

Sveučilište Jurja Dobrile u Puli  
Zagrebačka 30, 52100 Pula, Hrvatska

## "Sustav Za Upravljanje Skladištem"

Dokumentacija projekta iz kolegija Baze Podataka 1

**Nositelj kolegija:** doc. dr. sc. Goran Oreški

**Asistent:** Romeo Šajina mag. inf.

1. godina prijediplomskog sveučilišnog studija informatike

**Riješenje projekta za temu tima "Tim-10"**

Akademska godina 2023./2024.



Fakultet informatike u Puli

**Izradio:** Igor Tadić

**JMBAG:** 0152210311

**Datum:** 31. svibnja 2024. godine

# Sadržaj

- [1. Uvod](#)
- [2. Opis poslovnog procesa](#)
- [3. Entity Relationship \(ER\) Diagram](#)
- [3.1. Veze između entiteta, te njihov opis](#)
- [4. Sheme relacijskog modela](#)
- [5. EER dijagram \(MySQL Workbench\)](#)
- [6. Tablice](#)
  - [6.1. Tablica 'warehouseinventory'](#)
    - [Opis Atributa](#)
    - [Ključevi i Ograničenja](#)
    - [Relacije i Domene](#)
  - [6.2. Tablica 'warehouse'](#)
    - [Opis Atributa](#)
    - [Ključevi i Ograničenja](#)
    - [Relacije i Domene](#)
  - [6.3. Tablica 'article'](#)
    - [Opis Atributa](#)
    - [Ključevi i Ograničenja](#)
    - [Relacije i Domene](#)
  - [6.4. Tablica 'cartitem'](#)
    - [Opis Atributa](#)
    - [Ključevi i Ograničenja](#)
    - [Relacije i Domene](#)
  - [6.5. Tablica 'product'](#)
    - [Opis Atributa](#)
    - [Ključevi i Ograničenja](#)
    - [Relacije i Domene](#)
  - [6.6. Tablica 'category'](#)
    - [Opis Atributa](#)
    - [Pregled Hijerarhije](#)
    - [Ključevi i Ograničenja](#)
    - [Relacije i Domene](#)
  - [6.7. Tablica 'shippingaddress'](#)
    - [Opis Atributa](#)
    - [Indeksi i Ograničenja](#)
    - [Primjer database indeksa u praksi](#)

- [Relacije i Domene](#)
- [6.8. Tablica 'customer'](#)
  - [Opis Atributa](#)
  - [Ključevi i Ograničenja](#)
  - [Relacije i Domene](#)
- [6.9. Tablica 'producttrait'](#)
  - [Opis Atributa](#)
  - [Ključevi i Ograničenja](#)
  - [Relacije i Domene](#)
- [6.10. Tablica 'producttoproducttrait'](#)
  - [Opis Atributa](#)
  - [Ključevi i Ograničenja](#)
  - [Relacije i Domene](#)
- [6.11. Tablica 'customerorder'](#)
  - [Opis Atributa](#)
  - [Procedura ili trigger](#)
  - [Ključevi i Ograničenja](#)
  - [Relacije i Domene](#)
- [6.12. Tablica 'cart'](#)
  - [Ključevi i Ograničenja](#)
  - [Relacije i Domene](#)
- [7. Upiti](#)
  - [7.1. Upit 1 - Rezultat Upita](#)
  - [7.2. Upit 2 - Rezultat Upita](#)
  - [7.3. Upit 3 - Rezultat Upita](#)
  - [7.4. Upit 4 - Rezultat Upita](#)
  - [7.5. Upit 5 - Rezultat Upita](#)
  - [7.6. Upit 6 - Rezultat Upita](#)
  - [7.7. Upit 7 - Rezultat Upita](#)
  - [7.8. Upit 8 - Rezultat Upita](#)
  - [7.9. Upit 9 - Rezultat Upita](#)
  - [7.10. Upit 10 - Rezultat Upita](#)
  - [7.11. Upit 11 - Rezultat Upita](#)
  - [7.12. Upit 12 - Rezultat Upita](#)
  - [7.13. Upit 13 - Rezultat Upita](#)
  - [7.14. Upit 14 - Rezultat Upita](#)
  - [7.15. Upit 15](#)
  - [8. Korištene metode i tehnologije](#)

- [9. Zaključak](#)

# 1. Uvod

Ovaj projekt bio je pokušaj modeliranja sustava skladišta. Slični sustavi koriste se u velikom spektru poslovanja, te je zbog kontinuiranog unaprijeđenja dosegao razinu standarda koji se od takvog sistema očekuju. Činjenica je da je razlika između uspješnog poslovanja i ostalih, puno puta proporcionalna efikasnosti poslovnih sustava koje to poslovanje ima na raspolaganje. Sustav skladištenja opisuje realne radnje poslovanja te kao takav ih mora moći što točnije opisati, a da pri tome ne gubi na efikasnosti, fleksibilnosti, integritetu podataka, i skalabilnosti sustava u budućnosti.

Pokušao sam izraditi model sustava mreže skladišta imajući na umu da je izgradnja takvog poslovnog sustava jedan niz neprekidnih nadodavanja funkcionalnosti i iteracija te rast zahtjeva od strane poduzeća. Prema tome, bazu podataka sam dizajnirao koristeći strogo definirane entitete sa jasnim atributima i zadaćama sa nadom da bih na taj način održao modularnost sustava i sveo nepotrebnu redundanciju podataka na minimum.

## 2. Opis poslovnog procesa

Kao početak poslovnog procesa mogu se uzeti nabava robe i distribucija po skladištima, ili izrada narudžbe od strane potrošača, tj. klijenta, te izručivanje narudžbe. S obzirom da nabava robe i njena distribucija zahtjeva donošenje odluka utemeljenih obradom i analizom podataka dobivenih tokom samog rada sistema, npr. predviđanje vrste i količine robe za svako skladište analizom prijašnjih narudžbi, uzimajući pri tome u obzir da u idealnom slučaju narudžba treba biti isporučena iz geografski najbližeg skladišta.

Prema tome, za početak procesa uzimamo registraciju korisnika na nekome portalu. Pri registraciji korisnik može, ali ne mora unjeti svoju adresu za dostavu, te se adresa može naknadno dodati. Korisnik odabire artikle koje želi kupiti, te ih stavlja u košaru. Kad je korisnik izabrao artikle i želi finalizirati narudžbu, ista će se finalizirati samo ako postoji adresa za dostavu koja je asocirana sa korisnikom. Ako ona je postoji, korisnik upisuje adresu za dostavu, te dovršava narudžu.

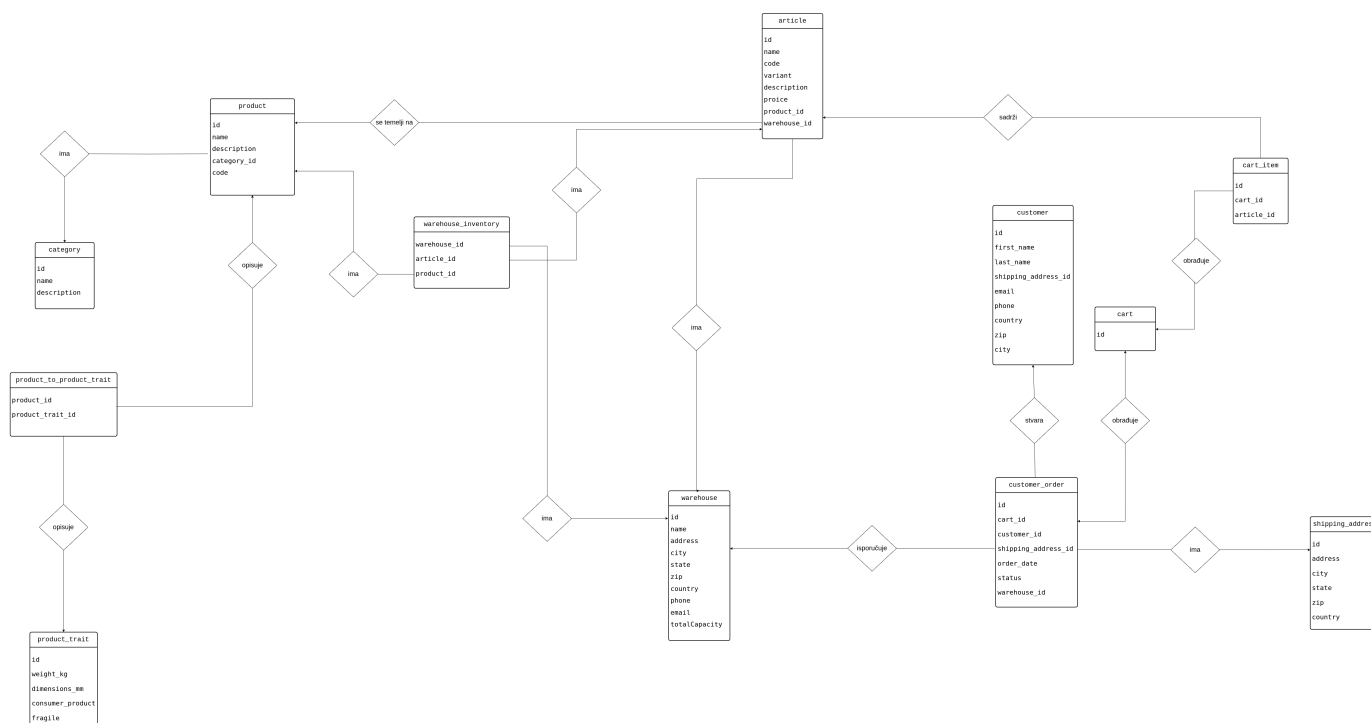
U planu daljem planu mi je razviti sustav za odlučivanje, koji bi trebao samostalno odlučiti iz kojeg skladišta izvršiti narudžu. Tu funkcionalnost za sada simuliram programatički, kodom koji nasumično bira iz kojeg skladišta će se isporučiti narudžba. Nakon što se stvori nova narudžba, status narudžbe dobiva vrijednost "processing".

```
1  create trigger set_status_and_warehouse_id
2  before insert on customer_order
3  for each row
4  begin
5  set new.status = 'processing';
6  set new.warehouse_id = (select id from warehouse order by rand() limit 1);
7  end;
```

S time je zaključena jedna narudžba. Svojstva entiteta i veze između njih će detaljno biti objašnjene u daljnim poglavljima.

### 3. Entity Relationship (ER) Diagram

Sljedeći ER dijagram prikazuje relacijsku strukturu entiteta, njihove atribute, te veze sa ostalim entitetima. Kardinalnost između entiteta je prikazana linijom koje spajaju vezane entitete. Smjer strelice očituje jeli entitet u povezan sa jednim ili više drugih entitea u promatranoj vezi.



### 3.1. Veze između entiteta, te njihov opis

Vlasnik veze	Zavisni entitet	Kardinalnost	Objašnjenje
cart_item	cart	Many-To-One	Do finalizacije naruđbe jedan te isti predmet kupovine može se istodobno nalaziti u više elektroničkih košara.
cart_item	article	Many-To-One	Pošto <b>cart_item</b> može biti u isto vrijeme u više košara, a <b>cart_item</b> je abstrakcija <b>article</b> entiteta, unutar košare znaci da se jedan <b>article</b> može unutar <b>cart_item</b> tablice ponavljati u više zapisa, dok će se u <b>article</b> pojaviti samo jednom.
warehouse_inventory	warehouse	Many-To-One	warehouse_inventory ispisuje sve <b>article</b> objekte koji se nalaze u svim skladištima. Svaki zapis u <b>warehouse_inventory</b> tablicu ima polje "warehouse_id". prema tome, broj ponavljanja warehouse_id polja odgovarat će broju artikala u pojedinom skladištu, dok <b>warehouse</b> tablica prikazuje listu svih unikatnih <b>warehouse</b> objekata koji postoje.
warehouse_inventory	article	Many-To-One	Pošto se je moguće da će korisnik odlučiti vratiti dostavljeni artikl, a <b>warehouse_inventory</b> prati prošlost svih uvezenih artikla, u tome slučaju će se jedan te isti artikl pojaviti u više zapisa unutar <b>warehouse_inventory</b> tablice, dok se tablica <b>article</b> ne mjenja.
warehouse_inventory	product	Many-To-One	<b>product</b> je generalan opis grupe proizvoda, te se kao takav može ponavljati. npr. jedno ili više skladišta može imati n broj proizvoda "mobitel". Dok će u tablici <b>product</b> postojati samo jedan zapis koji predstavlja taj proizvod "mobitel".
article	warehouse	Many-To-One	Svaki artikl može biti unutar samo jednog skladišta, dok svako skladište može sadržavati nula ili više artikala.
article	product	Many-To-One	Više različitih artikala može biti asocirano sa jednim "proizvodom" - npr. dva 'računala' <i>računalo-1</i> i <i>računalo-2</i> su dva unikatna objekta, iako oba spadaju u proizvod "računalo".



Vlasnik veze	Zavisni entitet	Kardinalnost	Objašnjenje
product	category	Many-To-One	Jedan ili više različitih proizvoda mogu pripadati istoj kategoriji proizvoda. - npr. jabuka i dunja pripadaju u kategoriju "voće".
customer_order	shipping_address	Many-To-One	Jedna narudžba će uvijek imati samo jednu adresu za dostavu, dok se jedna adresa za dostavu može biti vezana za više narudžbi.
customer_order	customer	Many-To-One	Jedna narudžba je vezana za jednog korisnika, dok jedan korisnik može obaviti više narudžbi.
customer_order	cart	One-To-One	Košarica je uvijek generirana samo za jednu narudžbu, te se nakon svakog <b>customer_order</b> zapisa stvara nova košara, a stara se ne koristi.
customer_order	warehouse	Many-To-One	Jedna narudžba će se isporučiti iz jednog skladišta. Dok jedno skladište može isporučiti više narudžbi.
product_to_product_trait	product	Many-To-One	<b>product</b> može imati više svojstava ( <b>product_trait</b> ), ali kombinacija ključeva <b>product_id</b> i <b>product_trait_id</b> će uvijek biti unikatan te je prema tome jedan proizvod uvijek asociran sa samo jednim <b>product_to_product_trait</b> objektom.
product_to_product_trait	product_trait	Many-To-One	<b>product_to_product_trait</b> veže jedan ili više <b>product_trait_id</b> sa <b>product</b> entitetom, no više <b>product</b> entiteta ne može dijeliti isti <b>product_to_product_trait.id</b> koji je unikatan za svaki proizvod.

## 4. Sheme relacijskog modela

**cart\_item**(id, cart\_id, article\_id)

**warehouse\_inventory**(warehouse\_id, article\_id, product\_id)

**article**(id, name, code, variant, description, price, product\_id, warehouse\_id)

**product**(id, name, description, category\_id, code)

**customer\_order**(id, cart\_id, customer\_id, shipping\_address\_id, order\_date, status, warehouse\_id)

**category**(id, name, description)

**shipping\_address**(id, address, city, state, zip, country)

**warehouse**(id, name, address, city, state, zip, country, phone, email, totalCapacity)

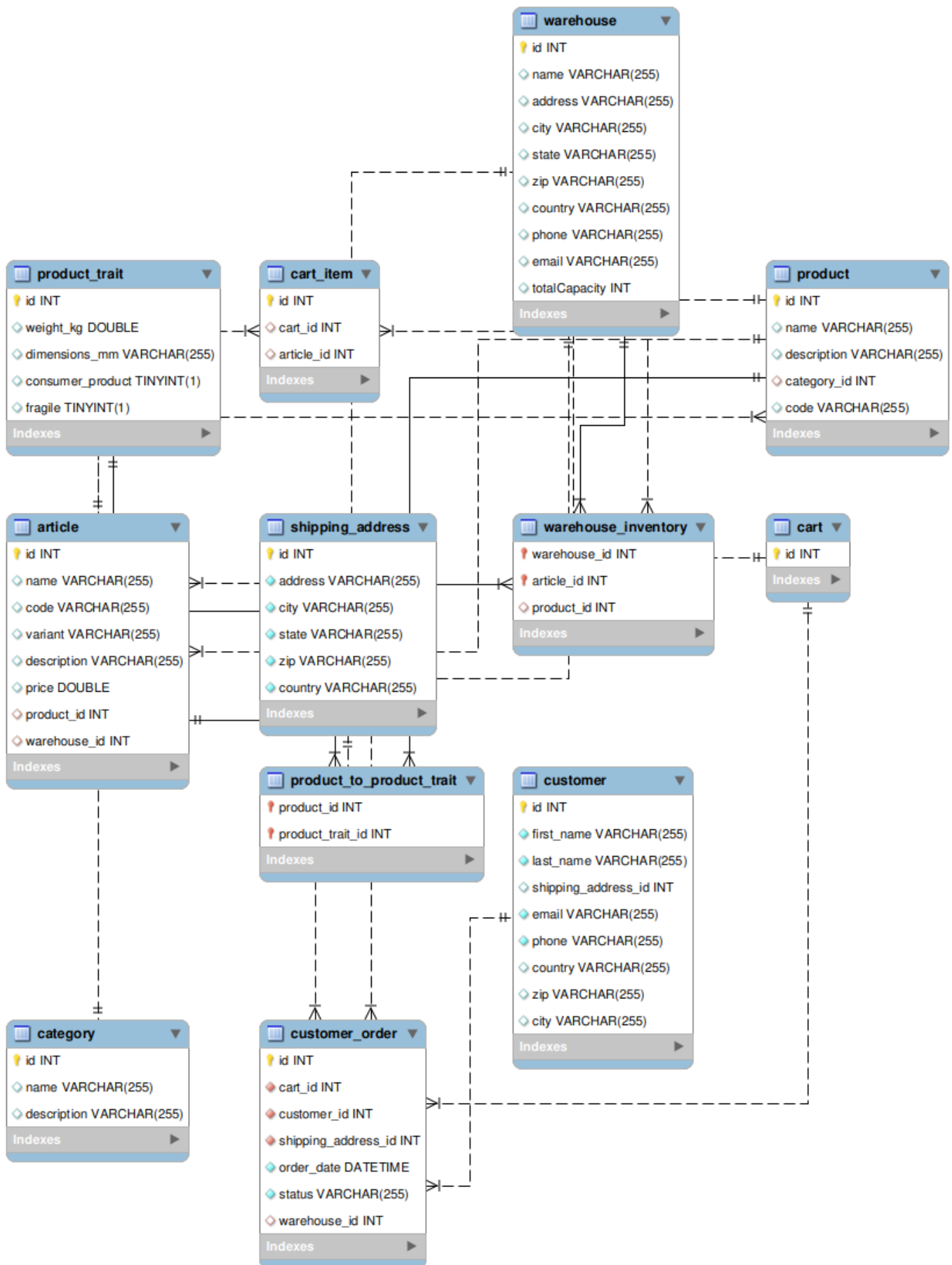
**cart**(id)

**product\_to\_product\_trait**(product\_id, product\_trait)

**product\_trait**(id, weight\_kg, dimensions\_mm, consumer\_product, fragile)

**customer**(id, first\_name, last\_name, shipping\_address\_id, email, phone, country, zip, city)

## 5. EER dijagram (MySQL Workbench)



## 6. Tablice

## 6.1. Tablica 'warehouse\_inventory'

Tablica **'warehouse\_inventory'** ključna je komponenta sustava za upravljanje skladištem, te služi kao "most" između tablica 'warehouse', 'article' i 'product'. Ona omogućuje da pratimo koji su artikli, odnosno proizvodi pohranjeni u kojim skladištima. To je bitno kod upravljanja razinama zaliha i osiguranja dostupnosti zaliha. Znajući koji se artikli nalaze u kojim skladištima, možemo optimizirati distribuciju inventara, npr. može se osigurati da su popularni proizvodi dobro opskrbljeni u svim skladištima ili možemo skladištiti određene proizvode bliže mjestima gdje se često prodaju. Isto tako, tablica **warehouse\_inventory** je korisna kod analize zaliha proizvoda, tj. može služiti da bih identificirali kojih proizvoda često nema na zalihama, koja skladišta imaju višak zaliha ili kako razine zaliha variraju tijekom vremena. Analizirajući razine zaliha, može se odrediti kada ponovno naručiti proizvode, i koliko naručiti. Upotreba primarnih i stranih ključeva u tablici 'warehouse\_inventory' osigurava integritet podataka. Strani ključevi uspostavljaju odnose s tablicama 'skladište', 'artikl' i 'proizvod', osiguravajući da se samo ispravni podaci mogu unijeti u tablicu 'skladište\_zaliha'. Svi atributi su integer podatkovnog tipa. Sadrži attribute: **warehouse\_id**, **article\_id** i **product\_id**. Tablica sadržava kompozitni **PRIMARY KEY** koji je kombinacija **warehouse\_id** i **article\_id**. To znači da je svaki zapis u warehouse\_inventory tablici jedinstveno identificiran kombinacijom **warehouse\_id** i **article\_id** atributa. Atribut **warehouse\_id** je **FOREIGN KEY** (strani ključ) koji referira **id** iz **warehouse** tablici. **warehouse\_id** omogućava many-to-one vezu između **warehouse\_inventory** i **warehouse** tablica. Atribut **article\_id** je **FOREIGN KEY** koji referira **id** iz **article** tablici, te omogućuje many-to-one vezu između **warehouse\_inventory** i **article** tablica. Atribut **product\_id** je **FOREIGN KEY** koji referira **id** iz tablice **product** te omogućuje many-to-one vezu između **warehouse\_inventory** i **product** tablica.

### Opis Atributa

1. **warehouse\_id** ( int , NOT NULL)
  - **Opis:** Identifikator skladišta.
  - **Svrha:** Povezuje artikle i proizvode s određenim skladištem. Ovo polje je dio složenog primarnog ključa.
2. **article\_id** ( int , NOT NULL)
  - **Opis:** Identifikator artikla.
  - **Svrha:** Povezuje artikle s određenim skladištem. Ovo polje je dio složenog primarnog ključa.
3. **product\_id** ( int , DEFAULT NULL)
  - **Opis:** Identifikator proizvoda.
  - **Svrha:** Opcionalno polje koje povezuje artikle s odgovarajućim proizvodom, omogućujući dodatnu razinu praćenja inventara.

### Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - Kombinacija **warehouse\_id** i **article\_id** osigurava da svaki par skladišta i artikla bude jedinstven unutar tablice.

- **Strani ključevi (FOREIGN KEYS):**

- `warehouse_id`:
  - **Opis:** Referencira `id` iz tablice `warehouse`.
  - **Svrha:** Osigurava da artikli povezani s ovim skladištem postoje u tablici `warehouse`.
- `article_id`:
  - **Opis:** Referencira `id` iz tablice `article`.
  - **Svrha:** Osigurava da artikli u skladištu postoje u tablici `article`.
- `product_id`:
  - **Opis:** Referencira `id` iz tablice `product`.
  - **Svrha:** Osigurava da proizvodi povezani s artiklima postoje u tablici `product`.

## Relacije i Domene

- **Relacije:**

- **Skladište ( warehouse ):**
  - Svako skladište može imati više artikala ( 1:M relacija s `warehouse_inventory` ).
- **Artikl ( article ):**
  - Svaki artikl može biti pohranjen u više skladišta ( M:1 relacija s `warehouse_inventory` ).
- **Proizvod ( product ):**
  - Svaki proizvod može biti povezan s više artikala u različitim skladištima ( M:1 relacija s `warehouse_inventory` ).

- **Domene:**

- `warehouse_id` : Mora biti valjani identifikator skladišta, kako je definirano u tablici `warehouse`.
- `article_id` : Mora biti valjani identifikator artikla, kako je definirano u tablici `article`.
- `product_id` : Mora biti valjani identifikator proizvoda, kako je definirano u tablici `product`.

```
1 CREATE TABLE `warehouse_inventory` (  
2   `warehouse_id` int NOT NULL,  
3   `article_id` int NOT NULL,  
4   `product_id` int DEFAULT NULL,  
5   PRIMARY KEY (`warehouse_id`,`article_id`),  
6   KEY `article_id` (`article_id`),  
7   KEY `product_id` (`product_id`),  
8   CONSTRAINT `warehouse_inventory_ibfk_1` FOREIGN KEY (`warehouse_id`)   
REFERENCES `warehouse` (`id`),  
9   CONSTRAINT `warehouse_inventory_ibfk_2` FOREIGN KEY (`article_id`) REFERENCES   
`article` (`id`),  
10  CONSTRAINT `warehouse_inventory_ibfk_3` FOREIGN KEY (`product_id`) REFERENCES   
`product` (`id`)  
11 );
```

## 6.2. Tablica 'warehouse'

Tablica '**warehouse**' evidentira sva skladišta koja se nalaze unutar ovoga sustava. Pomoću tablice **warehouse** možemo brzo doći do osnovnih informacija o skladištima unutar sistema., npr. lokaciju skladišta. informacije potrebne za kontaktirati skladište, i ukupan kapacitet skladišta. Tablica sadrži attribute **id**, **name**, **address**, **city**, **state**, **zip**, **country**, **phone**, **email**, **totalCapacity**. Atribut **id** je integer podatkovnog tipa, koji jedinstveno brojevno definira svako pojedino skladište. Atribut **id** je ujedno i PRIMARY KEY, odnosno primarni ključ ove tablice. Vrijednost **id** se unosi inkrementalno i automatski, pri svakom zapisu u tablicu. Unošenje vrijednosti za ostale attribute tablice nije neophodna. Tablica **warehouse** nema niti jedan FOREIGN KEY constraint, ali je ključna tablica potrebna za izradu ostalih tablica kao što su: **article**, **customer\_order**, **warehouse\_inventory**. Ovom strogom vezom potiče se unos ispravnih i smislenih podataka širem cijele baze podataka.

```
1 CREATE TABLE `warehouse` (  
2   `id` int NOT NULL AUTO_INCREMENT,  
3   `name` varchar(255) DEFAULT NULL,  
4   `address` varchar(255) DEFAULT NULL,  
5   `city` varchar(255) DEFAULT NULL,  
6   `state` varchar(255) DEFAULT NULL,  
7   `zip` varchar(255) DEFAULT NULL,  
8   `country` varchar(255) DEFAULT NULL,  
9   `phone` varchar(255) DEFAULT NULL,  
10  `email` varchar(255) DEFAULT NULL,  
11  `totalCapacity` int DEFAULT NULL,  
12  PRIMARY KEY (`id`)  
13 );
```

## Opis Atributa

1. **id** ( int , NOT NULL, AUTO\_INCREMENT)
  - **Opis:** Jedinstveni identifikator za svako skladište. Ovaj primarni ključ osigurava da se svako skladište može jedinstveno referencirati unutar sustava.
  - **Svrha:** Služi kao primarni ključ za tablicu, pružajući jedinstveni identifikator za svaki zapis.
2. **name** ( varchar(255) , DEFAULT NULL)
  - **Opis:** Naziv skladišta.
  - **Svrha:** Pruža korisnički prepoznatljiv naziv za svako skladište.
3. **address** ( varchar(255) , DEFAULT NULL)
  - **Opis:** Adresa skladišta.
  - **Svrha:** Pruža fizičku lokaciju skladišta.
4. **city** ( varchar(255) , DEFAULT NULL)
  - **Opis:** Grad u kojem se skladište nalazi.
  - **Svrha:** Pruža dodatne informacije o lokaciji skladišta.
5. **state** ( varchar(255) , DEFAULT NULL)
  - **Opis:** Država ili pokrajina u kojoj se skladište nalazi.
  - **Svrha:** Pruža dodatne informacije o administrativnoj lokaciji skladišta.
6. **zip** ( varchar(255) , DEFAULT NULL)

- **Opis:** Poštanski broj skladišta.
- **Svrha:** Pruža informacije potrebne za preciznu identifikaciju lokacije skladišta.

7. **country** ( `varchar(255)` , DEFAULT NULL)

- **Opis:** Država u kojoj se skladište nalazi.
- **Svrha:** Pruža informacije o zemlji u kojoj se skladište nalazi.

8. **phone** ( `varchar(255)` , DEFAULT NULL)

- **Opis:** Telefonski broj skladišta.
- **Svrha:** Pruža kontakt informacije za skladište.

9. **email** ( `varchar(255)` , DEFAULT NULL)

- **Opis:** Email adresa skladišta.
- **Svrha:** Pruža dodatne kontakt informacije za skladište.

10. **totalCapacity** ( `int` , DEFAULT NULL)

- **Opis:** Ukupni kapacitet skladišta.
- **Svrha:** Pruža informacije o maksimalnom kapacitetu skladišta u smislu skladištenja artikala.

## Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - `id` : Osigurava jedinstvenost svakog zapisa u tablici `warehouse` .

## Relacije i Domene

- **Relacije:**
  - **Artikl ( `article` ):**
    - Svako skladište može pohranjivati više artikala, što čini relaciju 1 s tablicom `article` (pomoću `warehouse_id` ).
  - **Narudžba ( `customer_order` ):**
    - Svaka narudžba može biti povezana s jednim skladištem, što čini relaciju 1 s tablicom `customer_order` (pomoću `warehouse_id` ).
  - **Inventar Skladišta ( `warehouse_inventory` ):**
    - Svako skladište može imati više artikala u svom inventaru, što čini relaciju 1 s tablicom `warehouse_inventory` (pomoću `warehouse_id` ).
- **Domene:**
  - `id` : Mora biti jedinstveni identifikator za svako skladište.
  - `name` : Tekstualno polje koje može sadržavati naziv skladišta.
  - `address` : Tekstualno polje koje može sadržavati adresu skladišta.
  - `city` : Tekstualno polje koje može sadržavati grad skladišta.
  - `state` : Tekstualno polje koje može sadržavati državu ili regiju skladišta.
  - `zip` : Tekstualno polje koje može sadržavati poštanski broj skladišta.
  - `country` : Tekstualno polje koje može sadržavati zemlju skladišta.

- **phone** : Tekstualno polje koje može sadržavati kontakt telefon skladišta.
- **email** : Tekstualno polje koje može sadržavati kontakt email skladišta.
- **totalCapacity** : Numeričko polje koje može sadržavati ukupan kapacitet skladišta.

## 6.3. Tablica 'article'

Tablica '**article**' pohranjuje informacije o pojedinim artiklima kojima se upravlja unutar skladišta. Svaki redak u tablici predstavlja jedinstveni artikal, koji može biti bilo koji uskladišteni predmet, od određenog proizvoda do specifične varijante tog proizvoda. Atribut **id** je jedinstveni identifikator za svaki artikal. Ovaj primarni ključ osigurava da se svaki artikal može jedinstveno referencirati unutar sustava.

Atribut **name** predstavlja naziv artikla. To može biti uobičajeni naziv proizvoda ili specifična oznaka koja identificira artkl. Ovaj atribut radi opisivanja artikla korištenjem identifikatora koji je korisnicima lakše zapamtiti.

Atribut **code** jest jedinstveni kod dodijeljen artiklu. Ovaj kod se može koristiti za brzo pretraživanje i identifikaciju, te pruža alternativni identifikator koji se može koristiti u sustavima za bar kodove ili unutarnje praćenje.

Atribut **description** predstavlja kratki opis artikla. Ovaj atribut pruža dodatne informacije o proizvodu, poput njegovih karakteristika i specifikacija.

Atribut **price** predstavlja cijenu artikla. Pošto artikal može predstavljati varijante nekog proizvoda, potrebno je cijenu precizirati na razini pojedine varijante, odnosno artikla.

Atribut **product\_id** je strani ključ koji povezuje s tablicom 'product'. Ovo polje povezuje artikal s širom kategorijom proizvoda kojemu pripada, te omogućuje hijerarhijsko upravljanje podacima.

Atribut **warehouse\_id** je strani ključ koji povezuje s tablicom 'warehouse'. Ovo polje označava u kojem se skladištu artikal nalazi, te na taj način omogućuje praćenje lokacije artikla unutar sustava, podržavajući učinkovito upravljanje zalihama i logistikom.

## Opis Atributa

### 1. **id** ( `int` , NOT NULL, AUTO\_INCREMENT)

- **Opis:** Jedinstveni identifikator za svaki artikal. Ovaj primarni ključ osigurava da se svaki artikal može jedinstveno referencirati unutar sustava.
- **Svrha:** Služi kao primarni ključ za tablicu, pružajući jedinstveni identifikator za svaki zapis.

### 2. **name** ( `varchar(255)` , DEFAULT NULL)

- **Opis:** Naziv artikla. To može biti uobičajeni naziv proizvoda ili specifična oznaka koja identificira artikal.
- **Svrha:** Pomaže korisnicima da lako prepoznaju i referenciraju artikal.

### 3. **code** ( `varchar(255)` , DEFAULT NULL)

- **Opis:** Jedinstveni kod dodijeljen artiklu. Ovaj kod može se koristiti za brzo pretraživanje i identifikaciju.



- **Svrha:** Pruža alternativni identifikator koji se može koristiti u sustavima za bar kodove ili unutarnje praćenje.
4. **variant** ( `varchar(255)` , `DEFAULT NULL` )
- **Opis:** Specifikacija varijante proizvoda. Na primjer, može označavati različite veličine, boje ili modele istog proizvoda.
  - **Svrha:** Omogućuje razlikovanje između više verzija proizvoda, osiguravajući precizno upravljanje zalihama.
5. **description** ( `varchar(255)` , `DEFAULT NULL` )
- **Opis:** Kratak opis artikla. Ovo polje pruža dodatne informacije o proizvodu, poput njegovih karakteristika ili specifikacija.
  - **Svrha:** Pomaže u razumijevanju karakteristika artikla i donošenju odluka u vezi s upravljanjem zalihama.
6. **price** ( `double` , `DEFAULT NULL` )
- **Opis:** Cijena artikla. To može biti nabavna cijena, maloprodajna cijena ili bilo koja druga relevantna informacija o cijeni.
  - **Svrha:** Koristi se za financijske izračune, strategije određivanja cijena i vrednovanje zaliha.
7. **product\_id** ( `int` , `DEFAULT NULL` )
- **Opis:** Strani ključ koji povezuje s tablicom `product` . Ovo polje povezuje artikal s širom kategorijom ili porodicom proizvoda.
  - **Svrha:** Uspostavlja vezu između artikla i njegovog proizvoda, omogućujući organizirano i hijerarhijsko upravljanje podacima.
8. **warehouse\_id** ( `int` , `DEFAULT NULL` )
- **Opis:** Strani ključ koji povezuje s tablicom `warehouse` . Ovo polje označava u kojem se skladištu artikal nalazi.
  - **Svrha:** Omogućuje praćenje lokacije artikla unutar skladišnog sustava, podržavajući učinkovito upravljanje zalihama i logistikom.

## Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - `id` : Osigurava jedinstvenost svakog zapisa u tablici `article` .
- **Vanjski ključevi (FOREIGN KEYS):**
  - `product_id` :
    - **Opis:** Povezuje `article` s tablicom `product` .
    - **Ograničenje:** `CONSTRAINT article_ibfk_1 FOREIGN KEY (product_id) REFERENCES product (id)` .
  - `warehouse_id` :
    - **Opis:** Povezuje `article` s tablicom `warehouse` .
    - **Ograničenje:** `CONSTRAINT article_ibfk_3 FOREIGN KEY (warehouse_id) REFERENCES warehouse (id)` .

- **Indeksi (KEYS):**
  - `product_id` : Indeks za ubrzavanje pretraživanja prema `product_id` .
  - `warehouse_id` : Indeks za ubrzavanje pretraživanja prema `warehouse_id` .

## Relacije i Domene

- **Relacije:**
  - **Proizvod ( product ):**
    - Svaki artikal je povezan s jednim proizvodom, što čini relaciju M:1 s tablicom `product` (pomoću `product_id` ).
  - **Skladište ( warehouse ):**
    - Svaki artikal je povezan s jednim skladištem, što čini relaciju M:1 s tablicom `warehouse` (pomoću `warehouse_id` ).
- **Domene:**
  - `id` : Mora biti jedinstveni identifikator za svaki artikal.
  - `name` : Tekstualno polje koje može sadržavati naziv artikla.
  - `code` : Tekstualno polje koje može sadržavati jedinstveni kod artikla.
  - `variant` : Tekstualno polje koje može sadržavati varijantu artikla.
  - `description` : Tekstualno polje koje može sadržavati opis artikla.
  - `price` : Numeričko polje koje može sadržavati cijenu artikla.
  - `product_id` : Numeričko polje koje može sadržavati referencu na `product` .
  - `warehouse_id` : Numeričko polje koje može sadržavati referencu na `warehouse` .

```

1 CREATE TABLE `article` (
2   `id` int NOT NULL AUTO_INCREMENT,
3   `name` varchar(255) DEFAULT NULL,
4   `code` varchar(255) DEFAULT NULL,
5   `variant` varchar(255) DEFAULT NULL,
6   `description` varchar(255) DEFAULT NULL,
7   `price` double DEFAULT NULL,
8   `product_id` int DEFAULT NULL,
9   `warehouse_id` int DEFAULT NULL,
10  PRIMARY KEY (`id`),
11  KEY `product_id` (`product_id`),
12  KEY `warehouse_id` (`warehouse_id`),
13  CONSTRAINT `article_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `product`
   (`id`),
14  CONSTRAINT `article_ibfk_3` FOREIGN KEY (`warehouse_id`) REFERENCES
   `warehouse` (`id`)
15 );

```

## 6.4. Tablica 'cart\_item'

Tablica '**cart\_item**' pohranjuje informacije o pojedinim artiklima koji su dodani u "košaricu". Svaki redak u tablici predstavlja jedinstveni artikl koji se nalazi unutar određene košarice.

Atribut **id** je jedinstveni identifikator za svaki artikl u "košarici". Ovaj primarni ključ osigurava da se svaki **cart\_item**, (artikl u košarici) može jedinstveno referencirati unutar sustava.

Atribut **cart\_id** je strani ključ koji s tablicom 'cart', i jednoznačno osigurava vezu artikla sa jednom i samo jednom košaricom kojoj taj artikl pripada.

Atribut **article\_id** je strani ključ koji povezuje s tablicom 'article'. Ovo polje povezuje artikl u košarici s konkretnim artiklom iz skladišta.

Povezivanjem s tablicama 'cart' i 'article,' tablica 'cart\_item' omogućuje praćenje i upravljanje artiklima unutar košarica, te osigurava točnost zaliha s obzirom na artikle koji se rezervirani u košaricama i olakšava proces ispunjavanja narudžbi na temelju artikala u košaricama.

Održavanjem ove tablice, osiguravamo točnost, učinkovitost i pouzdanost procesa zaduženih za upravljanjem sustavom za skladište.

## Opis Atributa

### 1. **id** ( `int` , NOT NULL, AUTO\_INCREMENT)

- **Opis:** Jedinstveni identifikator za svaki artikl u košarici. Ovaj primarni ključ osigurava da se svaki artikl može jedinstveno referencirati unutar sustava.
- **Svrha:** Služi kao primarni ključ za tablicu, pružajući jedinstveni identifikator za svaki zapis.

### 2. **cart\_id** ( `int` , DEFAULT NULL)

- **Opis:** Strani ključ koji povezuje s tablicom `cart` . Ovo polje povezuje artikl s određenom košaricom.
- **Svrha:** Uspostavlja vezu između artikla u košarici i same košarice, omogućujući organizirano praćenje stavki po košaricama.

### 3. **article\_id** ( `int` , DEFAULT NULL)

- **Opis:** Strani ključ koji povezuje s tablicom `article` . Ovo polje povezuje artikl u košarici s konkretnim artiklom iz skladišta.
- **Svrha:** Povezuje stavku u košarici s detaljnim informacijama o artiklu, kao što su naziv, opis i cijena.

## Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - `id` : Osigurava jedinstvenost svakog zapisa u tablici `cart_item` .
- **Vanjski ključevi (FOREIGN KEYS):**
  - `cart_id` :
    - **Opis:** Povezuje `cart_item` s tablicom `cart` .

- **Ograničenje:** `CONSTRAINT cart_item_ibfk_1 FOREIGN KEY (cart_id) REFERENCES cart (id)`.
- `article_id`:
  - **Opis:** Povezuje `cart_item` s tablicom `article`.
  - **Ograničenje:** `CONSTRAINT cart_item_ibfk_2 FOREIGN KEY (article_id) REFERENCES article (id)`.
- **Indeksi (KEYS):**
  - `cart_id`: Indeks za ubrzavanje pretraživanja prema `cart_id`.
  - `article_id`: Indeks za ubrzavanje pretraživanja prema `article_id`.

## Relacije i Domene

- **Relacije:**
  - **Košarica ( `cart` ):**
    - Svaka stavka u košarici je povezana s jednom košaricom, što čini relaciju M:1 s tablicom `cart` (pomoću `cart_id`).
  - **Artikal ( `article` ):**
    - Svaka stavka u košarici je povezana s jednim artiklom, što čini relaciju M:1 s tablicom `article` (pomoću `article_id`).
- **Domene:**
  - `id`: Mora biti jedinstveni identifikator za svaku stavku u košarici.
  - `cart_id`: Numeričko polje koje može sadržavati referencu na `cart`.
  - `article_id`: Numeričko polje koje može sadržavati referencu na `article`.

```

1 CREATE TABLE `cart_item` (
2   `id` int NOT NULL AUTO_INCREMENT,
3   `cart_id` int DEFAULT NULL,
4   `article_id` int DEFAULT NULL,
5   PRIMARY KEY (`id`),
6   KEY `cart_id` (`cart_id`),
7   KEY `article_id` (`article_id`),
8   CONSTRAINT `cart_item_ibfk_1` FOREIGN KEY (`cart_id`) REFERENCES `cart`
   (`id`),
9   CONSTRAINT `cart_item_ibfk_2` FOREIGN KEY (`article_id`) REFERENCES `article`
   (`id`)
10 );

```

## 6.5. Tablica 'product'

Tablica **'product'** pohranjuje osnovne informacije o svim proizvodima dostupnim u sustavu. Svaki redak u tablici predstavlja jedan proizvod s jedinstvenim identifikatorom, nazivom, opisom, kategorijom i kodom.

### Opis Atributa

1. **id** ( `int` , NOT NULL, AUTO\_INCREMENT)

- **Opis:** Jedinstveni identifikator za svaki proizvod. Ovaj primarni ključ osigurava da se svaki proizvod može jedinstveno referencirati unutar sustava.
- **Svrha:** Služi kao primarni ključ za tablicu, pružajući jedinstveni identifikator za svaki zapis.

2. **name** ( `varchar(255)` , DEFAULT NULL)

- **Opis:** Naziv proizvoda.
- **Svrha:** Pruža lako prepoznatljiv naziv za svaki proizvod, što olakšava identifikaciju i upravljanje proizvodima u sustavu.

3. **description** ( `varchar(255)` , DEFAULT NULL)

- **Opis:** Kratak opis proizvoda.
- **Svrha:** Pruža dodatne informacije o proizvodu, kao što su njegove karakteristike, namjena ili specifičnosti.

4. **category\_id** ( `int` , DEFAULT NULL)

- **Opis:** Strani ključ koji povezuje s tablicom `category` . Atribut 'category\_id' kategorizira proizvod unutar određene kategorije.
- **Svrha:** Omogućuje organizaciju proizvoda u kategorije, što olakšava njihovo pretraživanje i filtriranje unutar sustava.

5. **code** ( `varchar(255)` , DEFAULT NULL)

- **Opis:** Jedinstveni kod proizvoda.
- **Svrha:** Pruža jedinstveni identifikator ili šifru za proizvod, koja se može koristiti za brzu identifikaciju ili integraciju s vanjskim sustavima.

## Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - `id` : Osigurava jedinstvenost svakog zapisa u tablici `product` .
- **Vanjski ključ (FOREIGN KEY):**
  - `category_id` :
    - **Opis:** Povezuje `product` s tablicom `category` .
    - **Ograničenje:** `CONSTRAINT product_ibfk_1 FOREIGN KEY (category_id) REFERENCES category (id)` .
- **Indeks (KEY):**
  - `category_id` : Indeks za ubrzavanje pretraživanja prema `category_id` .

## Relacije i Domene

- **Relacije:**
  - **Kategorija ( category ):**
    - Svaki proizvod može biti povezan s jednom kategorijom, što čini relaciju M:1 s tablicom `category` (pomoću `category_id` ).
- **Domene:**

- `id` : Mora biti jedinstveni identifikator za svaki proizvod.
- `name` : Alfaničeričko polje koje može sadržavati naziv proizvoda.
- `description` : Alfaničeričko polje koje može sadržavati opis proizvoda.
- `category_id` : Numeričko polje koje može sadržavati referencu na `category` .
- `code` : Alfaničeričko polje koje može sadržavati jedinstveni kod proizvoda.

```

1 CREATE TABLE `product` (
2   `id` int NOT NULL AUTO_INCREMENT,
3   `name` varchar(255) DEFAULT NULL,
4   `description` varchar(255) DEFAULT NULL,
5   `category_id` int DEFAULT NULL,
6   `code` varchar(255) DEFAULT NULL,
7   PRIMARY KEY (`id`),
8   KEY `category_id` (`category_id`),
9   CONSTRAINT `product_ibfk_1` FOREIGN KEY (`category_id`) REFERENCES `category`
10  (`id`));

```

## 6.6. Tablica 'category'

Tablica '**category**' pohranjuje informacije o kategorijama u koje svrstavamo proizvode unutar ovog sustava. Svaki redak u tablici predstavlja jednu kategoriju s jedinstvenim identifikatorom, nazivom i opisom.

### Opis Atributa

#### 1. **id** ( int , NOT NULL, AUTO\_INCREMENT)

- **Opis:** Jedinstveni identifikator za svaku kategoriju. Ovaj primarni ključ osigurava da se svaka kategorija može jedinstveno referencirati unutar sustava.
- **Svrha:** Služi kao primarni ključ za tablicu, pružajući jedinstveni identifikator za svaki zapis.

#### 2. **name** ( varchar(255) , DEFAULT NULL)

- **Opis:** Naziv kategorije.
- **Svrha:** Sadrži jedinstvene identifikatore koji slijede određenu konvenciju za označavanje hijerarhije kategorija.

#### 3. **description** ( varchar(255) , DEFAULT NULL)

- **Opis:** Opis kategorije.
- **Svrha:** Pruža opisnu oznaku svake kategorije. Kategorije su organizirane hijerarhijski, gdje su identifikatori kategorija strukturirani na način da odražavaju odnose roditelj-dijete među kategorijama.

Htio bih dalje razjasniti razliku između polja *name* i *description* i koju ulogu igraju u efikasnoj kategorizaciji velike količine proizvoda kao što je to slučaj u sustavima za upravljanje skladištem. Kako bih to postigao, prikazati ću trenutne podatke iz tablice 'category' koji će služiti kao primjer.

name	description
A	Consumer Electronics
A2	Phones and Telecommunication
A3a	Nokia Mobile Phones
A3b	Xiaomi Mobile Phones
A3c	Huawei Mobile Phones

name	description
A3d	Apple Mobile Phones
A3e	Samsung Mobile Phones
A3f	LG Mobile Phones
A4	Computers and Tablets
A4a	Desktop Computers
A4b	Laptops
A4c	Tablets
A4d	2-in-1 Laptops
A4e	Gaming Laptops
B	Home Appliances
B2	Kitchen Appliances
B2a	Refrigerators
B2b	Ovens
B2c	Microwaves
B2d	Dishwashers
B2e	Toasters

Primjećujemo da polje `name` sadrži jedinstvene identifikatore koji slijede određenu konvenciju za označavanje hijerarhije kategorija, dok polje `description` pruža opisnu oznaku svake kategorije. Kategorije su organizirane hijerarhijski, gdje su identifikatori kategorija strukturirani na način da odražavaju odnose roditelji-dijete među kategorijama.

## Pregled Hijerarhije

### 1. Glavne Kategorije:

- Glavne kategorije imaju oznake poput `A` i `B`, što predstavljaju najviše razine u hijerarhiji.
- **Primjer:**
  - `A` - Consumer Electronics
  - `B` - Home Appliances

### 2. Podkategorije:

- Podkategorije dodaju dodatne znakove glavnoj kategoriji kako bi označile hijerarhijski položaj unutar te glavne kategorije.
- **Primjer:**
  - `A2` - Phones and Telecommunication
  - `A4` - Computers and Tablets
  - `B2` - Kitchen Appliances

### 3. Specifične Podkategorije:



- Specifične podkategorije dodatno specificiraju proizvode unutar podkategorija pomoću dodatnih slova ili brojeva.
- **Primjer:**
  - A3a - Nokia Mobile Phones
  - A4a - Desktop Computers
  - B2a - Refrigerators

Hijerarhijska struktura polja `name` omogućava jasno definiranu organizaciju kategorija i podkategorija. Slijedeći ovu strukturu, svaki identifikator u polju `name` predstavlja određeni stupanj specifičnosti, počevši od širokih glavnih kategorija (poput `A` i `B`), pa sve do vrlo specifičnih podkategorija (poput `A3a` i `B2a`). Polje `description` pruža jasne, ljudski čitljive nazive koji opisuju svaki nivo hijerarhije. Ova struktura omogućuje lako pretraživanje i filtriranje proizvoda prema njihovim kategorijama, te pomaže u organizaciji skladišta na logičan i učinkovit način.

## Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - `id` : Osigurava jedinstvenost svakog zapisa u tablici `category`.

## Relacije i Domene

- **Relacije:**
  - Tablica `category` može imati relaciju 1 s tablicom `product`, gdje jedna kategorija može sadržavati više proizvoda.
- **Domene:**
  - `id` : Mora biti jedinstveni identifikator za svaku kategoriju.
  - `name` : Alfnumeričko polje koje može sadržavati naziv kategorije.
  - `description` : Alfnumeričko polje koje može sadržavati opis kategorije.

```
1 CREATE TABLE `category` (
2   `id` int NOT NULL AUTO_INCREMENT,
3   `name` varchar(255) DEFAULT NULL,
4   `description` varchar(255) DEFAULT NULL,
5   PRIMARY KEY (`id`)
6 );
```

## 6.7. Tablica 'shipping\_address'

Tablica `'shipping_address'` pohranjuje informacije o adresama dostave za narudžbe. Svaki redak u tablici predstavlja jednu adresu dostave s jedinstvenim identifikatorom, adresom, gradom, državom, poštanskim brojem i zemljom. Tablica `'shipping_address'` omogućuje sustavu da osigura točne i pravovremene dostave, te integraciju s modulima za upravljanje narudžbama omogućava sustavu da automatski povuče podatke o adresama dostave prilikom obrade narudžbi, smanjujući rizik od grešaka i povećavajući učinkovitost dostave.

Održavanjem dobro strukturirane tablice 'shipping\_address', sustav osigurava točnost, učinkovitost i pouzdanost u upravljanju adresama dostave unutar skladišta.

Adrese unutar 'shipping\_address' tablice, su direktno vezane samo uz 'customer\_order', a indirektno vezane sa 'customer' tablicu. Razlog tome je što kupac može mjenjati svoju dostavnu adresu beskonačno mnogo puta, no u kontekstu ispunjavanja narudžbe važna je jedino ona dostavna adresa koja je bila vezana uz 'customer' dok se narudžba izvodila.

## Opis Atributa

### 1. **id** ( `int` , NOT NULL, AUTO\_INCREMENT)

- **Opis:** Jedinstveni identifikator za svaku adresu dostave. Ovaj primarni ključ osigurava da se svaka adresa može jedinstveno referencirati unutar sustava.
- **Svrha:** Služi kao primarni ključ za tablicu, pružajući jedinstveni identifikator za svaki zapis.

### 2. **address** ( `varchar(255)` , NOT NULL)

- **Opis:** Ulica i broj adrese dostave.
- **Svrha:** Pruža specifične informacije o lokaciji gdje će narudžba biti dostavljena.

### 3. **city** ( `varchar(255)` , NOT NULL)

- **Opis:** Grad dostave.
- **Svrha:** Specifikacija grada unutar adrese dostave.

### 4. **state** ( `varchar(255)` , NOT NULL)

- **Opis:** Država/regija dostave.
- **Svrha:** Pruža informacije o državi ili regiji unutar adrese dostave.

### 5. **zip** ( `varchar(255)` , NOT NULL)

- **Opis:** Poštanski broj adrese dostave.
- **Svrha:** Pruža poštanski broj koji pomaže u preciznom određivanju lokacije dostave.

### 6. **country** ( `varchar(255)` , NOT NULL)

- **Opis:** Država dostave.
- **Svrha:** Pruža informacije o zemlji gdje će narudžba biti dostavljena

## Indeksi i Ograničenja

- **Primarni Ključ ( `id` ):** Osigurava da svaka adresa dostave ima jedinstveni identifikator.
- **Indeks ( `idx_shipping_address` ):** Kreiran je na polju `address` kako bi se ubrzalo pretraživanje i filtriranje po adresi.

```
1 CREATE TABLE `shipping_address` (  
2   `id` int NOT NULL AUTO_INCREMENT,  
3   `address` varchar(255) NOT NULL,  
4   `city` varchar(255) NOT NULL,  
5   `state` varchar(255) NOT NULL,  
6   `zip` varchar(255) NOT NULL,  
7   `country` varchar(255) NOT NULL,
```

```

8     PRIMARY KEY (`id`),
9     KEY `idx_shipping_address` (`address`)
10 );

```

**Indeks ('idx\_shipping\_address')** koji je ovdje prikazan kao **KEY** kreiran je na polju 'address' kako bi se ubrzalo pretraživanje i filtriranje po adresi. Korištenjem indeksa, baza podataka stvara strukturu podataka (B-stablo) koja omogućava brže pronalaženje vrijednosti unutar tog stupca. Umjesto da pretražuje svaki redak u tablici da bi našla odgovarajuću adresu, baza podataka koristi indeks kako bi brzo pronašla retke koji odgovaraju kriterijima pretrage. U slučaju tablice 'shipping\_address', stupac koji će se najviše pretraživati je **address** tako da korištenjem indeksa povećavamo efikasnost pretraživanja informacija.

## Primjer database indeksa u praksi

Pretpostavimo da imate sljedeće retke u tablici `shipping_address` :

id	address	city	state	zip	country
1	123 Maple Street	CityA	StateA	11111	CountryA
2	456 Oak Avenue	CityB	StateB	22222	CountryB
3	789 Pine Road	CityC	StateC	33333	CountryC

Upitom `SELECT * FROM shipping_address WHERE address = '456 Oak Avenue'`; Baza podataka će koristiti indeks 'idx\_shipping\_address' kako bi brzo pronašla redak s adresom "456 Oak Avenue" umjesto da pretražuje cijelu tablicu.

## Relacije i Domene

- **Relacije:**
  - **Narudžba Kupca ( customer\_order ):**
    - Svaka narudžba kupca može biti povezana s jednom adresom dostave, čineći relaciju M:1.
  - **Kupac ( customer ):**
    - Svaki kupac može imati jednu ili više adresa dostave, čineći relaciju 1 .
- **Domene:**
  - `id` : Mora biti jedinstveni identifikator za svaku adresu dostave.
  - `address` : Mora biti valjana adresa koja precizira ulicu i broj.
  - `city` : Mora biti valjani naziv grada.
  - `state` : Mora biti valjani naziv države ili provincije.
  - `zip` : Mora biti valjani poštanski broj.
  - `country` : Mora biti valjani naziv države.

## 6.8. Tablica 'customer'

Tablica '**customer**' pohranjuje informacije o kupcima. Svaki redak predstavlja jednog kupca s jedinstvenim identifikatorom, imenom, prezimenom, adresom za dostavu, e-mail adresom, brojem telefona, državom,

poštanskim brojem i gradom.

## Opis Atributa

1. **id** ( `int` , NOT NULL, AUTO\_INCREMENT)
  - **Opis:** Jedinstveni identifikator za svakog kupca. Ovaj primarni ključ osigurava da se svaki kupac može jedinstveno referencirati unutar sustava.
  - **Svrha:** Služi kao primarni ključ za tablicu, pruža jedinstveni identifikator za svaki zapis.
2. **first\_name** ( `varchar(255)` , NOT NULL)
  - **Opis:** Ime kupca.
  - **Svrha:** Pohranjuje ime kupca za identifikaciju i komunikaciju.
3. **last\_name** ( `varchar(255)` , NOT NULL)
  - **Opis:** Prezime kupca.
  - **Svrha:** Pohranjuje prezime kupca za identifikaciju i komunikaciju.
4. **shipping\_address\_id** ( `int` , DEFAULT NULL)
  - **Opis:** Referenca na adresu dostave iz tablice `shipping_address` .
  - **Svrha:** Korištenjem stranog ključa, povezuje kupca s određenom adresom dostave.
5. **email** ( `varchar(255)` , NOT NULL)
  - **Opis:** E-mail adresa kupca.
  - **Svrha:** Pruža kontakt informacije za komunikaciju s kupcem.
6. **phone** ( `varchar(255)` , NOT NULL)
  - **Opis:** Telefonski broj kupca.
  - **Svrha:** Pruža dodatne kontakt informacije za komunikaciju s kupcem.
7. **country** ( `varchar(255)` , DEFAULT NULL)
  - **Opis:** Država kupca.
  - **Svrha:** Pruža informacije o zemlji prebivališta kupca.
8. **zip** ( `varchar(255)` , DEFAULT NULL)
  - **Opis:** Poštanski broj kupca.
  - **Svrha:** Pruža poštanski broj za identifikaciju lokacije kupca.
9. **city** ( `varchar(255)` , DEFAULT NULL)
  - **Opis:** Grad kupca.
  - **Svrha:** Pruža informacije o gradu prebivališta kupca.

## Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - `id` : Osigurava jedinstvenost svakog zapisa u tablici `customer` .
- **Indeks (KEY):**
  - `fk_customer_shipping_address ( shipping_address_id )` : Indeks za ubrzavanje pretraživanja prema polju `shipping_address_id` .

- **Strani ključ (FOREIGN KEY):**

- `shipping_address_id` : Povezuje se s poljem `id` u tablici `shipping_address` kako bi se osigurala referencijalna integritet.

## Relacije i Domene

- **Relacije:**

- **Adresa Dostave ( `shipping_address` ):**
  - Svaki kupac može biti povezan s jednom adresom dostave, čineći relaciju 1 .
- **Narudžba Kupca ( `customer_order` ):**
  - Svaka narudžba kupca mora biti povezana s jednim kupcem, čineći relaciju M:1.

- **Domene:**

- `id` : Mora biti jedinstveni identifikator za svakog kupca.
  - `first_name` : Mora biti valjano ime kupca.
  - `last_name` : Mora biti valjano prezime kupca.
  - `shipping_address_id` : Ako je definirano, mora biti valjani identifikator postojeće adrese dostave.
  - `email` : Mora biti valjana email adresa.
  - `phone` : Mora biti valjani telefonski broj.
  - `country` : Mora biti valjani naziv države.
  - `zip` : Mora biti valjani poštanski broj.
  - `city` : Mora biti valjani naziv grada.
- **Primarni Ključ ( `id` ):** Osigurava da svaki kupac ima jedinstveni identifikator.
  - **Indeks ( `fk_customer_shipping_address` ):** Kreiran je na polju `shipping_address_id` kako bi se ubrzalo pretraživanje i filtriranje po adresi dostave.

```
1 CREATE TABLE `customer` (  
2   `id` int NOT NULL AUTO_INCREMENT,  
3   `first_name` varchar(255) NOT NULL,  
4   `last_name` varchar(255) NOT NULL,  
5   `shipping_address_id` int DEFAULT NULL,  
6   `email` varchar(255) NOT NULL,  
7   `phone` varchar(255) NOT NULL,  
8   `country` varchar(255) DEFAULT NULL,  
9   `zip` varchar(255) DEFAULT NULL,  
10  `city` varchar(255) DEFAULT NULL,  
11  PRIMARY KEY (`id`),  
12  KEY `fk_customer_shipping_address` (`shipping_address_id`)  
13 );
```

## 6.9. Tablica 'product\_trait'

Tablica '**product\_trait**' pohranjuje informacije o specifičnim karakteristikama proizvoda. Svaki redak u tablici predstavlja karakteristike jednog proizvoda s jedinstvenim identifikatorom, težinom, dimenzijama, informacijom o tome je li proizvod namjenjen direktnoj prodaji potrošaču (b2c) ili ne, te je li proizvod lako lomljiv, pa se sa istim mora rukovati na siguran način. Tablica `product_trait` se popunjava neovisno od tablice `product`, tj. ne postoji direktna veza između tih entitet, te nije strogo definiran redoslijed popunjavanja tablica. Prema tome, vezivanje **product.id** i **product\_trait.id** je opcionalno i izvodi se indirektno preko tablice `product_to_product_trait`.

## Opis Atributa

1. **id** ( `int` , NOT NULL, AUTO\_INCREMENT)
  - **Opis:** Jedinstveni identifikator za svaku skupinu karakteristika proizvoda. Ovaj primarni ključ osigurava da se svaka skupina karakteristika može jedinstveno referencirati unutar sustava.
  - **Svrha:** Služi kao primarni ključ za tablicu, pružajući jedinstveni identifikator za svaki zapis.
2. **weight\_kg** ( `double` , DEFAULT NULL)
  - **Opis:** Težina proizvoda u kilogramima.
  - **Svrha:** Pruža informacije o težini proizvoda, što je važno za transport i skladištenje.
3. **dimensions\_mm** ( `varchar(255)` , DEFAULT NULL)
  - **Opis:** Dimenzije proizvoda u milimetrima (duljina, širina, visina).
  - **Svrha:** Pruža informacije o veličini proizvoda, što je važno za pakiranje, transport i skladištenje.
4. **consumer\_product** ( `tinyint(1)` , DEFAULT NULL)
  - **Opis:** Oznaka da li je proizvod namijenjen potrošačima (1 za da, 0 za ne).
  - **Svrha:** Identificira proizvode namijenjene za krajnje korisnike, što može utjecati na način skladištenja i rukovanja proizvodom.
5. **fragile** ( `tinyint(1)` , DEFAULT NULL)
  - **Opis:** Oznaka da li je proizvod lomljiv (1 za da, 0 za ne).
  - **Svrha:** Pruža informacije o tome je li proizvod lomljiv, što je važno za pravilno rukovanje i skladištenje kako bi se spriječila oštećenja.

## Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - `id` : Osigurava jedinstvenost svakog zapisa u tablici `product_trait`.

## Relacije i Domene

- **Relacije:**
  - **Proizvod** ( `product` ):
    - Svaka karakteristika proizvoda može biti povezana s jednim ili više proizvoda putem posredne tablice `product_to_product_trait`, čineći relaciju M.
- **Domene:**
  - `id` : Mora biti jedinstveni identifikator za svaku karakteristiku proizvoda.

- `weight_kg` : Mora biti pozitivna decimalna vrijednost ili NULL ako nije primjenjivo.
- `dimensions_mm` : Mora biti valjana tekstualna vrijednost koja opisuje dimenzije proizvoda ili NULL ako nije primjenjivo.
- `consumer_product` : Mora biti 1 (da) ili 0 (ne), ili NULL ako nije primjenjivo.
- `fragile` : Mora biti 1 (da) ili 0 (ne), ili NULL ako nije primjenjivo.

```

1 CREATE TABLE `product_trait` (
2   `id` int NOT NULL AUTO_INCREMENT,
3   `weight_kg` double DEFAULT NULL,
4   `dimensions_mm` varchar(255) DEFAULT NULL,
5   `consumer_product` tinyint(1) DEFAULT NULL,
6   `fragile` tinyint(1) DEFAULT NULL,
7   PRIMARY KEY (`id`)
8 );

```

## 6.10. Tablica 'product\_to\_product\_trait'

Tablica **'product\_to\_product\_trait'** povezuje proizvode s njihovim karakteristikama, omogućujući kreiranje **many-to-many** odnosa između proizvoda i njihovih specifičnih karakteristika. Svaki redak u ovoj tablici predstavlja vezu između jednog proizvoda i jedne njegove karakteristike.

Pružajući strukturu za pohranjivanje i upravljanje vezama između proizvoda i njihovih karakteristika, tablica `product_to_product_trait` omogućuje sustavu da osigura učinkovitost i točnost u upravljanju zalihama. Na primjer, povezivanjem proizvoda s njihovim karakteristikama, sustav može automatski prilagoditi skladištenje lomljivih proizvoda na sigurna mjesta ili optimizirati pakiranje proizvoda na temelju njihovih dimenzija i težine.

### Opis Atributa

#### 1. `product_id` ( `int` , NOT NULL)

- **Opis:** Jedinstveni identifikator proizvoda.
- **Svrha:** Povezuje karakteristiku s određenim proizvodom koristeći strani ključ prema tablici `product`.

#### 2. `product_trait_id` ( `int` , NOT NULL)

- **Opis:** Jedinstveni identifikator karakteristike proizvoda.
- **Svrha:** Povezuje proizvod s određenom karakteristikom koristeći strani ključ prema tablici `product_trait`.

### Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - `product_id`, `product_trait_id` : Kombinirani primarni ključ osigurava jedinstvenost svake relacije između proizvoda i karakteristike.
- **Strani ključevi (FOREIGN KEY):**

- `product_id` : Strani ključ koji referencira `id` u tablici `product` .
- `product_trait_id` : Strani ključ koji referencira `id` u tablici `product_trait` .

## Relacije i Domene

- **Relacije:**
  - **Proizvod ( `product` ):**
    - Svaki zapis u tablici `product_to_product_trait` povezuje jedan proizvod s jednom karakteristikom.
    - Relacija: Jedan proizvod može imati više karakteristika (M relacija).
  - **Karakteristika Proizvoda ( `product_trait` ):**
    - Svaki zapis u tablici `product_to_product_trait` povezuje jednu karakteristiku s jednim proizvodom.
    - Relacija: Jedna karakteristika može biti povezana s više proizvoda (M relacija).
- **Domene:**
  - `product_id` : Mora biti validan `id` iz tablice `product` .
  - `product_trait_id` : Mora biti validan `id` iz tablice `product_trait` .

```

1 CREATE TABLE `product_to_product_trait` (
2   `product_id` int NOT NULL,
3   `product_trait_id` int NOT NULL,
4   PRIMARY KEY (`product_id`,`product_trait_id`),
5   KEY `product_trait_id` (`product_trait_id`),
6   CONSTRAINT `product_to_product_trait_ibfk_1` FOREIGN KEY (`product_id`)
REFERENCES `product` (`id`),
7   CONSTRAINT `product_to_product_trait_ibfk_2` FOREIGN KEY (`product_trait_id`)
REFERENCES `product_trait` (`id`)
8 );

```

## 6.11. Tablica 'customer\_order'

Tablica **'customer\_order'** pohranjuje informacije o narudžbama kupaca, uključujući detalje o košarici, kupcu, adresi dostave, datumu narudžbe, statusu narudžbe i skladištu koje obrađuje narudžbu.

Pri kreiranju narudžbe, status te narudžbe poprima vrijednost **"processing"** što označava početak obradnog procesa narudžbe. Sistem zatim automatski odabire optimalno skladište iz kojega će se narudžba isporučiti. U stvarnosti donošenje takve odluke bi bilo potkrijepljeno obradom velike količine podataka koje sustav skuplja tokom svog radnog vijeka. Prema tome, za potrebe ovog projekta tu funkcionalnost ću zamjeniti sa procedurom koja će popuniti vrijednost `'warehouse_id'` atributa sa nasumično odabranim jedinstvenim identifikatorom jednog od skladišta.

```

1 CREATE TABLE `customer_order` (
2   `id` int NOT NULL AUTO_INCREMENT,
3   `cart_id` int NOT NULL,

```



```

4  `customer_id` int NOT NULL,
5  `shipping_address_id` int NOT NULL,
6  `order_date` datetime NOT NULL,
7  `status` varchar(255) NOT NULL,
8  `warehouse_id` int DEFAULT NULL,
9  PRIMARY KEY (`id`),
10 UNIQUE KEY `unique_cart_id` (`cart_id`),
11 KEY `fk_customer_order_customer` (`customer_id`),
12 KEY `fk_customer_order_warehouse` (`warehouse_id`),
13 KEY `fk_customer_order_shipping_address` (`shipping_address_id`),
14 CONSTRAINT `fk_customer_order_cart` FOREIGN KEY (`cart_id`) REFERENCES `cart`
   (`id`),
15 CONSTRAINT `fk_customer_order_customer` FOREIGN KEY (`customer_id`) REFERENCES
   `customer` (`id`),
16 CONSTRAINT `fk_customer_order_shipping_address` FOREIGN KEY
   (`shipping_address_id`) REFERENCES `shipping_address` (`id`),
17 CONSTRAINT `fk_customer_order_warehouse` FOREIGN KEY (`warehouse_id`)
   REFERENCES `warehouse` (`id`)
18 );

```

## Opis Atributa

### 1. `id` ( `int` , NOT NULL, AUTO\_INCREMENT)

- **Opis:** Jedinstveni identifikator za svaku narudžbu. Ovaj primarni ključ osigurava da se svaka narudžba može jedinstveno referencirati unutar sustava.
- **Svrha:** Služi kao primarni ključ za tablicu, pružajući jedinstveni identifikator za svaki zapis.

### 2. `cart_id` ( `int` , NOT NULL)

- **Opis:** Identifikator povezane košarice.
- **Svrha:** Povezuje narudžbu s pripadajućom košaricom, koristeći strani ključ prema tablici `cart`.

### 3. `customer_id` ( `int` , NOT NULL)

- **Opis:** Identifikator kupca koji je izvršio narudžbu.
- **Svrha:** Povezuje narudžbu s kupcem, koristeći strani ključ prema tablici `customer`.

### 4. `shipping_address_id` ( `int` , NOT NULL)

- **Opis:** Identifikator adrese dostave.
- **Svrha:** Povezuje narudžbu s adresom dostave, koristeći strani ključ prema tablici `shipping_address`.

### 5. `order_date` ( `datetime` , NOT NULL)

- **Opis:** Datum i vrijeme kada je narudžba izvršena.
- **Svrha:** Pruža vremensku oznaku za narudžbu, što je važno za praćenje i upravljanje narudžbama.

### 6. `status` ( `varchar(255)` , NOT NULL)

- **Opis:** Status narudžbe (npr. 'na čekanju', 'obrađeno', 'isporučeno').
- **Svrha:** Pruža informacije o trenutnom statusu narudžbe, što je važno za praćenje napretka i obavješćavanje kupaca.

## 7. `warehouse_id` ( `int` , `DEFAULT NULL` )

- **Opis:** Identifikator skladišta koje obrađuje narudžbu.
- **Svrha:** Povezuje narudžbu sa skladištem, koristeći strani ključ prema tablici `warehouse` .

## Procedura ili trigger

```
1  -- create set_status_and_warehouse_id trigger --
2
3  create trigger set_status_and_warehouse_id
4
5  before insert on customer_order
6
7  for each row
8
9  begin
10
11  set new.status = 'processing';
12
13  set new.warehouse_id = (select id from warehouse order by rand() limit 1);
14
15  end;
```

## Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**
  - `id` : Osigurava jedinstvenost svakog zapisa u tablici `customer_order` .
- **Jedinstveni ključ (UNIQUE KEY):**
  - `unique_cart_id` ( `cart_id` ): Osigurava da svaka košarica može biti povezana samo s jednom narudžbom.
- **Vanjski ključevi (FOREIGN KEYS):**
  - `cart_id` :
    - **Opis:** Povezuje `customer_order` s tablicom `cart` .
    - **Ograničenje:** `CONSTRAINT fk_customer_order_cart FOREIGN KEY (cart_id) REFERENCES cart (id)` .
  - `customer_id` :
    - **Opis:** Povezuje `customer_order` s tablicom `customer` .
    - **Ograničenje:** `CONSTRAINT fk_customer_order_customer FOREIGN KEY (customer_id) REFERENCES customer (id)` .
  - `shipping_address_id` :
    - **Opis:** Povezuje `customer_order` s tablicom `shipping_address` .
    - **Ograničenje:** `CONSTRAINT fk_customer_order_shipping_address FOREIGN KEY (shipping_address_id) REFERENCES shipping_address (id)` .
  - `warehouse_id` :

- **Opis:** Povezuje `customer_order` s tablicom `warehouse`.
- **Ograničenje:** `CONSTRAINT fk_customer_order_warehouse FOREIGN KEY (warehouse_id) REFERENCES warehouse (id)`.
- **Indeksi (KEY):**
  - `fk_customer_order_customer (customer_id)`: Indeks za ubrzavanje pretraživanja prema `customer_id`.
  - `fk_customer_order_warehouse (warehouse_id)`: Indeks za ubrzavanje pretraživanja prema `warehouse_id`.
  - `fk_customer_order_shipping_address (shipping_address_id)`: Indeks za ubrzavanje pretraživanja prema `shipping_address_id`.

## Relacije i Domene

- **Relacije:**
  - **Košarica ( `cart` ):**
    - Svaka narudžba je povezana s jednom košaricom, čineći relaciju 1:1.
  - **Kupac ( `customer` ):**
    - Svaka narudžba je povezana s jednim kupcem, čineći relaciju M:1.
  - **Adresa Dostave ( `shipping_address` ):**
    - Svaka narudžba je povezana s jednom adresom dostave, čineći relaciju M:1.
  - **Skladište ( `warehouse` ):**
    - Svaka narudžba može biti povezana s jednim skladištem, čineći relaciju M:1.
- **Domene:**
  - `id` : Mora biti jedinstveni identifikator za svaku narudžbu.
  - `cart_id` : Mora biti valjani identifikator postojeće košarice.
  - `customer_id` : Mora biti valjani identifikator postojećeg kupca.
  - `shipping_address_id` : Mora biti valjani identifikator postojeće adrese dostave.
  - `order_date` : Mora biti valjani datum i vrijeme narudžbe.
  - `status` : Mora biti alfanumeričko polje koje predstavlja status narudžbe.
  - `warehouse_id` : Mora biti valjani identifikator postojećeg skladišta (ako je specificirano).

## 6.12. Tablica 'cart'

Tablica '**cart**' pohranjuje informacije o košaricama koje kupci koriste prilikom obavljanja kupnje. Svaki redak u ovoj tablici predstavlja jednu košaricu. Tablica 'cart' omogućuje sustavu da prati kupnje kupaca od trenutka dodavanja artikala u košaricu do trenutka dovršetka narudžbe. Na primjer, kada kupac dodaje artikle u svoju košaricu, sustav generira jedinstveni identifikator košarice koji se može koristiti za praćenje svih artikala u toj košarici i za povezivanje te košarice s narudžbom kada kupac završi kupnju.

## Ključevi i Ograničenja

- **Primarni ključ (PRIMARY KEY):**

- `id` : Primarni ključ osigurava jedinstvenost svakog zapisa u tablici.

## Relacije i Domene

- **Relacije:**
  - Tablica `cart` može imati relaciju s drugim tablicama kao što su `cart_item` i `customer_order`. Iako ove relacije nisu direktno navedene u DDL-u tablice `cart`, one se ostvaruju putem drugih tablica koje sadrže strani ključ `cart_id`.
    - `cart_item` : Svaka košarica može imati više artikala povezanih putem tablice `cart_item`.
    - `customer_order` : Košarica može biti povezana s narudžbom putem tablice `customer_order`.
- **Domena:**
  - `id` : Mora biti jedinstven i automatski se povećava za svaki novi zapis.

```
1 CREATE TABLE `cart` (
2   `id` int NOT NULL AUTO_INCREMENT,
3   PRIMARY KEY (`id`)
4 );
```

## 7. Upiti

- Igor Tadić (7.1.-7.26.)

### 7.1. Upit 1

Pogled nazvan "customer\_order\_view" omogućava sveobuhvatan pregled svih narudžbi koje su primljene. Ovaj pogled omogućava pregled svih narudžbi kupaca uključujući informacije o kupcu, datum narudžbe, adresu dostave i informacije o skladištu koje obradžuje narudžbu.

Kod:

```
1 CREATE VIEW customer_order_view AS
2 SELECT c.first_name, c.last_name, o.order_date, o.status, s.address, w.name AS
   warehouse_name, w.address AS warehouse_address
3 FROM customer_order o
4      JOIN customer c ON o.customer_id = c.id
5      JOIN shipping_address s ON o.shipping_address_id = s.id
6      JOIN warehouse w ON o.warehouse_id = w.id;
```

Ovaj pogled koristi SELECT naredbu za dohvaćanje podataka iz nekoliko povezanih tablica. Proces započinje s tablicom `customer_order`, kojoj se dodjeljuje alias `o` radi lakšeg referenciranja.

Zatim, slijedi spajanje tablica kako bi se prikupili svi potrebni podaci. Prva povezana tablica je `customer`, kojoj se dodjeljuje alias `c`. Ova tablica se spaja s `customer_order` na temelju zajedničkog stupca

`customer_id` . Na taj način dohvaćaju se podaci o imenu ( `first_name` ) i prezimenu ( `last_name` ) kupca za svaku narudžbu.

Sljedeći korak uključuje spajanje tablice `shipping_address` , kojoj se dodjeljuje alias `s` . Ova tablica se povezuje s `customer_order` na temelju stupca `shipping_address_id` . To omogućuje dohvaćanje adrese za dostavu ( `address` ) za svaku narudžbu, čime se osigurava da je svaka narudžba povezana s točnom dostavnom adresom.

Konačno, tablica `warehouse` , s aliasom `w` , spaja se s `customer_order` preko stupca `warehouse_id` . Ovo spajanje omogućuje dohvaćanje imena skladišta ( `name` , kojem se dodjeljuje alias `warehouse_name` ) i adrese skladišta ( `address` , kojem se dodjeljuje alias `warehouse_address` ). Time se osigurava da svaka narudžba ima pridružene informacije o skladištu iz kojeg je isporučena.

Kombinacijom svih ovih spajanja, rezultat je pogled `customer_order_view` koji sadrži sve relevantne informacije o narudžbama, uključujući podatke o kupcima, adresama za dostavu i skladištima. Ovaj pogled omogućuje jednostavan pristup i pregled svih potrebnih informacija na jednom mjestu.

## Rezultat Upita

	ABC first_name	ABC last_name	🕒 order_date	ABC status	ABC address	ABC warehouse_name	ABC warehouse_address
1	John	Doe	'2024-05-05 12:03:42'	PROCESSING	123 Main St	Warehouse Name	Warehouse Address
2	John	Doe	'2024-05-11 00:31:02'	PROCESSING	123 Main St	Portugal_Lisbon	741 Spruce St
3	Jane	Doe	'2024-05-11 00:31:02'	PROCESSING	123 Main St	Poland_Warsaw	369 Cedar St
4	igor	tadić	'2024-05-11 00:31:02'	PROCESSING	Avenue Street 1	Sweden_Stockholm	258 Maple St
5	Vinko	Horvat	'2024-05-11 00:31:02'	SHIPPED	Cooney Village 32	UK_London	123 Main St
6	Dobrožizna	Pribislava	'2024-05-11 00:31:02'	PROCESSING	City Street 51	Portugal_Lisbon	741 Spruce St
7	John	Doe	'2024-05-11 00:33:01'	PROCESSING	123 Main St	Croatia_Zagreb	Radnicka 25
8	Jane	Doe	'2024-05-11 00:33:01'	PROCESSING	123 Main St	Croatia_Zagreb	Radnicka 25
9	igor	tadić	'2024-05-11 00:33:01'	PROCESSING	Avenue Street 1	Poland_Warsaw	369 Cedar St
10	Vinko	Horvat	'2024-05-11 00:33:01'	PROCESSING	Cooney Village 32	France_Paris	456 Market St
11	Dobrožizna	Pribislava	'2024-05-11 00:33:01'	PROCESSING	City Street 51	Croatia_Zagreb	Radnicka 25

## 7.2. Upit 2

Pogled "customer\_order\_status\_shipped" omogućuje prikaz informacija o narudžbama koje su poslane.

Kod:

```
1 CREATE VIEW customer_order_status_shipped AS
2     SELECT id, order_date, status, warehouse_id
3     FROM customer_order
4     WHERE status = "SHIPPED";
```

Ovaj pogled koristi `SELECT` naredbu za dohvaćanje podataka iz tablice `customer_order` . Iz ove tablice odabrani su stupci `id` , `order_date` , `status` i `warehouse_id` .

Uvjet `WHERE` se koristi za filtriranje rezultata tako da se prikazuju samo one narudžbe koje imaju status "SHIPPED". Time se osigurava da pogled sadrži isključivo informacije o narudžbama koje su već poslane, omogućujući jednostavan pregled takvih narudžbi zajedno s datumom narudžbe, statusom i pripadajućim skladištem.

## Rezultat Upita

	123 id	order_date	ABC status	123 warehouse_id
1	8	2024-05-11 00:31:02	SHIPPED	2

### 7.3. Upit 3

Pogled "customer\_to\_shipping\_address" omogućuje prikaz informacija o kupcima zajedno s njihovim adresama za dostavu.

Kod:

```
1 CREATE VIEW customer_to_shipping_address AS
2     SELECT c.first_name, c.last_name, c.phone, c.country, s.address
3     FROM customer c
4     JOIN shipping_address s
5     ON c.shipping_address_id = s.id;
```

Ovaj pogled koristi SELECT naredbu za dohvaćanje relevantnih podataka iz tablica `customer` i `shipping_address`. Tablica `customer`, označena aliasom `c`, sadrži informacije o imenu, prezimenu, telefonskom broju i zemlji kupca. Tablica `shipping_address`, označena aliasom `s`, sadrži podatke o adresi za dostavu.

Povezivanje ovih dviju tablica ostvaruje se pomoću stupca `shipping_address_id` iz tablice `customer`, koji se poklapa sa stupcem `id` u tablici `shipping_address`. Ovaj JOIN omogućuje pristup adresama za dostavu koje odgovaraju svakom kupcu. Rezultat je skup podataka koji uključuje ime, prezime, telefonski broj, zemlju kupca i adresu za dostavu, omogućavajući jednostavan pregled svih potrebnih informacija o kupcima i njihovim adresama za dostavu.

## Rezultat Upita

	ABC first_name	ABC last_name	ABC phone	ABC country	ABC address
1	John	Doe	1234567890	USA	123 Main St
2	Jane	Doe	1234567890	USA	123 Main St
3	igor	tadić	1234567890	Croatia	Avenue Street 1
4	Vinko	Horvat	41320841	Croatia	Cooney Village 32
5	Dobrožizna	Pribislava	413201321	Slovenia	City Street 51

### 7.4. Upit 4

Pogled "customers\_from\_croatia" omogućuje prikaz imena i prezimena kupaca koji su iz Hrvatske.

Kod:

```
1 CREATE VIEW customers_from_croatia AS
2     SELECT first_name, last_name
3     FROM customer
4     WHERE country = "Croatia";
```

Ovaj pogled koristi SELECT naredbu za dohvaćanje podataka iz tablice `customer`. Iz te tablice odabrani su stupci `first_name` i `last_name`, a podaci se filtriraju tako da se prikažu samo oni kupci koji imaju vrijednost "Croatia" u stupcu `country`. To omogućuje izdvajanje popisa kupaca koji dolaze iz Hrvatske.

## Rezultat Upita

	ABC first_name	ABC last_name
1	igor	tadić
2	Vinko	Horvat
3	Stephanie	White

## 7.5. Upit 5

Pogled "total\_product\_inventory" omogućuje prikaz ukupne količine svakog proizvoda u skladištu.

Kod:

```
1 CREATE VIEW total_product_inventory AS
2 SELECT product_id, COUNT(*) as total_quantity
3 FROM warehouse_inventory
4 GROUP BY product_id;
```

Ovaj pogled koristi SELECT naredbu za dohvaćanje podataka iz tablice `warehouse_inventory`. Iz te tablice odabran je stupac `product_id`, a pomoću funkcije `COUNT(*)` računa se ukupna količina svakog proizvoda. Rezultat se grupira po `product_id` kako bi se dobila ukupna količina za svaki proizvod. Na taj način pogled prikazuje broj dostupnih jedinica svakog proizvoda u skladištu.

## Rezultat Upita

	123 product_id	123 total_quantity
1	2	255
2	33	165

## 7.6. Upit 6

Pogled "cart\_details" omogućuje prikaz detalja o artiklima u košaricama korisnika.

Kod:

```
1 CREATE VIEW cart_details AS
2 SELECT c.id as cart_id, a.name as article_name, a.code as article_code,
3      a.variant as article_variant
4 FROM cart_item ci
5 JOIN article a ON ci.article_id = a.id
6 JOIN cart c ON ci.cart_id = c.id;
```

Ovaj pogled koristi SELECT naredbu za dohvaćanje podataka iz tablica `cart_item`, `article` i `cart`. Povezivanjem tablice `cart_item` s tablicom `article` putem `article_id`, te povezivanjem s tablicom

cart putem cart\_id , omogućuje se dohvaćanje detalja o artiklima koji se nalaze u pojedinim košaricama. Rezultat prikazuje ID košarice ( cart\_id ), naziv artikla ( article\_name ), šifru artikla ( article\_code ) i varijantu artikla ( article\_variant ). Na taj način, ovaj pogled olakšava praćenje sadržaja košarica korisnika.

## Rezultat Upita

	123 cart_id ▼	ABC article_name ▼	ABC article_code ▼	ABC article_variant ▼
1	1	Article Name	Article Code	Article Variant
2	1	Article Name	Article Code	Article Variant
3	1	Article Name	Article Code	Article Variant
4	2	[NULL]	[NULL]	[NULL]

Pošto nisam davao artiklima specifična imena, ovaj upit vraća 44 redaka podataka, od koja su 3 artikli sa testnim vrijednostima za imena i varijanti, a ostalih 41 imaju vrijednosti NULL.

## 7.7. Upit 7.

Pogled "cart\_value" omogućuje prikaz ukupne vrijednosti artikala u svakoj košarici korisnika.

Kod:

```
1 CREATE VIEW cart_value AS
2 SELECT ci.cart_id, SUM(a.price) as total_value
3 FROM cart_item ci
4 JOIN article a ON ci.article_id = a.id
5 GROUP BY ci.cart_id;
```

Ovaj pogled koristi SELECT naredbu za dohvaćanje podataka iz tablica cart\_item i article . Povezivanjem tablice cart\_item s tablicom article putem article\_id , omogućuje se dohvaćanje cijena artikala koji se nalaze u pojedinim košaricama. Zatim se ukupna vrijednost artikala za svaku košaricu izračunava pomoću funkcije SUM nad stupcem price . Podaci su grupirani prema cart\_id , čime se za svaku košaricu dobiva ukupan zbroj cijena artikala u njoj. Rezultat prikazuje ID košarice ( cart\_id ) i ukupnu vrijednost artikala u toj košarici ( total\_value ). Ovaj pogled olakšava praćenje ukupnih troškova korisničkih košarica.

## Rezultat Upita

	123 cart_id ▼	123 total_value ▼
1	1	300
2	2	1,200
3	3	1,000
4	4	1,000
5	5	1,000
6	6	1,000
7	7	1,000
8	8	1,000
9	9	1,000



## 7.8. Upit 8

Pogled "customer\_orders" omogućuje pregled svih narudžbi s osnovnim podacima o kupcima i statusima narudžbi.

Kod:

```
1 CREATE VIEW customer_orders AS
2 SELECT co.id as order_id, cu.first_name, cu.last_name, co.order_date, co.status
3 FROM customer_order co
4 JOIN customer cu ON co.customer_id = cu.id;
```

Ovaj pogled koristi SELECT naredbu za dohvaćanje podataka iz tablica `customer_order` i `customer`. Povezivanjem tablice `customer_order` s tablicom `customer` putem `customer_id`, omogućuje se dobivanje informacija o kupcima koji su napravili narudžbe. Rezultat prikazuje ID narudžbe (`order_id`), ime i prezime kupca (`first_name`, `last_name`), datum narudžbe (`order_date`) i status narudžbe (`status`). Ovaj pogled pruža cjelovit pregled svih narudžbi zajedno s relevantnim podacima o kupcima, što olakšava praćenje i upravljanje narudžbama.

### Rezultat Upita

	123 order_id	ABC first_name	ABC last_name	🕒 order_date	ABC status
1	4	John	Doe	2024-05-05 12:03:42	PROCESSING
2	5	John	Doe	2024-05-11 00:31:02	PROCESSING
3	6	Jane	Doe	2024-05-11 00:31:02	PROCESSING
4	7	igor	tadić	2024-05-11 00:31:02	PROCESSING
5	8	Vinko	Horvat	2024-05-11 00:31:02	SHIPPED
6	9	Dobrožizna	Pribislava	2024-05-11 00:31:02	PROCESSING
7	10	John	Doe	2024-05-11 00:33:01	PROCESSING
8	11	Jane	Doe	2024-05-11 00:33:01	PROCESSING
9	12	igor	tadić	2024-05-11 00:33:01	PROCESSING
10	13	Vinko	Horvat	2024-05-11 00:33:01	PROCESSING
11	14	Dobrožizna	Pribislava	2024-05-11 00:33:01	PROCESSING

## 7.9. Upit 9

Pogled "warehouse\_article\_count" omogućuje pregled ukupnog broja artikala po skladištima.

Kod:

```
1 CREATE VIEW warehouse_article_count AS
2 SELECT warehouse_id, COUNT(*) as total_articles
3 FROM warehouse_inventory
4 GROUP BY warehouse_id;
```

Ovaj pogled koristi SELECT naredbu za dohvaćanje podataka iz tablice `warehouse_inventory`. Grupiranjem podataka prema `warehouse_id`, omogućuje se brojanje ukupnog broja artikala u svakom skladištu. Rezultat prikazuje ID skladišta (`warehouse_id`) i ukupan broj artikala (`total_articles`). Ovaj

pogled pruža brz i jednostavan pregled zaliha u svakom skladištu, što pomaže u učinkovitijem upravljanju zalihama.

## Rezultat Upita

	123 warehouse_id ▼	123 total_articles ▼
1	1	60
2	2	45
3	3	30
4	4	45
5	5	30
6	7	45
7	8	45
8	9	30
9	10	30
10	11	30
11	12	30

## 7.10. Upit 10

Pogled "processed\_order\_count\_by\_warehouse" omogućuje pregled broja obrađenih narudžbi po skladištima.

Kod:

```
1 CREATE VIEW `processed_order_count_by_warehouse` AS
2 SELECT w.id AS warehouse_id, w.name AS warehouse_name, COUNT(o.id) AS
   total_orders_processed
3 FROM `warehouse` w
4     LEFT JOIN `customer_order` o ON w.id = o.warehouse_id
5 GROUP BY w.id, w.name
6 ORDER BY total_orders_processed DESC;
```

Ovaj pogled koristi SELECT naredbu za dohvaćanje podataka iz tablica `warehouse` i `customer_order`. Prvo se izvodi LEFT JOIN kako bi se sva skladišta iz tablice `warehouse` povezala s narudžbama iz tablice `customer_order` prema `warehouse_id`. Zatim se grupira prema `w.id` i `w.name` kako bi se dobio broj narudžbi obrađenih po skladištu. Konačni rezultat prikazuje ID skladišta (`warehouse_id`), naziv skladišta (`warehouse_name`), i ukupan broj obrađenih narudžbi (`total_orders_processed`). Podaci su sortirani silazno prema broju obrađenih narudžbi, što omogućuje identifikaciju skladišta s najviše obrađenih narudžbi.

## Rezultat Upita

	123 warehouse_id ▼	ABC warehouse_name ▼	123 total_orders_processed ▼
1	12	Croatia_Zagreb	3
2	10	Poland_Warsaw	2
3	11	Portugal_Lisbon	2
4	1	Warehouse Name	1
5	2	UK_London	1
6	3	France_Paris	1
7	9	Sweden_Stockholm	1
8	4	Germany_Berlin	0
9	5	Spain_Madrid	0
10	6	Italy_Rome	0
11	7	Netherlands_Amsterdam	0
12	8	Belgium_Brussels	0

## 7.11. Upit 11

Ovaj upit omogućuje pregled svih narudžbi kupaca zajedno s njihovom ukupnom cijenom i brojem artikala.

Kod:

```

1  SELECT co.id AS order_id,
2         c.first_name,
3         c.last_name,
4         SUM(a.price) AS total_price,
5         COUNT(ci.id) AS number_of_items,
6         co.order_date
7  FROM customer_order co
8       JOIN customer c ON co.customer_id = c.id
9       JOIN cart_item ci ON co.cart_id = ci.cart_id
10      JOIN article a ON ci.article_id = a.id
11 GROUP BY co.id, c.first_name, c.last_name, co.order_date
12 ORDER BY co.order_date DESC;
```

Ovaj upit povezuje nekoliko tablica kako bi dobio sve potrebne informacije o narudžbama kupaca. Tablica `customer_order` se povezuje s tablicom `customer` preko `customer_id`, čime se dobivaju podaci o kupcu kao što su ime i prezime. Zatim se `customer_order` povezuje s `cart_item` putem `cart_id`, a `cart_item` se povezuje s tablicom `article` putem `article_id`.

Pomoću agregatnih funkcija, kao što su `SUM` za ukupnu cijenu i `COUNT` za broj artikala, upit računa ukupnu cijenu svih artikala u narudžbi i broj artikala u svakoj narudžbi. Grupa se formira prema ID-u narudžbe, imenu i prezimenu kupca, te datumu narudžbe kako bi se osigurali točni agregirani podaci za svaku narudžbu. Konačno, rezultati su sortirani prema datumu narudžbe u silaznom redoslijedu, omogućujući prikaz najnovijih narudžbi na vrhu.

## Rezultat Upita

	123 order_id	ABC first_name	ABC last_name	123 total_price	123 number_of_items	order_date
1	10	John	Doe	1,000	5	2024-05-11 00:33:01
2	11	Jane	Doe	1,000	5	2024-05-11 00:33:01
3	12	igor	tadić	1,000	5	2024-05-11 00:33:01
4	5	John	Doe	1,200	6	2024-05-11 00:31:02
5	6	Jane	Doe	1,000	5	2024-05-11 00:31:02
6	7	igor	tadić	1,000	5	2024-05-11 00:31:02
7	8	Vinko	Horvat	1,000	5	2024-05-11 00:31:02
8	9	Dobrožizna	Pribislava	1,000	5	2024-05-11 00:31:02
9	4	John	Doe	300	3	2024-05-05 12:03:42

## 7.12. Upit 12

Ovaj upit omogućuje pronalaženje pet kupaca koji su potrošili najviše novca na narudžbe.

Kod:

```

1  SELECT c.id,
2         c.first_name,
3         c.last_name,
4         SUM(a.price) AS total_spent
5  FROM customer c
6       JOIN customer_order co ON c.id = co.customer_id
7       JOIN cart_item ci ON co.cart_id = ci.cart_id
8       JOIN article a ON ci.article_id = a.id
9  GROUP BY c.id, c.first_name, c.last_name
10 ORDER BY total_spent DESC
11 LIMIT 5;
```

Ovaj upit povezuje nekoliko tablica kako bi dobio podatke o ukupnoj potrošnji svakog kupca. Tablica `customer` se povezuje s tablicom `customer_order` preko `customer_id`, čime se dobivaju informacije o narudžbama koje je svaki kupac napravio. Zatim se `customer_order` povezuje s `cart_item` putem `cart_id`, a `cart_item` se povezuje s tablicom `article` putem `article_id`.

Koristeći agregatnu funkciju `SUM`, upit izračunava ukupnu vrijednost artikala kupljenih od strane svakog kupca. Grupiranje se vrši prema ID-u kupca, imenu i prezimenu kako bi se dobili točni podaci za svakog kupca. Konačno, rezultati su sortirani prema ukupnoj potrošnji u silaznom redoslijedu, a ograničenje `LIMIT 5` osigurava da se prikaže samo pet kupaca koji su potrošili najviše novca.

## Rezultat Upita

	123 id	ABC first_name	ABC last_name	123 total_spent
1	2	John	Doe	2,500
2	3	Jane	Doe	2,000
3	4	igor	tadić	2,000
4	5	Vinko	Horvat	1,000
5	6	Dobrožizna	Pribislava	1,000

## 7.13. Upit 13

Ovaj upit omogućuje prikaz svih proizvoda zajedno s njihovim karakteristikama.

Kod:

```
1  SELECT p.id AS product_id,
2         p.name AS product_name,
3         pt.weight_kg,
4         pt.dimensions_mm,
5         pt.consumer_product,
6         pt.fragile
7  FROM product p
8       JOIN product_to_product_trait ptpt ON p.id = ptpt.product_id
9       JOIN product_trait pt ON ptpt.product_trait_id = pt.id
10 ORDER BY p.name;
```

Ovim upitom dohvaćamo informacije o proizvodima i njihovim karakteristikama iz više tablica. Prvo se koristi tablica `product` koja sadrži osnovne informacije o proizvodima, kao što su ID i naziv proizvoda. Ta se tablica povezuje s međutablicom `product_to_product_trait` putem `product_id`, koja služi za povezivanje proizvoda s njihovim karakteristikama.

Međutablica `product_to_product_trait` povezuje se s tablicom `product_trait` putem `product_trait_id`, što omogućuje dohvaćanje detaljnih informacija o karakteristikama proizvoda kao što su težina, dimenzije, je li proizvod namijenjen potrošačima i je li lomljiv.

Konačno, rezultati su poredani prema nazivu proizvoda kako bi se dobio pregledan popis proizvoda zajedno s njihovim karakteristikama.

## Rezultat Upita

	123 product_id ▼	ABC product_name ▼	123 weight_kg ▼	ABC dimensions_mm ▼	123 consumer_product ▼	123 fragile ▼
1	2	Product Name	1.5	150x100x50	1	0

## 7.14. Upit 14

Ovaj upit omogućuje pronalaženje svih lomljivih proizvoda i njihovih pripadajućih skladišnih lokacija.

Kod:

```
1  SELECT p.id AS product_id,
2         p.name AS product_name,
3         w.name AS warehouse_name,
4         w.address AS warehouse_address
5  FROM product p
6       JOIN product_to_product_trait ptpt ON p.id = ptpt.product_id
7       JOIN product_trait pt ON ptpt.product_trait_id = pt.id
8       JOIN article a ON p.id = a.product_id
9       JOIN warehouse_inventory wi ON a.id = wi.article_id
10      JOIN warehouse w ON wi.warehouse_id = w.id
```

```
11 WHERE pt.fragile = 1
12 ORDER BY p.name;
```

Ovim upitom se dohvaćaju informacije o proizvodima koji su označeni kao lomljivi zajedno s njihovim lokacijama u skladištima.

Prvo se koriste tablice `product` i `product_to_product_trait` za povezivanje proizvoda s njihovim karakteristikama. Zatim se povezuje s tablicom `product_trait` kako bi se identificirali proizvodi koji su lomljivi (gdje je `pt.fragile = 1`).

Nakon toga, tablica `product` se povezuje s tablicom `article` da bi se dobila informacija o artiklima povezanim s proizvodima. Ti artikli se zatim povezuju s tablicom `warehouse_inventory` kako bi se dobile informacije o skladištima u kojima se nalaze.

Na kraju, tablica `warehouse` pruža detalje o skladištima, kao što su ime i adresa skladišta. Rezultati su poredani po nazivu proizvoda kako bi se dobio pregledan popis lomljivih proizvoda i njihovih lokacija.

## Rezultat Upita

	123 product_id ▼	ABC product_name ▼	ABC warehouse_name ▼	ABC warehouse_address ▼

U skladišnom sistemu jos ne postoji niti jedan proizvod koji je označen sa karakteristikom "lomljivo".

## 7.15. Upit 15

## 8. Korištene metode i tehnologije

Za izradu projekta koristio sam poprilično širok spektar alata i tehnologija. Većinom su to bili potpuno besplatni i Open Source alati primarno namjenjeni za Linux. Iznimka tome je bio MySQL Workbench preko kojega sam generirao EER dijagram, a za kojega nisam našao adekvatno dobru zamjenu koja je za Linux Fedoru.

Za izradu Entity Diagram-a koristio sam Draw.io, za planiranje i izradu entiteta prvotno sam koristio JPA u Java Spring Boot-u, dok je sam sustav, odnosno baza podataka izrađena u DBeaver CE-u.

Prve testne podatke sam generirao direktno preko MySQL konzole u DBeaver-u. Za generiranje velike količine realnih podataka koristio sam Python i biblioteku "faker". Za programatički unos tih podataka koristio sam Python biblioteku "mysql.connector"

Primjer jednih od skripti:

```
1 from faker import Faker
2 import mysql.connector
3
4
5
6 fake = Faker()
```

```

7
8
9
10 conn = mysql.connector.connect(
11     database="warehouse_management",
12     user="root",
13     password="030897",
14     host="localhost",
15     port="3306"
16 )
17
18
19 with conn.cursor() as cur:
20     for _ in range(100):
21         first_name = fake.first_name()
22         last_name = fake.last_name()
23         email = fake.email()
24         phone = fake.phone_number()
25         country = fake.country()
26         #zip = fake.zipcode()
27         city = fake.city()
28
29         cur.execute(
30             "INSERT INTO customer (first_name, last_name, email, phone, country,
zip, city) VALUES (%s, %s, %s, %s, %s, %s, %s)",
31             (first_name, last_name, email, phone, country, "42184858", city)
32         )
33     for _ in range(50):
34         address = fake.address()
35         city = fake.city()
36         state = fake.state()
37         #zip = fake.zipcode()
38         country = fake.country()
39
40         cur.execute(
41             "INSERT INTO shipping_address (address, city, state, zip, country)
VALUES (%s, %s, %s, %s, %s)",
42             (address, city, state, "90976", country)
43         )
44
45     conn.commit()

```

## 9. Zaključak

Za lokalno verzioniranje projekta koristio sam git, te za spremanje remo kopije projekta korišten je GitHub.

Za izradu projektne dokumentacije koristio sam markdown editor Obsidian. Sama dokumentacija formatirana je kombinacijom markdown sintakse i HTML jezika koji sam koristio za funkcionalnosti koje bi bilo teže izvesti

samim Markdown-om, kao sto je forsiranje loma stranice (page break) u Obsidian-u, centriranje teksta, i specificiranje boje teksta paragrafa.

U ovom projektu, cilj mi je bio dizajnirati i implementirati bazu podataka za upravljanje skladištem. Trudio sam se obuhvatiti ključne aspekte poslovanja skladišta, uključujući upravljanje proizvodima, artiklima, narudžbama, kupcima i samim skladištima. Nastojao sam dizajnirati bazu podataka koja bi omogućila učinkovito praćenje zaliha, obradu narudžbi i optimizaciju skladišnih operacija. Tijekom razvoja projekta, suočio sam se s različitim izazovima, poput modeliranja složenih odnosa između entiteta i osiguravanja integriteta podataka. Kroz proces planiranja i iterativnog dizajna, radio sam na stvaranju strukture baze podataka koja bi bila robusna i skalabilna. Svjestan sam da uvijek postoji prostor za poboljšanje i proširenje funkcionalnosti baze podataka. U budućnosti, mogao bih razmotriti integraciju s drugim sustavima, poput sustava za upravljanje transportom ili sustava za praćenje pošiljaka u stvarnom vremenu. Također, planiram izraditi i provesti implementacije napredne analitike podataka kako bih potencijalno dobio korisne uvide o trendovima zaliha, obrascima narudžbi i ponašanju kupaca. Unatoč izazovima s kojima sam se suočio tijekom razvoja projekta, nastojao sam postići postavljene ciljeve i stvoriti temelj za upravljanje skladišnim operacijama. Baza podataka koju sam dizajnirao ima za cilj omogućiti centralizirano skladištenje i pristup ključnim informacijama, s namjerom olakšavanja donošenja odluka i poboljšanja cjelokupne učinkovitosti poslovanja skladišta. Ovaj projekt mi je pružio vrijedne spoznaje i iskustvo u dizajniranju i implementaciji baze podataka u stvarnom poslovnom kontekstu. Vjerujem da će mi stečeno znanje i vještine biti korisni u budućim projektima i profesionalnom razvoju, te se nadam da sam uspio točno opisati dizajn i funkcionalnosti informacijskog sustava za upravljanje skladištima.