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

„Sustav upravljanja hotelom“

Dokumentacija projekta iz kolegija Baze podataka I

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

**Izvođač**: Romeo Šajina, mag. inf.



Autori projekta su studenti I.godine prijediplomskog sveučilišnog studija informatike (online)

1. **Iva Batur**
2. **Lea Beletić**
3. **Tanja Gattin-Zebić**
4. **Marta Kralj**
5. **Marko Valečić**  
   (JMBAG: 0135255772)

U Puli, 31.svibnja 2024. godine

Sadržaj

[1. Uvod 3](#_Toc167909886)

[2. Opis poslovnog procesa 4](#_Toc167909887)

[1. Upravljanje rezervacijama 4](#_Toc167909888)

[2. Upravljanje gostima 4](#_Toc167909889)

[3. Upravljanje zaposlenicima 4](#_Toc167909890)

[4. Upravljanje financijama 4](#_Toc167909891)

[5. Upravljanje uslugama 4](#_Toc167909892)

[6. Upravljanje održavanjem 5](#_Toc167909893)

[7. Upravljanje restoranima 5](#_Toc167909894)

[8. Upravljanje skladištem 5](#_Toc167909895)

[9. Upravljanje rasporedom čišćenja 5](#_Toc167909896)

[3. Entity Relationship (ER) dijagram 6](#_Toc167909897)

[4. Veze entiteta prema ER dijagramu 7](#_Toc167909898)

[5. Sheme relacijskog modela 9](#_Toc167909899)

[6. EER dijagram (MySQL Workbench) 10](#_Toc167909900)

[7. Pokretanje SQL tablice 11](#_Toc167909901)

[8. SQL tablice 12](#_Toc167909902)

[8.1 Tablica radnik 12](#_Toc167909903)

[8.2 Tablica skladiste 13](#_Toc167909904)

[8.3 Tablica radnik\_skladiste 14](#_Toc167909905)

[8.4 Tablica dobavljac 14](#_Toc167909906)

# Uvod

Tijekom razvoja našeg sustava upravljanja hotelom, projekt je značajno napredovao, postao je daleko opsežniji i detaljniji od prvobitne misli. Ovaj napredak je dokaz našeg rasta i boljeg razumijevanja tehnologija i tema s kojima radimo. Kako smo dublje ulazili u razvoj baze podataka, koncepti su nam postali jasniji.

Za djeljenje koda i podataka koristili smo GitHub. Za komunikaciju smo koristili WhatsApp poruke, dok smo za sastanke i razgovore koristili Zoom. ER dijagrame izrađivali smo u Lucidchartu, dok smo za pisanje shema, upita te generiranje automatskih EER dijagrama putem reverse enginering opcije koristili MySQL Workbench. Generiranje podataka ostvareno je kombinacijom programranja u Pythonu, korištenjem random data generatora, pomoći ChatGPT-a teručnim unosom i pregledavanjem podataka.

Naša baza podataka i poslovni procesi pokrivaju sve aspekte hotelskog poslovanja. Evidentiramo podatke o gostima, zaposlenicima, sobama, rezervacijama, restoranima, računima, zahtjevima održavanja, sadržajima, uslugama, plaćanjima, dobavljačima, skladištu, događajima, recenzijama, rasporedima čišćenja, smjenama radnika i radnim mjestima. Ovaj sustav omogućuje sveobuhvatno upravljanje hotelskim operacijama, od prijema gostiju i rezervacija do financijskog upravljanja i održavanja objekta.

Svi osobni podaci u sustavu su generirani nasumično, te je svaka sličnost s pravim podacima slučajna. Trudili smo se osigurati točnost podataka, kao što su opisi usluga i sadržaja, ali ne možemo jamčiti za njihovu potpunu preciznost. U sljedećim poglavljima ćemo predstaviti naš dijagram, detaljno opisati sve relacije te objasniti proces generiranja podataka.

# Opis poslovnog procesa

Naš sustav upravljanja hotelom obuhvaća sve ključne poslovne procese potrebne za učinkovito i uspješno vođenje hotela. Ti procesi uključuju upravljanje rezervacijama, gostima, zaposlenicima, financijama, uslugama, održavanjem, restoranima i skladištem. U nastavku je detaljan opis svakog od ovih poslovnih procesa:

## Upravljanje rezervacijama

Proces započinje kada gost napravi rezervaciju, odnosno kada radnik sve potrebne podatke unese u sustav. Rezervacija uključuje podatke kao što su datumi prijave i odjave, broj gostiju, tip sobe i posebne zahtjeve. Sustav automatski provjerava dostupnost soba i potvrđuje rezervaciju. Svaka potvrđena rezervacija povezuje se s odgovarajućim gostom i sobom.

## Upravljanje gostima

Podaci o gostima prikupljaju se tijekom procesa rezervacije i prijave. Ovi podaci uključuju osobne informacije, kontakte i povijest boravka. Sustav omogućava praćenje preferencija gostiju, što omogućava personaliziranu uslugu i bolju korisničku podršku. Nakon boravka, gost može ostaviti recenziju koja se također pohranjuje u sustavu.

## Upravljanje zaposlenicima

Podaci o zaposlenicima uključuju osobne informacije, radne pozicije, radne smjene i povijest zapošljavanja. Sustav omogućava vođenje evidencije o radnim smjenama, dodjeljivanje zadataka i praćenje učinkovitosti rada. Radnici su povezani s radnim mjestima, smjenama i odgovornim područjima poput održavanja ili čišćenja.

## Upravljanje financijama

Financijski procesi uključuju izdavanje računa za boravak, usluge i restoranske troškove. Računi su povezani s gostima i rezervacijama, a plaćanja se evidentiraju zajedno s načinom plaćanja. Sustav omogućava generiranje financijskih izvještaja koji pomažu u praćenju prihoda i troškova.

## Upravljanje uslugama

Hotel nudi razne dodatne usluge kao što su spa, fitness, najam vozila i organizacija izleta. Sve usluge su evidentirane u sustavu, zajedno s cijenama i opisima. Gost može rezervirati usluge tijekom boravka, a troškovi se dodaju na njegov račun.

## Upravljanje održavanjem

Sustav omogućava prijavu i praćenje zahtjeva za održavanje. Kada gost ili zaposlenik prijavi kvar ili potrebu za održavanjem, zahtjev se evidentira s opisom problema, datumom prijave i statusom. Zahtjevi se dodjeljuju odgovornim radnicima na temelju njihove radne pozicije i dostupnosti.

## Upravljanje restoranima

Hotel može imati više restorana, a svaki restoran vodi evidenciju o gostima, narudžbama i troškovima. Gost može objedovati u restoranu i troškovi se dodaju na njegov hotelski račun. Restoranski računi su povezani s gostima i centralnim financijskim sustavom hotela.

## Upravljanje skladištem

Sustav omogućava vođenje evidencije o zalihama u skladištu, uključujući informacije o dobavljačima, nabavi i stanju zaliha. Zaposlenici odgovorni za skladište evidentiraju provjere zaliha i narudžbe dobavljačima, što omogućava održavanje optimalne razine zaliha.

## Upravljanje rasporedom čišćenja

Raspored čišćenja soba i zajedničkih prostora upravlja se putem sustava. Zaposlenici zaduženi za čišćenje dobivaju dnevne zadatke, a status čišćenja soba se ažurira u stvarnom vremenu. Ovo osigurava da su sve sobe spremne za nove goste u pravom trenutku.

Ovi poslovni procesi integrirani su u jedinstveni sustav koji omogućava učinkovito upravljanje hotelom, poboljšava korisničko iskustvo i optimizira radne procese. U sljedećim poglavljima detaljno ćemo opisati naš ER dijagram, sve relacije između tablica i način generiranja podataka.

# Entity Relationship (ER) dijagram

ER dijagram našeg sustava za upravljanje hotelom prikazuje glavne entitete kao što su gosti, zaposlenici, sobe, rezervacije, računi, usluge i restorani, te njihove međusobne odnose. Korišteni su različiti tipovi odnosa, uključujući jedan-na-jedan, jedan-na-više i više-na-više. Dijagram jasno definira kako su entiteti povezani, na primjer, jedan gost može imati više rezervacija (1), dok jedna rezervacija može biti povezana s jednim računom (N:1) i jednom sobom (N:1). Ovaj dijagram omogućuje jednostavno razumijevanje i implementaciju strukture baze podataka.

# Veze entiteta prema ER dijagramu

|  |  |  |
| --- | --- | --- |
| Veze između entiteta | Kardinalnost | Opis |
| radnik - skladište | Many-to-Many | Jedan radnik provjerava više skladišta, dok jedno skladište može provjeravait više radnika. |
| skladište - dobavljač | Many-to-Many | U jedno skladište dostavlja više dobavljača, dok jedan dobavljač može dostavljati u više skladišta. |
| radnik – radno\_mjesto | One-to-Many | Jedan radnik radi na jednom radnom mjestu, na jednom radnom mjestu može raditi više radnika. |
| radnik – smjena\_radnika | Many-to-Many | Jedan radnik može raditi u više smjena, dok u jednoj smjeni može raditi više radnika. |
| radnik - raspored\_čišćenja | Many-to-Many | Jedan radnik može raditi prema više rasporeda čišćenja, također prema jednom rasporedu može čisiti više radnika. |
| radnik – rezervacija | One-to-Many | Radnik može napraviti više rezervacija, dok tu istu rezervaciju trenutno može napaviti samo jedan radnik. |
| radnik – restran | One-to-Many | U jendnom restoranu može raditi više radnika, dok jedan radnik može raditi samo u više restoranu. |
| rezrevacija - recenzija | One-to-Many | Jedna rezervacija može dobiti više recenzija, dok jedna recenzija uvijek pripada samo jednoj rezervaciji. |
| rezervacija - racun | One-to-One | Rezervacija može imat samo jedan račun, dok taj isti račun može pripadati samo toj rezervaciji. |
| rezervacija - gost | One-to-Many | Gost može napraviti više rezervacija, ali jedna rezervacija može pripadati samo jednom gostu, tj gostu koji rezervira. |
| soba – raspored\_ciscenja | Many-to-Many | Svaka soba se može čisiti prema više rasporeda, a svaki raspored u sebi može imati više soba |
| soba – zahtjev\_odrzavanja | Many-to-Many | Soba može imati više zahtjeva za održavanje, dok jedan zahjev može pripadati samo jednoj sobi |
| zahtjev\_odrzavanja - gost | One-to-Many | Jedan gost može podnijeti više zahtjeva za održavanje, dok jedan zahtjev se odnosi samo na tog gosta. |
| soba - sadrzaj | Many-to-Many | Soba može imati više sadržaja, isto tako jedan sadržaj se može dodijeliti na više soba |
| racun - usluge | One-to-Many | Jedan racun moze sadržavati više usluga, dok ista usluga može biti na više različitih računa |
| racun - placanje | One-to-Many | Svaki račun se može platiti na više načina |
| gost – restoran | Many-to-Many | Jedan restoran može korisiti više gostiju, dok jedan gost može posječivati više restorana. |

# Sheme relacijskog modela

**radnik**(id\_radnik, ime, prezime, odjel, telefon, email, datum\_zaposlenja, id\_radno\_mjesto, id\_raspored\_ciscenja)  
**skladiste**(id\_skladiste, odjel, naziv, kolicina, datum\_nabave)   
**radnik\_skladiste**(id\_radnik, id\_skladiste, datum\_provjere, napomena)   
**dobavljac**(id\_dobavljac, naziv, kontakt\_osoba, telefon, email)   
**skladiste\_dobavljac**(id\_skladiste, id\_dobavljac, datum\_dostave, napomena)   
**radno\_mjesto**(id\_radno\_mjesto, naziv, opis, odjel)   
**smjena\_radnika**(datum, vrijeme\_pocetka, vrijeme\_zavrsetka)   
**radnik\_smjena\_radnika**(id\_radnik, id\_smjena, vrijeme\_pocetka, vrijeme\_zavrsetka, pozicija)  
**raspored\_ciscenja**(datum, status)   
**soba**(id\_soba, broj\_sobe, tip, opis, cijena\_nocenja, id\_raspored\_ciscenja)   
**zahtjev\_odrzavanja**(id\_zahtjev, opis, datum\_zahtjeva, status, id\_soba, id\_gost)   
**sadrzaj**(id\_sadrzaj, naziv, opis)   
**soba\_sadrzaj**(id\_soba, id\_sadrzaj, kolicina)   
**rezervacija**(id\_rezervacija, datum\_prijave, datum\_odjave, broj\_gostiju, id\_gost, id\_radnik\_id\_racun)  
**recenzija**(id\_recenzija, datum, ocjena, komentar, id\_gost, id\_rezervacija   
**gost**(id\_gost, ime, prezime, datum\_rodenja, adresa, telefon, email)   
**racun**(id\_racun, datum, iznos)  
**usluge**(id\_usluga, naziv, opis, cijena)   
**racun\_usluge**(id\_racun, id\_usluga, kolicina)   
**vrsta\_placanja**(id\_placanje, nacin\_placanja)   
**restoran**(id\_restoran, ime, tip, lokacija, radno\_vrijeme)  
**racun\_restoran**(id\_ racun\_restoran, datum, cijena, id\_restoran)   
**gost\_restoran**(id\_gost, id\_restoran, status)  
**racun\_vrsta\_placanja**(id\_racun, id\_placanje)

# EER dijagram (MySQL Workbench)

# Pokretanje SQL tablice

# SQL tablice

## Tablica radnik

Tablica radnik služi za pohranu informacija o zaposlenicima hotela. Sadrži atribute: id\_radnik, ime, prezime, odjel, telefon, email, datum\_zaposlenja, id\_radno\_mjesto i id\_raspored\_ciscenja. Atribut id je tipa INT i koristi se za jedinstveno identificiranje svakog zaposlenika, čineći ga PRIMARY KEY atributom. Postavljen je na AUTO\_INCREMENT, što osigurava automatsko povećanje vrijednosti pri svakom novom unosu i time osigurava jedinstvenost svakog zapisa.

Atributi ime i prezime su tipa VARCHAR, što omogućava pohranu imena i prezimena zaposlenika u tekstualnom formatu. Atribut odjel je također tipa VARCHAR i koristi se za pohranu naziva odjela u kojem zaposlenik radi, omogućavajući fleksibilnost pri unosu različitih odjela. Atribut telefon je tipa VARCHAR, što omogućava unos različitih formata telefonskih brojeva, uključujući međunarodne brojeve.Atribut email je tipa VARCHAR i ima ograničenje UNIQUE, čime se osigurava da svaki zaposlenik ima jedinstvenu email adresu unutar baze podataka. Atribut datum\_zaposlenja je tipa DATE, omogućavajući pohranu datuma kada je zaposlenik počeo raditi u hotelu. Atribut id\_radno\_mjesto je tipa INT i koristi se kao strani ključ koji povezuje zaposlenika s tablicom radno\_mjesto, definirajući njegovu radnu poziciju.Atribut id\_raspored\_ciscenja je također tipa INT i koristi se kao strani ključ koji povezuje zaposlenika s tablicom raspored\_ciscenja, što omogućava praćenje odgovornosti za čišćenje i održavanje.

Primarni ključ ove tablice je id, a jedinstveno ograničenje je postavljeno na atribut email. Tablica također sadrži dva strana ključa, id\_radno\_mjesto i id\_raspored\_ciscenja, čime se osigurava integritet podataka između tablica. Neki atributi su označeni kao NOT NULL, što znači da moraju biti uneseni za svaki unos.

CREATE TABLE radnik (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 ime VARCHAR(50) NOT NULL,  
 prezime VARCHAR(50) NOT NULL,  
 odjel VARCHAR(50),  
 telefon VARCHAR(20),  
 email VARCHAR(50) NOT NULL UNIQUE,  
 datum\_zaposlenja DATE NOT NULL,  
 id\_radno\_mjesto INT,  
 id\_raspored\_ciscenja INT,  
 FOREIGN KEY (id\_radno\_mjesto) REFERENCES radno\_mjesto(id\_radno\_mjesto),  
 FOREIGN KEY (id\_raspored\_ciscenja) REFERENCES  
 raspored\_ciscenja(id\_raspored\_ciscenja)  
);

## Tablica skladiste

Tablica skladiste sadrži evidenciju potrošne robe za potrebne hotelu. Tablica radnik sadrži atribute: id\_skladiste, odjel, naziv, kolicina i datum\_nabave. Atribut id\_skladiste je tipa INT jer prima brojčanu vrijednost i koristi se za jedinstveno identificiranje svake skladišne jedinice, postavljen je za PRIMARY KEY kako bi imali jedinstvenu oznaku svakog skladišta. Dodajemo AUTO\_INCREMENT, što omogućava automatsko povećanje vrijednosti pri svakom novom unosu, osiguravajući jedinstvenost svakog zapisa.

Atribut odjel je tipa VARCHAR jer prima znakovnu vrijednost i koristi se za označavanje odjela kojem skladišna jedinica pripada, što omogućava kategorizaciju i organizaciju skladišnih jedinica po odjelima. Ne možemo unaprijed znati koliko znakova će sadržavati, no možemo postaviti maksimalan mogući broj znakova što smo u ovom slučaju postavili na 50. S druge strane atribut naziv je također tipa VARCHAR, no tu smo ograničili znakove na 100, a sadrži naziv pojedine robe. Atribut kolicina je tipa INT koji bilježi količinu dostupnih artikala ili proizvoda na skladištu. Ovaj atribut omogućuje praćenje trenutne zalihe stavke. Atribut datum\_nabave je stupac tipa DATE koji bilježi datum kada je stavka nabavljena ili dodana u skladište. Ovaj atribut pruža informacije o vremenu kada je stavka postala dostupna u skladištu.

Primarni ključ tablice je id\_skladiste, osiguravajući jedinstvenost svake stavke u skladištu. Osim toga, stupac naziv je označen kao NOT NULL kako bi se osiguralo da svaka stavka ima definiran naziv.

CREATE TABLE skladiste (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 odjel VARCHAR(50),  
 naziv VARCHAR(100) NOT NULL,  
 kolicina INT,  
 datum\_nabave DATE

);

## Tablica radnik\_skladiste

Tablica radnik\_skladiste je tablica napravljena kao nusprodukt kardinalnosti „više na više“ između tablica radnik i skladiste.

Sadrži atribute kao što su id\_radnik i id\_skladiste, koji predstavljaju strane ključeve koji povezuju zaposlenika i skladište.  
  
Atribut datum\_provjere je tipa DATE a, atribut napomena je tipa TEXT kako napomena nebi imala ograničenja u znakovima. U ovaj atribut se unose eventualne opaske ili potvrda provjere skladišta.

CREATE TABLE radnik\_skladiste (  
 id\_radnik INT,  
 id\_skladiste INT,  
 datum\_provjere DATE,  
 napomena TEXT,  
 PRIMARY KEY (id\_radnik, id\_skladiste),  
 FOREIGN KEY (id\_radnik) REFERENCES radnik(id\_radnik),  
 FOREIGN KEY (id\_skladiste) REFERENCES skladiste(id\_skladiste)

);

## Tablica dobavljac

Tablica dobavljac služi za pohranu informacija o dobavljačima hotela. Svaki dobavljač ima jedinstveni identifikator koji se automatski povećava. Ostali atributi su naziv dobavljača koji je tipa VARCHAR i ima ogranicenje na 100 znakova, što je sasvim dovoljno za potrebe naziva. Ukoliko postoje neke nejasnoće s dobavljačima posjedujemo atribut kontakt\_osoba odnosno ime osobe te telefon za uspostavu poziva. Kako bi sve potrebne dokumente, i po potrebi posiljke, mogli razmjenjivati, definirali smo atribut email te ga odredili kao VARCHAR i postavili UNIQUE za jedinstvenu vrijednost.

CREATE TABLE dobavljac (  
 id\_dobavljac INT AUTO\_INCREMENT PRIMARY KEY,  
 naziv VARCHAR(100) NOT NULL,  
 kontakt\_osoba VARCHAR(100),  
 telefon VARCHAR(20),  
 email VARCHAR(50) UNIQUE,  
 adresa VARCHAR(255),  
 napomena TEXT

);

## Tablica skladiste\_dobavljac

Ova tablica omogućuje povezivanje skladišta s dobavljačima, pružajući informacije o datumu dostave robe ili materijala određenog dobavljača u određeno skladište. Svaki zapis u tablici definira vezu između određenog skladišta i određenog dobavljača, čime se olakšava praćenje opskrbe robe ili materijala unutar hotela.

CREATE TABLE skladiste\_dobavljac (  
 id\_skladiste INT,  
 id\_dobavljac INT,  
 datum\_dostave DATE,  
 napomena TEXT,  
 PRIMARY KEY (id\_skladiste, id\_dobavljac),  
 FOREIGN KEY (id\_skladiste) REFERENCES skladiste(id\_skladiste),  
 FOREIGN KEY (id\_dobavljac) REFERENCES dobavljac(id\_dobavljac)

);

## Tablica radno\_mjesto

Tablica radno mjesto predstavlja skup informacija o različitim radnim mjestima. Svako radno mjesto ima svoj identifikator, automatski dodijeljen. Tablica sadrži atribut naziv tipa VARCHAR koji sadrzi informacije radnog mjesta. Atribut opis koji je tipa TEXT i u njega navaodimo opis posla koji radnik mora obavljati u svoje radno vrijeme. Odjel u kojem se nalazi radno mjesto omogućuje organizaciju osoblja unutar hotela prema različitim funkcionalnim područjima.

CREATE TABLE radno\_mjesto (  
 id\_radno\_mjesto INT AUTO\_INCREMENT PRIMARY KEY,  
 naziv VARCHAR(100) NOT NULL,  
 opis TEXT,  
 odjel VARCHAR(50)

);

## Tablica smjena\_radnika

Tablica smjena\_radnika predstavlja skup informacija o radnim smjenama zaposlenika u hotelu. Svaka smjena ima svoj jedinstveni identifikator, automatski dodijeljen i povećan s dodavanjem novog zapisa. Ostali atributi uključuju: datum tipa DATE, koji označava datum kada se smjena događa; vrijeme\_pocetka tipa TIME, koje označava vrijeme početka smjene; te vrijeme\_zavrsetka isto tipa TIME, koje označava vrijeme završetka smjene. Ova tablica omogućuje precizno praćenje rasporeda radnih smjena za osoblje hotela, olakšavajući organizaciju rada i pružajući jasne informacije o vremenima prisutnosti zaposlenika.

CREATE TABLE smjena\_radnika (  
 id\_smjena INT AUTO\_INCREMENT PRIMARY KEY,  
 datum DATE,  
 vrijeme\_pocetka TIME,  
 vrijeme\_zavrsetka TIME

);

## Tablica radnik\_smjena\_radnika

Tablica radnik\_smjena\_radnika predstavlja rezultat odnosa “više na više” između tablica radnik i smjena\_radnika, omogućujući povezivanje zaposlenika i radnih smjena unutar hotela.

Ovaj model omogućuje praćenje povezanosti između zaposlenika i njihovih radnih smjena, što olakšava organizaciju rasporeda rada i upravljanje prisutnošću zaposlenika tijekom smjena. Tablica sadrži atribute koji identificiraju zaposlenika (id\_radnik) i smjenu (id\_smjena), te dodatne informacije o vremenima početka i završetka smjene za određenog zaposlenika.

CREATE TABLE radnik\_smjena\_radnika (  
 id\_radnik INT,  
 id\_smjena INT,  
 vrijeme\_pocetka TIME,  
 vrijeme\_zavrsetka TIME,  
 PRIMARY KEY (id\_radnik, id\_smjena),  
 FOREIGN KEY (id\_radnik) REFERENCES radnik(id\_radnik),  
 FOREIGN KEY (id\_smjena) REFERENCES smjena\_radnika(id\_smjena)

);

## Tablica raspored\_ciscenja

Ova tablica omogućuje praćenje rasporeda čišćenja u hotelu. Svaki raspored čišćenja ima svoj jedinstveni identifikator (id\_raspored\_ciscenja), koji se automatski dodjeljuje i povećava s dodavanjem novog zapisa. Atributi uključuju datum tipa DATE, kada je raspored čišćenja planiran te atribut status, tipa VARCAHR, koji opisuje trenutno stanje rasporeda čišćenja (npr. "u tijeku", "završeno", "odgođeno" itd.).

Ova tablica omogućuje efikasno upravljanje rasporedom čišćenja u hotelu, pružajući jasne informacije o rasporedu i statusu čišćenja za svaki dan.

CREATE TABLE raspored\_ciscenja (  
 id\_raspored\_ciscenja INT AUTO\_INCREMENT PRIMARY KEY,  
 datum DATE,  
 status VARCHAR(50)

);

## Tablica soba

Tablica soba pohranjuje informacije o sobama unutar hotela, a svaki atribut je detaljno opisan u nastavku:  
  
Atribut “id\_soba” je jedinstveni identifikator za svkau sobu. Tip atributa je INT, a atribut se automatski povećava sa AUTO\_INCREMENT kako bi se osigurala jedinstvenost svake osobe. Atribut “id\_soba” je također primarni ključ (PRIMARY KEY) tablice.

Atribut “broj\_sobe” je broj sobe unutar hotela. Tip atributa je VARCHAR(10) što omogućuje pohranu brojeva soba u obliku niza znakova, obzirom da brojevi soba mogu sadržavati i slova(npr. “101A”). Ovo polje je obavezno NOT NULL

Atribut “tip\_sobe” je tip sobe koji može biti na primjer, jednokrevetna, dvokrevetna, apartman ili slično. Tip atributa je je VARCHAR(50). Ovo polje je također obavezno, odnosno NOT NULL.

“opis” sobe, pruža dodatne informacije o značajkama i pogodnostima sobe. Tip ovog atributa je TEXT, koji omogućuje pohranu dužih tekstualnih opisa.

Atribut “cijena\_noćenja” odnosi se na cijenu noćenja u sobi. Tip atributa smo odredili da bude DECIMAL(10, 2), što bi značilo da u tablicu možemo spremiti brojčanu vrijednost s dvije decimalne točke. Ovo polje mora biti obavezn popunjeno, NOT NULL.

Na kraju u tablici imamo atribut “id\_raspored\_ciscenja” koji nam je strani kljuc. Tip atributa je INT, a atribut je povezan s atributom “id\_raspored\_ciscenja” iz tablice “raspored\_ciscenja” putem FOREIGN KEY ograničenja. Ovo omogućuje da svaka soba bude povezana s određenim rasporedom čišćenja..

CREATE TABLE soba (  
 id\_soba INT AUTO\_INCREMENT PRIMARY KEY,  
 broj\_sobe VARCHAR(10) NOT NULL,  
 tip VARCHAR(50) NOT NULL,  
 opis TEXT,  
 cijena\_nocenja DECIMAL(10, 2) NOT NULL,  
 id\_raspored\_ciscenja INT,  
 FOREIGN KEY (id\_raspored\_ciscenja) REFERENCES  
 raspored\_ciscenja(id\_raspored\_ciscenja)

);

## Tablica zahtjev\_odrzavanja

Tablica zahtjev\_odrzavanja pohranjuje informacije o zahtjevima za održavanje u hotelu, a svaki atribut je detaljno opisan u nastavku:

Atribut “id­\_zahtjev” je jedinstveni identifikator za svaki zahtjev za održavanje. Tip atributa je INT, a atribut se automatski povećava (AUTO\_INCREMENT) kako bi se osigurala jedinstvenost svakog zahtjeva. Atribut “id\_zahtjev” je također primarni ključ (PRIMARY KEY) tablice.

Atribut “opis” detaljno opisuje problem ili radove koje treba obaviti. Tip atributa je TEXT, što omogućuje pohranu dužih tekstualnih opisa. Ovo polje je obavezno (NOT NULL).

Atribut “datum\_zahtjeva” označava datum kada je zahtjev za održavanje podnesen. Tip atributa je DATE, što omogućuje pohranu datuma u formatu godine, mjeseca i dana. Ovo polje je obavezno (NOT NULL).

Atribut “status” pokazuje trenutni status zahtjeva, koji može biti, na primjer, "u tijeku", "završeno", "odgođeno" i slično. Tip atributa je VARCHAR(50), što omogućuje pohranu različitih statusa kao niz znakova. Ovo polje je obavezno (NOT NULL).

Atribut “id\_soba” je strani ključ koji povezuje zahtjev s određenom sobom. Tip atributa je INT, a atribut je povezan s atributom “id\_soba” iz tablice soba putem FOREIGN KEY ograničenja. Ovo omogućuje praćenje koji zahtjev za održavanje se odnosi na koju sobu.

Atribut “id\_gost” je strani ključ koji povezuje zahtjev s određenim gostom. Tip atributa je INT, a atribut je povezan s atributom “id\_gost” iz tablice “gost” putem FOREIGN KEY ograničenja. Ovo omogućuje praćenje koji gost je podnio zahtjev za održavanje.

CREATE TABLE zahtjev\_odrzavanja (  
 id\_zahtjev INT AUTO\_INCREMENT PRIMARY KEY,  
 opis TEXT NOT NULL,  
 datum\_zahtjeva DATE NOT NULL,  
 status VARCHAR(50) NOT NULL,  
 id\_soba INT,  
 id\_gost INT,  
 FOREIGN KEY (id\_soba) REFERENCES soba(id\_soba),  
 FOREIGN KEY (id\_gost) REFERENCES gost(id\_gost)

);

## Tablica sdrzaj

Tablica sadrzaj pohranjuje informacije o sadržajima unutar hotela, a svaki atribut je detaljno opisan u nastavku:

Atribut “id\_sadrzaj” je jedinstveni identifikator za svaki sadržaj. Tip atributa je INT, a atribut se automatski povećava (AUTO\_INCREMENT) kako bi se osigurala jedinstvenost svakog sadržaja. Atribut id\_sadrzaj je također primarni ključ (PRIMARY KEY) tablice.

Atribut “naziv” predstavlja naziv sadržaja. Tip atributa je VARCHAR(100), što omogućuje pohranu naziva sadržaja kao niz znakova. Ovo polje je obavezno (NOT NULL).

Atribut “opis” pruža dodatne informacije o značajkama i pogodnostima sadržaja. Tip atributa je TEXT, što omogućuje pohranu dužih tekstualnih opisa. Ovo polje je također obavezno (NOT NULL).

CREATE TABLE sadrzaj (  
 id\_sadrzaj INT AUTO\_INCREMENT PRIMARY KEY,  
 naziv VARCHAR(100) NOT NULL,  
 opis TEXT NOT NULL

);

## Tablica soba\_sadrzaj

Tablica soba\_sadrzaj povezuje informacije o sadržajima dostupnim u pojedinim sobama unutar hotela, omogućujući odnos “više na više” između tablica “soba” i “sadrzaj”. Svaki atribut u tablici je detaljno opisan u nastavku:

Atribut ” id\_soba” je strani ključ koji povezuje ovu tablicu s tablicom “soba”. Tip atributa je INT, a atribut je povezan s atributom “id\_soba” iz tablice “soba” putem FOREIGN KEY ograničenja. Ovo omogućuje praćenje koji sadržaji su dostupni u kojoj sobi.

Atribut ” id\_sadrzaj” je strani ključ koji povezuje ovu tablicu s tablicom “sadrzaj”. Tip atributa je INT, a atribut je povezan s atributom “ id\_sadrzaj” iz tablice “sadrzaj” putem FOREIGN KEY ograničenja. Ovo omogućuje praćenje koji sadržaji su povezani s kojom sobom.

Atribut “kolicina” označava broj jedinica određenog sadržaja prisutnog u sobi. Tip atributa je INT i ovo polje je obavezno (NOT NULL). Ovaj atribut omogućuje precizno praćenje količine svakog sadržaja u pojedinim sobama.

Primarni ključ (PRIMARY KEY) za ovu tablicu je kombinacija atributa “id\_soba” i “id\_sadrzaj”, što osigurava jedinstvenost svakog unosa u smislu kombinacije sobe i sadržaja.

CREATE TABLE soba\_sadrzaj (  
 id\_soba INT,  
 id\_sadrzaj INT,  
 kolicina INT NOT NULL,  
 PRIMARY KEY (id\_soba, id\_sadrzaj),  
 FOREIGN KEY (id\_soba) REFERENCES soba(id\_soba),  
 FOREIGN KEY (id\_sadrzaj) REFERENCES sadrzaj(id\_sadrzaj)

);

## Tablica rezervacija