

Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli



Fakultet informatike u Puli

**DOKUMENTACIJA UZ PROJEKTNII ZADATAK
„ORGANIZACIJA U PROIZVODNJI METALNIH
KONSTRUKCIJA“**

TIM 4

Loren Bažon

Morena Martan

Bruno Rebić

Fran Barba

Josip Milković

Smjer : Informatika

Kolegij : Baze podataka I

Mentor : doc. Dr. sc. Goran Oreški

Pula, svibanj, 2024. godina

1. UVOD.....	3
2. OPIS POSLOVNOG PROCESA	3
3. ENTITY RELATIONSHIP (ER) DIJAGRAM	3
3.1. OPIS ER DIJAGRAMA	4
4. RELACIJSKI MODEL (SCHEMA)	5
5. EER DIJAGRAM (MYSQL)	6
6. TABLICE.....	6
6.1. TABLICA klijent	6
6.2. TABLICA zaposlenik	7
6.3. TABLICA projekt.....	7
6.4. TABLICA segment	8
6.5. TABLICA faze_izrade	8
6.6. TABLICA tip_materijala.....	9
6.7. TABLICA materijal	9
6.8. TABLICA proizvodnja	9
6.9. TABLICA utrosak_sati.....	11
6.10. TABLICA utrosak_materijala	11
6.11. TABLICA primka_materijala	12
6.12. TABLICA alat	12
6.13. TABLICA lager_materijala	13
6.14. TABLICA isporuka.....	13
6.15. TABLICA zaduženje alata.....	15
7. UPITI.....	16
7.1. UPIT 1.....	16
7.2. UPIT 2.....	17
7.3. UPIT 3.....	18
7.4. UPIT 4.....	20
7.5. UPIT 5.....	21
7.6. UPIT 6.....	22
7.7. UPIT 7.....	23
7.8. UPIT 8.....	25
7.9. UPIT 9.....	26
7.10. UPIT 10	27
7.11. UPIT 11	29
7.12. UPIT 12	31

7.13.	UPIT 13	33
7.14.	UPIT 14	35
7.15.	UPIT 15	36
7.16.	UPIT 16	38
8.	ZAKLJUČAK	40

1. UVOD

Ovaj projekt je od svoje početne verzije prošao kroz značajne promjene i unapređenja. Kako smo napredovali sa stjecanjem novih znanja i vještina tokom semestra, stalno smo uvodili poboljšanja koja su optimizirala strukturu i funkcionalnost elemenata našeg projekta. Zbog nemogućnosti okupljanja svih članova tima uslijed poslovnih i drugih obaveza, proces izrade projekta je koordiniran putem GitHuba, Whatsappa, Zooma i ostalih komunikacijskih alata.

Izradili smo plan i model organizacije proizvodnje metalnih konstrukcija, s naglaskom na precizno opisivanje svih proizvodnih procesa i konceptualnog modela, prikazanog putem ER dijagrama. Naš projekt obuhvaća samo dio stvarnog poslovanja, pažljivo reducirajući i specificirajući ključne aspekte za potrebe ovog zadatka. Krajnji cilj je prikazati kompletan proces od nabave sirovina, preko proizvodnje, do isporuke gotovih metalnih proizvoda.

Svaki korak proizvodnje, uključujući tablice, unose podataka, upite i ograničenja, temeljito je dokumentiran u narednim stranicama. EER dijagram, odnosno logička shema organizacije proizvodnje, generiran je u MySQL Workbench-u pomoću Reverse Engineering opcije. Svi nazivi zaposlenika, adrese, OIB-ovi i brojevi mobitela su generirani nasumično pomoću online alata, dok su nazivi alata i klijenata preuzeti iz stvarne industrijske prakse.

2. OPIS POSLOVNOG PROCESA

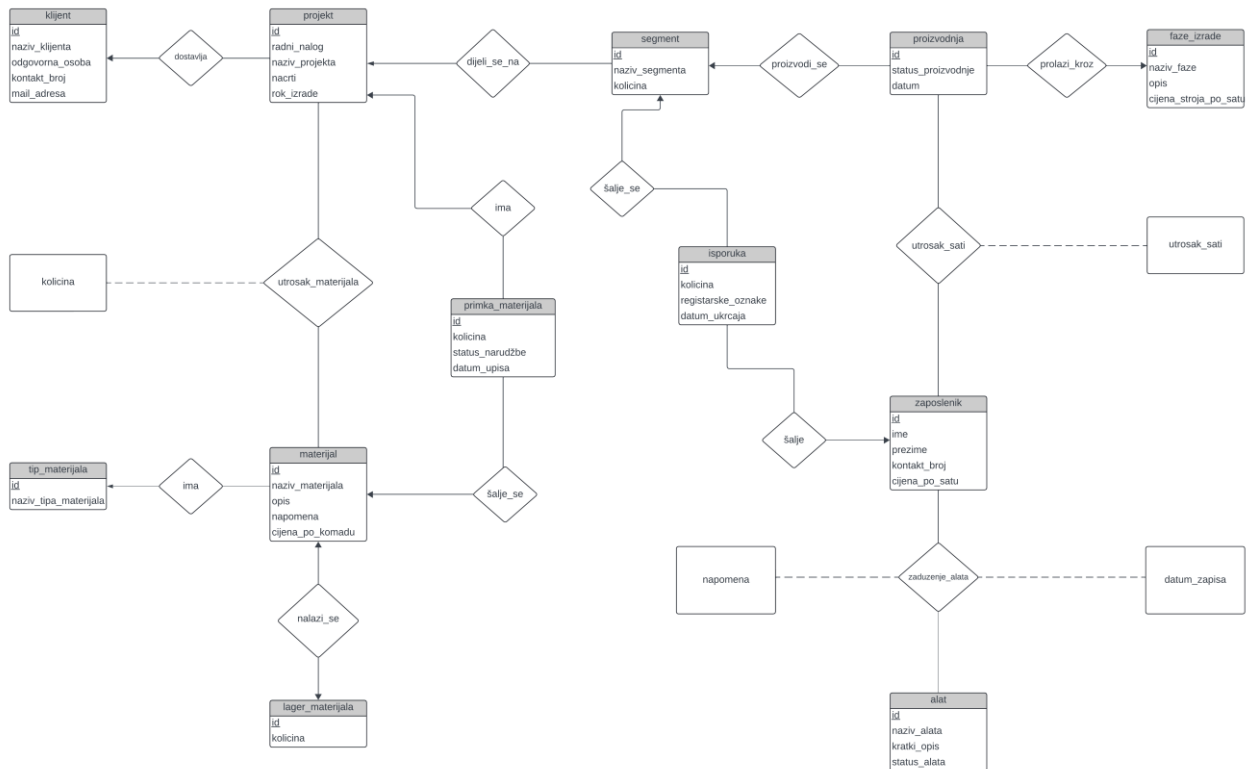
U proizvodnji sve kreće od klijenta koji daje neki projekt koji je potrebno izraditi s potrebnom dokumentacijom (nacrtima). Za potrebe ovog projekta smatra se da klijent ima svu potrebnu dokumentaciju, iako u praksi ne mora biti tako. Za klijenta pratimo naziv, odgovornu osobu, kontakt broj i mail adresu. Za projekt se prati radni nalog, naziv, nacrt i rok izrade.

Nakon dobivanja projekta, inženjeri proučavaju projekt te daju informacije o potrebnim materijalima za proizvodnju. Za materijale se prati naziv, opis, određena napomena i cijena po komadu. Računa se utrošak materijala za projekt, te se prati tip materijala, kao i njegovo količinsko stanje na lageru. Tada se radi dokument koji se zove primka, te se materijali naručuju i skladište do početka proizvodnje projekta.

Projekt se tada dijeli na segmente. Svakom segmentu se prati naziv i količina. Da bi određeni zaposlenik, kojem pratimo ime, prezime, kontakt broj i cijenu po satu, mogao započeti proizvodnju određenog segmenta on mora zadužiti određene alate. Tu pratimo sve alate koje imamo, te se za svaki alat prati naziv, kratki opis i status. Pri zaduženju svakog alata pratimo datum te kratku napomenu zaposlenika. U proizvodnji se prati status proizvodnje te datum izrade. Također u proizvodnji imamo različite faze izrade kojima se prati naziv, kratki opis te cijena stroja po satu. Nakon završetka proizvodnje, moramo izračunati cijenu rada zaposlenika. To radimo na način da pratimo utrošak sati rada određenog zaposlenika. Na kraju zaposlenik šalje segment koji je proizveo na isporuku u kojoj se prati količina, registarske oznake vozila kojim se šalje te datum ukrcaja.

3. ENTITY RELATIONSHIP (ER) DIJAGRAM

Slijedeći ER dijagram detaljno i pregledno opisuje sve skupove - entiteta (njihove atribute) kao i skupove - veza između njih. Kardinalnost mapiranja (strelice) predstavljaju koliko drugih entiteta može biti povezano s entitetom preko određenog skupa – veza.



Slika 1/ ER dijagram za našu bazu

3.1. OPIS ER DIJAGRAMA

- **KLIJENT** dostavlja **PROJEKT** (klijent može dostaviti više projekata, dok je jedan projekt specifično dostavljen od strane jednog klijenta) **jedan na više**
- **PROJEKTU** trebaju određeni **MATERIJALI** (jedan projekt treba više materijala za korištenje, a jedan materijal može biti korišten kod više projekata) **više na više**
- Stvara se dodatni entitet **UTROSAK_MATERIJALA** gdje se prati količina utrošenog materijala, a primarni ključevi entiteta **PROJEKT** i **MATERIJAL** se referenciraju kao strani ključ za **UTROSAK_MATERIJALA**
- **TIP MATERIJALA** opisuje **MATERIJAL** (jedan materijal se sastoji od jednog tipa materijala, dok isti taj tip materijala može biti kod više različitih materijala) **jedan na više**
- **MATERIJAL** se nalazi na **LAGERU** (jedan materijal se nalazi na jednom lageru, taj lager prati količinu i vezan je za taj jedan materijal) **jedan na jedan**
- **MATERIJAL** se šalje na **PRIMKU** (jedan materijal može biti na više različitih primki, dok je jedna primka vezana za jedan materijal) **jedan na više**
- **PROJEKT** ima **PRIMKU** (jedan projekt može imati više različitih primki, dok je jedna primka vezana za jedan projekt) **jedan na više**

- **PROJEKT** se dijeli na **SEGMENTE** (jedan projekt može imati više segmenata, dok je jedan segment specifično vezan za jedan projekt) **jedan na više**
- **SEGMENTU** se prati **PROIZVODNJA** (jedan segment može proći kroz više različitih dijelova proizvodnje, dok se u toj jednoj proizvodnji proizvodi jedan segment) **jedan na više**
- **PROIZVODNJA** prolazi kroz **FAZU IZRADE** (jedna specifična faza izrade može biti kod više različitih proizvodnji, dok u jednoj proizvodnji kada se odradi određena faza, ona ne može biti ponovljena, dakle jedinstvena je) **jedan na više**
- **ZAPOSLENIK** zadužuje **ALAT** (jedan zaposlenik može zadužiti više različitih alata, dok jedan alat može biti zadužen kod više različitih zaposlenika) **više na više**
- Stvara se dodatni entitet **ZADUZENJE_ALATA** gdje se referenciraju primarni ključevi od **ZAPOSLENIK** i **ALAT** kao strani ključ u **ZADUZENJE_ALATA**. Prati se još datum zapisa te određena napomena zaposlenika.
- **ZAPOSLENIK** odraduje **PROIZVODNJU** (jedan zaposlenik može odraditi više proizvodnji, dok se u jednoj proizvodnji može koristiti više zaposlenika) **više na više**
- Stvara se dodatni entitet **UTROSAK_SATI** koji prati utrošak sati zaposlenika pri proizvodnji, te referencira primarne ključeve od **ZAPOSLENIK** i **PROIZVODNJA** kao strani ključ.
- **SEGMENT** se nalazi na **ISPORUCI** (jedan segment može biti na više različitih isporuka, dok u jednoj isporuci se nalazi samo jedan segment) **jedan na više**
- **ZAPOSLENIK** šalje **ISPORUKU** (jedan zaposlenik može slati više isporuka, dok je jedna isporuka poslana od strane samo jednog zaposlenika) **jedan na više**

4. RELACIJSKI MODEL (SCHEMA)

klijent (id, naziv_klijenta, odgovorna_osoba, kontakt_broj, mail_adresa)

zaposlenik (id, ime, prezime, kontakt_broj, cijena_po_satu)

projekt (id, id_klijent, radni_nalog, naziv_projekta, nacrti, rok_izrade)

segment (id, id_projekt, naziv_segmenta, kolicina)

faze_izrade (id, naziv_faze, opis, cijena_stroja_po_satu)

tip_materijala (id, naziv_tipa_materijala)

materijal (id, id_tip_materijala, naziv_materijala, opis, napomena, cijena_po_komadu)

proizvodnja (id, id_segment, id_faza, status_proizvodnje, datum)

utrosak_sati (id, id_proizvodnja, id_zaposlenik, utrosak_sati)

utrosak_materijala (id, id_projekt, id_materijal, kolicina)

primka_materijala (id, id_projekt, id_materijal, kolicina, status_narudzbe, datum_upisa)

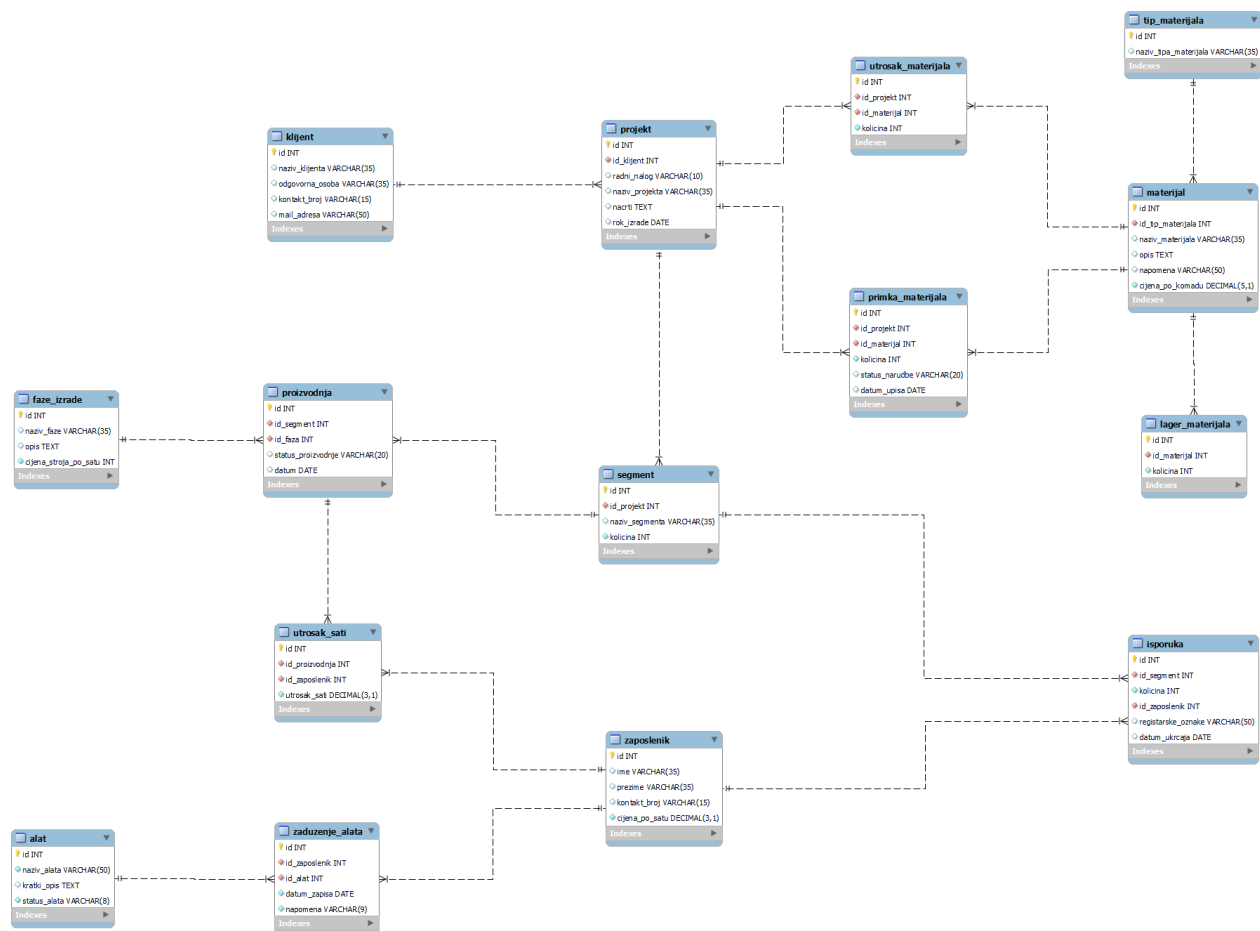
alat (id, naziv_alata, kratki_opis, status_alata)

lager_materijala (id, id_materijal, kolicina)

isporuka (id, id_segment, kolicina, id_zaposlenik, registarske_oznake, datum_ukrcaja)

zaduzenje_alata (id, id_zaposlenik, id_alat, datum_zapisa, napomena)

5. EER DIJAGRAM (MYSQL)



Slika 2/EER dijagram, napravljen u MySQL Workbenchu

6. TABLICE

6.1. TABLICA klijent

Pomoću tablice **klijent** stvaramo bazu podataka o klijentima te informacijama vezanih uz njih: **id**, **naziv_klijenta**, **odgovorna_osoba**, **kontakt_broj**, **mail_adresa**. Atribut **id** nam predstavlja interni **PRIMARY KEY** i njega smo stavili tipa **INTEGER**. **Naziv_klijenta** odnosno tvrtke koja naručuje izradu određenog projekta zapisat ćemo pomoću **VARCHAR(35)**. U našu bazu podataka ulazi i **odgovorna_osoba** koju ćemo upisivati pomoću **VARCHAR(35)**, a u slučaju nedoumice ili promjene u projektu dužni smo informirati odgovornu osobu pa ćemo **kontakt_broj** te osobe također upisati u tablicu i bit će nam tipa **VARCHAR(15)**. Broj telefona nismo stavili tipa integer zbog stranih klijenata čije brojeve moramo upisivati pomoću pozivnog broja zemlje iz koje dolazi a za to ponekad koristimo `+`. Radi lakše komunikacije i evidentiranih dogovora ili eventualnih izmjena, komunicirat ćemo sa odgovornom osobom i putem maila pa nam je potrebna njegova **mail_adresa** koju ćemo upisivati pomoću **VARCHAR(50)**.

```
CREATE TABLE klijent (
```

```

id INTEGER PRIMARY KEY,
naziv_klijenta VARCHAR(35),
odgovorna_osoba VARCHAR(35),
kontakt_broj VARCHAR(15),
mail_adresa VARCHAR(50)
);

```

6.2. TABLICA zaposlenik

Tablica **zaposlenik** nam sadrži attribute: **id**, **ime**, **prezime**, **kontakt_broj**, **cijena_po_satu**. Pomoću **id**-a koji je **PRIMARY KEY** tipa **INTEGER**, povezivat ćemo ovu relaciju sa drugim relacijama s kojima je u međuovisnosti. **Ime** svakog zaposlenika upisivat ćemo pomoću **VARCHAR** tipa koji smo ograničili na 35 znakova, za **prezime** ćemo također koristiti tip **VARCHAR(35)** sa istim ograničenjem. Za **kontakt_broj** koristimo **VARCHAR(15)** također da bismo omogućili korištenje znaka '+' kao dio pozivnog broja u slučaju da neki naš zaposlenik dolazi iz strane zemlje i ne koristi hrvatski broj telefona. Ovisno o stažu, znanju, iskustvu i zalaganju, svakom zaposleniku je uprava dodijelila i posebnu satnicu pomoću koje će kasnije biti kreirana završna cijena gotovog proizvoda pa nam je za ovu relaciju bitan i atribut **cijena_po_satu** koju ćemo unijeti tipa **NUMERIC(3,1) NOT NULL** zbog mogućeg decimalnog broja.

```

CREATE TABLE zaposlenik (
id INTEGER PRIMARY KEY,
ime VARCHAR(35),
prezime VARCHAR(35),
kontakt_broj VARCHAR(15),
cijena_po_satu NUMERIC(3,1) NOT NULL,
);

```

6.3. TABLICA projekt

Tablica **projekt** nam služi kako bismo lakše pratili cijeli proces proizvodnje od klijentovog zahtjeva, podjelu na segmente, utrošak materijala do gotovog krajnjeg proizvoda. Sastavljena je od atributa: **id** koji nam je i ovog puta **PRIMARY KEY** odnosno jedinstveni broj zapisan u **INTEGER** tip podatka. Svaki projekt povezujemo sa **FOREIGN KEY** relacije klijent odnosno, **id_klijent** tipa **INTEGER** i stavili smo ograničenje **NOT NULL** kako bismo spriječili eventualnu pogrešku prilikom unosa podataka. Nadalje, atribut **radni_nalog** tipa **VARCHAR(10)** smo ograničili na 10 simbola radi lakše interne kontrole. Svakom projektu smo dodijelili i ime atributom **naziv_projekta** koji je tipa **VARCHAR(35)**. Atribut **nacrti** je tipa **TEXT** jer će tamo biti pohranjena adresa nacрта. Za svaki projekt imamo dogovoreno i vrijeme isporuke pa taj podatak pratimo atributom **rok_izrade** koji ćemo zapisivati **DATE** tipom.


```
CREATE TABLE projekt (
id INTEGER PRIMARY KEY,
id_klijent INTEGER NOT NULL,
FOREIGN KEY (id_klijent) REFERENCES klijent (id),
radni_nalog VARCHAR(10),
naziv_projekta VARCHAR(35),
nacrti TEXT,
rok_izrade DATE
);
```

6.4. TABLICA segment

U ovoj tablici vodimo evidenciju i stanje izrade pojedinog **segmenta**. Atributi potrebni za ažurno praćenje ovih podataka su: **id**, **id_projekt**, **naziv_segmenta**, **kolicina**. **Id** nam je **PRIMARY KEY** tipa **INTEGER**, **id_projekt** je **FOREIGN KEY** u ovoj relaciji (odnosno, **PRIMARY KEY** relacije projekt) tipa **INTEGER NOT NULL**. Ovaj podatak nam ne smije izostati, pa smo stavili ograničenje **NOT NULL** iz tog razloga. Svaki segment imenujemo atributom **naziv_segmenta** tipom **VARCHAR(35)** i unosimo **kolicina** **INTEGER** također **NOT NULL**.

```
CREATE TABLE segment (
id INTEGER PRIMARY KEY,
id_projekt INTEGER NOT NULL,
FOREIGN KEY (id_projekt) REFERENCES projekt (id),
naziv_segmenta VARCHAR(35),
kolicina INTEGER NOT NULL
);
```

6.5. TABLICA faze_izrade

Ova tablica služi za evidentiranje različitih **faza izrade** koje se koriste u proizvodnji metalnih konstrukcija. Svaka faza ima svoj **naziv**, **opis** i **cijenu stroja po satu**, što je ključno za planiranje i praćenje troškova proizvodnje. Tablica sadrži attribute : „**id**“ – **INTEGER PRIMARY KEY** – što znači da je atribut primarni ključ te da je podatak broj. On nam služi kao jedinstveni identifikator za svaku fazu izrade; „**naziv_faze**“ - **VARCHAR(35)** – fazu izrade imenujemo znakovnim nizom ne dužim od 35 znakova; „**opis**“ – **TEXT** - detaljni opis faze izrade tipa podatka tekst; „**cijena_stroja_po_satu**“ – **INTEGER NOT NULL** – cijena stroja po satu tijekom faze izrade koja ne smijem biti 0. Tablica omogućava praćenje troškova proizvodnje po fazama, identifikaciju različitih faza izrade i njihovu uporabu u projektima.

```
CREATE TABLE faze_izrade (
id INTEGER PRIMARY KEY,
naziv_faze VARCHAR(35),
opis TEXT,
cijena_stroja_po_satu INTEGER NOT NULL
);
```

6.6. TABLICA tip_materijala

Ova tablica služi za kategorizaciju materijala koji se koriste u proizvodnji. Omogućuje identifikaciju tipova materijala, što pomaže u organizaciji i upravljanju skladištem. Njezini atributi su: „**id**“ koji nam služi kao primarni ključ za svaki tip materijala i tip podatka mu je INTEGER te „**naziv_tipa_materijala**“ koji sadrži naziv tipa materijala i tip podatka mu je VARCHAR(35). Omogućava praćenje i upravljanje različitim tipovima materijala te njihovo povezivanje s konkretnim materijalima u skladištu.

```
CREATE TABLE tip_materijala (  
id INTEGER PRIMARY KEY,  
naziv_tipa_materijala VARCHAR(35)  
);
```

6.7. TABLICA materijal

Ova tablica evidentira pojedinačne materijale koji se koriste u proizvodnji. Sadrži informacije o vrsti materijala, količini, opisu i cijeni po komadu, što je ključno za upravljanje troškovima i zalihe materijala. Tablica ima attribute: „**id**“ koji je njezin primarni ključ tipa INTEGER; „**id_tip_materijala**“ koji je strani ključ koji povezuje materijal s njegovim tipom te je on INTEGER NOT NULL; „**naziv_materijala**“ sadrži naziv materijala i tipa je VARCHAR(50); „**opis**“ detaljan opis materijala tip TEXT; „**napomena**“ dodatne napomene o materijalu; „**cijena_po_komadu**“ sadrži cijenu materijala po komadu i tipa je NUMERIC(5,1) NOT NULL što znači da ne može pohraniti vrijednosti koje imaju više od ukupno 5 znamenki (npr. 123456.0) ili vipe od 1 znamenke iza decimalne točke (npr. 123.45). Omogućava praćenje zaliha, troškova materijala i specifikacija materijala koji se koriste u projektima.

```
CREATE TABLE materijal (  
id INTEGER PRIMARY KEY,  
id_tip_materijala INTEGER NOT NULL,  
FOREIGN KEY (id_tip_materijala) REFERENCES tip_materijala (id),  
naziv_materijala VARCHAR(35),  
opis TEXT,  
napomena VARCHAR(50),  
cijena_po_komadu NUMERIC(5,1) NOT NULL  
);
```

6.8. TABLICA proizvodnja

Ova tablica prati status proizvodnje svakog segmenta projekta kroz različite faze izrade. Evidentira trenutni status i datum svake proizvodne aktivnosti. Tablica ima attribute: „**id**“ koji nam je primarni ključ te je tipa INTEGER; „**id_segment**“ strani ključ koji povezuje proizvodnju s odgovarajućim segmentom projekta i tip je INTEGER NOT NULL; „**id_faza**“ strani ključ koji povezuje proizvodnju s fazom izrade i tipa je INTEGER NOT NULL; „**status_proizvodnje**“ status trenutne proizvodne aktivnosti i tipa je VARCHAR(20); „**datum**“ datum aktivnosti u proizvodnji. Omogućava praćenje statusa proizvodnje, identifikaciju zastoja u proizvodnji i planiranje daljnjih aktivnosti na temelju trenutnog statusa i datuma.

```
CREATE TABLE proizvodnja (  
id INTEGER PRIMARY KEY,  
id_segment INTEGER NOT NULL,  
FOREIGN KEY (id_segment) REFERENCES segment (id),  
id_faza INTEGER NOT NULL,  
FOREIGN KEY (id_faza) REFERENCES faze_izrade (id),  
status_proizvodnje VARCHAR(20),  
datum DATE  
);
```

6.9. TABLICA utrosak_sati

Tablica **utrosak_sati** pomaže u evidenciji potrošenih sati na izradu pojedinog segmenta projekta. Ta informacija će nam biti potrebna u kreiranju konačne cijene proizvoda krajnjem kupcu. Tablica sadrži attribute: **id** i **utrosak_sati** te primarne ključeve tablica o kojima ovisi: **id_proizvodnje** i **id_zaposlenik**. Atribut **id** je **PRIMARY KEY** tablice tipa **INTEGER** jer se u njega unosi jedinstvena brojučana vrijednost za identifikaciju utrošenih sati za pojedini segment. Nisu nam potrebna dodatna ograničenja za jedinstvenost jer upravo to i jest primarni ključ po svojoj definiciji. Razlog stvaranje ove relacije jest upravo **utrosak_sati** tipa **INTEGER** kako bismo pratili zauzetost naših radnika i njihovu angažiranost za pojedini segment. **FOREIGN KEY id_proizvodnja** spaja relacije **segment**, **utrosak_sati** i **faza_izrade**, dok **id_zaposlenik** kao strani ključ spaja broj utrošenih sati sa zaposlenikom koji radi na određenom segmentu projekta.

```
CREATE TABLE utrosak_sati (  
id INTEGER PRIMARY KEY,  
id_proizvodnja INTEGER NOT NULL,  
FOREIGN KEY (id_proizvodnja) REFERENCES proizvodnja (id),  
id_zaposlenik INTEGER NOT NULL,  
FOREIGN KEY (id_zaposlenik) REFERENCES zaposlenik (id),  
Utrosak_sati NUMERIC(3,1) NOT NULL  
);
```

6.10. TABLICA utrosak_materijala

Tablica **utrosak_materijala** služi evidenciji materijala koji je potrošen na određenom projektu kako bismo kasnije mogli kreirati cijenu. Ova relacija sadrži attribute: **id**, **id_projekt**, **id_materijal**, **kolicina**. Atribut **id** je tipa **INTEGER** jer je jedinstveni broj za identifikaciju količine materijala potrošene na određeni projekt. Zbog međuovisnosti projekta i materijala, u ovoj relaciji navodimo i njihove **PRIMARY KEY** koji nam ovdje postaju **FOREIGN KEY**. Stavili smo ograničenje **NOT NULL** jer nam materijal mora biti povezan nekim projektom i materijalom čije stanje pratimo. Atribut **kolicina** smo također ograničili na neki pozitivan broj jer u suprotnom ova relacija nema smisla.

```
CREATE TABLE utrosak_materijala (  
id INTEGER PRIMARY KEY,  
id_projekt INTEGER NOT NULL,  
FOREIGN KEY (id_projekt) REFERENCES projekt (id),
```

TABLICA

```
id_materijal INTEGER NOT NULL,  
FOREIGN KEY (id_materijal) REFERENCES materijal (id),  
kolicina INTEGER NOT NULL CHECK (kolicina > 0)  
);
```

6.11. TABLICA primka_materijala

Tablica primka_materijala sadrži attribute: **id**, **id_projekt**, **id_materijal**, **kolicina**, **status_narudzbe**, **datum_upisa**. I ovdje nam je **id** kao i kod ostalih relacija tipa **INTEGER**, **PRIMARY KEY** po svojoj definiciji, jedinstveni broj za identifikaciju primke. **FOREIGN KEY** koji povezuje ovu relaciju s relacijama koje vode evidenciju o projektu i materijalima nam je također tipa **INTEGER**. Atribut **kolicina** označuje količinu materijala kojeg smo naručili, odnosno ima li potrebe naručivati još materijala ili je onaj na lageru dovoljan te ćemo za njega koristiti **INTEGER NOT NULL**. Stavljamo ograničenje kako bismo spriječili moguće pogreške prilikom unosa podataka. **Status_narudzbe** nam označava ako je materijal dostavljen ili tek naručen. **datum_upisa** koristimo kako bismo lakše predvidjeli sam datum isporuke obzirom na potrebne sate za izradu ovog projekta i projekata koji su u procesu izrade, odnosno na pravilan poredak na „listi čekanja“. Za ovaj atribut koristit ćemo **DATETIME NOT NULL**.

```
CREATE TABLE primka_materijala(  
id INTEGER PRIMARY KEY,  
id_projekt INTEGER NOT NULL,  
FOREIGN KEY (id_projekt) REFERENCES projekt (id),  
id_materijal INTEGER NOT NULL,  
FOREIGN KEY (id_materijal) REFERENCES materijal (id),  
kolicina INTEGER NOT NULL,  
status_narudzbe VARCHAR(20),  
datum_upisa DATE  
);
```

6.12. TABLICA alat

Relacija alat sadrži attribute: **id**, **naziv_alata**, **kratki_opis**, **status**. Atribut **id** nam je **INTEGER**, **PRIMARY KEY**. **Naziv_alata** smo stavili **VARCHAR(50)**, tj. znakovni niz koji smo ograničili na 50 znakova. Atribut **kratki_opis** smo stavili tipa **TEXT**. Ovaj stupac nam služi kako bismo korisnicima naše baze podataka dali kratki opis alata i oznaku kako bi se laše prepoznalo ako se radi baš o tom alatu. Status će nam davati informaciju o eventualnoj nedostupnosti određenog alata. Bitno je znati koji su alati dostupni u kojem trenutku kako bi voditelj što kvalitetnije iskoristio i koordinirao radnike obzirom na dostupnost alata. Atribut

TABLICA

status je tipa VARCHAR (8) te se nude samo dvije mogućnosti upisa: „ispravno“ ili „servis“ i nije potrebno zauzimati više memorije.

```
CREATE TABLE alat(  
id INTEGER PRIMARY KEY,  
naziv_alata VARCHAR (50) NOT NULL,  
kratki_opis TEXT,  
status_alata VARCHAR (8) NOT NULL  
);
```

6.13. TABLICA lager_materijala

Tablica **lager materijala** je dizajnirana za praćenje lagera materijala u skladištu. Struktura tablice obuhvaća sljedeća polja:

id: Ovo je jedinstveni identifikator za svaki zapis u tablici. Polje je tipa INTEGER i postavljeno je kao PRIMARY KEY, što znači da svaka vrijednost mora biti jedinstvena i ne može biti NULL.

id_materijala: Ovo polje služi za identifikaciju materijala. Također je tipa INTEGER i ne može biti NULL. Očekuje se da će ovo polje vjerojatno biti povezano s drugim tablicama koje detaljnije opisuju materijale, iako u ovom primjeru ta relacija nije eksplicitno navedena.

količina: Ovo polje predstavlja količinu materijala koja je trenutno na lageru. Polje je tipa INTEGER i ne može biti NULL, što znači da svaki zapis mora imati definiranu količinu materijala.

```
CREATE TABLE lager_materijala (  
id INTEGER PRIMARY KEY,  
id_materijal INTEGER NOT NULL,  
FOREIGN KEY (id_materijal) REFERENCES materijal (id),  
kolicina INTEGER NOT NULL  
);
```

6.14. TABLICA isporuka

Tablica isporuka je dizajnirana za praćenje isporuka segmenta proizvoda. Struktura tablice obuhvaća sljedeća polja:

id: Ovo je jedinstveni identifikator za svaki zapis u tablici. Polje je tipa INTEGER i postavljeno je kao PRIMARY KEY, što znači da svaka vrijednost mora biti jedinstvena i ne može biti NULL.

id_segment: Ovo polje služi za identifikaciju segmenta proizvoda koji je isporučen. Također je tipa INTEGER i ne može biti NULL.

TABLICA

kolicina: Ovo polje predstavlja količinu segmenta proizvoda koja je isporučena. Polje je tipa INTEGER i ne može biti NULL.

id_zaposlenik: Ovo polje služi za identifikaciju zaposlenika koji je odgovoran za isporuku. Također je tipa INTEGER i ne može biti NULL.

registarske_oznake: Ovo polje služi za unos registarskih oznaka vozila koje je korišteno za isporuku. Polje je tipa VARCHAR s maksimalnom duljinom od 50 znakova.

datum_ukrcaja: Ovo polje predstavlja datum kada je segment proizvoda ukrcan za isporuku. Polje je tipa DATE

```
CREATE TABLE isporuka (  
  id INTEGER PRIMARY KEY,  
  id_segment INTEGER NOT NULL,  
  FOREIGN KEY (id_segment) REFERENCES segment (id),  
  kolicina INTEGER NOT NULL,  
  id_zaposlenik INTEGER NOT NULL,  
  FOREIGN KEY (id_zaposlenik) REFERENCES zaposlenik (id),  
  registarske_oznake VARCHAR(10),  
  datum_ukrcaja DATE  
);
```

TABLICA

6.15. TABLICA zaduženje alata

Tablica **zaduženje alata** je dizajnirana za praćenje zaduživanja alata od strane zaposlenika. Struktura tablice obuhvaća sljedeća polja:

id: Ovo je jedinstveni identifikator za svaki zapis u tablici. Polje je tipa INTEGER i postavljeno je kao PRIMARY KEY, što znači da svaka vrijednost mora biti jedinstvena i ne može biti NULL.

id_zaposlenik: Ovo polje služi za identifikaciju zaposlenika koji je zadužio alat. Također je tipa INTEGER i ne može biti NULL.

id_alat: Ovo polje služi za identifikaciju alata koji je zadužen. Također je tipa INTEGER i ne može biti NULL.

datum_zapisa: Ovo polje predstavlja datum kada je alat zadužen. Polje je tipa DATE i ne može biti NULL.

napomena: Ovo polje služi za dodatne napomene vezane za zaduženje alata. Polje je tipa VARCHAR s maksimalnom duljinom od 9 znakova, odnosno 'zaduzeno' ili 'razduzeno'. Ova tablica je ključna za upravljanje zaduživanjem alata, omogućujući praćenje koji je alat zadužen, od koga i kada, kao i eventualne dodatne napomene vezane za zaduženje.

```
CREATE TABLE zaduzenje_alata (  
  id INTEGER PRIMARY KEY,  
  id_zaposlenik INTEGER NOT NULL,  
  FOREIGN KEY (id_zaposlenik) REFERENCES zaposlenik (id),  
  id_alat INTEGER NOT NULL,  
  FOREIGN KEY (id_alat) REFERENCES alat (id),  
  datum_zapisa DATE NOT NULL,  
  napomena VARCHAR(9) NOT NULL
```


7. UPITI

- Morena Martan (7.1 - 7.5)
- Loren Bažon (7.6 – 7.10)
- Josip Milković (7.11 – 7.13)
- Fran Barba (7.14 – 7.16)

7.1. UPIT 1

Količina velikih kutnih brusilica i količina malih kutnih brusilica.

S obzirom na veliku količinu projekata ove godine i uzimajući u obzir da su nam velika i mala kutna brusilica najčešći alat, voditelja proizvodnje zanima količina ispravnih brusilica. Kako bi se moglo nabaviti nove velike i male kutne brusilice ukoliko je potrebno.

TRAŽENO RJEŠENJE:

tip_brusilice, broj_komada

KOD ZA UPIT:

```
SELECT
CASE
    WHEN kratki_opis LIKE '%Mala kutna brusilica%' THEN 'Mala kutna brusilica'
    WHEN kratki_opis LIKE '%Velika kutna brusilica%' THEN 'Velika kutna brusilica'
END AS tip_brusilice,
COUNT(*) AS broj_komada
FROM
    alat
WHERE
    kratki_opis LIKE '%Mala kutna brusilica%' AND status_alata = 'ispravno'
    OR kratki_opis LIKE '%Velika kutna brusilica%' AND status_alata = 'ispravno'
GROUP BY
    tip_brusilice;
```

OPIS UPITA:

Ovaj upit prebrojava koliko ima 'Malih kutnih brusilica' i 'Velikih kutnih brusilica' u relaciji **alat** na temelju sadržaja atributa **kratki_opis**. Prvo koristi **CASE** izraz za klasifikaciju brusilica kao '**Mala kutna brusilica**' ili '**Velika kutna brusilica**', a zatim koristi **COUNT(*)**

za brojanje N-torki u svakoj grupi. Konačni rezultat prikazuje broj komada za svaki tip brusilice.

- **SELECT** naredba odabire attribute koji će biti uključeni u rezultat upita.
- **CASE** je uvjetna logika koja se koristi za određivanja tipa busilice, u nastavku koda je **WHEN kratki_opis LIKE '%Mala kutna brusilica%' THEN 'Mala kutna brusilica'** koji nam govori ako neka N-torka sadrži tekst 'Mala kutna brusilica', tada će vrijednost atributa **tip_brusilice** biti 'Mala kutna brusilica'. Nakon toga ista stvar za 'Velika kutna brusilica' te na kraju **END AS tip_brusilice** koji nam rezultat **CASE** izraza daje aliasu **tip_brusilice**.
- **COUNT(*) AS broj_komada** nam broji svaku n-torku koja odgovara jednom od 2 izraza prije i te brojeve prikazuje u atributu **broj_komada**.
- **FROM** alat specificira iz koje se relacije preuzimaju podaci. **WHERE** filtrira n-torke koje će se uključiti u grupiranje, zatim **kratki_opis LIKE '%Mala kutna brusilica%' AND status_alata = 'ispravno' OR kratki_opis LIKE '%Velika kutna brusilica%' AND status_alata = 'ispravno'** uključuje samo one n-torke gdje gdje **kratki_opis** sadrži tekst 'Mala kutna brusilica' ili 'Velika kutna brusilica' i **status_alata 'ispravno'** kako bi filtrirali ispravne od onih koje su na servisu.
- **GROUP BY tip_brusilice** nam grupira rezultate prema vrijednosti koja je određena u **CASE** izrazu. Tj. Sve n-torke koje imaju isti tip brusilice bit će grupirane zajedno.

REZULTAT:

	tip_brusilice	broj_komada
►	Mala kutna brusilica	7
	Velika kutna brusilica	3

7.2. UPIT 2

Popis svih zaposlenika s trenutnom i povećanom satnicom za 0,6 €

Obzirom na porast BDP-a, rastu i plaće u državi. Računovodstvo nas je tražilo popis svih zaposlenika te ispis trenutne satnice za svakog pojedinog radnika. Obzirom na veliku dobit, planiraju povećanje satnice svakom radniku za 0.60EUR pa su tražili da dodamo i taj podatak.

TRAŽENO RJEŠENJE:

id, ime, prezime, cijena_po_satu, nova_cijena_po_satu

KOD ZA UPIT:

```
SELECT id, ime, prezime, cijena_po_satu, cijena_po_satu + 0.6 AS nova_cijena_po_satu  
FROM zaposlenik;
```

OPIS UPITA:

Ovaj upit pretražuje tablicu **zaposlenik** i odabire attribute **id, ime, prezime, cijena_po_satu**. Prikazuje trenutnu satnicu za svakog zaposlenika i izračunava novu satnicu dodavanjem 0,6

EUR na trenutnu satnicu. Konačni rezultat je popis zaposlenika s trenutnom i novom satnicom.

- **SELECT** **id** odabire jedinstveni identifikator svakog zaposlenika. **ime** odabire ime zaposlenika. **prezime** odabire prezime zaposlenika. **cijena_po_satu** odabire trenutnu satnicu zaposlenika. **cijena_po_satu + 0.6 AS nova_cijena_po_satu** izračunava novu satnicu povećanjem trenutne satnice za 0,6 EUR i daje ovom izračunu alias **nova_cijena_po_satu**.
- **FROM** **zaposlenik** specificira tablicu **zaposlenik** kao izvor podataka za upit.

REZULTAT:

	id	ime	prezime	cijena_po_satu	nova_cijena_po_satu
▶	101	Mihovil	Bulić	8.0	8.6
	102	Manuel	Matić	12.0	12.6
	103	Andrea	Čelebić	11.5	12.1
	104	Ivica	Pomazan	9.5	10.1
	105	Tihomir	Matejčić	7.0	7.6
	106	Matteo	Marić	8.5	9.1
	107	Martin	Modrušan	9.0	9.6
	108	Junuz	Bajrami	10.0	10.6
	109	Marko	Milohanović	7.0	7.6
	110	Petar	Šegon	15.0	15.6
	111	Bruno	Galant	10.5	11.1
	112	Karlo	Juranić	11.0	11.6
	113	Ivan	Zelić	8.5	9.1
	114	Tihomir	Matić	12.0	12.6
	115	Goran	Horvat	11.5	12.1
	116	Marijan	Kovačević	9.5	10.1
	117	Luka	Tomić	7.0	7.6
	118	Dario	Nikolić	8.5	9.1
	119	Ivor	Perić	9.0	9.6
	120	Viktor	Vuković	10.0	10.6
	121	Marin	Babić	7.0	7.6
	122	Nino	Živković	15.0	15.6
	123	Filip	Radovan	10.5	11.1
	124	Ante	Savić	11.0	11.6
	125	Hrvoje	Knežević	8.5	9.1
	126	Rino	Petrović	12.0	12.6
	127	Marko	Jurišić	11.5	12.1
	128	Nikola	Barišić	9.5	10.1
	129	Mateo	Marinović	7.0	7.6
	130	Karlo	Novak	8.5	9.1

7.3. UPIT 3

Popis projekata s rokom do kraja 2024. godine i potrebnim materijalom
 Projektni menadžer ide na godišnji odmor. Zatražio je da mu dostavimo popis svih projekata s rokom do kraja 2024 i potrebnim materijalom kako bi ga naručio prije nego ode, da bi se mogao neometano odmoriti.

TRAŽENO RJEŠENJE:

naziv_projekta, rok_izrade, kolicina, naziv_materijala, opis, napomena

KOD ZA UPIT:

```
SELECT p.naziv_projekta, p.rok_izrade, u.kolicina, m.naziv_materijala, m.opis,  
m.napomena  
FROM utrosak_materijala AS u  
JOIN projekt AS p ON p.id = u.id_projekt  
JOIN materijal AS m ON u.id_materijal = m.id  
WHERE p.rok_izrade <= '2024-12-31';
```

OPIS UPITA:

Ovaj upit pretražuje bazu podataka kako bi pronašao sve projekte s rokom izrade do kraja 2024. godine i potrebnim materijalom za te projekte. Upit odabire naziv projekta, rok izrade, količinu potrebnog materijala, naziv materijala, opis materijala i napomenu vezanu za materijal. Konačni rezultat je popis projekata s potrebnim materijalom, što omogućava projektnom menadžeru da naruči sav potreban materijal prije nego ode na godišnji odmor.

- **SELECT p.naziv_projekta** odabire naziv projekta iz tablice **projekt**. **p.rok_izrade** odabire rok izrade projekta iz tablice **projekt**. **u.kolicina** odabire količinu materijala potrebnu za projekt iz tablice **utrosak_materijala**. **m.naziv_materijala** odabire naziv materijala iz tablice **materijal**. **m.opis** odabire opis materijala iz tablice **materijal**. **m.napomena** odabire napomenu vezanu za materijal iz tablice **materijal**.
- **FROM utrosak_materijala AS u** specificira tablicu **utrosak_materijala** kao izvor podataka za upit i daje joj alias **u**.
- **JOIN projekt AS p ON p.id = u.id_projekt** pridružuje tablicu **projekt** tablici **utrosak_materijala** na temelju zajedničkog atributa id projekta.
- **JOIN materijal AS m ON u.id_materijal = m.id** pridružuje tablicu **materijal** tablici **utrosak_materijala** na temelju zajedničkog atributa id materijala.
- **WHERE p.rok_izrade <= '2024-12-31'** filtrira rezultate tako da se prikažu samo projekti s rokom izrade do kraja 2024. godine.

REZULTAT:

	naziv_projekta	rok_izrade	kolicina	naziv_materijala	opis	napomena
►	Silos	2024-03-30	4	10mm P265GH	Debljina stijenke i kvaliteta	dimenzija 3000x1500
	Silos	2024-03-30	11	UNP120 S235	120x60mm su gabariti profila	6 metara je kom cijevi
	Silos	2024-03-30	5	5mm S355	Debljina stijenke i kvaliteta	dimenzija 2000x1000
	Torre Tension	2024-04-30	9	UNP120 S235	120x60mm su gabariti profila	6 metara je kom cijevi
	Utezi	2024-04-30	8	20mm S235	Debljina stijenke i kvaliteta	dimenzija 2500x1250
	Pregrada skladišta	2024-05-25	6	10mm P265GH	Debljina stijenke i kvaliteta	dimenzija 3000x1500
	Pregrada skladišta	2024-05-25	7	5mm S355	Debljina stijenke i kvaliteta	dimenzija 2000x1000
	Pregrada skladišta	2024-05-25	12	UNP120 S235	120x60mm su gabariti profila	6 metara je kom cijevi
	Inox bubanj	2024-05-30	9	20mm S235	Debljina stijenke i kvaliteta	dimenzija 2500x1250
	Plodine - Varaždinske Toplice	2024-06-01	10	UNP120 S235	120x60mm su gabariti profila	6 metara je kom cijevi
	Totem	2024-06-10	5	10mm P265GH	Debljina stijenke i kvaliteta	dimenzija 3000x1500
	Totem	2024-06-10	8	5mm S355	Debljina stijenke i kvaliteta	dimenzija 2000x1000
	Totem	2024-06-10	15	UNP120 S235	120x60mm su gabariti profila	6 metara je kom cijevi
	Protupožarno stepenište	2024-08-10	10	20mm S235	Debljina stijenke i kvaliteta	dimenzija 2500x1250
	Rešetka za kanal	2024-08-30	11	UNP120 S235	120x60mm su gabariti profila	6 metara je kom cijevi
	Nadstrešnica	2024-09-30	8	IPE300 S235	300x150mm su gabariti profila	12 metara je kom cijevi
	Platforma	2024-10-20	2	4/5mm S235	Debljina stijenke i kvaliteta	dimenzija 3000x1500
	Platforma	2024-10-20	6	IPE300 S235	300x150mm su gabariti profila	12 metara je kom cijevi
	Totem	2024-10-25	10	f1 18mm S235	Pronjler i kvaliteta	6 metara je kom
	Totem	2024-10-25	2	HEB100 S235	100x100mm su gabariti profila	6 metara je kom cijevi
	Stalak	2024-11-05	5	UNP120 S235	120x60mm su gabariti profila	6 metara je kom cijevi
	Metalna kutija	2024-11-15	15	5mm AISI304	Debljina stijenke i kvaliteta	dimenzija 2000x1000
	Nosač kablova	2024-11-25	3	4/5mm S235	Debljina stijenke i kvaliteta	dimenzija 3000x1500
	Nosač kablova	2024-11-25	1	HEB100 S235	100x100mm su gabariti profila	6 metara je kom cijevi
	Čelična posuda	2024-12-05	4	HEB100 S235	100x100mm su gabariti profila	6 metara je kom cijevi
	Okvir filtera	2024-12-15	2	8mm S235	Debljina stijenke i kvaliteta	dimenzija 3000x1500
	Metalni okvir	2024-12-20	4	12mm S235	Debljina stijenke i kvaliteta	dimenzija 2500x1250

7.4. UPIT 4

Prikaz popisa svih alata koji su dostavljeni na servis.

Neki alati nam se nisu vratili sa servisa, pa je serviser Ivan tražio popis svih naših alata koji su mu dostavljeni kako bi ih lakše našao u svojoj radioni

TRAŽENO RJEŠENJE:

naziv_alata, kratki_opis

KOD ZA UPIT:

SELECT naziv_alata, kratki_opis

FROM alat

WHERE status_alata = 'servis';

OPIS UPITA:

Ovaj upit pretražuje bazu podataka kako bi pronašao sve alate koji su trenutno na servisu.

Upit odabire naziv i kratki opis alata iz tablice **alat** i filtrira rezultate tako da se prikažu samo oni alati koji imaju status 'servis'. Konačni rezultat je popis alata koji su trenutno na servisu, što omogućava serviseru Ivanu da ih lakše pronađe u svojoj radioni.

- **SELECT naziv_alata** odabire naziv alata iz tablice **alat**. **kratki_opis** odabire kratki opis alata iz tablice **alat**.
- **FROM alat** specificira tablicu **alat** kao izvor podataka za upit.
- **WHERE status_alata = 'servis'** filtrira rezultate tako da se prikažu samo one n-torke gdje je status alata 'servis'.

REZULTAT:

	naziv_alata	kratki_opis
►	Milwaukee 6122-30	Velika kutna brusilica ,oznaka B3
	Metabo W820-RT	Velika kutna brusilica ,oznaka B4
	Hugong Extremig 180	Mig aparat za zavarivanje ,oznaka B12
	Bosch GSB 570	Ručna bušilica ,oznaka B17
	DeWalt DWE4887N	Ručni brusni alat za obradu metala, oznaka B19
	Bosch GWS 18V-10 C	Aku kutna brusilica za obradu metala, oznaka B23
	Hugong Extremig 180	Mig aparat za zavarivanje, oznaka B30
	Bosch GSB 570	Ručna bušilica, oznaka B35
	Hugong Extremig 180	Mig aparat za zavarivanje, oznaka B40
	Bosch GSB 570	Ručna bušilica, oznaka B45
	Hugong Extremig 180	Mig aparat za zavarivanje, oznaka B50

7.5. UPIT 5

Prikaz svih ukrcaja kamiona registarskih tablica PU-5566-C sa popisom segmenata, količinom i datumom ukrcaja.

Šofer Luka je napustio skladište i zaboravio ponijeti papir s popisom segmenata, količinom i datumom. Trenutno je na granici sa Slovenijom i granična policija ga traži dokument tereta i sve predhodne ukrcaje koji su ukrcani iz proizvodnje . Moramo mu hitno dostaviti tražene podatke.

TRAŽENO RJEŠENJE:

kolicina, datum_ukrcaja, registarske_oznake, naziv_segmenta

KOD ZA UPIT:

```
SELECT u.kolicina, u.datum_ukrcaja, u.registarske_oznake, s.naziv_segmenta
FROM isporuka AS u
JOIN segment AS s ON u.id_segment = s.id
WHERE u.registarske_oznake = 'PU-5566-C';
```

OPIS UPITA:

Ovaj upit pretražuje bazu podataka kako bi pronašao sve relevantne informacije o ukrcaju kamiona s registarskim oznakama "PU-5566-C". Upit spaja (**JOIN**) tablice **isporuka** i **segment** kako bi dobio nazive segmenata, a zatim filtrira rezultate prema registarskim oznakama. Konačni rezultat je popis segmenata, količina, datuma ukrcaja i registarskih oznaka, što omogućava vozaču da graničnoj policiji pruži potrebne informacije o teretu.

- **SELECT u.kolicina** odabire količinu iz tablice **isporuka**. **u.datum_ukrcaja** odabire datum ukrcaja iz tablice **isporuka**. **u.registarske_oznake** odabire registarske oznake iz tablice **isporuka**. **s.naziv_segmenta** odabire naziv segmenta iz tablice **segment**.
- **FROM isporuka AS u** specificira tablicu **isporuka** kao početnu točku za upit i daje joj alias **u**.

- **JOIN** segment **AS** s **ON** **u.id_segment = s.id** povezuje tablicu **segment** s tablicom **isporuka** na temelju **id_segment** iz tablice **isporuka** i **id** iz tablice **segment**, te daje tablici **segment** alias **s**.
- **WHERE** **u.registarske_oznake = 'PU-5566-C'** filtrira rezultate tako da se prikažu samo one n-torke gdje su registarske oznake kamiona "PU-5566-C".

REZULTAT:

	kolicina	datum_ukrcaja	registarske_oznake	naziv_segmenta
▶	4	2024-02-21	PU-5566-C	Noseći stupovi
	1	2024-02-24	PU-5566-C	Perforirani lim
	4	2024-03-04	PU-5566-C	Noseći stupovi
	4	2024-03-16	PU-5566-C	Poklopac posude

7.6. UPIT 6

Prikaz projekata po klijentima, uključujući broj segmenata za svaki projekt

Naša firma ima dvije podružnice proizvodnje sa istim brojem radnika. Odlučili su nagraditi sve radnike proizvodnje koja je napravila više projekata pa su nas tražili popis projekata po klijentima, uključujući broj segmenata za svaki projekt.

TRAŽENO RJEŠENJE:

naziv_klijenta, naziv_projekta, broj_segmenata

KOD ZA UPIT:

```
SELECT k.naziv_klijenta, p.naziv_projekta, COUNT(s.id) AS broj_segmenata
FROM klijent k
JOIN projekt AS p ON k.id = p.id_klijent
JOIN segment AS s ON p.id = s.id_projekt
GROUP BY k.naziv_klijenta, p.naziv_projekta;
```

OPIS UPITA:

Ovaj upit pretražuje bazu podataka kako bi prikazao sve projekte grupirane po klijentima, uključujući broj segmenata za svaki projekt. Upit koristi spajanje (**JOIN**) tablica **klijent**, **projekt**, i **segment** kako bi dobio sve potrebne informacije. Konačni rezultat je popis klijenata s njihovim projektima i brojem segmenata za svaki projekt, što pomaže u ocjeni proizvodnih pogona na temelju broja dovršenih projekata.

- **SELECT k.naziv_klijenta** odabire naziv klijenta iz tablice **klijent**. **p.naziv_projekta** odabire naziv projekta iz tablice **projekt**. **COUNT(s.id) AS broj_segmenata** broji broj segmenata (komponenti) u svakom projektu i daje tom broju alias **broj_segmenata**.
- **FROM klijent k** specificira tablicu **klijent** kao početnu točku za upit i daje joj alias **k**.

- **JOIN** **projekt AS p ON k.id = p.id_klijent** povezuje tablicu **projekt** s tablicom **klijent** na temelju **id_klijent** iz tablice **projekt** i **id** iz tablice **klijent**, te daje tablici **projekt** alias **p**.
- **JOIN** **segment AS s ON p.id = s.id_projekt** povezuje tablicu **segment** s tablicom **projekt** na temelju **id_projekt** iz tablice **segment** i **id** iz tablice **projekt**, te daje tablici **segment** alias **s**.
- **GROUP BY k.naziv_klijenta, p.naziv_projekta** grupira rezultate prema nazivu klijenta i nazivu projekta. Ovo osigurava da se broj segmenata računa za svaki pojedini projekt.

REZULTAT:

	naziv_klijenta	naziv_projekta	broj_segmenata
►	Holcim	Silos	3
	Holcim	Torre Tension	1
	Holcim	Utezi	1
	British American Tobacco	Pregrada skladišta	2
	Rockwool Adriatic	Inox bubanj	2
	Plodine	Plodine - Varaždinske Toplice	3
	Plodine	Totem	1
	Plodine	Protupožarno stepenište	3
	INA - Naftaplin d.d.	Rešetka za kanal	1
	INA - Naftaplin d.d.	Nadstrešnica	2
	HEP Proizvodnja	Platforma	3
	Kaufland	Totem	1
	Kaufland	Stalak	1
	AD Plastik d.o.o.	Metalna kutija	2
	A1 Hrvatska	Nosač kablova	2
	Podravka Vegeta	Čelična posuda	2
	Franck	Okvir filtera	1
	T-com Hrvatska	Metalni okvir	1
	Philip Morris Zagreb	Mašina za pakiranje	1
	Fortenova Grupa	Ormarić	2
	Rimac Automobili	Metalni okvir za auto	2
	Tommy	Polica	2
	Orbico	Metalna rešetka	2
	INA - Industrija nafte	Čelična ploča	2
	Decathlon	Metalni okvir za reklamu	2
	Ledo	Kutija	2
	Agrokor	Metalni stalak za proizvode	2
	Belupo	Čelični medicinski ormarić	2
	Atlantic Trade	Okvir za pult	2
	Emmezeta	Čelična konstrukcija	2

7.7. UPIT 7

Prikaz svih projekata koje treba završiti u narednih 30 dana, uključujući informacije o klijentu i odgovornoj osobi

Zbog planiranja godišnjih odmora, potrebno je utvrditi koje projekte je potrebno dovršiti u narednih 30 dana. Uz to, voditelj je zatražio i informacije o klijentu i odgovornoj osobi za te projekte.

TRAŽENO RJEŠENJE:

id_projekt, naziv_projekta, rok_izrade, naziv_klijenta, odgovorna_osoba

KOD ZA UPIT:

```
SELECT p.id AS id_projekta, p.naziv_projekta, p.rok_izrade, k.naziv_klijenta,  
k.odgovorna_osoba  
FROM projekt AS p  
JOIN klijent AS k ON p.id_klijent = k.id  
WHERE p.rok_izrade BETWEEN CURDATE() AND DATE_ADD(CURDATE(),  
INTERVAL 30 DAY);
```

OPIS UPITA:

Ovaj upit pretražuje bazu podataka kako bi pronašao sve projekte koje je potrebno dovršiti u narednih 30 dana. Spaja tablice **projekt** i **klijent** kako bi uključio informacije o klijentu i odgovornoj osobi za svaki projekt. Konačni rezultat je popis projekata s njihovim ID-evima, nazivima, rokovima izrade, nazivima klijenata i odgovornim osobama, što omogućava planiranje rada i godišnjih odmora s obzirom na nadolazeće rokove.

- **SELECT p.id AS id_projekta** odabire **id** projekta iz tablice **projekt** i daje mu alias **id_projekta**. **p.naziv_projekta** odabire naziv projekta iz tablice **projekt**. **p.rok_izrade** odabire rok izrade projekta iz tablice **projekt**. **k.naziv_klijenta** odabire naziv klijenta iz tablice **klijent**. **k.odgovorna_osoba** odabire odgovornu osobu iz tablice **klijent**.
- **FROM projekt AS p** specificira tablicu **projekt** kao početnu točku za upit i daje joj alias **p**.
- **JOIN klijent AS k ON p.id_klijent = k.id** povezuje tablicu **klijent** s tablicom **projekt** na temelju **id_klijent** iz tablice **projekt** i **id** iz tablice **klijent**, te daje tablici **klijent** alias **k**.
- **WHERE p.rok_izrade BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 30 DAY)** filtrira projekte prema roku izrade tako da se prikažu samo oni projekti čiji je rok izrade između današnjeg datuma (**CURDATE()**) i datuma koji je 30 dana od današnjeg dana (**DATE_ADD(CURDATE(), INTERVAL 30 DAY)**)

REZULTAT:

	id_projekta	naziv_projekta	rok_izrade	naziv_klijenta	odgovorna_osoba
▶	204	Pregrada skladišta	2024-05-25	British American Tobacco	Toni Belić
	205	Inox bubanj	2024-05-30	Rockwool Adriatic	Ivana Marić
	206	Plodine - Varaždinske Toplice	2024-06-01	Plodine	Lucas Perić
	207	Totem	2024-06-10	Plodine	Lucas Perić

7.8. UPIT 8

Ukupan trošak materijala po projektu

Zbog mjesečne inventure, koordinator za nabavu materijala je zatražio ukupan trošak materijala po pojedinom projektu.

TRAŽENO RJEŠENJE:

naziv_projekta, ukupan_trosak_materijala, naziv_klijenta

KOD ZA UPIT:

```
SELECT klijent.naziv_klijenta, projekt.naziv_projekta, SUM(materijal.cijena_po_komadu *  
utrosak_materijala.kolicina) AS ukupno_trosak_materijala  
FROM projekt  
JOIN utrosak_materijala ON projekt.id = utrosak_materijala.id_projekt  
JOIN materijal ON utrosak_materijala.id_materijal = materijal.id  
JOIN klijent ON projekt.id_klijent = klijent.id  
GROUP BY projekt.id;
```

OPIS UPITA:

Ovaj upit pretražuje bazu podataka kako bi izračunao ukupan trošak materijala za svaki projekt i povezanog klijenta. Prvo spaja tablice **projekt**, **utrosak_materijala**, **materijal**, i **klijent** na temelju zajedničkih ključeva. Zatim koristi funkciju **SUM** za zbrajanje troškova materijala po projektu, gdje se trošak izračunava množenjem cijene po komadu materijala sa količinom utrošenog materijala za taj projekt. Konačni rezultat je popis projekata s njihovim nazivima, nazivima klijenata, i ukupnim troškovima materijala. Ove informacije omogućuju koordinatoru za nabavu materijala da provede mjesečnu inventuru i prati troškove po projektima.

- **SELECT klijent.naziv_klijenta** odabire naziv klijenta iz tablice **klijent**.
projekt.naziv_projekta odabire naziv projekta iz tablice **projekt**.
- **SUM(materijal.cijena_po_komadu * utrosak_materijala.kolicina) AS ukupno_trosak_materijala** izračunava ukupan trošak materijala za svaki projekt. Množi cijenu po komadu materijala sa količinom materijala utrošenog za taj projekt, a zatim sve te vrijednosti zbraja za svaki projekt.
- **FROM projekt** specificira tablicu **projekt** kao početnu točku za upit.
- **JOIN utrosak_materijala ON projekt.id = utrosak_materijala.id_projekt** povezuje tablicu **utrosak_materijala** s tablicom **projekt** na temelju **id_projekt** iz tablice **utrosak_materijala** i **id** iz tablice **projekt**.
- **JOIN materijal ON utrosak_materijala.id_materijal = materijal.id** povezuje tablicu **materijal** s tablicom **utrosak_materijala** na temelju **id_materijal** iz tablice **utrosak_materijala** i **id** iz tablice **materijal**.
- **JOIN klijent ON projekt.id_klijent = klijent.id** povezuje tablicu **klijent** s tablicom **projekt** na temelju **id_klijent** iz tablice **projekt** i **id** iz tablice **klijent**.

- **GROUP BY projekt.id** grupira rezultate prema **id** iz tablice **projekt**. Ovaj korak je potreban kako bi se dobio ukupan trošak materijala za svaki pojedini projekt.

REZULTAT:

	naziv_klijenta	naziv_projekta	ukupno_trosak_materijala
►	Holcim	Silos	1352.1
	Holcim	Torre Tension	568.8
	Holcim	Utezi	1644.8
	British American Tobacco	Pregrada skladišta	1717.5
	Rockwool Adriatic	Inox bubanj	1850.4
	Plodine	Plodine - Varaždinske Toplice	632.0
	Plodine	Totem	1861.0
	Plodine	Protupožarno stepenište	2056.0
	INA - Naftaplin d.d.	Rešetka za kanal	695.2
	INA - Naftaplin d.d.	Nadstrešnica	2356.8
	HEP Proizvodnja	Platforma	1904.8
	Kaufland	Totem	362.4
	Kaufland	Stalak	316.0
	AD Plastik d.o.o.	Metalna kutija	2287.5
	A1 Hrvatska	Nosač kablova	281.0
	Podravka Vegeta	Čelična posuda	300.8
	Franck	Okvir filtera	320.4
	T-com Hrvatska	Metalni okvir	922.0
	Philip Morris Zagreb	Mašina za pakiranje	512.4
	Fortenova Grupa	Ormarić	683.2
	Rimac Automobili	Metalni okvir za auto	1922.4
	Tommy	Polica	953.5
	Orbico	Metalna rešetka	887.2
	INA - Industrija nafte	Čelična ploča	2027.7
	Decathlon	Metalni okvir za reklamu	2419.9
	Ledo	Kutija	875.4
	Agrokor	Metalni stalak za proizvode	357.6
	Belupo	Čelični medicinski ormarić	482.4
	Atlantic Trade	Okvir za pult	758.7
	Emmezeta	Čelična konstrukcija	45088.4

7.9. UPIT 9

Svi projekte koji su prekoračili rok izrade

Upravitelj je dobio mail nezadovoljnog klijenta zbog kašnjenja isporuke projekta. Projektni menadžer tražio nas je popis svih projekata koji su prekoračili rok izrade kako bi mogao lakše procijeniti od kojeg nezadovoljnog klijenta je stigla kritika.

TRAŽENO RJEŠENJE:

naziv_klijenta, naziv_projekta, naziv_segmenta, kolicina, rok_izrade

KOD ZA UPIT:

```
SELECT DISTINCT k.naziv_klijenta, p.naziv_projekta, s.naziv_segmenta, s.kolicina,
p.rok_izrade
FROM projekt AS p
```

```

JOIN segment AS s ON p.id = s.id_projekt
JOIN proizvodnja AS pr ON pr.id_segment = s.id
JOIN klijent AS k ON k.id = p.id_klijent
WHERE p.rok_izrade < CURDATE() AND pr.status_proizvodnje = 'u izradi';

```

OPIS UPITA:

Ovaj upit pretražuje bazu podataka kako bi pronašao sve projekte koji su prekoračili rok izrade i trenutno su u izradi. Prvo spaja tablice **projekt**, **segment**, **proizvodnja**, i **klijent** na temelju zajedničkih ključeva. Zatim filtrira projekte čiji je rok izrade prošao i trenutno su u statusu 'u izradi'. Konačni rezultat je popis jedinstvenih projekata s nazivima klijenata, nazivima projekata, nazivima segmenata, količinama i rokovima izrade. Ovaj popis omogućava projektnom menadžeru da lakše identificira projekte koji kasne i potencijalno nezadovoljne klijente.

- **SELECT DISTINCT** koristi se za odabir jedinstvenih redaka, eliminirajući duplikate iz rezultata. **k.naziv_klijenta** odabire naziv klijenta iz tablice **klijent**. **p.naziv_projekta** odabire naziv projekta iz tablice **projekt**. **s.naziv_segmenta** odabire naziv segmenta iz tablice **segment**. **s.kolicina** odabire količinu iz tablice **segment**. **p.rok_izrade** odabire rok izrade iz tablice **projekt**.
- **FROM projekt AS p** specificira tablicu **projekt** kao početnu točku za upit.
- **JOIN segment AS s ON p.id = s.id_projekt** povezuje tablicu **segment** s tablicom **projekt** na temelju **id_projekt** iz tablice **segment** i **id** iz tablice **projekt**.
- **JOIN proizvodnja AS pr ON pr.id_segment = s.id** povezuje tablicu **proizvodnja** s tablicom **segment** na temelju **id_segment** iz tablice **proizvodnja** i **id** iz tablice **segment**.
- **JOIN klijent AS k ON k.id = p.id_klijent** povezuje tablicu **klijent** s tablicom **projekt** na temelju **id** iz tablice **klijent** i **id_klijent** iz tablice **projekt**.
- **WHERE p.rok_izrade < CURDATE()** filtrira projekte čiji je rok izrade prošao (manji od trenutnog datuma). **AND pr.status_proizvodnje = 'u izradi'** filtrira samo one segmente koji su trenutno u izradi.

REZULTAT:

	naziv_klijenta	naziv_projekta	naziv_segmenta	kolicina	rok_izrade
▶	Holcim	Torre Tension	Glavna konstrukcija	2	2024-04-30
	Holcim	Utezi	Utezi	52	2024-04-30

7.10. UPIT 10

Ukupna količina materijala na lageru

Dobili smo poveći broj zahtjeva za nove projekte, voditelj traži popis i količinu materijala na lageru kako bi znao koje projekte prihvatiti a koje mora ostaviti na čekanju dok ne stigne još materijala.

TRAŽENO RJEŠENJE:

naziv_materijala, ukupna_kolicina, napomena

KOD ZA UPIT:

```
SELECT materijal.naziv_materijala, SUM(lager_materijala.kolicina) AS ukupna_kolicina,  
materijal.napomena  
FROM materijal  
JOIN lager_materijala ON materijal.id = lager_materijala.id_materijal  
GROUP BY materijal.id;
```

OPIS UPITA:

Ovaj upit prebrojava ukupnu količinu svakog materijala koji je na lageru, grupirajući rezultate prema tipu materijala. Prvo spaja tablice **materijal** i **lager_materijala** na temelju zajedničkog **id_materijal** i **id**. Zatim koristi **SUM** za zbrajanje količina materijala za svaki tip i prikazuje naziv materijala, ukupnu količinu i napomenu o materijalu. Konačni rezultat je popis materijala s njihovim ukupnim količinama na lageru, što pomaže voditelju da donese odluke o prihvatanju novih projekata na temelju dostupnosti materijala.

- **SELECT materijal.naziv_materijala, SUM(lager_materijala.kolicina) AS ukupna_kolicina, materijal.napomena** nam odabire **naziv_materijala** i **napomena** iz tablice **materijal** uz to koristi agregatnu funkciju **SUM** za zbrajanje količine materijala iz tablice **lager_materijala** i daje toj sumi alias **ukupna_kolicina**.
- **FROM materijal** nam specificira tablicu **materijal** kao izvor podataka.
- **JOIN lager_materijala ON materijal.id = lager_materijala.id_materijal** koristi **JOIN** za spajanje dviju tablica. Povezuje tablicu **materijal** s tablicom **lager_materijala** koristeći **id** iz tablice **materijal** i **id_materijal** iz tablice **lager_materijala**.
- **GROUP BY materijal.id** grupira rezultate prema **id** iz tablice **materijal**. To znači da će se za svaki materijal izračunati ukupna količina.

REZULTAT:

	naziv_materijala	ukupna_kolicina	napomena
►	15mm S235	25	dimenzija 3000x1500
	20mm S235	17	dimenzija 2500x1250
	10mm P265GH	47	dimenzija 3000x1500
	5mm S355	24	dimenzija 2000x1000
	HEA120 S235	19	12 metara je kom cijevi
	UNP120 S235	19	6 metara je kom cijevi
	IPE300 S235	8	12 metara je kom cijevi
	HEB100 S235	17	6 metara je kom cijevi
	4/5mm S235	5	dimenzija 3000x1500
	fi 18mm S235	9	6 metara je kom
	5mm AISI304	11	dimenzija 2000x1000
	8mm S235	16	dimenzija 3000x1500
	12mm S235	7	dimenzija 2500x1250
	5mm AISI316	14	dimenzija 2000x1000
	20mm S355	19	dimenzija 3000x1500
	15mm S275	8	dimenzija 2500x1250
	10mm S235JR	23	dimenzija 3000x1500
	HEA140 S235	10	12 metara je kom cijevi
	IPE360 S235	18	12 metara je kom cijevi
	HEB160 S235	6	6 metara je kom cijevi
	UNP180 S235	12	6 metara je kom cijevi
	4/5mm S235	15	dimenzija 2500x1250
	6/7mm S235	9	dimenzija 2500x1250

7.11. UPIT 11

Prikaz ukupnog troška rada zaposlenika po projektu

Radi se evidencija ukupnog troska rada zaposlenika po projektu.

TRAŽENO RJEŠENJE:

naziv_projekta, ukupan_trosak_rada

KOD ZA UPIT:

```
SELECT p.naziv_projekta, SUM(us.utrosak_sati * z.cijena_po_satu) AS
ukupni_trosak_rada
```

```
FROM projekt AS p
```

```
JOIN segment AS s ON p.id = s.id_projekt
```

```
JOIN proizvodnja AS pr ON s.id = pr.id_segment
```

```
JOIN utrosak_sati AS us ON pr.id = us.id_proizvodnja
```

```
JOIN zaposlenik AS z ON us.id_zaposlenik = z.id
```



```
GROUP BY p.naziv_projekta
ORDER BY ukupni_trosak_rada DESC;
```

OPIS UPITA:

Ovaj upit koristi nekoliko **JOIN** operacija za pridruživanje više tablica i dobivanje potrebnih podataka. Cilj je izračunati ukupni trošak rada za svaki projekt. Kroz više **JOIN** operacija, upit spaja tablice **projekt**, **segment**, **proizvodnja**, **utrosak_sati**, i **zaposlenik** kako bi se dobili podaci o nazivima projekata, utrošenim satima rada, cijenama po satu zaposlenika, te ukupnim troškovima rada. Rezultati su grupirani po nazivima projekata i sortirani prema ukupnim troškovima rada u silaznom redoslijedu.

```
SELECT p.naziv_projekta, SUM(us.utrosak_sati * z.cijena_po_satu) AS ukupni_trosak_rada
```

Ova naredba odabire dva podatka:

- **p.naziv_projekta**: Naziv projekta iz tablice projekt.
- **SUM(us.utrosak_sati * z.cijena_po_satu) AS ukupni_trosak_rada**: Izračunava ukupni trošak rada za svaki projekt. Ovdje se množe utrošeni sati (us.utrosak_sati) s cijenom po satu zaposlenika (z.cijena_po_satu) i zbrajaju rezultati za sve zaposlenike koji su radili na projektu.

```
FROM projekt AS p
```

Definira tablicu projekt kao osnovnu tablicu za upit i dodjeljuje joj alias p za lakše referenciranje.

```
JOIN segment AS s ON p.id = s.id_projekt
```

Pridružuje tablicu segment s tablicom projekt na temelju podudaranja stupaca id_projekt iz tablice segment i id iz tablice projekt. Ovo znači da se za svaki projekt pridružuju svi njegovi segmenti.

```
JOIN proizvodnja AS pr ON s.id = pr.id_segment
```

Pridružuje tablicu proizvodnja s tablicom segment na temelju podudaranja stupaca id_segment iz tablice proizvodnja i id iz tablice segment. Ovo znači da se za svaki segment pridružuju sve faze proizvodnje koje se odnose na taj segment.

```
JOIN utrosak_sati AS us ON pr.id = us.id_proizvodnja
```

Pridružuje tablicu utrosak_sati s tablicom proizvodnja na temelju podudaranja stupaca id_proizvodnja iz tablice utrosak_sati i id iz tablice proizvodnja. Ovo znači da se za svaku fazu proizvodnje pridružuju svi zapisi o utrošenim satima rada zaposlenika.

```
JOIN zaposlenik AS z ON us.id_zaposlenik = z.id
```

Pridružuje tablicu zaposlenik s tablicom utrosak_sati na temelju podudaranja stupaca id_zaposlenik iz tablice utrosak_sati i id iz tablice zaposlenik. Ovo znači da se za svaki zapis o utrošenim satima pridružuju podaci o odgovarajućem zaposleniku.

GROUP BY p.naziv_projekta

Grupira rezultate prema nazivu projekta (p.naziv_projekta). Ovo omogućuje agregaciju podataka (korištenje funkcije SUM) za svaki projekt zasebno.

ORDER BY ukupni_trosak_rada **DESC**;

Sortira rezultate prema izračunatom ukupnom trošku rada u silaznom redoslijedu (DESC). To znači da će projekti s najvećim ukupnim troškom rada biti prikazani prvi.

REZULTAT:

	naziv_projekta	ukupni_trosak_rada
►	Silos	251.50
	Pregrada skladišta	119.50
	Inox bubanj	112.25
	Plodine - Varaždinske Toplice	92.50
	Utezi	41.50
	Torre Tension	27.50

7.12. UPIT 12

Prikaz materijala koji se najviše koristi po segmentima

Radi se evidencija utroska materijala po segmentima

TRAŽENO RJEŠENJE:

naziv_segmenta, naziv_materijala, ukupno_koristeno

KOD ZA UPIT:

```
SELECT s.naziv_segmenta, m.naziv_materijala, SUM(um.kolicina) AS  
ukupno_koristeno  
FROM segment AS s  
JOIN utrosak_materijala AS um ON s.id_projekt = um.id_projekt  
JOIN materijal AS m ON um.id_materijal = m.id  
GROUP BY s.naziv_segmenta, m.naziv_materijala  
ORDER BY ukupno_koristeno DESC;
```


OPIS UPITA:

Ovaj upit koristi nekoliko **JOIN** operacija za pridruživanje više tablica i dobivanje potrebnih podataka. Cilj je izračunati ukupnu količinu korištenog materijala za svaki segment. Kroz više **JOIN** operacija, upit spaja tablice **segment**, **utrosak_materijala**, i **materijal** kako bi se dobili podaci o nazivima segmenata, nazivima materijala, i ukupnoj količini korištenog materijala. Rezultati su grupirani po nazivima segmenata i materijala, te sortirani prema ukupnoj količini korištenog materijala u silaznom redoslijedu.

SELECT s.naziv_segmenta, m.naziv_materijala, SUM(um.kolicina) AS ukupno_koristeno

Ova naredba odabire tri podatka:

- **s.naziv_segmenta**: Naziv segmenta iz tablice segment.
- **m.naziv_materijala**: Naziv materijala iz tablice materijal.
- **SUM(um.kolicina) AS ukupno_koristeno**: Izračunava ukupnu količinu materijala korištenog za svaki segment. Funkcija SUM zbraja količine materijala (um.kolicina) za svaki segment i materijal.

FROM segment AS s

Definira tablicu segment kao osnovnu tablicu za upit i dodjeljuje joj alias s za lakše referenciranje.

JOIN utrosak_materijala AS um ON s.id_projekt = um.id_projekt

Pridružuje tablicu utrosak_materijala s tablicom segment na temelju podudaranja stupaca id_projekt iz tablice segment i id_projekt iz tablice utrosak_materijala. Ovo znači da se za svaki segment pridružuju svi zapisi o utrošku materijala koji pripadaju istom projektu.

JOIN materijal AS m ON um.id_materijal = m.id

Pridružuje tablicu materijal s tablicom utrosak_materijala na temelju podudaranja stupaca id_materijal iz tablice utrosak_materijala i id iz tablice materijal. Ovo znači da se za svaki utrošeni materijal pridružuju podaci o materijalu.

GROUP BY s.naziv_segmenta, m.naziv_materijala

Grupira rezultate prema nazivu segmenta (s.naziv_segmenta) i nazivu materijala (m.naziv_materijala). Ovo omogućuje agregaciju podataka (korištenje funkcije SUM) za svaki segment i materijal zasebno.

ORDER BY ukupno_koristeno DESC

Sortira rezultate prema izračunatoj ukupnoj količini korištenog materijala u silaznom redoslijedu (DESC). To znači da će segmenti i materijali s najvećom ukupnom količinom korištenog materijala biti prikazani prvi.

REZULTAT:

	naziv_segmenta	naziv_materijala	ukupno_koristeno
►	Konstrukcija baza	12mm P355GH	26
	Konstrukcija dodatak	12mm P355GH	26
	Konstrukcija baza	30mm S235	24
	Konstrukcija dodatak	30mm S235	24
	Konstrukcija baza	25mm S235	22
	Konstrukcija dodatak	25mm S235	22
	Glavna konstrukcija	UNP120 S235	20
	Konstrukcija baza	fi 32mm Monel 400	20
	Konstrukcija dodatak	fi 32mm Monel 400	20
	Konstrukcija dodatak	fi 28mm Hastelloy C276	18
	Konstrukcija baza	fi 28mm Hastelloy C276	18
	Konstrukcija baza	fi 30mm S235	16

Napomena: Tablica ima 88 redova ,ali zbog veličine samo zalijepio prvih desetak redova.

7.13. UPIT 13

Prikaz alata koji su najviše zaduživani od strane zaposlenika

Radi se evidencija najviše korištenih alata

TRAŽENO RJEŠENJE:

naziv_alata, broj_zaduzenja

KOD ZA UPIT:

```
SELECT a.naziv_alata, COUNT(za.id) AS broj_zaduzenja
FROM alat AS a
JOIN zaduzenje_alata AS za ON a.id = za.id_alat
GROUP BY a.naziv_alata
ORDER BY broj_zaduzenja DESC;
```

OPIS UPITA:

Ovaj upit koristi **JOIN** operaciju za pridruživanje tablica **alat** i **zaduzenje_alata** kako bi se dobili potrebni podaci. Cilj je izračunati koliko puta je svaki alat zadužen. Kroz **JOIN** operaciju, upit spaja tablice **alat** i **zaduzenje_alata** na temelju podudaranja ID-ova alata.

SELECT a.naziv_alata, COUNT(za.id) AS broj_zaduzenja

Ova naredba odabire dva podatka:

- **a.naziv_alata**: Naziv alata iz tablice alat.
- **COUNT(za.id) AS broj_zaduzenja**: Izračunava broj zaduženja za svaki alat. Funkcija COUNT broji koliko puta se svaki alat pojavljuje u tablici zaduzenje_alata.

FROM alat AS a

Definira tablicu alat kao osnovnu tablicu za upit i dodjeljuje joj alias a za lakše referenciranje.

JOIN zaduzenje_alata AS za ON a.id = za.id_alat

Pridružuje tablicu zaduzenje_alata s tablicom alat na temelju podudaranja stupaca id iz tablice alat i id_alat iz tablice zaduzenje_alata. Ovo znači da se za svaki alat pridružuju svi zapisi o njegovom zaduženju.

GROUP BY a.naziv_alata

Grupira rezultate prema nazivu alata (a.naziv_alata). Ovo omogućuje agregaciju podataka (korištenje funkcije COUNT) za svaki alat zasebno.

ORDER BY broj_zaduzenja DESC

Sortira rezultate prema izračunatom broju zaduženja u silaznom redoslijedu (DESC). To znači da će alati s najvećim brojem zaduženja biti prikazani prvi.

REZULTAT:

	naziv_alata	broj_zaduzenja
►	Hugong INVERMIG 500 III IGBT	4
	Hugong MIG 250	4
	Lincoln Electric V205T	3
	Hugong Extremig 180	2
	Bosch GSB 2100 RE	1
	Bosch GSB 18V-150 C	1

7.14. UPIT 14

Prikaz svih projekata po statusu proizvodnje

traženi rezultat:

naziv_projekt, status_proizvodnje

kod :

```
SELECT p.naziv_projekta, pr.status_proizvodnje
FROM projekt AS p
JOIN segment AS s ON p.id = s.id_projekt
JOIN proizvodnja AS pr ON s.id = pr.id_segment
GROUP BY p.naziv_projekta, pr.status_proizvodnje
ORDER BY p.naziv_projekta, pr.status_proizvodnje;
```

Opis:

Ovaj upit je koristan za dobivanje pregleda o stanju proizvodnje svih projekata. Može se koristiti za identifikaciju koliko segmenata je trenutno u određenom statusu proizvodnje (npr. "u izradi", "završeno") za svaki projekt. Na taj način, menadžeri mogu pratiti napredak svakog projekta i vidjeti koji projekti ili segmenti zahtijevaju više pažnje ili resursa.

Objasnjene koda:

```
SELECT p.naziv_projekta, pr.status_proizvodnje
```

Ovaj dio upita odabire dva podatka:

p.naziv_projekta: Naziv projekta iz tablice projekt.
pr.status_proizvodnje: Status proizvodnje iz tablice proizvodnja.

```
FROM projekt AS p
```

Definira tablicu projekt kao osnovnu tablicu za upit i dodjeljuje joj alias p za lakše referenciranje.
JOIN segment AS s ON p.id = s.id_projekt

Pridružuje tablicu segment s tablicom projekt na temelju podudaranja stupaca id_projekt iz tablice segment i id iz tablice projekt. Ovo znači da se za svaki projekt pridružuju svi njegovi segmenti.

JOIN segment AS s ON p.id = s.id_projekt: Ovo spaja tablicu projekt s tablicom segment, povezujući projekte s njihovim segmentima.

```
JOIN proizvodnja AS pr ON s.id = pr.id_segment
```

Pridružuje tablicu proizvodnja s tablicom segment na temelju podudaranja stupaca id_segment iz tablice proizvodnja i id iz tablice segment. Ovo znači da se za svaki segment pridružuju sve faze proizvodnje koje se odnose na taj segment.

JOIN proizvodnja AS pr ON s.id = pr.id_segment: Ovo spaja tablicu segment s tablicom proizvodnja, povezujući segmente s njihovim proizvodnim fazama.

GROUP BY p.naziv_projekta, pr.status_proizvodnje

Grupira rezultate prema nazivu projekta (p.naziv_projekta) i statusu proizvodnje (pr.status_proizvodnje). Ovo omogućuje agregaciju podataka (korištenje funkcije COUNT) za svaki projekt i status proizvodnje zasebno.

GROUP BY p.naziv_projekta, pr.status_proizvodnje: Ovo grupira rezultate tako da svaki jedinstveni par naziv_projekta i status_proizvodnje ima svoj vlastiti redak u rezultatu upita.

ORDER BY p.naziv_projekta, pr.status_proizvodnje

Sortira rezultate prema nazivu projekta i statusu proizvodnje. To znači da će rezultati biti prikazani abecednim redoslijedom po nazivu projekta, a zatim po statusu proizvodnje unutar svakog projekta.

ORDER BY p.naziv_projekta, pr.status_proizvodnje: Ovo sortira rezultate prema nazivu projekta, a zatim prema statusu proizvodnje unutar svakog projekta.

Rezultat:

naziv_projekta	status_proizvodnje
Inox bubanj	u izradi
Plodine - Varaždinske Toplice	u izradi
Pregrada skladišta	u izradi
Protupožarno stepenište	u izradi
Silos	izrađeno
Torre Tension	u izradi
Totem	u izradi
Utezi	u izradi

7.15. UPIT 15

Prikaz zaduženih alata po zaposlenicima

Trazeni rezultat:

ime, prezime, naziv_alata, broj_zaduzenja

Kod:

```
SELECT z.ime, z.prezime, a.naziv_alata, COUNT(za.id) AS broj_zaduzenja
FROM zaposlenik AS z
JOIN zaduzenje_alata AS za ON z.id = za.id_zaposlenik
JOIN alat AS a ON za.id_alat = a.id
GROUP BY z.ime, z.prezime, a.naziv_alata
ORDER BY z.prezime, z.ime, broj_zaduzenja DESC;
```

Opis:

Ovaj upit je koristan za dobivanje pregleda o tome koji su alati najviše zaduživani od strane zaposlenika. Može se koristiti za identifikaciju zaposlenika koji najviše koriste određene alate, kao i alata koji su najviše korišteni. Na taj način, menadžeri mogu pratiti upotrebu alata i potencijalno planirati održavanje ili nabavu dodatnih alata.

Opis koda:

```
SELECT z.ime, z.prezime, a.naziv_alata, COUNT(za.id) AS broj_zaduzenja
```

Ovaj dio upita odabire četiri podatka:

z.ime: Ime zaposlenika iz tablice zaposlenik.

z.prezime: Prezime zaposlenika iz tablice zaposlenik.

a.naziv_alata: Naziv alata iz tablice alat.

COUNT(za.id) AS broj_zaduzenja: Izračunava broj zaduženja svakog alata po zaposleniku. Funkcija COUNT broji koliko puta se svaki alat pojavljuje u tablici zaduzenje_alata.

```
FROM zaposlenik AS z
```

Definira tablicu zaposlenik kao osnovnu tablicu za upit i dodjeljuje joj alias z za lakše referenciranje. JOIN zaduzenje_alata AS za ON z.id = za.id_zaposlenik

Pridružuje tablicu zaduzenje_alata s tablicom zaposlenik na temelju podudaranja stupaca id iz tablice zaposlenik i id_zaposlenik iz tablice zaduzenje_alata. Ovo znači da se za svakog zaposlenika pridružuju svi zapisi o njegovim zaduženjima alata.

JOIN zaduzenje_alata AS za ON z.id = za.id_zaposlenik: Ovo spaja tablicu zaposlenik s tablicom zaduzenje_alata, povezujući zaposlenike s njihovim zaduženjima alata.

```
JOIN alat AS a ON za.id_alat = a.id
```

Pridružuje tablicu alat s tablicom zaduzenje_alata na temelju podudaranja stupaca id iz tablice alat i id_alat iz tablice zaduzenje_alata. Ovo znači da se za svako zaduženje pridružuju podaci o alatu koji je zadužen.

JOIN alat AS a ON za.id_alat = a.id: Ovo spaja tablicu zaduzenje_alata s tablicom alat, povezujući zaduženja s podacima o alatima.

```
GROUP BY z.ime, z.prezime, a.naziv_alata
```

Grupira rezultate prema imenu i prezimenu zaposlenika te nazivu alata. Ovo omogućuje agregaciju podataka (korištenje funkcije COUNT) za svaki alat po zaposleniku zasebno.

GROUP BY z.ime, z.prezime, a.naziv_alata: Ovo grupira rezultate tako da svaki jedinstveni par ime, prezime, i naziv_alata ima svoj vlastiti redak u rezultatu upita.

```
ORDER BY z.prezime, z.ime, broj_zaduzenja DESC
```

Sortira rezultate prema prezimenu i imenu zaposlenika, a zatim prema broju zaduženja u silaznom redoslijedu. To znači da će rezultati biti prikazani abecednim redoslijedom po prezimenu i imenu zaposlenika, a zatim sortirani prema broju zaduženja alata u silaznom redoslijedu.

ORDER BY z.prezime, z.ime, broj_zaduzenja DESC: Ovo sortira rezultate prema prezimenu i imenu zaposlenika, a zatim prema broju zaduženja u silaznom redoslijedu.

Rerezultat:

ime	prezime	naziv_alata	broj_zaduzenja
Junuz	Bajrami	Bosch GSB 2100 RE	1
Bruno	Galant	Hugong MIG 250	3
Karlo	Juranić	Lincoln Electric V205T	2
Matteo	Marić	Hugong Extremig 180	2
Matteo	Marić	Hugong INVERMIG 500 III IGBT	2
Marko	Milohanović	Lincoln Electric V205T	1
Martin	Modrušan	Hugong INVERMIG 500 III IGBT	2
Martin	Modrušan	Hugong MIG 250	1
Petar	Šegon	Bosch GSB 18V-150 C	1

7.16. UPIT 16

Prikaz broja zaposlenika po projektu

Trazeni rezultat:

naziv_projekta, broj_zaposlenika

Kod:

```
SELECT p.naziv_projekta, COUNT(DISTINCT us.id_zaposlenik) AS broj_zaposlenika
FROM projekt AS p
JOIN segment AS s ON p.id = s.id_projekt
JOIN proizvodnja AS pr ON s.id = pr.id_segment
JOIN utrosak_sati AS us ON pr.id = us.id_proizvodnja
GROUP BY p.naziv_projekta
ORDER BY broj_zaposlenika DESC;
```

Opis:

Ovaj upit je koristan za dobivanje pregleda o tome koliko je zaposlenika radilo na svakom projektu. Može se koristiti za identifikaciju projekata koji zahtijevaju najviše radne snage i za planiranje resursa. Na taj način, menadžeri mogu pratiti angažman zaposlenika na različitim projektima i optimizirati dodjelu radne snage.

Opis koda:

```
SELECT p.naziv_projekta, COUNT(DISTINCT us.id_zaposlenik) AS broj_zaposlenika
```

Ovaj dio upita odabire dva podatka:

p.naziv_projekta: Naziv projekta iz tablice projekt.

COUNT(DISTINCT us.id_zaposlenik) AS broj_zaposlenika: Izračunava broj jedinstvenih zaposlenika koji su radili na svakom projektu. Funkcija COUNT(DISTINCT ...) broji broj jedinstvenih zaposlenika (us.id_zaposlenik) za svaki projekt.

FROM projekt AS p

Definira tablicu projekt kao osnovnu tablicu za upit i dodjeljuje joj alias p za lakše referenciranje.
JOIN segment AS s ON p.id = s.id_projekt

Pridružuje tablicu segment s tablicom projekt na temelju podudaranja stupaca id_projekt iz tablice segment i id iz tablice projekt. Ovo znači da se za svaki projekt pridružuju svi njegovi segmenti.

JOIN segment AS s ON p.id = s.id_projekt: Ovo spaja tablicu projekt s tablicom segment, povezujući projekte s njihovim segmentima.

JOIN proizvodnja AS pr ON s.id = pr.id_segment

Pridružuje tablicu proizvodnja s tablicom segment na temelju podudaranja stupaca id_segment iz tablice proizvodnja i id iz tablice segment. Ovo znači da se za svaki segment pridružuju sve faze proizvodnje koje se odnose na taj segment.

JOIN proizvodnja AS pr ON s.id = pr.id_segment: Ovo spaja tablicu segment s tablicom proizvodnja, povezujući segmente s njihovim proizvodnim fazama.

JOIN utrosak_sati AS us ON pr.id = us.id_proizvodnja

Pridružuje tablicu utrosak_sati s tablicom proizvodnja na temelju podudaranja stupaca id_proizvodnja iz tablice utrosak_sati i id iz tablice proizvodnja. Ovo znači da se za svaku fazu proizvodnje pridružuju svi zapisi o utrošenim satima rada zaposlenika.

JOIN utrosak_sati AS us ON pr.id = us.id_proizvodnja: Ovo spaja tablicu proizvodnja s tablicom utrosak_sati, povezujući proizvodne faze s utrošenim satima rada zaposlenika.

GROUP BY p.naziv_projekta

Grupira rezultate prema nazivu projekta (p.naziv_projekta). Ovo omogućuje agregaciju podataka (korištenje funkcije COUNT) za svaki projekt zasebno.

GROUP BY p.naziv_projekta: Ovo grupira rezultate tako da svaki projekt ima svoj vlastiti redak u rezultatu upita.

ORDER BY broj_zaposlenika DESC

Sortira rezultate prema broju zaposlenika u silaznom redoslijedu (DESC). To znači da će projekti s najvećim brojem zaposlenika biti prikazani prvi.

ORDER BY broj_zaposlenika DESC: Ovo sortira rezultate prema broju zaposlenika u silaznom redoslijedu.

Rezultat:

naziv_projekta	broj_zaposlenika
Silos	12
Inox bubanj	6
Plodine - Varaždinske Toplice	6
Pregrada skladišta	6
Utezi	3
Torre Tension	2

8. ZAKLJUČAK

Smatramo da smo napravili solidnu bazu podataka za temu 'Organizacija u proizvodnji metalnih konstrukcija'. Naravno, svi smo svjesni da ima još mjesta za napredak. Ova tema je veoma opširna i još uvijek postoji mnogo situacija koje smo zanemarili u ovom projektu. Isto tako smatramo da smo ispunili sve zahtjeve prema pravilima za izradu projekta. Za izradu ovog projekta koristili smo:

- za komunikaciju: Discord , Whatsapp , Zoom
- za pisanje SQL skripte: MySQL Workbench
- za kreiranje ER dijagrama: Lucidchart
- za izradu dokumentacije: GitHub ,MS Office Word