-06-13', 'Great way to start the day!', 'Online yoga session at home'),
(21, 11, '2024-06-14', 'Peaceful experience!', 'Online yoga session at home'),
(21, 12, '2024-06-15', 'Loved the session!', 'Online yoga session at home'),
(21, 13, '2024-06-16', 'Stress-relieving!', 'Online yoga session at home'),
(21, 14, '2024-06-17', 'Enjoyable workout!', 'Online yoga session at home'),
(21, 15, '2024-06-18', 'Great for flexibility!', 'Online yoga session at home');

```

Slika 7.

4. JEDNOSTAVNI UPITI

U ovom poglavlju opisat ću jednostavne upite nad danom bazom podataka.

Prvi jednostavan upit (Slika 8.):

--Ovaj upit vraća sve podatke (sve stupce) iz tablice klient.
--To znači da će vratiti sve redove i sve informacije o svakom klientu.
SELECT * FROM klient;

Slika 8.

- Rezultat upita (Slika 9.) izgleda ovako:

	id_klijenta [PK] integer	ime character varying (255)	prezime character varying (255)	datum_rođenja date	spol character	email character varying (255)	broj_telefona character varying (20)	visina double precision	cilj character varying (100)
1	1	Petar	Petrović	1995-05-05	M	petar.p@example.com	0956781234	180	Lose Weight
2	2	Karla	Karla	1993-06-06	F	karla.k@example.com	0945671234	165	Gain Muscle
3	3	Luka	Lukić	1991-07-07	M	luka.l@example.com	0934561234	175	Improve Endurance
4	4	Ema	Emić	1988-08-08	F	ema.e@example.com	0923451234	170	Maintain Fitness
5	5	Tomislav	Tomić	1992-09-09	M	tomislav.t@example.com	0912341234	185	Lose Weight
6	6	Ivana	Ivić	1994-10-10	F	ivana.i@example.com	0919876543	160	Improve Flexibility
7	7	Sara	Sarić	1996-11-11	F	sara.s@example.com	0928765432	168	Gain Muscle
8	8	Ante	Antić	1989-12-12	M	ante.a@example.com	0937654321	178	Improve Strength
9	9	Nina	Ninić	1990-01-01	F	nina.n@example.com	0946543210	167	Lose Weight
10	10	Filip	Filipović	1987-02-02	M	filip.f@example.com	0955432109	182	Maintain Fitness
11	11	Marta	Martić	1986-03-03	F	marta.m@example.com	0964321098	164	Gain Muscle
12	12	Domagoj	Domagojić	1985-04-04	M	domagoj.d@example.co...	0973210987	179	Improve Endurance
13	13	Lucija	Lucić	1984-05-05	F	lucija.l@example.com	0982109876	161	Maintain Fitness
14	14	Kristijan	Kristić	1983-06-06	M	kristijan.k@example.com	0991098765	180	Gain Muscle
15	15	Tina	Tinić	1992-07-07	F	tina.t@example.com	0910987654	172	Lose Weight

Slika 9.

Drugi jednostavan upit (Slika 10.):

--Ovaj upit traži redak iz tablice trener gdje je email jednak 'ivan.ivic@example.com'.
--Vraća sve stupce za trenera koji ima taj specifični email.
SELECT * FROM trener WHERE email = 'ivan.ivic@example.com';

Slika 10.

- Rezultat ovog upita (Slika 11.) može se vidjeti ovdje:

	id_trenera [PK] integer	ime character varying (255)	prezime character varying (255)	datum_rođenja date	email character varying (255)	broj_telefona character varying (20)	specijalizacija character varying (255)
1	1	Ivan	Ivić	1980-01-01	ivan.ivic@example.com	0912345678	Fitness

Slika 11.

Treći jednostavan upit (Slika 12.):

--Ovaj upit spaja tablice grupa i trener na temelju trener_id u tablici grupa i ID_trenera u tablici trener.
--Vraća naziv grupe te ime i prezime trenera koji vodi tu grupu.

```
SELECT g.naziv, t.ime, t.prezime FROM grupa g JOIN trener t ON g.trener_id = t.ID_trenera;
```

Slika 12.

- Rezultat upita vidljiv je na sljedećoj slici (Slika 13.):

	naziv character varying (255)	ime character varying (255)	prezime character varying (255)
1	Morning Fitness	Ivan	Ivić
2	Yoga Class	Ana	Anić
3	Bodybuilding Basics	Marko	Markić

Slika 13.

Četvrti jednostavan upit (Slika 14.):

--Ovaj upit vraća sve podatke iz tablice rad_s_grupama gdje je datum_odrzavanja jednak '2023-01-01'.

```
SELECT * FROM rad_s_grupama WHERE datum_odrzavanja = '2024-06-16';
```

Slika 14.

- Rezultat upita (Slika 15.):

	trener_id [PK] integer	grupa_id [PK] integer	datum_odrzavanja [PK] date	vrijeme_pocetka double precision	vrijeme_zavrsetka double precision	vrsta_treninga character varying (100)	broj_polaznika integer	ocjena_trenera numeric (5,2)	ocjena_grupe numeric (5,2)
1	1	1	2024-06-16	8	9	Fitness	19	4.40	4.30

Slika 15.

Peti jednostavan upit (Slika 16.):

--Ovaj upit vraća sve podatke iz tablice program gdje je trener_id jednak 1,
--odnosno sve programe čiji je autor trener koji ima ID 1.

```
SELECT * FROM program WHERE trener_id = 1;
```

Slika 16.

- Rezultat upita (Slika 17. i Dodatak slici 17.):

	id_programa [PK] integer	naziv character varying (255)	vrsta character varying (50)	datum_kreiranja date	opis text	nivo_tezine integer	trajanje_programa double precision	cijena double precision	lokacija character varying (200)
1	1	Weight Loss Program	Uživo	2024-01-01	Weight loss program description	3	12	500	Gym
				link_zapristup text	izazovi text	upozorenja text	trener_id integer		
				[null]	Follow diet plan	Consult physician	1		

Slika 17. i Dodatak slici 17.

5. UPITI NAD VIŠE TABLICA

Sljedeće poglavlje sadrži upite nad više tablica unutar baze.

Prvi složeni upit (Slika 18.):

```
--Ovaj upit spaja tablice klijent_program, klijent i program.  
--Povezuje podatke o klijentima (ime, prezime) s programima (naziv) koje su ti klijenti pohađali.  
SELECT k.ime, k.prezime, p.naziv FROM klijent_program kp  
JOIN klijent k ON kp.klijent_id = k.ID_klijenta  
JOIN program p ON kp.program_id = p.ID_programa;
```

Slika 18.

- Rezultat upita (Slika 19. i Dodatak slici 19.):

	ime character varying (255) 🔒	prezime character varying (255) 🔒	naziv character varying (255) 🔒
1	Petar	Petrović	Weight Loss Program
2	Karla	Karla	Yoga for Beginners
3	Luka	Lukić	Advanced Bodybuilding
4	Ema	Emić	Weight Loss Program
5	Tomislav	Tomić	Yoga for Beginners
6	Ivana	Ivić	Advanced Bodybuilding
7	Sara	Sarić	Weight Loss Program
8	Ante	Antić	Yoga for Beginners
9	Nina	Ninić	Advanced Bodybuilding
10	Filip	Filipović	Weight Loss Program
11	Marta	Martić	Yoga for Beginners
12	Domagoj	Domagojić	Advanced Bodybuilding
13	Lucija	Lucić	Weight Loss Program
14	Kristijan	Kristić	Yoga for Beginners
15	Tina	Tinić	Advanced Bodybuilding
16	Ante	Antić	Online Yoga
17	Nina	Ninić	Online Yoga
18	Filip	Filipović	Online Yoga
19	Marta	Martić	Online Yoga
20	Domagoj	Domagojić	Online Yoga
21	Lucija	Lucić	Online Yoga
22	Kristijan	Kristić	Online Yoga
23	Tina	Tinić	Online Yoga

Slika 19. i Dodatak slici 19.

Drugi složeni upit (Slika 20.):

```
--Ovaj upit spaja tablice klijent_trening i trening, te vraća naziv odradenih treninga,  
--opis i vrijeme izvođenja treninga za klijenta s ID-jem 1.  
SELECT t.naziv, t.opis, kt.vrijeme_izvođenja_treninga FROM klijent_trening kt  
JOIN trening t ON kt.trening_id = t.ID_treninga  
WHERE kt.klijent_id = 1;
```

Slika 20.

- Rezultat upita vidljiv je na sljedećoj slici (Slika 21.):

	naziv character varying (255) 🔒	opis text 🔒	vrijeme_izvođenja_treninga time without time zone 🔒
1	Live Cardio Class	In-person cardio session	01:15:00

Slika 21.

Treći složeni upit (Slika 22.):

```
--Ovaj upit spaja tablice suradnja_recenzija, klijent i trener, te vraća informacije o recenzijama  
--(ocjena, dodatne bilješke) koje su klijenti dali trenerima.  
SELECT s.trener_id, s.klijent_id, s.ocjena, s.dodatne_biljeske FROM suradnja_recenzija s  
JOIN klijent k ON s.klijent_id = k.ID_klijenta  
JOIN trener t ON s.trener_id = t.ID_trenera;
```

Slika 22.

- Rezultat upita (Slika 23.):

	trener_id integer 🔒	klijent_id integer 🔒	ocjena double precision 🔒	dodatne_biljeske text 🔒
1	1	1	5	Great progress
2	2	2	4	Improved flexibility
3	3	3	5	Increased strength

Slika 23.

Četvrti složeni upit (Slika 24.):

```
--Ovaj upit spaja tablice rad_s_grupama i grupa, te vraća informacije o treninzima ili radu s grupama
--(trener ID, grupa ID, vrsta treninga) za grupu koja se zove 'Morning Fitness'.
SELECT DISTINCT r.trener_id, r.grupa_id, r.vrsta_treninga FROM rad_s_grupama r
JOIN grupa g ON r.grupa_id = g.GRUPA_ID
WHERE g.naziv = 'Morning Fitness';
```

Slika 24.

- Rezultat upita može se vidjeti na sljedećoj slici (Slika 25.):

	trener_id integer	grupa_id integer	vrsta_treninga character varying (100)
1	1	1	Fitness

Slika 25.

Peti složeni upit (Slika 26.):

```
--Ovaj će upit za svakog klijenta, prvo prikazati podatke o početnom stanju iz pocetno_stanje tablice,
--a zatim će biti prikazani podaci o svakom osobnom napretku iz osobni_napredak tablice
SELECT k.ime, k.prezime, ps.tezina AS pocetna_tezina, ps.postotak_masti AS pocetni_postotak_masti,
ps.obujam_struka AS pocetni_obujam_struka, ps.obujam_prsa AS pocetni_obujam_prsa,
ps.obujam_bokova AS pocetni_obujam_bokova, ps.BMI AS pocetni_BMI, o.datum_mjerenja AS datum_napretka,
o.tezina AS tezina_napretka, o.postotak_masti AS postotak_masti_napretka,
o.obujam_struka AS obujam_struka_napretka, o.obujam_prsa AS obujam_prsa_napretka,
o.obujam_bokova AS obujam_bokova_napretka, o.BMI AS BMI_napretka
FROM klijent k
JOIN pocetno_stanje ps ON k.ID_klijenta = ps.klijent_id
JOIN osobni_napredak o ON k.ID_klijenta = o.klijent_id;
```

Slika 26.

- Rezultat upita (Slika 27. i Dodatak slici 27.):

	ime character varying (255)	prezime character varying (255)	pocetna_tezina numeric (5,2)	pocetni_postotak_masti numeric (5,2)	pocetni_obujam_struka numeric (5,2)	pocetni_obujam_prsa numeric (5,2)	pocetni_obujam_bokova numeric (5,2)	pocetni_bmi numeric (5,2)	datum_napretka date
1	Petar	Petrović	80.00	25.00	90.00	100.00	95.00	24.70	2024-03-31
2	Karla	Karla	65.00	20.00	75.00	85.00	80.00	23.90	2024-06-27
3	Luka	Lukić	70.00	18.00	80.00	90.00	85.00	22.90	2024-04-01
4	Ema	Emić	60.00	30.00	100.00	110.00	105.00	29.50	2024-06-15

pocetni_bmi numeric (5,2)	datum_napretka date	tezina_napretka numeric (5,2)	postotak_masti_napretka numeric (5,2)	obujam_struka_napretka numeric (5,2)	obujam_prsa_napretka numeric (5,2)	obujam_bokova_napretka numeric (5,2)	bmi_napretka numeric (5,2)
24.70	2024-03-31	85.00	25.00	90.00	100.00	95.00	27.50
23.90	2024-06-27	70.00	20.00	75.00	85.00	80.00	23.40
22.90	2024-04-01	75.00	18.00	80.00	90.00	85.00	24.50
29.50	2024-06-15	95.00	30.00	100.00	110.00	105.00	29.50

Slika 27. i Dodatak slici 27.

6. UPITI S AGREGIRAJUĆIM FUNKCIJAMA

Na kraju ostaju s agregirajućim funkcijama koji se mogu provesti nad ovom bazom podataka.

Prvi upit (Slika 28.):

```
--Ovaj upit vraća ime i prezime trenera te broj klijenata koji su trenirali pod njihovim vodstvom.  
--Može se koristiti za praćenje popularnosti trenera među klijentima.  
SELECT t.ime, t.prezime, COUNT(kp.klijent_id) AS broj_klijenata FROM klijent_program kp  
JOIN program p ON kp.program_id = p.ID_programa  
JOIN trener t ON p.trener_id = t.ID_trenera  
GROUP BY t.ime, t.prezime;
```

Slika 28.

- Rezultat upita (Slika 29.):

	ime character varying (255)	prezime character varying (255)	broj_klijenata bigint
1	Marko	Markić	5
2	Ana	Anić	13
3	Ivan	Ivić	5

Slika 29.

Drugi upit (Slika 30.):

```
--Ovaj upit vraća ime i prezime trenera te prosječnu ocjenu koju su dobili na temelju recenzija.  
--Koristi se za procjenu zadovoljstva klijenata s radom trenera.  
SELECT t.ime, t.prezime, AVG(s.ocjena) AS prosjecna_ocjena FROM suradnja_recenzija s  
JOIN trener t ON s.trener_id = t.ID_trenera  
GROUP BY t.ime, t.prezime;
```

Slika 30.

- Rezultati upita (Slika 31.):

	ime character varying (255)	prezime character varying (255)	prosjecna_ocjena double precision
1	Marko	Markić	5
2	Ana	Anić	4
3	Ivan	Ivić	5

Slika 31

Treći upit (Slika 32.):

```
--Ovaj upit vraća naziv grupe te broj treninga održanih za svaku grupu.  
--Koristi se za praćenje aktivnosti grupa u centru.  
SELECT g.naziv, COUNT(r.grupa_id) AS broj_treninga FROM rad_s_grupama r  
JOIN grupa g ON r.grupa_id = g.GRUPA_ID  
GROUP BY g.naziv;
```

Slika 32.

- Rezultati upita (Slika 33.):

	naziv character varying (255) 🔒	broj_treninga bigint 🔒
1	Yoga Class	9
2	Bodybuilding Basics	9
3	Morning Fitness	9

Slika 33.

Četvrti upit (Slika 34.):

```
--Ovaj upit vraća naziv programa te prosječni BMI (Body Mass Index)  
--klijenata koji su sudjelovali u tom programu.  
--Može koristiti za procjenu učinka programa na zdravlje klijenata.  
SELECT p.naziv, ROUND(AVG(o.BMI),2) AS prosjecni_BMI FROM osobni_napredak o  
JOIN klijent_program kp ON o.klijent_id = kp.klijent_id  
JOIN program p ON kp.program_id = p.ID_programa  
GROUP BY p.naziv;
```

Slika 34.

- Rezultati upita (Slika 35.):

	naziv character varying (255) 🔒	prosjecni_bmi numeric 🔒
1	Advanced Bodybuilding	24.50
2	Weight Loss Program	28.50
3	Yoga for Beginners	23.40

Slika 35.

Peti upit (Slika 36.):

```
--Ovaj upit vraća broj treninga održanih svakog mjeseca, grupiran po mjesecima godine i sortiran kronološki.  
--Pomaže u analizi aktivnosti tijekom godine.  
SELECT EXTRACT(YEAR FROM datum_odrzavanja) AS godina,  
       EXTRACT(MONTH FROM datum_odrzavanja) AS mjesec,  
       COUNT(*) AS broj_treninga  
FROM rad_s_grupama  
GROUP BY godina, mjesec  
ORDER BY godina, mjesec;
```

Slika 36.

- Rezultati upita (Slika 37.):

	godina numeric	mjesec numeric	broj_treninga bigint
1	2024	6	27

Slika 37.

7. PODUPITI I OSTALI UPITI



U ovom poglavlju bit će opisani primjeri podupita, ugniježđenih upita i upita pomoću skupovnih operacija.

Prvi upit (Slika 38.):

```
--Ovaj upit koristi ugniježđeni upit unutar glavnog upita kako bi izračunao  
--broj klijenata u svakom programu iz tablice klijent_program.  
SELECT p.naziv,  
       (SELECT COUNT(kp.klijent_id) FROM klijent_program kp WHERE kp.program_id = p.ID_programa) AS broj_klijenata  
FROM program p;
```

Slika 38.

- Rezultati upita (Slika 39.):

	naziv character varying (255) 	broj_klijenata bigint 
1	Weight Loss Program	5
2	Yoga for Beginners	5
3	Advanced Bodybuilding	5
4	Online Yoga	8

Slika 39.

Drugi upit (Slika 40.):

```
--Upit s podupitom koji izračunava broj treninga po klijentu
SELECT k.ime, k.prezime,
       (SELECT COUNT(*)
        FROM work_log wl
        WHERE wl.klijent_id = k.ID_klijenta) AS broj_treninga
FROM klijent k;
```

Slika 40.

- Rezultat upita (Slika 41.):

	ime character varying (255) 🔒	prezime character varying (255) 🔒	broj_treninga bigint 🔒
1	Petar	Petrović	0
2	Karla	Karla	0
3	Luka	Lukić	0
4	Ema	Emić	0
5	Tomislav	Tomić	0
6	Ivana	Ivić	0
7	Sara	Sarić	0
8	Ante	Antić	1
9	Nina	Ninić	1
10	Filip	Filipović	1
11	Marta	Martić	1
12	Domagoj	Domagojić	1
13	Lucija	Lucić	1
14	Kristijan	Kristić	1
15	Tina	Tinić	1

Slika 41.

Treći upit (Slika 42.):

```
--Ovaj upit koristi ugniježdene upite kako bi usporedio maksimalnu početnu težinu (iz tablice pocetno_stanje) i
--minimalnu težinu nakon napretka (iz tablice osobni_napredak) za svakog klijenta.
SELECT k.ime, k.prezime,
       (SELECT MAX(ps.tezina) FROM pocetno_stanje ps WHERE ps.klijent_id = k.ID_klijenta) AS maksimalna_pocetna_tezina,
       (SELECT MIN(o.tezina) FROM osobni_napredak o WHERE o.klijent_id = k.ID_klijenta) AS minimalna_tezina_nakon_napretka
FROM klijent k;
```

Slika 42.

- Rezultat upita (Slika 43.):

	ime character varying (255) 🔒	prezime character varying (255) 🔒	maksimalna_pocetna_tezina numeric 🔒	minimalna_tezina_nakon_napretka numeric 🔒
1	Petar	Petrović	80.00	85.00
2	Karla	Karla	65.00	70.00
3	Luka	Lukić	70.00	75.00
4	Ema	Emić	60.00	95.00
5	Tomislav	Tomić	85.00	[null]
6	Ivana	Ivić	62.00	[null]
7	Sara	Sarić	75.00	[null]
8	Ante	Antić	78.00	[null]
9	Nina	Ninić	67.00	[null]
10	Filip	Filipović	80.00	[null]
11	Marta	Martić	70.00	[null]
12	Domagoj	Domagojić	72.00	[null]
13	Lucija	Lucić	63.00	[null]
14	Kristijan	Kristić	77.00	[null]
15	Tina	Tinić	68.00	[null]

Slika 43.

Četvrti upit (Slika 44.):

```
--Ovaj upit koristi INTERSECT kako bi pronašao klijente koji su unutar baze uključeni u  
--online programe (iz tablice program) i grupne treninge (iz tablice rad_s_grupama).  
--INTERSECT vraća samo one retke koji se pojavljuju u oba upita.  
SELECT k.ime, k.prezime  
FROM klijent_program kp  
JOIN klijent k ON kp.klijent_id = k.ID_klijenta  
WHERE kp.program_id IN (SELECT ID_programa FROM program WHERE vrsta = 'Online')  
INTERSECT  
SELECT k.ime, k.prezime  
FROM rad_s_grupama rg  
JOIN grupa g ON rg.grupa_id = g.grupa_id  
JOIN grupa_klijent kg ON kg.grupa_id = g.grupa_id  
JOIN klijent k ON k.ID_klijenta = kg.klijent_id;
```

Slika 44.

- Rezultat upita (Slika 45.):

	ime character varying (255) 🔒	prezime character varying (255) 🔒
1	Nina	Ninić
2	Tina	Tinić
3	Lucija	Lucić
4	Domagoj	Domagojić
5	Ante	Antić
6	Kristijan	Kristić
7	Marta	Martić
8	Filip	Filipović

Slika 45.

Peti upit (Slika 46.):

```
--Ovaj upit koristi UNION ALL kako bi spojio informacije o trenerima koji su sudjelovali u kreiranju
--u programima (iz tablice program) i radili s grupama (iz tablice rad_s_grupama).
--Dodatni stupac 'tip' označava iz kojeg dijela tablice dolaze podaci.
--UNION ALL vraća sve retke iz oba upita, uključujući i duplicirane retke ako postoje,
--sortirane prema imenu i prezimenu trenera.
-- Ovaj upit koristi UNION ALL kako bi spojio informacije o trenerima koji su sudjelovali u programima i radili s grupama.
SELECT t.ime, t.prezime, 'Program' AS tip
FROM program p
JOIN trener t ON p.trener_id = t.ID_trenera
UNION ALL
SELECT t.ime, t.prezime, 'Grupa' AS tip
FROM rad_s_grupama rg
JOIN trener t ON rg.trener_id = t.ID_trenera
ORDER BY ime, prezime, tip;
```

Slika 46.

- Dio rezultata upita (Slika 47.):

	ime character varying (255)	prezime character varying (255)	tip text
1	Ana	Anić	Grupa
2	Ana	Anić	Grupa
3	Ana	Anić	Grupa
4	Ana	Anić	Grupa
5	Ana	Anić	Grupa
6	Ana	Anić	Grupa
7	Ana	Anić	Grupa
8	Ana	Anić	Grupa
9	Ana	Anić	Grupa
10	Ana	Anić	Program
11	Ana	Anić	Program
12	Ivan	Ivić	Grupa
13	Ivan	Ivić	Grupa
14	Ivan	Ivić	Grupa
15	Ivan	Ivić	Grupa
16	Ivan	Ivić	Grupa
17	Ivan	Ivić	Grupa
18	Ivan	Ivić	Grupa
19	Ivan	Ivić	Grupa
20	Ivan	Ivić	Grupa
21	Ivan	Ivić	Program
22	Marko	Markić	Grupa

Slika 47.

8. ZADANE VRIJEDNOSTI, UVJETI I KOMENTARI

Zadane vrijednosti (Slika 48.):

- 1. zadana vrijednost (Slika 48.): svaki novi redak koji se unese u tablicu program automatski će imati zadano upozorenje, osim ako nije drugačije navedeno.
- 2. zadana vrijednost (Slika 48.): pri unosu različitih treninga, ukoliko se ne navedu potrebni rekviziti, uvijek će se unijeti zadana poruka o upotrebi rekvizita.

```
--postavljanje defaultnih vrijednosti

-- Dodavanje zadane vrijednosti za stupac upozorenja
ALTER TABLE program
ALTER COLUMN upozorenja SET DEFAULT 'Pazite prilikom izvođenja i ne prekoračujte svoje mogućnosti.';

-- Dodavanje zadane vrijednosti za stupac potrebni_rekviziti u tablici trening
ALTER TABLE trening
ALTER COLUMN potrebni_rekviziti SET DEFAULT 'Svaki trening se može odraditi i sa i bez rekvizita, ovisno o osobnoj spremi.';
```

Slika 48.

Primjeri nekih od uvjeta koji su dio kreiranja tablica:

Napomena: svi uvjeti su uneseni već prilikom kreiranja tablica.

- Prvi primjer uvjeta (Slika 49.): ovo ograničenje provjerava da se u stupac spol može unijeti samo vrijednost koja je ili 'M' (muški) ili 'F' (ženski). Ako se pokuša unijeti neka druga vrijednost, SQL će vratiti grešku. Ograničenje CHECK na stupcu visina osigurava da se unese samo pozitivna vrijednost (visina > 0). Ovo osigurava da se u tablicu ne mogu unijeti negativne vrijednosti visine ili vrijedost 0.

```
-- Table: klijent
CREATE TABLE klijent (
  ID_klijenta SERIAL PRIMARY KEY,
  ime VARCHAR(255) NOT NULL,
  prezime VARCHAR(255) NOT NULL,
  datum_rodjenja DATE NOT NULL,
  spol CHAR(1) CHECK (spol IN ('M', 'F')),
  email VARCHAR(255) NOT NULL UNIQUE,
  broj_telefona VARCHAR(20),
  visina FLOAT NOT NULL CHECK (visina > 0),
  cilj VARCHAR(100)
);
```

Slika 49.

- Drugi primjer uvjeta (Slika 50.): ovo ograničenje osigurava da se u stupac ocjena može unijeti samo vrijednost koja je tipa INT i u rasponu od 1 do 5, uključujući i granice.

```
-- Table: suradnja_recenzija
CREATE TABLE suradnja_recenzija (
  ID_recenzije SERIAL NOT NULL,
  trener_id INT NOT NULL,
  klijent_id INT NOT NULL,
  CONSTRAINT RECENZIJ_ID PRIMARY KEY (ID_recenzije, trener_id, klijent_id),
  datum_pocetka DATE NOT NULL,
  datum_zavrsetka DATE,
  vrsta_suradnje VARCHAR(100) NOT NULL,
  status VARCHAR(50) NOT NULL,
  ocjena FLOAT NOT NULL CHECK (ocjena BETWEEN 1 AND 5),
  dodatne_biljeske TEXT,
  ostvareni_napredak TEXT,
  FOREIGN KEY (klijent_id) REFERENCES klijent(ID_klijenta),
  FOREIGN KEY (trener_id) REFERENCES trener(ID_trenera)
);
```

Slika 50.

Komentari (Slika 51.):

```
COMMENT ON TABLE trener IS 'Tablica koja sadrži informacije o trenerima.';
COMMENT ON TABLE klijent IS 'Tablica koja sadrži informacije o klijentima.';
COMMENT ON TABLE grupa IS 'Tablica koja sadrži informacije o fitness grupama koje vodi trener.';
COMMENT ON TABLE grupa_klijent IS 'Poveznica tablica između grupa i klijenata.';
COMMENT ON TABLE rad_s_grupama IS 'Tablica koja bilježi rad trenera s grupama.';
COMMENT ON TABLE suradnja_recenzija IS 'Tablica koja bilježi recenzije suradnje između trenera i klijenata.';
COMMENT ON TABLE program IS 'Tablica koja sadrži informacije o fitness programima koje kreiraju treneri.';
COMMENT ON TABLE trening IS 'Tablica koja sadrži informacije o pojedinačnim treninzima u programima.';
COMMENT ON TABLE klijent_program IS 'Poveznica tablica između klijenata i programa.';
COMMENT ON TABLE klijent_trening IS 'Poveznica tablica između klijenata i pojedinačnih treninga.';
COMMENT ON TABLE pocetno_stanje IS 'Tablica koja sadrži početno stanje mjerenja klijenata.';
COMMENT ON TABLE osobni_napredak IS 'Tablica koja bilježi osobni napredak klijenata.';
COMMENT ON TABLE work_log IS 'Tablica koja bilježi radne logove između trenera i klijenata.';
```

Slika 51.

9. INDEKSI

Koristeći indekse, moguće je ubrzati određene upite. Budući da se često treba pretraživati klijente po imenima i prezimenima, kreirala sam indeks na tablici klijent za atribut prezime i ime jer se očekuje velik broj polaznika koji će biti i u bazi. Također, zbog raznovrsnosti ponude indeks bi bilo dobro imati i na tablici program koji će imati brzi pristup svim nazivima programa. Kreiran su i hash indeksi; na tablici klijent budući da atribut " spol " može poprimiti 2 specifične vrijednosti odnosno podudaranja te također na tablici program na stupcu vrsta koji također ima određene 2 mogućnosti vrijednosti.

```
-- indeksi
CREATE INDEX klijenti ON klijent(ime, prezime);

CREATE INDEX programi ON program (naziv);

CREATE INDEX klijent_spol ON klijent USING HASH(spol);

CREATE INDEX program_vrsta ON program USING HASH(vrsta);
|
```

Slika 52.

10. PROCEDURE

Procedure olakšavaju specifične upite, unose i ažuriranja koja se često izvode nad bazom. U nastavku slijede dvije procedure koje su kreirane za ovu bazu podataka.

Prva procedura (Slika 54.):

- Ova procedura omogućuje korisniku ili aplikaciji da dodaju novi trening u program s minimalnim naporom, a istovremeno osigurava da se dodajući trening pravilno zabilježi i da se korisniku pruži obavijest o uspješnom unosu.

```
513 -- Procedura za dodavanje novog treninga u program pri čemu se dobije obavijest
514 CREATE OR REPLACE PROCEDURE dodaj_trening_u_program(
515     p_naziv VARCHAR(255),
516     p_opis TEXT,
517     p_nivo_tezine INT,
518     p_trajanje FLOAT,
519     p_potrebni_rekviziti TEXT,
520     p_program_id INT
521 )
522 LANGUAGE PLPGSQL
523 AS $$
524 BEGIN
525     INSERT INTO trening (naziv, opis, nivo_tezine, trajanje, potrebni_rekviziti, program_id)
526     VALUES (p_naziv, p_opis, p_nivo_tezine, p_trajanje, p_potrebni_rekviziti, p_program_id);
527
528     RAISE NOTICE 'Dodan je novi trening "%".', p_naziv;
529 END $$;
530
531 CALL dodaj_trening_u_program('Trening za snagu',
532                               'Intenzivan trening za razvoj snage i izdržljivosti.',
533                               4, 90.5, 'Bučice, traka za trčanje', 1);
534
535
536
```

Data Output Messages Notifications

NOTICE: Dodan je novi trening "Trening za snagu".
CALL

Query returned successfully in 44 msec.

Slika 54.

Druga procedura (Slika 57.):

- Dana procedura, `treninzi_po_tipu`, prima naziv tipa treninga kao ulazni parametar (`tip`). Nakon toga, provjerava postoji li taj tip treninga u tablici `trening`. Ako postoji, izvršava upit koji vraća sve treninge s tim nazivom i njihovo trajanje te podiže obavijest o uspješnom prikazu. Ako tip treninga pak ne bude pronađen, procedura podiže iznimku s odgovarajućom porukom.

```
535 --Procedura koja ispisuje treninge željenog tipa ukoliko oni postoje u bazi
536 CREATE OR REPLACE PROCEDURE treninzi_po_tipu(IN tip VARCHAR(50))
537 LANGUAGE plpgsql
538 AS $$
539 BEGIN
540     RAISE NOTICE 'Prikazujem treninge za tip: %', tip;
541
542     IF NOT EXISTS (SELECT 1 FROM trening WHERE naziv = tip) THEN
543         RAISE EXCEPTION 'Tip treninga "%" nije pronađen.', tip;
544     END IF;
545
546     PERFORM ID_treninga, naziv, trajanje
547     FROM trening
548     WHERE naziv = tip;
549
550     RAISE NOTICE 'Prikazani su treninzi za tip: %', tip;
551 END $$;
552 CALL treninzi_po_tipu('Cardio Workout');
553
554
```

Data Output Messages Notifications

NOTICE: Prikazujem treninge za tip: Cardio Workout
NOTICE: Prikazani su treninzi za tip: Cardio Workout
CALL

Query returned successfully in 75 msec.

Slika 57.

11. OKIDAČI

U ovom poglavlju opisani su okidači kreirani za ovu bazu podataka.

Prvi okidač (Slika 58.):

- Ovaj okidač osigurava da se u work_log mogu unositi samo zapisi za online treninge, kako je definirano prema vrsti programa povezanoj s treningom u tablici program.

```
557 --Unos work_log-a samo za online tip treninga.
558 CREATE OR REPLACE FUNCTION provjeri_vrstu_programa()
559 RETURNS TRIGGER AS $$
560 DECLARE
561     vrsta_programa VARCHAR(50);
562 BEGIN
563     SELECT p.vrsta INTO vrsta_programa
564     FROM trening t
565     JOIN program p ON t.program_id = p.ID_programa
566     WHERE t.ID_treninga = NEW.trening_id;
567     IF vrsta_programa = 'Online' THEN
568         RETURN NEW;
569     ELSE
570         RAISE EXCEPTION 'Samo online trening može biti zapisan u work_log.';
571         RETURN NULL;
572     END IF;
573 END $$ LANGUAGE plpgsql;
574
575 CREATE OR REPLACE TRIGGER provjeri_online_trening
576 BEFORE INSERT ON work_log
577 FOR EACH ROW
578 EXECUTE FUNCTION provjeri_vrstu_programa();
579
580 -- Ovdje je primjer unosa u work_log, unosi se trening koji se izvodi uživo
581 INSERT INTO work_log (trening_id, klijent_id, vrijeme_kreiranja, komentar, opis_odradenog)
582 VALUES (10, 8, '2024-06-30', 'Dobar napredak postignut!', 'Odradili smo sve vježbe prema planu i klijent je bio iznimno motiviran.');
```

Data Output Messages Notifications

ERROR: Samo online trening može biti zapisan u work_log.
CONTEXT: PL/pgSQL function provjeri_vrstu_programa() line 15 at RAISE

Slika 58.

Drugi okidač (Slika 59.):

- S obzirom da postoji limit na broj polaznika grupe, prije dodavanja klijenta u grupu, ovaj okidač provjerava da li grupa već ima maksimalan broj klijenata.

```
584 --Prije dodavanja klijenta u grupu, okidač provjerava da li grupa već ima maksimalan broj klijenata.
585 CREATE OR REPLACE FUNCTION provjeri_max_br_polaznika()
586 RETURNS TRIGGER AS $$
587 DECLARE
588     broj_klijenata INT;
589 BEGIN
590     SELECT COUNT(*)
591     INTO broj_klijenata
592     FROM grupa_klijent
593     WHERE GRUPA_ID = NEW.GRUPA_ID;
594
595     IF broj_klijenata >= (SELECT max_br_polaznika FROM grupa WHERE GRUPA_ID = NEW.GRUPA_ID) THEN
596         RAISE EXCEPTION 'Grupa već ima maksimalan broj polaznika.';
597         RETURN NULL;
598     END IF;
599
600     RETURN NEW;
601 END $$ LANGUAGE plpgsql;
602
603 CREATE TRIGGER provjeri_polaznike
604 BEFORE INSERT ON grupa_klijent
605 FOR EACH ROW
606 EXECUTE FUNCTION provjeri_max_br_polaznika();
607
608 -- Ovdje je primjer dodavanja 6. člana u grupu od mogućih 5
609 INSERT INTO grupa_klijent (GRUPA_ID, klijent_id) VALUES (2, 12);
610
```

Data Output Messages Notifications

ERROR: Grupa već ima maksimalan broj polaznika.
CONTEXT: PL/pgSQL function provjeri_max_br_polaznika() line 11 at RAISE

Slika 59.

12. ZAKLJUČAK

Dakle, iako se svakodnevno čuje i gleda o njemu, fitnes sada dolazi u još jednom pogledu, u pogledu baza podataka. Kroz ovaj projekt, koji se možda čini kompleksnim, može se zaključiti da svaki dodatni slobodni izbor i proširenje liste entiteta te entiteta s više različitih atributa i veza bitno komplicira uspostavljanje optimalnog modela baze podataka. Od samog početka i zamisli pa do provedbe ideje u djelo uslijedilo je mnoštvo promjena stoga je zaključak da jednostavnost, kojom se ispunjavaju određeni zahtjevi, najviše doprinosi kvalitetnoj bazi podataka čija će korisnost biti mnogostruka.