

SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

SEMINARSKI RAD

kolegij Baze podataka

BAZA PODATAKA ZA RESTORAN

Računarstvo – preddiplomski studij (120)
Akademska godina 2020/21

Veronika Dumić
Božo Durdov
David Čeko

Sadržaj

1. UVOD	3
1.1 Korisnički zahtjevi.....	3
2. MODELIRANJE PODATAKA	4
2.1 Uočavanje tipova entiteta i veza	4
2.2 Kreiranje ERD	5
2.3 Relacijska shema	6
2.4 Tipovi entiteta.....	7
2.5 Tipovi veza	8
2.6 Normalizacija	9
3. STVARANJE BAZE PODATAKA.....	10
4. RAD S BAZOM PODATAKA	14
4.1 Upiti.....	14
4.2 Pogledi.....	18
4.3 Transakcije	20

1. UVOD

Cilj ovog rada je dokumentirati proces izrade baze podataka za praćenje poslovanja jednog restorana. U radu su opisani svi koraci izrade baze podataka – od analize korisničkih zahtjeva do završne baze koja udovoljava tim zahtjevima. Pri izradi baze podataka koristile su se zakonitosti izrade relacijskih baza podataka i besplatan sustav za upravljanje bazama podataka otvorenog koda PostgreSQL.

1.1 Korisnički zahtjevi

Napraviti bazu podataka za praćenje poslovanja jednog restorana. Potrebno je voditi evidenciju o svim jelima koja se u restoranu spravljaju i namirnicama koje su potrebne za njihovu pripremu. Nadalje, potrebno je omogućiti planiranje nabave namirnica, vodeći računa o količinama namirnica koje odlaze u pripravu pojedinog jela, ali i njihovom predviđenom roku trajanja. Kako bi to bilo moguće, zahtijeva se bilježenje podataka o jelima koja se nalaze na dnevnom meniju restorana i količinama porcija pojedinog jela koje se pripravljaju.

Iz postojeće evidencije očekuje se da se mogu dobiti sljedeći izvještaji:

- izvještaj o sastojcima koji odlaze u pripravu pojedinog jela i njihovoj trenutnoj raspoloživosti.
- izvještaj o sastojcima koje je bilo/će biti potrebno nabaviti unutar zadanog vremenskog perioda.
- izvještaj o svim jelima u kojima se koristi određena namirnica.
- popis svih namirnica kojih nema u dovoljnim količinama za pripravu određenog jela.
- popis svih jela koja se mogu pripremiti od raspoloživih namirnica.
- pregled svih jela pripremljenih u određenom vremenskom periodu.
- ...

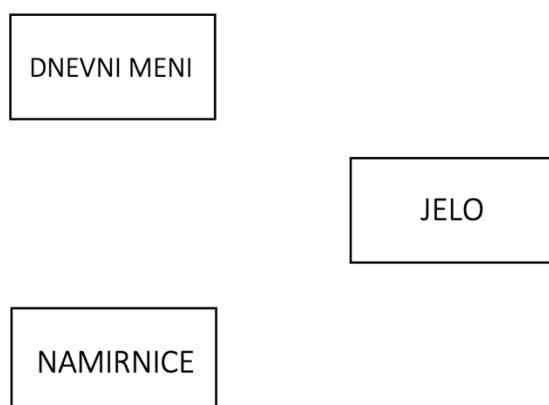
2. MODELIRANJE PODATAKA

Modeliranje podataka je tehnika organiziranja i dokumentiranja podataka sustava. Metodologija korištena u ovom radu naziva se metoda entiteti-veze (Entity-Relationship model). Dobivena shema je ER dijagram koji se vrlo lako transformira u relacijsku logičku shemu baze podataka. Za kreiranje ERD-a korišten je alat za izradu dijagrama toka draw.io.

2.1 Uočavanje tipova entiteta i veza

U tekstu koji specificira korisničke zahtjeve pokušat ćemo uočiti entitete (crveno) i veze među njima (plavo).

Napraviti bazu podataka za praćenje poslovanja jednog restorana. Potrebno je voditi evidenciju o **svim jelima** koja se u restoranu **spravlja** i **namirnicama** koje su **potrebne** za njihovu pripremu. Nadalje, potrebno je omogućiti planiranje nabave namirnica, vodeći računa o količinama namirnica koje odlaze u pripravu pojedinog jela, ali i njihovom predviđenom roku trajanja. Kako bi to bilo moguće, zahtijeva se bilježenje podataka o **jelima** koja **se nalaze** na **dnevnom meniju** restorana i količinama porcija pojedinog jela koje se pripravlja.



Slika 2.1.1 . Kandidati za tipove entiteta

2.2 Kreiranje ERD

Nakon daljnje analize zahtjeva kreiran je Entity-Relationship dijagram koji sadrži sljedeće tipove entiteta s pripadajućim atributima:

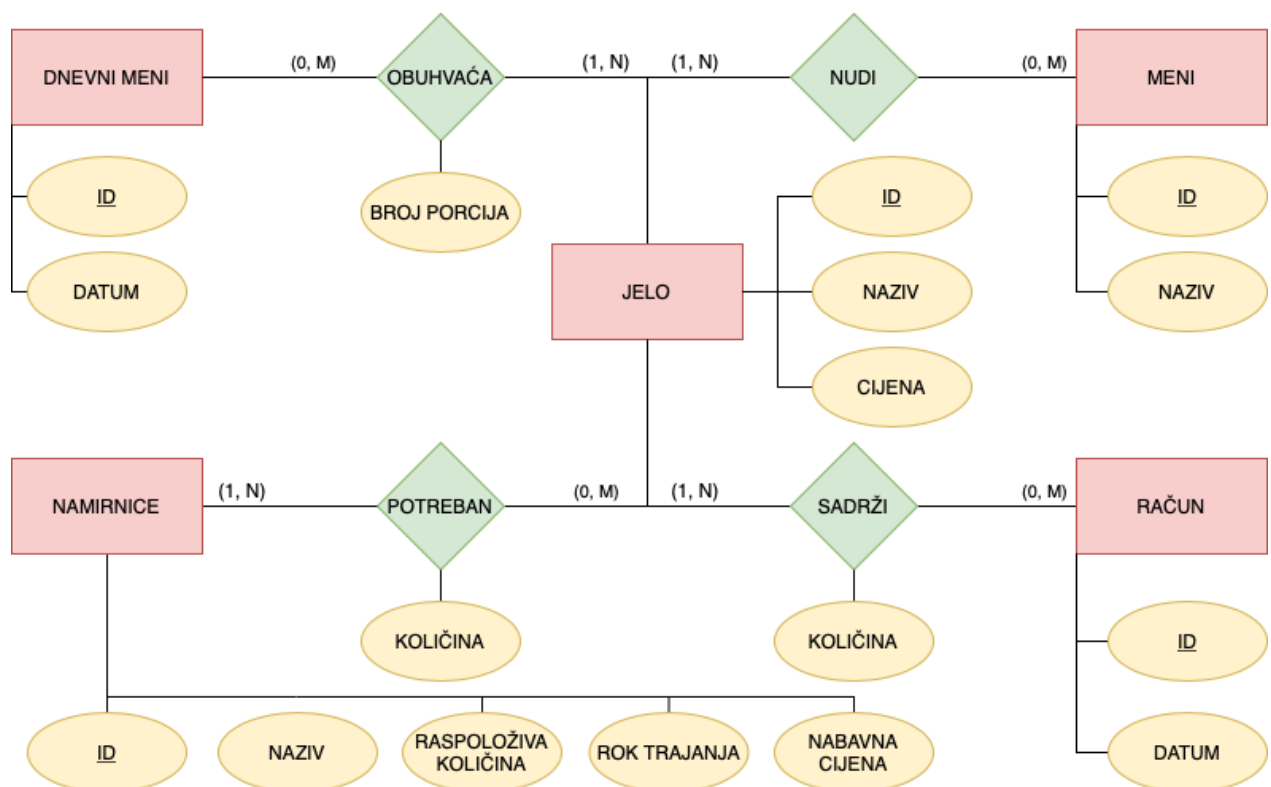
DNEVNI MENI (ID, datum)

MENI (ID, naziv)

JELO (ID, naziv, cijena)

NAMIRNICE (ID, naziv, raspoloživa_količina, datum_isteka_roka_trajanja, nabavna_cijena)

RAČUN (ID, datum)



Slika 2.2.1 . Završni ER dijagram

2.3 Relacijska shema

Dobiveni ER dijagram se vrlo lako transformira u relacijsku logičku shemu baze podataka. Veze među podacima u relacijskoj shemi su potpuno logičke. Veza između relacija uspostavlja se primjenom stranih ključeva. Ključ predstavlja atribut ili skup atributa koji nedvosmisleno određuje sve članove zapisa. U procesu modeliranja podataka svakom tipu entiteta pridijeljen je ID – brojčana vrijednost koja predstavlja primarni ključ. Razlog tomu jest što za pohranu IDa za ključeve i generirane strane ključeve treba manje prostora, što omogućava efikasniji rad baze podataka. Povezivanja će obavljati sustav za upravljanje bazama podataka, a njemu je svejedno barata li s ID ili nazivom. ID se također sam generira i tako se čuva integritet primarnog ključa.

RELACIJSKA SHEMA:

DnevniMeni (id, datum)

Meni (id, naziv)

Jelo (id, naziv, cijena)

DnevniMeniJelo (id_dnevni_meni, id_jelo, broj_porcija)

MeniJelo (id_meni, id_jelo)

Namirnice (id, naziv, raspoloziva_kolicina, datum_isteka_roka_trajanja, nabavna_cijena)

Racun (id, datum)

NamirniceJelo (id_namirnice, id_jelo, kolicina)

RacunJelo (id_racun, id_jelo, kolicina)

2.4 Tipovi entiteta

TIP ETITETA: JELO

ATRIBUT	KLJUČ	TIP	PRIMJER UPISA	OGRANIČENJA	PRIMJEDBA
ID	PK	serial	001		
Naziv		char	Pizza	not null, unique	
Cijena		real	50	not null, >0	

TIP ETITETA: MENI

ATRIBUT	KLJUČ	TIP	PRIMJER UPISA	OGRANIČENJA	PRIMJEDBA
ID	PK	serial	001		
Naziv		char	Juhe	not null, unique	

TIP ETITETA: DNEVNI MENI

ATRIBUT	KLJUČ	TIP	PRIMJER UPISA	OGRANIČENJA	PRIMJEDBA
ID	PK	serial	001		
Datum		date	2021-08-15	not null, unique, >=now()	

TIP ETITETA: NAMIRNICE

ATRIBUT	KLJUČ	TIP	PRIMJER UPISA	OGRANIČENJA	PRIMJEDBA
ID	PK	serial	001		
Naziv		char	Brašno	not null, unique	
Raspoloživa količina		real	20	>=0	
Rok trajanja		date	2021-08-15	>now()	
Nabavna cijena		real	5.5	>0	

TIP ENTITETA: **RAČUN**

ATRIBUT	KLJUČ	TIP	PRIMJER UPISA	OGRANIČENJA	PRIMJEDBA
ID	PK	serial	001		
Datum		timestamp	2021-08-15	not null	default now()

2.5 Tipovi veza

TIP VEZA: **nudi**

POVEZUJE TIPOVE ENTITETA: **MENI** i **JELO**

Nema atributa

TIP VEZA: **obuhvaća**

POVEZUJE TIPOVE ENTITETA: **DNEVNI MENI** i **JELO**

ATRIBUT	KLJUČ	TIP	PRIMJER UPISA	OGRANIČENJA	PRIMJEDBA
Broj porcija		integer	40	>0	

TIP VEZA: **potreban**

POVEZUJE TIPOVE ENTITETA: **JELO** i **NAMIRNICE**

ATRIBUT	KLJUČ	TIP	PRIMJER UPISA	OGRANIČENJA	PRIMJEDBA
Količina		integer	5	>0	

TIP VEZA: **sadrži**

POVEZUJE TIPOVE ENTITETA: **RAČUN** i **JELO**

ATRIBUT	KLJUČ	TIP	PRIMJER UPISA	OGRANIČENJA	PRIMJEDBA
Količina		integer	5	>0	

2.6 Normalizacija

Postupak normalizacije relacijskog modela ili prevođenje relacijskog modela u normalne forme je postupak dorađivanja logičke sheme baze podataka. Cilj je uočiti i otkloniti potencijalne probleme te svesti redundanciju podataka na minimum. Ovaj postupak proveden je na svakoj relaciji baze s istim ishodom, prikazanim na primjeru jedne relacije:

JELO

ID	Naziv	Cijena
1	Goveđa juha	35
2	Pizza miješana	50
3	Palačinke	20
4	Čokoladna torta	25

- Relacija Jela je u Prvoj normalnoj formi (1NF) jer su sve vrijednosti njenih atributa atomske.
- Relacija Jela je u Drugoj normalnoj formi (2NF) jer je u 1NF i svi njeni ne ključni atributi potpuno i funkcionalno zavise od primarnog ključa.
- Relacija Jela je u Trećoj normalnoj formi (3NF) jer je u 2NF i u njoj ne postoji ni jedan atribut koji je tranzitivno funkcionalno zavisao o primarnom ključu i svi njeni atributi daju jednoznačne činjenice o cijelom ključu i samo o cijelom ključu.
- Relacija Jela je u Boyce-Codd-ovoj normalnoj formi (BCNF) jer su sve determinante u relaciji i kandidati za ključ.

3. STVARANJE BAZE PODATAKA

Realizacija baze podataka na računalu obavlja se kroz pgAdmin – upravljački alat za PostgreSQL. pgAdmin pokreće se kao web ili desktop aplikacija koja kroz grafičko korisničko sučelje omogućava izradu cjelokupne baze podataka. Osim grafičkog sučelja, za izradu baze podataka, mogu se koristiti i SQL naredbe kroz konzolu. Za svaku relaciju priložen je isječak koda iz konzole kojim je ona stvorena.

Stvaranje nove baze:

```
CREATE DATABASE Restoran;
```

Stvaranje relacije Jelo:

```
CREATE TABLE Jelo (  
  id          SERIAL PRIMARY KEY,  
  naziv       CHARACTER VARYING(30) NOT NULL UNIQUE,  
  cijena      REAL NOT NULL,  
  CONSTRAINT POZITIVNA_VRIJEDNOST CHECK (cijena>0)  
);
```

Stvaranje relacije Račun:

```
CREATE TABLE Racun (  
  id          SERIAL PRIMARY KEY,  
  datum       TIMESTAMP NOT NULL DEFAULT now()  
);
```

Stvaranje relacije Namirnice:

```
CREATE TABLE Namirnice (  
  id                SERIAL PRIMARY KEY,  
  naziv             CHARACTER VARYING(30) NOT NULL UNIQUE,  
  raspoloziva_kol   REAL,  
  rok_trajanja      DATE,  
  nabavna_cijena    REAL,  
  CONSTRAINT POZITIVNA_VRIJEDNOST CHECK (raspoloziva_kol>=0 AND  
  nabavna_cijena>0),  
  CONSTRAINT DATUM_NAKON CHECK (rok_trajanja>now())  
);
```

Stvaranje relacije DnevniMeni:

```
CREATE TABLE DnevniMeni (  
  id                SERIAL PRIMARY KEY,  
  datum            DATE NOT NULL UNIQUE,  
  CONSTRAINT DATUM_NAKON CHECK (datum>now())  
);
```

Stvaranje relacije Meni:

```
CREATE TABLE Meni (  
  id                SERIAL PRIMARY KEY,  
  naziv            CHARACTER VARYING(30) NOT NULL UNIQUE  
);
```

Stvaranje relacije JeloRacun:

```
CREATE TABLE JeloRacun (  
  id_jelo          BIGINT references Jelo(id),  
  id_racun         BIGINT references Racun(id),  
  kolicina         REAL NOT NULL DEFAULT 1,  
  PRIMARY KEY      (id_jelo, id_racun),  
  CONSTRAINT POZITIVNA_VRIJEDNOST CHECK (kolicina>0)  
);
```

Stvaranje relacije JeloNamirnice:

```
CREATE TABLE JeloNamirnice (  
  id_jelo          BIGINT references Jelo(id),  
  id_namirnice     BIGINT references Namirnice(id),  
  kolicina         REAL NOT NULL DEFAULT 1,  
  PRIMARY KEY      (id_jelo, id_namirnice),  
  CONSTRAINT POZITIVNA_VRIJEDNOST CHECK (kolicina>0)  
);
```

Stvaranje relacije JeloDnevniMeni:

```
CREATE TABLE JeloDnevniMeni (  
  id_jelo          BIGINT references Jelo(id),  
  id_dnevnimeni    BIGINT references DnevniMeni(id),  
  broj_porcija     INTEGER NOT NULL DEFAULT 1,  
  PRIMARY KEY      (id_jelo, id_dnevnimeni),  
  CONSTRAINT POZITIVNA_VRIJEDNOST CHECK (broj_porcija>0)  
);
```

Stvaranje relacije JeloMeni:

```
CREATE TABLE JeloMeni (  
  id_jelo          BIGINT references Jelo(id),  
  id_meni          BIGINT references Meni(id),  
  PRIMARY KEY      (id_jelo, id_meni)  
);
```

4. RAD S BAZOM PODATAKA

Upitima se izvršavaju različite funkcije u bazi podataka. Njihova najčešća funkcija je dohvaćanje određenih podataka iz tablica. Neke od drugih mogućnosti upita su: kombiniranje podataka iz povezanih relacija u jednu podatkovnu tablicu, provođenje izračuna i sažetaka te automatizacija provođenja izmjena.

4.1 Upiti

Izvještaj o sastojcima koji odlaze u pripravu pojedinog jela i njihovoj trenutnoj raspoloživosti:

```
SELECT namirnice.naziv, namirnice.raspoloziva_kol FROM jelo
JOIN jelonamirnice ON jelo.id = jelonamirnice.id_jelo
JOIN namirnice ON namirnice.id = jelonamirnice.id_namirnice
WHERE jelo.naziv = 'Mijesana pizza';
```

Izvještaj o sastojcima koje je bilo/će biti potrebno nabaviti unutar zadanog vremenskog perioda:

```
SELECT namirnice.naziv, SUM(jelonamirnice.kolicina *
jelodnevnimeni.broj_porcija) AS potrebna_kolicina FROM jelo
JOIN jelonamirnice ON jelo.id = jelonamirnice.id_jelo
JOIN namirnice ON jelonamirnice.id_namirnice = namirnice.id
JOIN jelodnevnimeni ON jelo.id = jelodnevnimeni.id_jelo
JOIN dnevnimeni ON jelodnevnimeni.id_dnevnimeni = dnevnimeni.id
WHERE dnevnimeni.datum > '2021-10-01'::date AND dnevnimeni.datum <
'2021-10-05'::date

GROUP BY namirnice.naziv;
```

Izvještaj o svim jelima u kojima se koristi određena namirnica:

```
SELECT jelo.naziv FROM jelo
JOIN jelonamirnice ON jelo.id = jelonamirnice.id_jelo
JOIN namirnice ON namirnice.id = jelonamirnice.id_namirnice
WHERE namirnice.naziv = 'ulje';
```

Popis svih namirnica kojih nema u dovoljnim količinama za pripravu određenog jela:

```
SELECT namirnice.naziv, namirnice.raspoloziva_kol AS dostupno,
jelonamirnice.kolicina AS potrebno, (jelonamirnice.kolicina -
namirnice.raspoloziva_kol) AS nedostaje FROM jelo
JOIN jelonamirnice ON jelo.id = jelonamirnice.id_jelo
JOIN namirnice ON jelonamirnice.id_namirnice = namirnice.id
WHERE namirnice.raspoloziva_kol < jelonamirnice.kolicina AND
jelo.naziv = 'Mijesana pizza';
```

Popis svih jela koja se mogu pripremiti od raspoloživih namirnica:

```
SELECT DISTINCT jelo.naziv FROM jelo
JOIN jelonamirnice ON jelo.id = jelonamirnice.id_jelo
JOIN namirnice ON namirnice.id = jelonamirnice.id_namirnice
WHERE (jelo.naziv) <> ALL (
    SELECT DISTINCT jelo.naziv FROM jelo
    JOIN jelonamirnice ON jelo.id = jelonamirnice.id_jelo
    JOIN namirnice ON namirnice.id = jelonamirnice.id_namirnice
    WHERE jelonamirnice.kolicina > namirnice.raspoloziva_kol
);
```

Pregled svih jela pripremljenih u određenom vremenskom periodu:

```
SELECT jelo.naziv, SUM (jeloracun.kolicina) AS broj_pripremljenih_jela
FROM jelo
JOIN jeloracun ON jelo.id = jeloracun.id_jelo
JOIN racun ON racun.id = jeloracun.id_racun
WHERE racun.datum >= '2021-05-25'::date
GROUP BY jelo.naziv;
```

Pregled ukupne zarade u određenom periodu:

```
SELECT SUM(jelo.cijena * jeloracun.kolicina) AS ukupna_zarada FROM jelo
JOIN jeloracun ON jelo.id = jeloracun.id_jelo
JOIN racun ON racun.id = jeloracun.id_racun
WHERE racun.datum > '2021-05-28'::date;
```

Izvještaj o količini iskorištenih namirnica u određenom periodu:

```
SELECT DISTINCT namirnice.naziv, SUM(jelonamirnice.kolicina *
jeloracun.kolicina) AS iskorisena_kolicina FROM jelo
JOIN jelonamirnice ON jelo.id = jelonamirnice.id_jelo
JOIN namirnice ON jelonamirnice.id_namirnice = namirnice.id
JOIN jeloracun ON jelo.id = jeloracun.id_jelo
JOIN racun ON racun.id = jeloracun.id_racun
WHERE racun.datum > '2021-05-28'::date
GROUP BY namirnice.naziv;
```


Pregled prioriternih jela za dnevni meni s obzirom na datum isteka roka namirnica u određenom broju dana:

```
SELECT * FROM (
    SELECT DISTINCT ON (distinct_datum.prioritetna_jela) * FROM (
        SELECT jelo.naziv AS prioritetna_jela, namirnice.naziv AS
        namirnica_u_pripravi, namirnice.rok_trajanja FROM jelo
        JOIN JeloNamirnice ON jelo.id = JeloNamirnice.id_jelo
        JOIN namirnice ON namirnice.id = JeloNamirnice.id_namirnice
        WHERE namirnice.rok_trajanja BETWEEN now()::date AND
        now()::date+50
        ORDER BY (namirnice.rok_trajanja)
    )distinct_datum
    )distinct_jela
ORDER BY (distinct_jela.rok_trajanja);
```

Uvid u količinu namirnica koja će ostati raspoloživa nakon pripravljanja određenog jela s dnevnog menija u navedenom broju porcija:

```
SELECT namirnice.naziv, SUM (namirnice.raspoloziva_kol -
jelonamirnice.kolicina * jelodnevnimeni.broj_porcija) AS ostaje
FROM jelo
JOIN jelonamirnice ON jelo.id = jelonamirnice.id_jelo
JOIN namirnice ON jelonamirnice.id_namirnice = namirnice.id
JOIN jelodnevnimeni ON jelo.id = jelodnevnimeni.id_jelo
JOIN dnevnimeni ON dnevnimeni.id = jelodnevnimeni.id_dnevnimeni
WHERE dnevnimeni.datum = '2021-10-05'::date AND jelo.naziv = 'Mijesana
pizza'
GROUP BY namirnice.naziv;
```

4.2 Pogledi

Vlasnica/vlasnik restorana želi imati pregled prometa po danima:

```
CREATE VIEW DnevniPromet AS

SELECT SUM(jelo.cijena * jeloracun.kolicina) AS promet,
make_date(date_part('year', racun.datum)::int, date_part('month',
racun.datum)::int, date_part('day', racun.datum)::int) AS datum
FROM racun

JOIN jeloracun ON racun.id = jeloracun.id_racun
JOIN jelo ON jeloracun.id_jelo = jelo.id
GROUP BY date_part('day', racun.datum), date_part('month',
racun.datum), date_part('year', racun.datum);
```

Gost želi imati uvid u današnji dnevni meni:

```
CREATE VIEW NaDnevnomMeniju AS

SELECT jelo.naziv, jelo.cijena, jelodnevnimeni.broj_porcija
FROM dnevnimeni

JOIN jelodnevnimeni ON dnevnimeni.id = jelodnevnimeni.id_dnevnimeni
JOIN jelo ON jelodnevnimeni.id_jelo = jelo.id
WHERE dnevnimeni.datum = now()::date;
```

Gost želi uvid u sva jela na meniju Juhe:

```
CREATE VIEW MeniJuhe AS

SELECT jelo.naziv, jelo.cijena FROM meni
JOIN jelomeni ON meni.id = jelomeni.id_meni
JOIN jelo ON jelomeni.id_jelo = jelo.id
WHERE meni.naziv = ('Juhe');
```

Vlasnik restorana želi imati uvid u ukupnu dnevnu zaradu za pojedini mjesec u godini:

```
CREATE VIEW dnevna_zarada AS
SELECT SUM (jelo.cijena * jeloracun.kolicina) AS zarada,
date_part('day', racun.datum) AS dan FROM racun
JOIN  jeloracun ON jeloracun.id_racun = racun.id
JOIN jelo ON jeloracun.id_racun = jelo.id
WHERE date_part('month', racun.datum) = 05
GROUP BY date_part('day', racun.datum);
```

4.3 Transakcije

Konobar želi kreirati novi račun navodeći samo jela i njihove količine:

```
BEGIN;

with rows as (INSERT INTO racun DEFAULT VALUES RETURNING id)

INSERT INTO jeloracun(id_jelo,id_racun,kolicina)
VALUES

((SELECT jelo.id FROM jelo WHERE jelo.naziv='Goveda juha'),
(SELECT id FROM rows),1),

((SELECT jelo.id FROM jelo WHERE jelo.naziv='Juha od rajcice'),
(SELECT id FROM rows),3),

((SELECT jelo.id FROM jelo WHERE jelo.naziv='Mijesana pizza'),
(SELECT id FROM rows),4),

((SELECT jelo.id FROM jelo WHERE jelo.naziv='Pizza margarita'),
(SELECT id FROM rows),2);

COMMIT;
```

Kuhar želi kreirati novi meni navodeći samo naziv menija i jela:

```
BEGIN;

with rows as (INSERT INTO meni(naziv) VALUES ('Juhe2') RETURNING id)

INSERT INTO jelomeni(id_jelo,id_meni)
VALUES

((SELECT jelo.id FROM jelo WHERE jelo.naziv='Juha od rajcice'),
(SELECT id FROM rows)),

((SELECT jelo.id FROM jelo WHERE jelo.naziv='Goveda juha'),
(SELECT id FROM rows));

COMMIT;
```

Kuhar želi kreirati novi dnevni meni navodeći samo nazive jela i broj porcija:

```
BEGIN;

with rows as (INSERT INTO dnevnimeni(datum) VALUES (now()::date)
RETURNING id)

INSERT INTO jelodnevnimeni(id_jelo,id_dnevnimeni,broj_porcija)
VALUES

((SELECT jelo.id FROM jelo WHERE jelo.naziv='Goveda juha'),
(SELECT id FROM rows),10),

((SELECT jelo.id FROM jelo WHERE jelo.naziv='Mijesana pizza'),
(SELECT id FROM rows),10),

((SELECT jelo.id FROM jelo WHERE jelo.naziv='Cokoladna torta'),
(SELECT id FROM rows),10);

COMMIT;
```