

SVEUČILIŠTE J. J. STROSSMAYERA U OSIJEKU
FAKULTET PRIMIJENJENE MATEMATIKE I INFORMATIKE

MODERNI SUSTAVI BAZA PODATAKA

SEMINARSKI RAD

BAZA PODATAKA ZA CAFFE BAR

LANA MALTAR

OSIJEK, 2024.

SADRŽAJ

1. UVOD	3
2. MODEL BAZE PODATAKA	4
MEV model	4
Relacijski model	5
3. KREIRANJE TABLICA I UNOS PODATAKA	6
4. UPITI	7
Jednostavni upiti	7
Upiti nad više tablica	8
Upiti sa agregirajućim funkcijama	10
Podupiti i skupovne operacije	11
5. ZADANE VRIJEDNOSTI, UVJETI I KOMENTARI	13
6. INDEKSI	14
7. OKIDAČI	14
8. PROCEDURE	16
9. ZAKLJUČAK	17

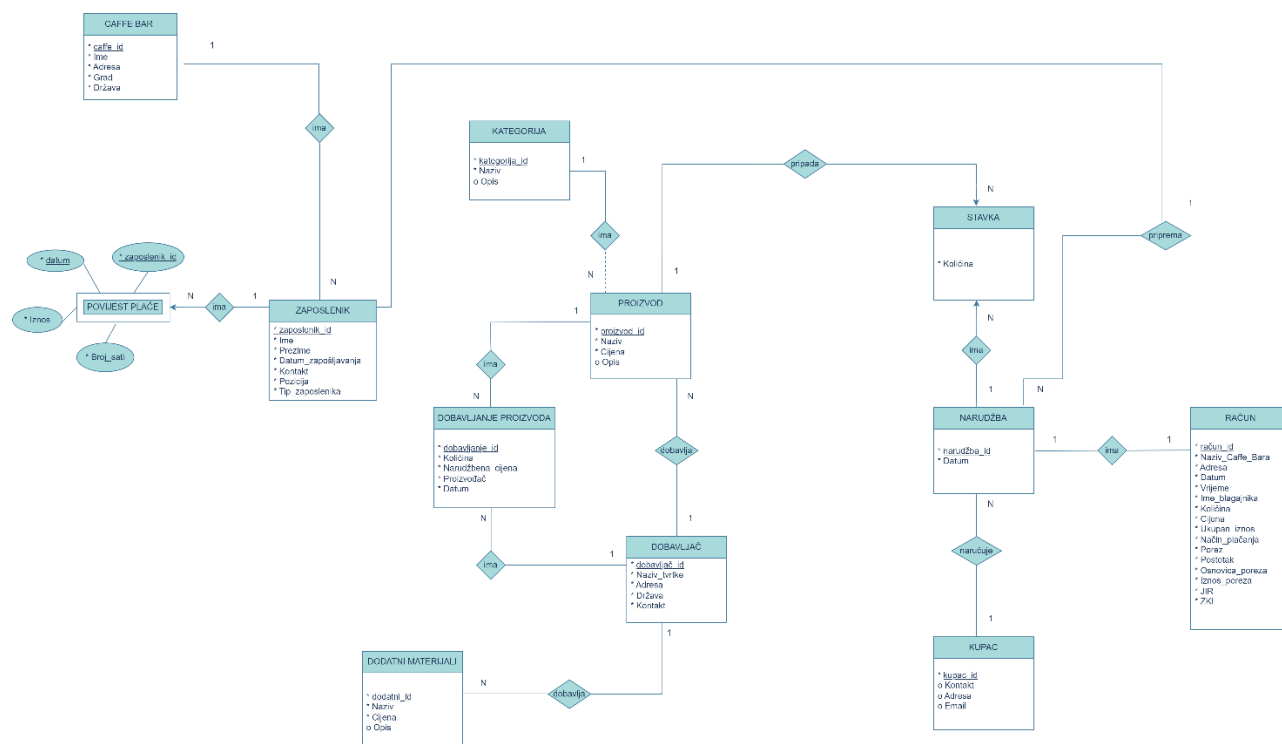
1. Uvod

U današnjem dinamičnom poslovnom okruženju, caffè barovi, kao i mnogi drugi ugostiteljski objekti, suočavaju se s izazovima upravljanja velikim količinama podataka. Ti podaci uključuju informacije o zalihama, narudžbama, zaposlenicima i kupcima. Upravljanje ovim podacima na učinkovit i organiziran način ključno je za održavanje visokog nivoa usluge, povećanje produktivnosti i ostvarivanje poslovnog uspjeha.

Baza podataka omogućava centraliziranu pohranu informacija, brzo i precizno pretraživanje te sigurnu manipulaciju podacima. Iz tih razloga odlučila sam izraditi i predstaviti bazu podataka koja će unaprijediti poslovanje jednog caffè bara.

2. Model baze podataka

MEV model



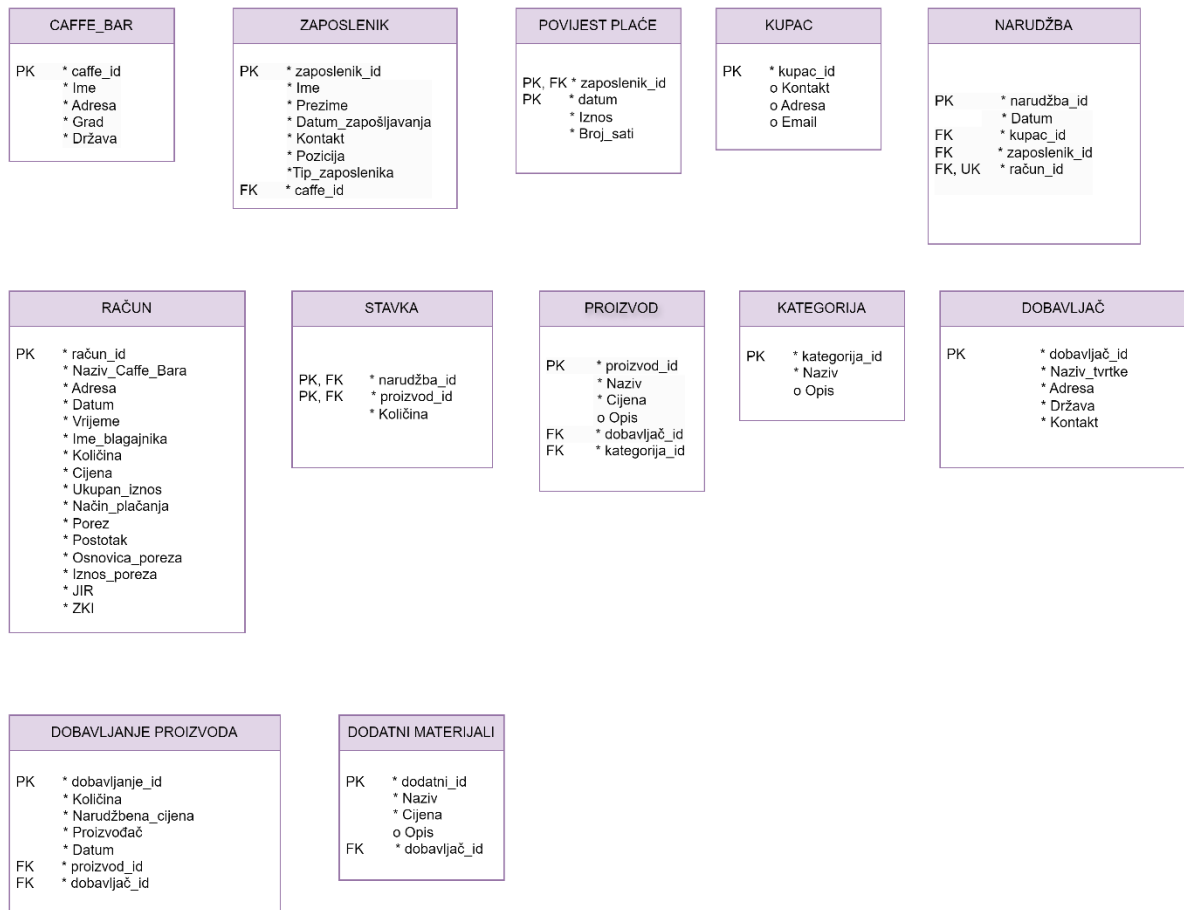
Model entiteta i veza sastoji se od 12 entiteta. Veze između entiteta su uglavnom jedan naprema više te jedan naprema jedan.

Za bazu podataka prvenstveno je potrebno imati entitet `caffe_bar` koji sadrži informacije o imenu i lokaciji caffe bara. Svaki caffe bar ima zaposlenike pa tako stvaramo entitet `zaposlenik` koji sadrži osnovne informacije o zaposleniku kao što su ime, prezime, datum zapošljavanja, kontakt, radna pozicija te tip zaposlenika koji može biti osoblje ili upravitelj. Svaki zaposlenik ima povijest plaće koja predstavlja egzistencijalno i identifikacijski slab entitet. Zaposlenik priprema narudžbu, a entitet `narudžba` povezan je sa entitetima `kupac` i `racun` koji sadrže osnovne informacije vezane uz ove entite. Svako narudžbi pripadaju proizvodi pa pomoću entiteta `stavka` pratimo količinu proizvoda koja pripada određenoj narudžbi. Proizvod ima jednu od sedam mogućih kategorija (bezalkoholna pića, topli napitci, pivo, vino, alkoholna pića, cideri, kokteli) te dobavljača. Pomoću entiteta `dobavljanje_proizvoda` pratimo količinu te narudžbenu cijenu proizvoda kao i datum dobavljanja. Dobavljač također dobavlja dodatne materijale poput šećera, mlijeka i voća.

Caffe bar može imati više zaposlenika dok zaposlenik mora pripadati jednom radnom mjestu. Budući da je povijest plaće egzistencijalno i identifikacijski slab entitet povezan je sa zaposlenikom strijelicom koja ide iz smjera entiteta zaposlenik. Zaposlenik priprema više

narudžbi, ali jednu narudžbu priprema samo jedan zaposlenik. Računu pripada jedna narudžba, dok kupac može napraviti više narudžbi, također narudžbi može pripadati više proizvoda. Proizvodi i dodatni materijali imaju jednog dobavljača, dok dobavljač dobavlja više proizvoda i dodatnih materijala. Proizvod također mora imati jednu kategoriju, dok određenim kategorijama ne mora pripadati nijedan proizvod zbog opcionalnosti veze.

Relacijski model



Prema pravilima za pretvorbu MEV modela u relacijski dobivamo relacijski model. Zbog veze između entiteta zaposlenik i caffe_bar, u tablicu zaposlenik dodajemo strani ključ caffe_id. U tablicu povijest_plaće dodajemo primarni i strani ključ zaposlenik_id. Tablici narudžba dodajemo strane ključeve zaposlenik_id i kupac_id te strani i jedinstveni ključ račun_id radi veze jedan naprema jedan između ovih entiteta. Proizvod ima strane ključeve dobavljač_id i kategorija_id, a entitet stavka sadrži dva primarna i dva strana ključa narudžba_id i proizvod_id zbog načina na koji je stavka povezana sa tim entitetima. Tablica dodatni_materijali također ima strani ključ dobavljač_id, dok tablica dobavljanje_proizvoda ima strane ključeve proizvod_id i dobavljač_id.

3. Kreiranje tablica i unos podataka

Tablice su kreirane u potpunosti u skladu s definiranim relacijskim modelom. U nastavku se nalaze primjeri SQL koda za kreiranje i unos podataka. Sve tablice i unosi prikazani su u priloženoj SQL skripti.

Kreiranje

```
CREATE TABLE ZAPOSLENIK (  
    zaposlenik_id INTEGER PRIMARY KEY,  
    ime VARCHAR(50) NOT NULL,  
    prezime VARCHAR(50) NOT NULL,  
    datum_zaposljavanja DATE NOT NULL,  
    kontakt VARCHAR(50) NOT NULL,  
    pozicija VARCHAR(50) NOT NULL,  
    tip_zaposlenika VARCHAR(50) NOT NULL,  
    caffe_id INTEGER NOT NULL CONSTRAINT caffe_id_fk REFERENCES CAFFE_BAR(caffe_id)  
);
```

Unos podataka

```
INSERT INTO KATEGORIJA (kategorija_id, naziv)  
VALUES  
    (1, 'Bezalkoholna pića'),  
    (2, 'Topli napitci'),  
    (3, 'Vino'),  
    (4, 'Pivo'),  
    (5, 'Alkoholna pića'),  
    (6, 'Cider'),  
    (7, 'Kokteli');
```

4. Upiti

Jednostavni upiti

- Upit 1: Ispis svih dodatnih materijala.

```
SELECT *  
FROM DODATNI_MATERIJALI;
```

	dodatni_id [PK] integer	naziv character varying (50)	cijena numeric (10,2)	opis text	dobavljac_id integer
1	1	Bijeli šećer	1.00	Bijeli šećer	1
2	2	Smeđi šećer	1.00	Smeđi šećer	1
3	3	Mlijeko	0.75	Mlijeko	1
4	4	Naranče	1.25	Naranče za cijeđeni sok	1

- Upit 2: Ispis svih informacija o zaposlenicima koji pripadaju osoblju.

```
SELECT *  
FROM ZAPOSLENIK  
WHERE tip_zaposlenika = 'Osoblje';
```

	zaposlenik_id [PK] integer	ime character varying (50)	prezime character varying (50)	datum_zaposljavanja date	kontakt character varying (50)	pozicija character varying (50)	tip_zaposlenika character varying (50)	caffe_id integer
1	1	Lana	Ivanić	2023-05-05	095456123	Konobar	Osoblje	1
2	2	Ivan	Ivanić	2023-07-23	097257152	Konobar	Osoblje	1

- Upit 3: Ispis svih proizvoda kojima je cijena veća od 2 eura.

```
SELECT proizvod_id, naziv, cijena, kategorija_id  
FROM PROIZVOD  
WHERE cijena > 2;
```

	proizvod_id [PK] integer	naziv character varying (100)	cijena numeric (10,2)	kategorija_id integer
1	8	Red Bull	2.50	1
2	28	Vermouth	20.00	3
3	29	Ožujsko	2.20	4
4	30	Ožujsko radler	2.20	4
5	31	Staropramen	2.20	4
6	32	Tomislav	2.20	4
7	33	Becks	2.20	4
8	34	Stella Artois	2.20	4
9	35	Osječko	2.20	4
10	36	Heineken	2.20	4
11	37	Corona	3.00	4
12	38	Strano craft pivo	3.00	4
13	39	Točeno pivo	3.00	4
14	40	Somersby	2.40	6
15	41	Mojito	2.39	7
16	42	Cuba Libre	2.39	7
17	43	Sex on the beach	2.39	7
18	44	Cosmopolitan	2.39	7
19	45	Vodka	2.10	5
20	48	Vilijamovka	2.50	5
21	49	Pelinkovac Antique	2.20	5
22	50	Stock	2.70	5
23	51	Amaro	2.40	5
24	52	Gin	2.60	5
25	53	Jagermeister	2.30	5
26	56	Whiskey	2.70	5
27	57	Jack Daniels	2.20	5

- Upit 4: Naziv proizvoda koji pripadaju kategoriji “Topli napitci”.

```
SELECT naziv
FROM PROIZVOD
WHERE kategorija_id = 2;
```

	naziv character varying (100) 🔒
1	Kava mala
2	Kava produžena
3	Caffe Latte
4	Kava šlag-mlijeko
5	Cappuccino
6	Bijela kava
7	Nescafe
8	Topla čokolada
9	Ledena kava
10	Čaj
11	Kakao
12	Kava bez kofeina

- Upit 5: Ispis svih računa koji su izdani 2024. godine.

```
SELECT *
FROM RACUN
WHERE EXTRACT(YEAR FROM datum) = 2024;
```

	racun_id [PK] integer 🔒	naziv_caffe_bara character varying (50) 🔒	adresa character varying (100) 🔒	datum date 🔒	vrijeme time without time zone 🔒	ime_blagajnika character varying (50) 🔒	kolicina integer 🔒	ukupan_iznos numeric (10,2) 🔒	nacin_placanja character varying (50) 🔒	porez numeric (5,2) 🔒	postotak numeric (5,2) 🔒	osnovica_poreza numeric (10,2) 🔒	iznos_poreza numeric (10,2) 🔒
1	2	Caffe Bar Passage	Trg kralja Tomislava 16	2024-06-01	10:14:00	Lana Ivanic	5	9.50	Gotovina	10.00	3.00	50.00	2.50
2	1	Caffe Bar Passage	Trg kralja Tomislava 16	2024-05-31	18:30:00	Lana Ivanic	5	14.20	Gotovina	10.00	3.00	3.20	0.16

Upiti nad više tablica

- Upit 1: Ime, prezime i pozicija zaposlenika koji radi u određenom caffe baru.

```
SELECT Z.ime, Z.prezime, Z.pozicija, CB.ime "Caffe_bar", CB.grad
FROM ZAPOSLENIK Z JOIN CAFFE_BAR CB ON Z.caffe_id = CB.caffe_id;
```

	ime character varying (50) 🔒	prezime character varying (50) 🔒	pozicija character varying (50) 🔒	Caffe_bar character varying (50) 🔒	grad character varying (50) 🔒
1	Lana	Ivanić	Konobar	Caffe Bar Passage	Osijek
2	Ivan	Ivanić	Konobar	Caffe Bar Passage	Osijek
3	Marko	Marković	Konobar	Caffe Bar Passage	Osijek

- Upit 2: Ispis naziva proizvoda, količine i ukupne cijene proizvoda koji pripada određenoj narudžbi.

```
SELECT N.narudzba_id, P.naziv "Proizvod", S.kolicina, P.cijena, S.kolicina * P.cijena "Ukupna_cijena"
FROM NARUDZBA N JOIN STAVKA S USING(narudzba_id)
JOIN PROIZVOD P USING(proizvod_id)
ORDER BY N.narudzba_id, P.naziv;
```


	narudzba_id integer	Proizvod character varying (100)	kolicina integer	cijena numeric (10,2)	Ukupna_cijena numeric
1	1	Coca-Cola	2	2.00	4.00
2	1	Coca-Cola Zero Sugar	1	2.00	2.00
3	1	Fanta	1	2.00	2.00
4	1	Kava mala	3	1.40	4.20
5	2	Schweppes	2	2.00	4.00
6	2	Sprite	3	2.00	6.00

- Upit 3: Ispis svih proizvoda po kategorijama.

```
SELECT K.naziv "Kategorija", P.naziv "Proizvod", P.cijena
FROM KATEGORIJA K JOIN PROIZVOD P USING(kategorija_id)
ORDER BY K.naziv, P.naziv;
```

	Kategorija character varying (50)	Proizvod character varying (100)	cijena numeric (10,2)
1	Alkoholna pića	Amaro	2.40
2	Alkoholna pića	Gin	2.60
3	Alkoholna pića	Jack Daniels	2.20
4	Alkoholna pića	Jagermeister	2.30
5	Alkoholna pića	Likeri	1.90
6	Alkoholna pića	Pelinkovac Antique	2.20
7	Alkoholna pića	Rakije	2.00
8	Alkoholna pića	Rum	1.50
9	Alkoholna pića	Stock	2.70
10	Alkoholna pića	The Famous Grouse	1.70
11	Alkoholna pića	Vilijamovka	2.50
12	Alkoholna pića	Vodka	2.10
13	Alkoholna pića	Vodka voćna	1.80
14	Alkoholna pića	Whiskey	2.70
15	Bezalkoholna pića	Cedevita	1.50
16	Bezalkoholna pića	Coca-Cola	2.00
17	Bezalkoholna pića	Coca-Cola Zero Sugar	2.00
18	Bezalkoholna pića	Fanta	2.00
19	Bezalkoholna pića	FUZETEA	2.00
20	Bezalkoholna pića	Red Bull	2.50
21	Bezalkoholna pića	Romeequelle	1.60
22	Bezalkoholna pića	Schweppes	2.00

Napomena; zbog velikog broja proizvoda izdvojila sam samo određeni dio rezultata upita

- Upit 4: Ispis imena i prezimena zaposlenika koji su pripremili određenu narudžbu.

```
SELECT N.narudzba_id, Z.ime || ' ' || Z.prezime "Ime prezime zaposlenika"
FROM NARUDZBA N JOIN ZAPOSLENIK Z ON N.zaposlenik_id = Z.zaposlenik_id;
```

	narudzba_id [PK] integer	Ime prezime zaposlenika text
1	1	Lana Ivanić
2	2	Lana Ivanić

- Upit 5: Ispis ukupne sume svih plaća određenog zaposlenika.

```
SELECT Z.ime, Z.prezime, SUM(PP.iznos) "Ukupna_zarada"
FROM ZAPOSLENIK Z JOIN POVIJEST_PLACE PP ON Z.zaposlenik_id = PP.zaposlenik_id
GROUP BY Z.ime, Z.prezime;
```

	ime character varying (50)	prezime character varying (50)	Ukupna_zarada numeric
1	Ivan	Ivanić	650.00
2	Lana	Ivanić	950.00
3	Marko	Marković	1070.00

Upiti sa agregirajućim funkcijama

- Upit 1: Ispis ukupne cijene narudžbe.

```
SELECT N.narudzba_id, SUM(S.kolicina * P.cijena) "Ukupna_cijena_narudzbe"
FROM NARUDZBA N JOIN STAVKA S ON N.narudzba_id = S.narudzba_id
      JOIN PROIZVOD P ON S.proizvod_id = P.proizvod_id
GROUP BY N.narudzba_id;
```

	narudzba_id [PK] integer	Ukupna_cijena_narudzbe numeric
1	2	10.00
2	1	12.20

- Upit 2: Broj proizvoda po kategorijama.

```
SELECT N.narudzba_id, SUM(S.kolicina * P.cijena) "Ukupna_cijena_narudzbe"
FROM NARUDZBA N JOIN STAVKA S ON N.narudzba_id = S.narudzba_id
      JOIN PROIZVOD P ON S.proizvod_id = P.proizvod_id
GROUP BY N.narudzba_id;
```

	Kategorija character varying (50)	Broj_proizvoda bigint
1	Cider	1
2	Pivo	11
3	Bezalkoholna pića	9
4	Alkoholna pića	14
5	Topli napitci	12
6	Vino	7
7	Kokteli	4

- Upit 3: Prosječni broj odrađenih sati svih zaposlenika.

```
SELECT AVG(broj_sati) "Prosjecni_broj_sati"
FROM POVIJEST_PLACE;
```

	Prosjecni_broj_sati numeric
1	130.000000000000000000

- Upit 4: Ispis maksimalne cijene proizvoda po kategoriji.

```
SELECT K.naziv "Kategorija", MAX(P.cijena) "Maksimalna_cijena"
FROM KATEGORIJA K LEFT JOIN PROIZVOD P USING(kategorija_id)
GROUP BY K.naziv;
```

	Kategorija character varying (50)	Maksimalna_cijena numeric
1	Cider	2.40
2	Pivo	3.00
3	Bezalkoholna pića	2.50
4	Alkoholna pića	2.70
5	Topli napitci	2.00
6	Vino	20.00
7	Kokteli	2.39

- Upit 5: Ukupan broj prodanih proizvoda po kategoriji.

```
SELECT K.naziv "Kategorija", SUM(S.kolicina) "Ukupan_broj_prodanih"
FROM KATEGORIJA K JOIN PROIZVOD P USING(kategorija_id)
      JOIN STAVKA S USING(proizvod_id)
GROUP BY K.naziv;
```

	Kategorija character varying (50)	Ukupan_broj_prodanih bigint
1	Bezalkoholna pića	9
2	Topli napitci	3

Podupiti i skupovne operacije

- Upit 1: Podupit koji vraća ime i prezime zaposlenika kojima je broj odrađenih sati veći od 80.

```
SELECT ime, prezime
FROM ZAPOSLENIK
WHERE zaposlenik_id IN (
  SELECT zaposlenik_id
  FROM POVIJEST_PLACE
  WHERE broj_sati > 80
);
```

	ime character varying (50)	prezime character varying (50)
1	Lana	Ivanić
2	Ivan	Ivanić
3	Marko	Marković

- Upit 2: Podupit koji vraća ID kupca te ukupan iznos njegove narudžbe.

```
SELECT K.kupac_id, (
  SELECT SUM(R.ukupan_iznos)
  FROM NARUDZBA N JOIN RACUN R ON N.racun_id = R.racun_id
  WHERE N.kupac_id = K.kupac_id
) "Ukupan_iznos"
FROM KUPAC K;
```

	kupac_id [PK] integer	Ukupan_iznos numeric
1	1	14.20
2	5	9.50

- Upit 3: Podupit koji ispisuje naziv i cijenu proizvoda koji pripada kategoriji “Bezalkoholna pića” i ima cijenu veću od prosječne cijene svih proizvoda.

```
SELECT naziv, cijena
FROM PROIZVOD
WHERE kategorija_id = (
    SELECT kategorija_id
    FROM KATEGORIJA
    WHERE naziv = 'Bezalkoholna pića' )
    AND cijena > (
    SELECT AVG(cijena)
    FROM PROIZVOD
);
```

	naziv character varying (100)	cijena numeric (10,2)
1	Red Bull	2.50

- Upit 4: Primjer upita koji koristi skupovnu operaciju union i ispisuje kontakte zaposlenika i kupaca.

```
SELECT kontakt
FROM ZAPOSLENIK
UNION
SELECT kontakt
FROM KUPAC;
```

	kontakt character varying (50)
1	[null]
2	095251054
3	095456123
4	097257152
5	095632425

- Upit 5: Primjer upita koji koristi skupovnu operaciju except i ispisuje ime i prezime svih zaposlenika koji nisu pripremili niti jednu narudžbu.

```
SELECT ime, prezime
FROM ZAPOSLENIK
EXCEPT
SELECT Z.ime, Z.prezime
FROM NARUDZBA N JOIN ZAPOSLENIK Z USING (zaposlenik_id);
```

	ime character varying (50)	prezime character varying (50)
1	Ivan	Ivanić
2	Marko	Marković

5. Zadane vrijednosti, uvjeti i komentari

Zadane vrijednosti

- Postavljanje zadane vrijednosti stupca datum u tablici NARUDZBA na trenutni datum.

```
ALTER TABLE NARUDZBA  
ALTER COLUMN datum SET DEFAULT CURRENT_DATE;
```

- Postavljanje zadane vrijednosti stupca drzava u tablici CAFFE_BAR na "Hrvatska".

```
ALTER TABLE CAFFE_BAR  
ALTER COLUMN drzava SET DEFAULT 'Hrvatska';
```

- Stupac opis u tablici KATEGORIJA postavljen je na zadanu vrijednost "Nema opisa".

```
ALTER TABLE KATEGORIJA  
ALTER COLUMN opis SET DEFAULT 'Nema opisa';
```

Uvjeti

- Uvjet 1: Koristimo CHECK kako bi provjerili je li vrijednost stupca kolicina u tablici STAVKA veća od 0.

```
ALTER TABLE STAVKA  
ADD CONSTRAINT stavka_kolicina_ck  
CHECK (kolicina > 0);
```

- Uvjet 2: Koristimo CHECK kako bi provjerili je li cijena proizvoda veća od 0.

```
ALTER TABLE PROIZVOD  
ADD CONSTRAINT proizvod_cijena_ck  
CHECK (cijena > 0);
```

- Uvjet 3: Osiguravamo da svaki proizvod ima naziv korištenjem ograničenja SET NOT NULL.

```
ALTER TABLE PROIZVOD  
ALTER COLUMN naziv SET NOT NULL;
```

- Uvjet 4: Vrijednost u stupcu email mora biti jedinstvena za svaki redak i koristimo UNIQUE ograničenje.

```
ALTER TABLE KUPAC  
ADD CONSTRAINT kupac_email_unique  
UNIQUE (email);
```

Komentari

```
COMMENT ON TABLE CAFFE_BAR IS 'Tablica sadrži informacije o Caffè Baru.';  
COMMENT ON TABLE DOBAVLJAC IS 'Tablica sadrži informacije o dobavljaču proizvoda.';  
COMMENT ON TABLE DOBAVLJANJE_PROIZVODA IS 'Tablica sadrži informacije o dobavljanju proizvoda.';  
COMMENT ON TABLE DODATNI_MATERIJALI IS 'Tablica sadrži informacije o dodatnim materijalima.';  
COMMENT ON TABLE KATEGORIJA IS 'Tablica sadrži informacije o kategorijama proizvoda.';  
COMMENT ON TABLE KUPAC IS 'Tablica sadrži informacije o kupcima.';  
COMMENT ON TABLE NARUDZBA IS 'Tablica sadrži informacije o narudžbi.';  
COMMENT ON TABLE POVIJEST_PLACE IS 'Tablica sadrži povijest plaće zaposlenika.';  
COMMENT ON TABLE PROIZVOD IS 'Tablica sadrži detalje proizvoda.';  
COMMENT ON TABLE RACUN IS 'Tablica sadrži informacije računa.';  
COMMENT ON TABLE STAVKA IS 'Tablica sadrži informacije o količini proizvoda u narudžbi.';  
COMMENT ON TABLE ZAPOSLENIK IS 'Tablica sadrži informacije o zaposlenicima.';
```

6. Indeksi

- Indeks 1: Indeks je kreiran na tablici POVIJEST_PLACE i na stupcu datum. Služi kako bi se ubrzali upiti koji filtriraju ili sortiraju podatke na temelju datuma.

```
CREATE INDEX povijest_place_datum_index ON POVIJEST_PLACE (datum);
```

- Indeks 2: Ubrzava upite koji filtriraju proizvode po kategoriji. Ovaj indeks sam odabrala radi čestih korištenja upita nad kategorijom proizvoda.

```
CREATE INDEX proizvod_kategorija_index ON PROIZVOD (kategorija_id);
```

- Indeks 3: Namjena ovog indeksa je ubrzavanje upita koji traže proizvode po cijeni ili sortiraju proizvode prema cijeni. Indeks je napravljen u tablici PROIZVOD na stupcu cijena.

```
CREATE INDEX proizvod_cijena_index ON PROIZVOD (cijena);
```

7. Okidači

- Okidač 1: Ovaj okidač osigurava da nijedan proizvod ne može biti unesen ili ažuriran s cijenom manjom ili jednakom nuli. Funkcija provjeri_cijenu provjerava vrijednost stupca cijena za novi ili ažurirani redak.

```
CREATE OR REPLACE FUNCTION provjeri_cijenu()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
    IF NEW.cijena <= 0 THEN
        RAISE EXCEPTION 'Cijena proizvoda mora biti veća od 0!';
    END IF;
    RETURN NEW;
END;
$$;
```

```
CREATE TRIGGER provjeri_cijenu_trigger
BEFORE INSERT OR UPDATE ON PROIZVOD
FOR EACH ROW
EXECUTE FUNCTION provjeri_cijenu();
```

- Okidač 2: Okidač služi kako bi osigurali da svaka nova narudžba mora biti povezana s postojećim računom.

```
CREATE OR REPLACE FUNCTION provjeri_racun()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
    IF NOT EXISTS (
        SELECT 1
        FROM RACUN
        WHERE racun_id = NEW.racun_id)
    THEN
        RAISE EXCEPTION 'Narudžba mora imati povezan račun!';
    END IF;
```

```

    RETURN NEW;
END;
$$;

CREATE TRIGGER provjeri_racun_trigger
BEFORE INSERT ON NARUDZBA
FOR EACH ROW
EXECUTE FUNCTION provjeri_racun();

```

- Okidač 3: Ovaj okidač osigurava da ne može biti unesena stavka narudžbe ako nema dovoljno dostupne količine proizvoda. Funkcija provjeri_kolicinu provjerava ukupnu dostupnu količinu proizvoda, a ako je dostupna količina manja od količine u novoj stavci, pokreće se iznimka s porukom.

```

CREATE OR REPLACE FUNCTION provjeri_kolicinu()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
DECLARE
    dostupno INTEGER;
BEGIN
    SELECT SUM(kolicina) INTO dostupno
    FROM DOBAVLJANJE_PROIZVODA
    WHERE proizvod_id = NEW.proizvod_id;
    IF dostupno < NEW.kolicina THEN
        RAISE EXCEPTION 'Nema dovoljne količine proizvoda sa ID-jem %.', NEW.proizvod_id;
    END IF;
    RETURN NEW;
END;
$$;

CREATE TRIGGER kolicina_trigger
BEFORE INSERT ON STAVKA
FOR EACH ROW
EXECUTE FUNCTION provjeri_kolicinu();

```

- Okidač 4: Okidač automatski ažurira ukupnu cijenu računa nakon umetanja novog proizvoda u narudžbu. Funkcija azuriraj_ukupnu_cijenu računa iznos nove stavke kao umnožak količine i cijene proizvoda te taj iznos dodaje ukupnoj cijeni računa.

```

CREATE OR REPLACE FUNCTION azuriraj_ukupnu_cijenu()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE RACUN
    SET ukupan_iznos = ukupan_iznos + (NEW.kolicina * (SELECT cijena
                                                         FROM PROIZVOD
                                                         WHERE proizvod_id = NEW.proizvod_id))
    WHERE racun_id = NEW.narudzba_id;
    RETURN NEW;
END;
$$;

CREATE TRIGGER cijena_trigger
AFTER INSERT ON STAVKA
FOR EACH ROW
EXECUTE FUNCTION azuriraj_ukupnu_cijenu();

```

8. Procedure

- Procedura 1:

```
CREATE OR REPLACE PROCEDURE dodaj_proizvod(
    IN p_id INTEGER,
    IN p_naziv VARCHAR(100),
    IN p_cijena DECIMAL(10, 2),
    IN p_opis TEXT,
    IN p_dobavljac_id INTEGER,
    IN p_kategorija_id INTEGER
)
LANGUAGE plpgsql
AS $$
BEGIN
    INSERT INTO PROIZVOD (proizvod_id, naziv, cijena, opis, dobavljac_id, kategorija_id)
    VALUES (p_id, p_naziv, p_cijena, p_opis, p_dobavljac_id, p_kategorija_id);
END;
$$;
```

- Procedura 2:

```
CREATE OR REPLACE PROCEDURE obrisi_proizvod(
    IN p_proizvod_id INTEGER
)
LANGUAGE plpgsql
AS $$
BEGIN
    DELETE FROM PROIZVOD
    WHERE proizvod_id = p_proizvod_id;
END;
$$;
```

Za svoju bazu podataka izradila sam procedure koje olakšavaju unos i brisanje podataka, nudeći lakše rukovanje s podacima i poboljšavajući njihovu sigurnost i točnost.

Procedura `dodaj_proizvod` definirana je za unos novih proizvoda u bazu podataka, a prihvaća šest ulaznih parametara: `p_id`, `p_naziv`, `p_cijena`, `p_opis`, `p_dobavljac_id`, i `p_kategorija_id`. Unutar procedure, ovi parametri se koriste za izvršavanje `INSERT` naredbe koja dodaje novi redak u tablicu `PROIZVOD`.

Procedura `obrisi_proizvod` služi za brisanje postojećih proizvoda iz baze podataka. Prihvaća jedan ulazni parametar, `p_proizvod_id`, koji predstavlja identifikator proizvoda koji se želi ukloniti. Ova procedure izvršava `DELETE` naredbu koja uklanja redak iz tablice `PROIZVOD` gdje je `proizvod_id` jednak vrijednosti `p_proizvod_id`.

9. Zaključak

Implementacija baze podataka za caffe bar predstavlja ključni korak prema efikasnijem poslovanju i unaprjeđenju korisničkog iskustva. Omogućuje precizno praćenje i brzu obradu podataka te točan i brz pristup informacijama. Nadalje, bazu podataka je moguće nadograditi dodatnim funkcionalnostima poput integracije s mobilnim aplikacijama za online narudžbe ili implementacijom sustava za povratne informacije gostiju, čime bi se dodatno unaprijedila usluga. Unatoč mogućnostima za daljnje unapređenje, trenutna implementacija bila je dovoljna za ispunjenje osnovnih potreba projekta.