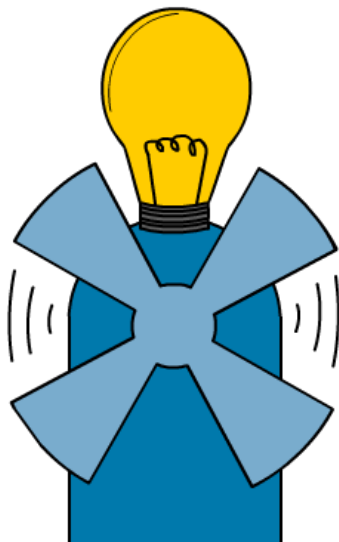


studirko.com.hr
predstavlja



Baze podataka za FPZ (31. 07. 2024.)

Za studirko.com.hr napisao:
Slaven Špigel

Uvodna riječ

Poštovani,

skripta koju čitate dio je popratnih materijala koji dolaze uz serijal video tutorijala za učenje baza podataka. Cijeli serijal „Baze podatka za FPZ“, kao i posljednju inačicu ove skripte, možete naći na web stranici studirko.com.hr u sekciji „Tutorijali“.

Iako je skripta napravljena prvenstveno za kolegij „Baze podataka“ na Fakultetu prometnih znanosti, skriptom se mogu koristiti i ostali učenici i studenti koji žele naučiti više o bazama podataka.

Skripta je trenutno samo popis svih zadataka koji su mi bili dostupni, a njezino nadopunjavanje i rješenja zadataka bit će naknadno dodani ako za to bude dovoljno interesa. Obzirom da je teško doći do kompletnih ispita, stavljani su i neki nepotpuni ispiti.

Vjerojatno je da će se skripta i materijali mijenjati kako budemo nadograđivali postojeće i dodavali nove zadatke. U naslovu skripte je datum na koji je skripta objavljena kako bi mogli pratiti imate li najnoviju inačicu skripte.

Ako nađete grešku u skripti, ako imate prijedlog, kritiku ili zadatak koji bi htjeli znati riješiti javite nam se preko kontakt forme na stranici studirko.com.

Želim Vam ugodno učenje,

Slaven Špigl

Sadržaj

Pismeni ispiti	1
Pismeni ispit #1 – grupa A (06. 02. 2014.).....	1
Pismeni ispit #2 – grupa A (06. 02. 2014.).....	4
Pismeni ispit #3 – grupa A (12.04.2016.).....	7
Pismeni ispit #4 – grupa A (16.06.2016.).....	8
Pismeni ispit #5 – grupa A (13.07.2017.).....	9
Pismeni ispit #6 – grupa B (15.02.2018.).....	10
Pismeni ispit #7 – grupa A (17.11.2018.).....	11
Pismeni ispit #8 – grupa A (21.02.2019.).....	13
Pismeni ispit #9 – grupa A (04.02.2021.).....	15
Pismeni ispit #10 – grupa A (18.02.2021.).....	17
Pismeni ispit #11 – grupa A (17.04.2021.).....	19
Pismeni ispit #12 – grupa A (28.06.2021.).....	21
Pismeni ispit #13 – grupa A (09.09.2020. i 08.07.2021.)	23
Pismeni ispit #14 – grupa A (02.09.2021.).....	25
Pismeni ispit #15 – grupa A (23.03.2022.).....	27
Pismeni ispit #16 – grupa A (23.06.2022.).....	29
Pismeni ispit #17 – grupa A (07.07.2022.).....	31
Pismeni ispit #18 – grupa A (08.09.2022.).....	33
Pismeni ispit #19 (11.07.2024.)	36
Prvi kolokviji	38
Prvi kolokvij #1.....	38
Prvi kolokvij #2 – grupa A (01. 12. 2010.)	39
Prvi kolokvij #3 – grupa A (18. 11. 2014.)	40
Prvi kolokvij #4 – grupa A (17. 11. 2015.)	41
Prvi kolokvij #5 – grupa B (17. 11. 2015.).....	42
Prvi kolokvij #6 – grupa B (06. 12. 2016.).....	43
Prvi kolokvij #7 – grupa A (21. 11. 2017.)	44
Prvi kolokvij #8 – grupa B (21. 11. 2017.).....	46
Prvi kolokvij #9 – grupa B (20. 11. 2018.).....	48
Drugi kolokviji	50
Drugi kolokvij #9 – grupa A (12. 01. 2016.).....	50
Drugi kolokvij #10 – grupa E (12. 01. 2016.)	51
Drugi kolokvij #11 – Grupa F (12. 01. 2016).....	52
Pismeni ispiti - rješenja	53

Pismeni ispit #12 – grupa A (28.06.2021.)	53
Pismeni ispit #13 – grupa A (09.09.2020. i 08.07.2021.)	54
Pismeni ispit #14 – grupa A (02.09.2021.)	55
Pismeni ispit #15 – grupa A (23.03.2022.)	56
Pismeni ispit #16 – grupa A (23.06.2022.)	57
Pismeni ispit #17 – grupa A (07.07.2022.)	63
Pismeni ispit #18 – grupa A (08.09.2022.)	68

Pismeni ispiti

Pismeni ispit #1 – grupa A (06. 02. 2014.)

1. (4 boda)

Analiza korisničkih zahtjeva i kreiranje baze podataka

Potrebno je kreirati bazu podataka koja će pamtit poddatke o vlasnicima nekretnina.

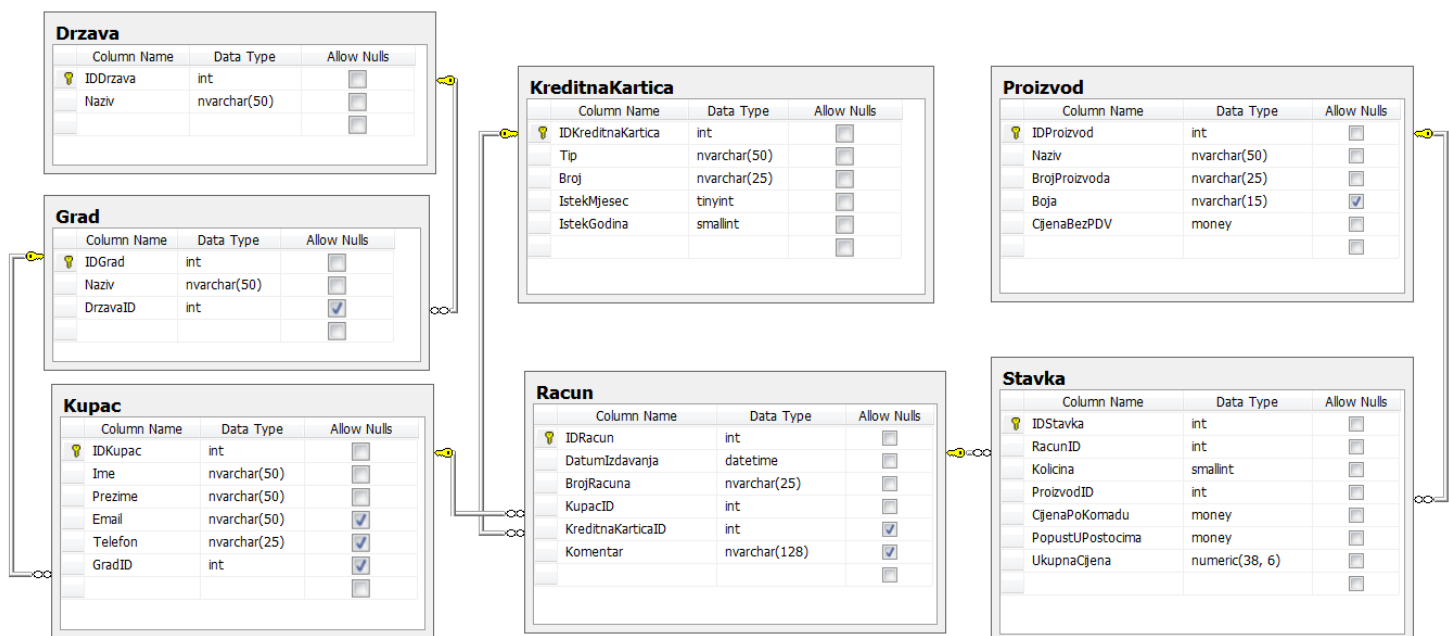
Za svaku nekretninu potrebno je pamtit adresu na kojoj se ona nalazi, kat na kojem se nalazi, površinu, te godinu gradnje koja je opcionalna za unos.

Za svakog vlasnika nekretnine potrebno je pamtit njegovo ime, prezime, adresu, grad i poštanski broj grada iz kojega dolazi, te naziv države kojoj taj grad pripada. Svaki grad pripada točno jednoj državi.

Svaki vlasnik nekretnine može posjedovati jednu ili više nekretnina dok jedna nekretnina može pripadati jednom ili više vlasnika

- kreirajte odgovarajući dijagram entiteta sa kardinalitetima atributa
- nacrtajte ER dijagram za prikaz veza među entitetima

Relacijski model baze na kojoj se izvode sljedeći upiti prikazan je na slici



2. (1 bod)

Dohvatite ukupan broj računa i preimenujte taj stupac u „Broj računa“.

3. (1 bod)

Dohvatite sve nazive i cijene proizvoda te ih sortirajte prema cijeni i prema broju proizvoda, oboje silazno.

4. (1 bod)

Obrišite proizvod čiji je ID jednak 100.

5. (1 bod)

Za svaki račun dohvatiti njegov broj i datum izdavanja, ime i prezime kupca te naziv grada iz kojega kupac dolazi.

6. (2 boda)

Za svaki mjesec isteka kreditne kartice dohvatite koliko je kartica tog tipa i taj stupac preimenujte u „Broj kartica“.

7. (2 boda)

Dohvatite brojeve svih kreditnih kartica sa kojima se nije kupovalo.

8. (2 boda)

Dohvatite sve stavke računa čija je ukupna cijena ispodprosječna i čija je boja različita od „Crvena“.

9. (2 boda)

Za svaki naziv grada dohvatite iz koje je države s tim da naziv države mora biti dohvaćen podupitom.



10. (3 boda)

Izgradite B-stablo neklasteriranog indeksa na stupcu *Boja* prema sljedećoj tablici, ako je primarni ključ postavljen na stupac pod nazivom *Id*, a kapacitet čvora u B-stablu je 2.

Napomena: potrebno je ispravno popuniti sve elemente B-stabla da bi se zadatak priznao.

Id	Naziv	Šifra	Boja
2	B Ball	BA-8327	Bijela
1	Lock Nut 16	AR-5381	Crna
5	Paint - Red	CN-6137	Crvena
7	Fork Crown	FC-3654	Ljubičasta
11	Front Derailleur Linkage	FL-2301	Narančasta
12	Guide Pulley	GP-0982	Narančasta
10	HL Nipple	HU-6280	Plava
6	Hex Nut 16	FW-9160	Roza
8	HL Grip Tape	GT-2908	Siva
13	LL Grip Tape	TT-0820	Srebrna
9	Thin-Jam Hex Nut 9	HJ-1213	Zelena
4	HL Road Rim	NN-6137	Zlatna
3	Metal Bar 2	BE-2349	Žuta

11. (1 bod)

U bazi postoje dvije tablice koje su u relaciji prikazanoj na slici ispod. Što će se dogoditi ako je uključena opcija stupnjevitog ili kaskadnog brisanja (*cascade*) kod referencijalnog integriteta, a pokušamo obrisati iz tablice *Grad* redak sa ID-em 3?

Grad		
ID	Model	DrzavaId
1	Zagreb	1
2	Maribor	2
3	Ljubljana	2

Drzava	
ID	Naziv
1	Hrvatska
2	Slovenija



Pismeni ispit #2 – grupa A (06. 02. 2014.)

1. (4 boda)

Analiza korisničkih zahtjeva i kreiranje baze podataka

Potrebno je kreirati bazu podataka koja će pamti podatke o vlasnicima vozila.

Za svako vozilo potrebno je pamti njegovu marku, model, godinu proizvodnje te boju koja je opcionalna za unos.

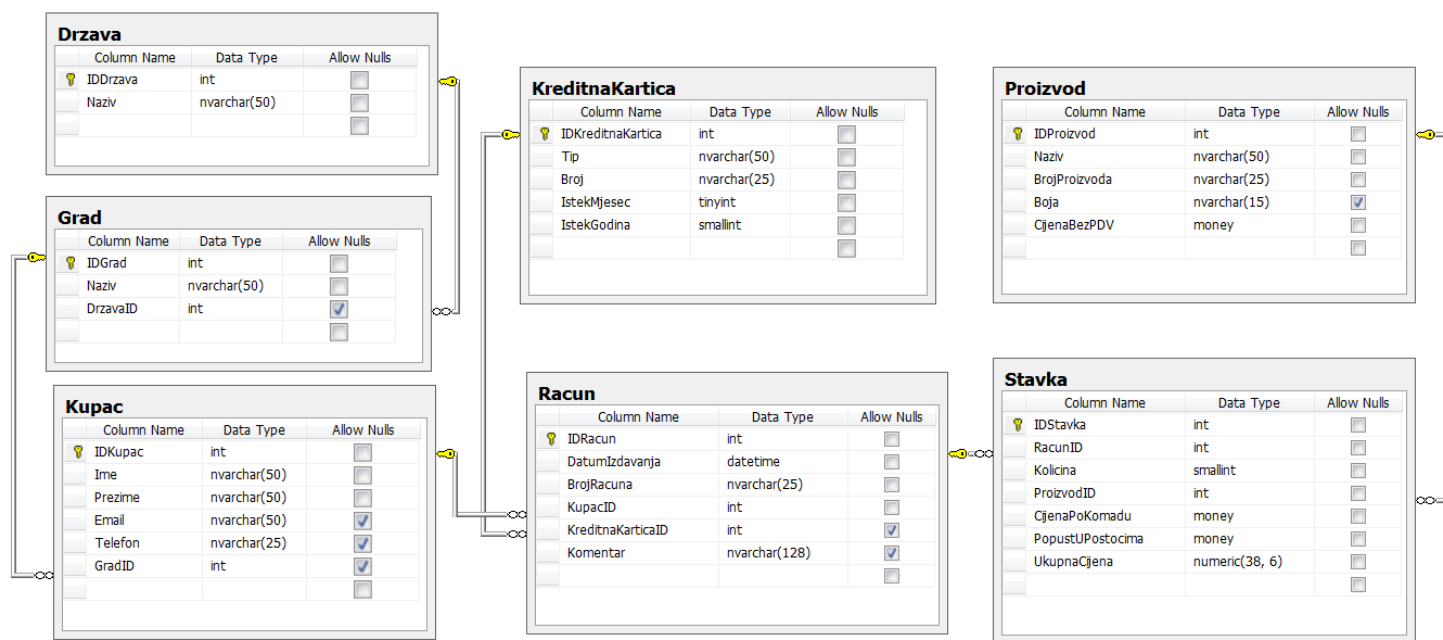
Za svakog vlasnika vozila potrebno je pamti njegovo ime, prezime, adresu, mjesto i poštanski broj mjesta iz kojega dolazi, te naziv županije kojoj to mjesto pripada.

Svako mjesto pripada točno jednoj županiji.

Svaki vlasnik vozila može posjedovati jedno ili više vozila dok jedno vozilo može pripadati jednom ili više vlasnika.

- kreirajte odgovarajući dijagram entiteta sa kardinalitetima atributa
- nacrtajte ER dijagram za prikaz veza među entitetima

Relacijski model baze na kojoj se izvode sljedeći upiti prikazan je na slici



2. (1 bod)

Dohvatite ukupan broj kupaca i preimenujte taj stupac u „Broj stupaca“.

3. (1 bod)

Dohvatite sve tipove i brojeve kreditne kartice, te ih sortirajte prema godini isteka i zatim prema mjesecu isteka oboje uzlazno.

4. (1 bod)

Obrišite kupca čiji je ID jednak 100.

5. (1 bod)

Za svakog kupca dohvatite njegovo ime i prezime, naziv grada iz kojeg dolazi te naziv države iz koje dolazi.

6. (2 boda)

Za svaku godinu isteka kreditne kartice dohvatite koliko je kartica tog tipa i taj stupac preimenujte u „Broj kartica“.

7. (2 boda)

Dohvatite nazive svih proizvoda koji nisu nikada prodani.

8. (2 boda)

Dohvatite sve stavke računa čija je ukupna cijena iznadprosječna i čija je boja proizvoda „Crna“.

9. (2 boda)

Za svaki naziv grada dohvatite iz koje je države s tim da naziv države mora biti dohvaćen podupitom.



10. (3 boda)

Izgradite B-stablo neklasteriranog indeksa na stupcu *Šifra* prema sljedećoj tablici, ako je primarni ključ postavljen na stupac pod nazivom *ID*, a kapacitet čvora u B-stablu je 4.

Napomena: potrebno je ispravno popuniti sve elemente B-stabla da bi se zadatak priznao.

Id	Naziv	Šifra	Boja
1	Lock Nut 16	AR-5381	Crna
2	B Ball	BA-8327	Bijela
3	Metal Bar 2	BE-2349	Žuta
5	Paint - Red	CN-6137	Crvena
7	Fork Crown	FC-3654	Ljubičasta
11	Front Derailleur Linkage	FL-2301	Narančasta
6	Hex Nut 16	FW-9160	Roza
12	Guide Pulley	GP-0982	Narančasta
8	HL Grip Tape	GT-2908	Siva
9	Thin-Jam Hex Nut 9	HJ-1213	Zelena
10	HL Nipple	HU-6280	Plava
4	HL Road Rim	NN-6137	Zlatna
13	LL Grip Tape	TT-0820	Srebna

11. (1 bod)

U bazi postoje dvije tablice koje su u relaciji prikazanoj na slici ispod. Što će se dogoditi ako je uključena opcija stupnjevitog ili kaskadnog brisanja (*cascade*) kod referencijalnog integriteta, a pokušamo obrisati iz tablice *Drzava* redak sa *ID*-em 2?

Grad		
ID	Model	DrzavaId
1	Zagreb	1
2	Maribor	2
3	Ljubljana	2

Drzava	
ID	Naziv
1	Hrvatska
2	Slovenija



Pismeni ispit #3 – grupa A (12.04.2016.)

ER modeliranje (8 bodova)

1. Izradite DIJAGRAM ENTITETA uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „Zadatak1.grapgm1“
2. Izradite ER DIJAGRAM uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „zadatak2,raphml“

Korisnički zahtjevi: kreirajte bazu podataka „Biciklist“ koja će pamtit i podatke o biciklima i njihovim vlasnicima te odvoženim trkama. Za bicikl je potrebno pamtit njegov naziv, serijski broj od točno 12 znakova, boju, veličinu kotača te vlasnika bicikla (biciklista). Biciklist može posjedovati više bicikala dok svaki bicikl pripada samo jednom biciklistu. Za biciklista je potrebno pamtit njegovo ime, prezime, OIB, e-mail koji je opcionalan za unos te utrke koje je biciklist odvozio. Za svaku utrku potrebno je bilježiti naziv utrke, kilometražu, datum održavanja i tip utrke. Za tip utrke potrebno je pamtit naziv i opis (do maksimalno 100 znakova). Svaka utrka može biti samo jednog tipa, ali istom tipu može pripadati više utrka.

Ubacivanje, ažuriranje podataka i dohvaćanje podataka iz jedne tablice (6 bodova)

1. U tablicu država ubacite novu državu pod nazivom 'Poljska'.
2. Svim proizvodima crvene boje smanjite cijenu bez PDV-a za 10%.
3. Dohvatite sve kupce čije ime počinje i završava slovom 'a'. Poredak ostalih slova nije bitan.
4. Dohvatite telefonski broj prvih 10 kupaca sortiranih prema Prezimenu uzlazno i GradID-u silazno.

Dohvaćanje podataka iz više tablica (5 bodova)

5. Za svaku stavku dohvatite količinu, ukupno cijenu, broj računa te naziv proizvoda koji se u njoj nalazi.
6. Za svakog kupca dohvatite koliko ima zabilježenih računa (uključujući kupce bez računa).

Postavljanje indeksa (2 boda)

7. Postavite neklasterirane indeks da ubrza upit ispod prema broju logičkih čitanja (uključivanje statistike: SET STATISTICS IO ON)

```
SELECT IDStavka, Kolicina, UkupnaCijena
FROM Stavka
WHERE Kolicina < 30 AND UkupnaCijena > 2000
```

Normalizacija baze podataka (4 boda)

8. Napišite skriptu (kreiranje novih tablica, te njihovo punjenje) koja normalizira sljedeće podatke

Ime i prezime	Godina rođenja	Kolegij 1	Kolegij 2
Miro Mirić	1987	Računalstvo	Algoritmi i programiranje
Iva Ivić	1976	Baze podataka	Napredne baze podataka
Maja Majić	1961	Optimizacija	Baze podataka



Pismeni ispit #4 – grupa A (16.06.2016.)

ER modeliranje (8 bodova)

1. Izradite DIJAGRAM ENTITETA uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „Zadatak1.grapgm1“

2. Izradite ER DIJAGRAM uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „zadatak2,raphml“

Korisnički zahtjevi: kreirajte bazu podataka „Restoran“ koja će pamtit i podatke o kuharima, gostima i jelima koja su u ponudi restorana. Za gosta je potrebno pamtit i ime, prezime, JMBAG te kontakt telefon koji je opcionalan za unos. Svaki gost mora naručiti neko jelo. Za pojedino jelo potrebno je bilježiti naziv, cijenu i kalorijsku vrijednost. Uprava restorana provodi politiku da gost može naručiti samo jedno jelo, dok jedno jelo može naručiti više gostiju. Jedno jelo pripravlja samo jedan kuhar za kojeg se pamti ime, prezime, broj licence i datum rođenja. Kuhar zna i može spremi više jela. Također je potrebno pamtit i od kojih se namirnica sastoji jelo. Pojedina namirnica se može pojavljivati u više jela i svako jelo se sastoji od više namirnica. Za namirnice je potrebno pamtit i naziv i cijenu po kilogramu.

Ubacivanje, ažuriranje podataka i dohvaćanje podataka iz jedne tablice (6 bodova)

3. U tablicu država ubacite novu državu pod nazivom 'Grčka'.

4. Svim računima bez upisanog komentara dodajte komentar 'Nema komentara'.

5. Dohvatite nazive država sa četvrtim slovom naziva 'a'. Poredak ostalih slova nije bitan.

6. Dohvatite telefonski broj prvih 10 kupaca sortiranih prema prezimenu uzlazno i GradID-u silazno.

Dohvaćanje podataka iz više tablica (5 bodova)

7. Za svaki stavku dohvatite količinu, ukupnu cijenu, broj računa te naziv proizvoda koji se u njoj nalazi.

8. Za svakog kupca dohvatite koliko ima zabilježenih računa (uključujući kupce bez računa).

Postavljanje indeksa (2 boda)

9. Postavite neklastirane indekse da ubrzate upit ispod prema broju logičkih čitanja (uključivanje statistike SET STATISTICS IO ON)

```
SELECT IDKreditnaKartica, IstekGodina, IstekMjesec  
FROM KreditnaKartica  
WHERE IstekMjesec < 6 AND IstekGodina > 2000
```

Normalizacija baze podataka (4 boda)

10. Napišite skriptu(kreiranje novih tablica, te njihovo punjenje) koja normalizira sljedeće podatke.

Ime i prezime	Godina rođenja	Kolegij 1	Kolegij 2
Miro Mirić	1987	Računalstvo	Algoritmi i programiranje
Iva Ivić	1976	Baze podataka	Napredne baze podataka
Maja Majić	1961	Optimizacija	Baze podataka



Pismeni ispit #5 – grupa A (13.07.2017.)

ER modeliranje (7 bodova)

1. Izradite DIJAGRAM ENTITETA i ER dijagram za bazu podataka „Lingvist“ koja će pamtit i podatke o jezicima, predavačima i polaznicima škole stranih jezika. Za polaznike je potrebno bilježiti ime, prezime, OIB (11 znakova), telefon neobavezan za unos i strane jezike koje je polaznik upisao. Jedan strani jezik može upisati više polaznika. Za strane jezike potrebno je bilježiti naziv, stupanj, cijenu (realan broj) i predavača koji ga predaje. Za predavača je potrebno bilježiti ime, prezime i datum rođenja. Jedan predavač može predavati više stranih jezika, a jedan strani jezik predaje isključivo jedan predavač. Svi strani jezici održavaju se u učionicama za koje je potrebno bilježiti naziv i broj sjedećih mjesta. Jedan strani jezik može se održavati u samo jednoj učionici, dok se u jednoj učionici može održavati više stranih jezika. Rješenje se predaje u datoteci pod nazivom „Zadatak1.graphml“.

Ubacivanje, ažuriranje podataka i dohvaćanje podataka iz jedne tablice (5 bodova)

Rješenje zadataka 2-10 predaje se u datoteci pod nazivom „Zadaci_2_10.sql“.

2. U tablicu proizvod dodajte proizvod sa proizvoljnim vrijednostima bez da unesete boju proizvoda. (1 bod)
3. Obrišite kupca kojem telefon počinje znamenkom 2, završava znamenkama 85 te živi u gradu s ID-em 8. (1 bod)
4. Svim stavkama koje imaju ukupnu cijenu veću od 25000, smanjite vrijednost cijene po komadu za 50%. (1 bod)
5. Dohvatite prvih 20% stavki s najvećom količinom te kojima je proizvodID između 700 i 800. (2 boda)

Dohvaćanje podataka iz više tablica, grupiranje, podupiti (7 bodova)

6. Za stavku s ID-em 121293 dohvatite cijenu po komadu, datum izdavanja računa i tip kreditne kartice s kojom je račun plaćen. (2,5 boda)
7. Dohvatite samo one države kojima pripada više od 3 grada. Potrebno je prikazati stupce IDDržava i Broj gradova. (2,5 boda)
8. Dohvatite ID stavke i količinu onih stavaka čija je količina dva puta veća od prosječne količine stavke. (2 boda)



Pismeni ispit #6 – grupa B (15.02.2018.)

2. zadatak (2 boda)

Dohvatite sve podatke proizvoda čiji naziv ne sadrži znakovni niz *set*, a predzadnje slovo mora biti *a*, *b* ili *c*.

3. zadatak (1 bod)

U tablicu kreditnih kartica ubacite novu kreditnu karticu tipa *Visa* koja ističe na sutrašnji dan s brojem kartice *2600001A*.

4. zadatak (1 bod)

Svim računima koji su izdani prvog dana u mjesecu promijenite komentar u „Prvi dan u mjesecu“.

7. zadatak (3 boda)

Dohvatite poredane ID-eve računa od onih koji u sumi imaju najveći prodanu količinu proizvoda do onih koji imaju najmanju. Potrebno je prikazati stupce IDRačun i ukupna količina.

8. zadatak (2 boda)

Korištenjem podupita dohvatite nazive proizvoda koji nisu prodani niti jedanput.

9. zadatak (2 boda)

Postavite neklasterirani indeks da što više ubrzate upit ispod prema broju logičkih čitanja. Potrebno je komentarom označiti odabir indeksa i broj logičkih čitanja, te napisati naredbu za brisanje indeksa.

```
SELECT PopustUPostocima, RacunID, IDStavka FROM Stavka  
WHERE PopustUPostocima > 0.2 AND RacunID < 46330
```

10. zadatak (3 boda)

Normalizirajte sljedeću tablicu do 3NF. Na kraju je potrebno pobrisati kreirane tablice!

Film 1	Film 2	Ime i prezime	OIB
Ono	Čudo	Mario Marić	1234
Coco	Kum	Ana Anić	2345

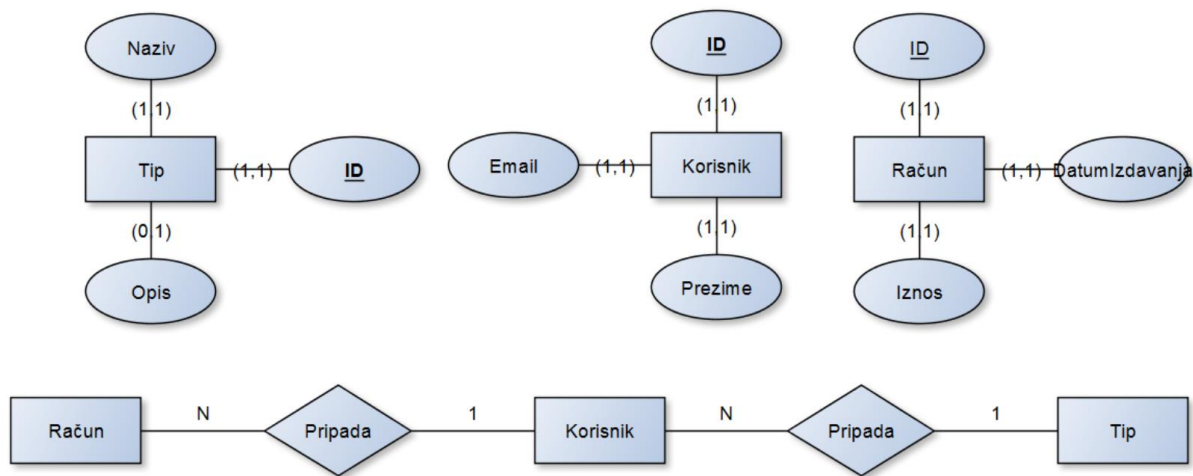


Pismeni ispit #7 – grupa A (17.11.2018.)

1. zadatak (6 bodova)

Na osnovu izrađenog ER modela prikazanog na slici 1 kreirajte relacijsku shemu baze podataka Zadatak1 u SQL Serveru (bilo pomoću GUI sučelja bilo vlastitom skriptom). Potrebno je postaviti da se DBMS sam brine oko jedinstvenih vrijednosti primarnog ključa. Potrebno je napraviti SQL skriptu koju DBMS sam generira. Rješenje se predaje u datoteci pod nazivom Zadatak1.sql. Potrebno je uzeti u obzir i sljedeće stavke:

- Opis – maksimalan broj znakova
- svaki tip ima barem jednog korisnika, no svaki korisnik ne mora pripadati u neki tip.



Slika 1. Dijagram entiteta i ER dijagram

2. zadatak (1 bod)

U tablicu kupac dodajte novog kupca Eva Ević iz grada sa ID-em 1.

3. zadatak (1 bod)

Svim stavkama čija je ukupna cijena zaokružena na dva decimalna mjesta veća od 25000 promijenite popust u postotcima na 5%.

4. zadatak (1 bod)

Obrišite kreditne kartice kojima broj kartice završava znakom A te koje ne sadrže dva uzastopna broja 1 u broju kartice.

5. zadatak (2 boda)

Dohvatite sve podatke o kreditnim karticama čiji broj započinje znakovima 55, a mjesec isteka im je u rasponu od 9 do 12.

6. zadatak (3 boda)

Dohvatite sve račune kupca sa ID-em kupca manjim od 300. U rezultatu upita prikazuju se ID računa, broj računa malim slovima te broj kreditne kartice kojom je račun plaćen. Svi stupci moraju biti odgovarajuće nazvani.



7. zadatak (3.5 bodova)

Dohvatite naziv proizvoda te maksimalan popust u postotcima odgovarajućeg proizvoda u tablici Stavka. Dohvaćaju se samo oni proizvodi koji se pojavljuju u tablici Stavka više od 10 puta. Svi stupci moraju biti odgovarajuće nazvani.

8. zadatak (2.5 boda)

Korištenjem podupita dohvatite ID računa, broj računa, datum izdavanja računa te tip kreditne kartice kojom je račun plaćen.

9. zadatak (2 boda)

Postavite neklasterirani indeks da što više ubrzate upit ispod prema broju logičkih čitanja. Potrebno je komentarom označiti odabir indeksa i broj logičkih čitanja, te napisati naredbu za brisanje indeksa.

```
SELECT IDRačun, DatumIzdavanja, KupacID FROM Račun  
WHERE DatumIzdavanja < GETDATE() AND KupacID > 16000
```

10. zadatak (3 boda)

Normalizirajte sljedeću tablicu do 3NF. Na kraju je potrebno pobrisati kreirane tablice!

Ime i prezime	OIB	Izložba 1	Izložba 2
Mario Marić	125125125	Muzej suvremene umjetnosti	Tehnički muzej
Ana Anić	785785785	Muzej iluzija	Muzej za umjetnost i obrt

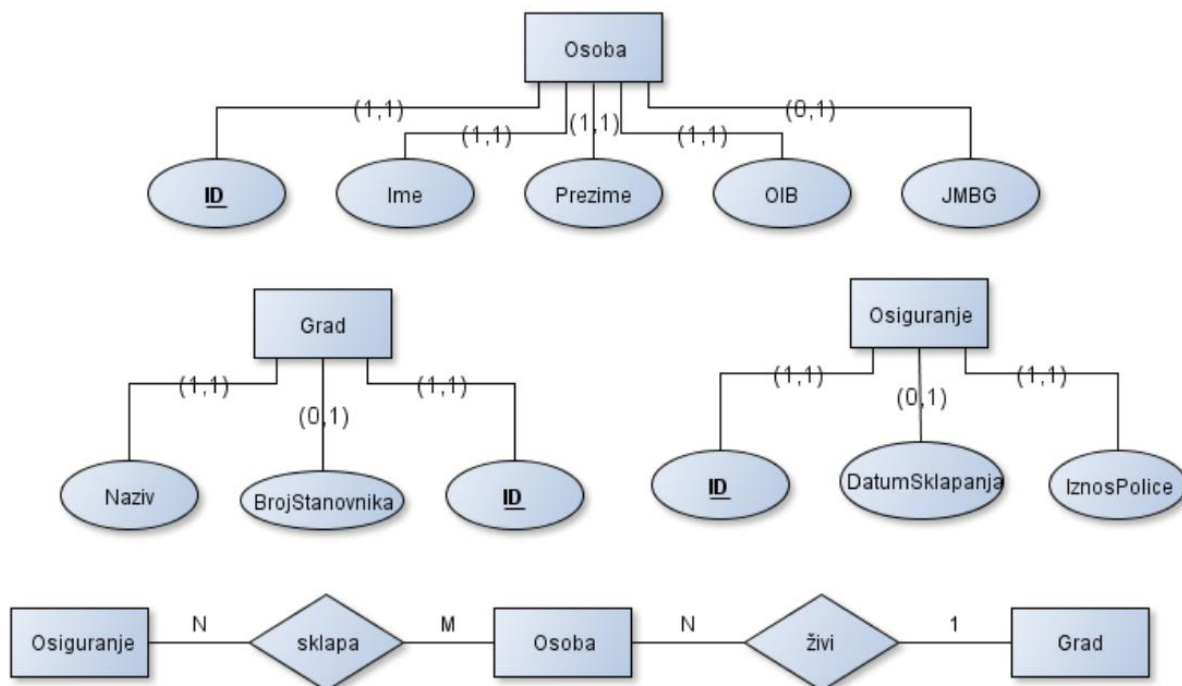


Pismeni ispit #8 – grupa A (21.02.2019.)

1. zadatak (7 bodova)

Na osnovu ER modela prikazanog na slici 1 kreirajte relacijsku shemu baze Osiguranje vlastitom skriptom u SQL jeziku. Potrebno je uzeti u obzir i sljedeće stavke:

- svaka osoba ne mora imati unesen propadajući grad
- IznosPolice – vrijednost mora biti veća od 0
- DatumSklapanja – predefiniрана vrijednost je trenutni datum



Slika 1. Dijagram entiteta i ER dijagram

2. zadatak (1 bod)

Koliko ima kupaca koji imaju različito ime i prezime. Rezultat navedite u komentaru, a mora biti vidljiva i SELECT naredba.

3. zadatak (1 bod)

U tablicu računa ubacite novi račun koji je izdan na današnji datum s brojem računa SO1234 (O je slovo, a ne broj nula) za kupca s ID-em 1 te s komentarom 'Problematičan račun!'.

4. zadatak (3 boda)

Dohvatite sve stavke kupaca čije prezime počinje znakom A. Potrebno je prikazati stupce ID stavke, cijena po komadu, komentar te prezime kupca. Cijenu po komadu potrebno je zaokružiti na jednu decimalu, a ako komentar nije poznat potrebno je ispisati 'Nema komentara!'. Svi stupci moraju biti odgovarajuće nazvani.



5. zadatak (3 boda)

Dohvatite ID države, naziv države i broj kupaca u državi onih država koje imaju više od 5000 kupaca. Svi stupci moraju biti odgovarajuće nazvani.

6. zadatak (3 boda)

Dohvatite poredane račune od najvećeg do najmanjeg ostvarenog prometa (ukupno plaćeno). Potrebno je prikazati stupce broj računa i ukupno plaćeno. Svi stupci moraju biti odgovarajuće nazvani.

7. zadatak (2.5 boda)

Korištenjem podupita za svaki grad dohvatite ID grada, naziv te odgovarajući naziv države. Svi stupci moraju biti odgovarajuće nazvani.

8. zadatak (3 boda)

Od svih proizvoda čiji naziv počinje znakom b i čija je cijena bez PDV-a veća od 0, dohvatite samo onaj najskuplji. Potrebno je prikazati stupce Naziv, IDProizvod i CijenaBezPDV-a. Potom postavite neklasterirani indeks da što više ubrzate navedeni upit prema broju logičkih čitanja. Potrebno je komentarom označiti odabir indeksa i broj logičkih čitanja, te napisati naredbu za brisanje indeksa.

9. zadatak (1.5 bod)

Unutar komentara u SQL-u nadopunite sljedeći C# kod tako da dohvatite sve stupce onih proizvoda čiji naziv počinje znakom koji se unosi preko TextBoxa naziv.

```
SqlCommand cmd = new SqlCommand(_____,sql);  
SqlDataReader reader = cmd._____;
```

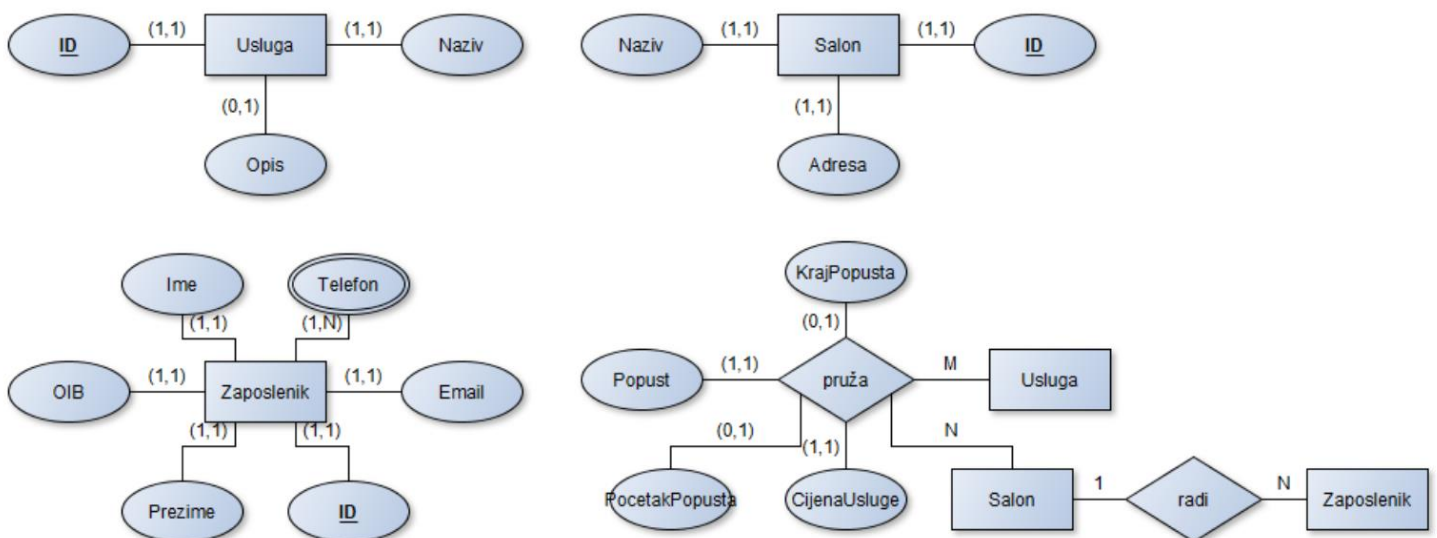


Pismeni ispit #9 – grupa A (04.02.2021.)

1. zadatak (7 bodova)

Na osnovu ER modela prikazanog na slici 1 kreirajte relacijsku shemu baze podataka Salon vlastitom skriptom u SQL jeziku. U skripti moraju biti vidljive naredbe za **kreiranje** baze podataka i tablica, te na kraju naredba za **brisanje** kreirane baze podataka. Potrebno je postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa te uzeti u obzir i sljedeće stavke:

- svaki zaposlenik mora imati unesen salon u kojem radi
- CijenaUsluge - realan broj unutar intervala [0–500]
- PocetakPopusta - predefinirana vrijednost je trenutni datum
- OIB – jedinstven



2. zadatak (1 bod)

Od svih proizvoda čiji naziv sadrži niz znakova *cap* i čija je cijena bez pdv-a veća od 0, dohvatite samo onaj najskuplji. Potrebno je prikazati stupce *Naziv*, *IDProizvod* i *CijenaBezPDV*.

3. zadatak (1 bod)

Svim računima koji imaju uneseni komentar i koji su izdani u 2001. ili 2003. godini, postavite komentar na vrijednost mjeseca u kojoj je račun izdan.

4. zadatak (3 boda)

Dohvatite tip kreditne kartice i ukupan plaćeni iznos svakim od tipa kreditnih kartica uz uvjet da se uzimaju u obzir samo one kreditne kartice koje ističu u 2005. godini. Rezultat je potrebno poredati od najvećeg do najmanjeg plaćenog iznosa te prikazati rezultat samo za one tipove koji imaju ukupan plaćeni iznos veći od 6000000. Svi stupci moraju biti odgovarajuće nazvani.



5. zadatak (3 boda)

Dohvatite one proizvode koji su prodani u više od 200 primjeraka. Stupac CijenaBezPDV potrebno je zaokružiti na jednu decimalu, a iz stupca BrojProizvoda potrebno je uzeti prva 4 znaka i spojiti ga sa stupcem Boja prikazanim velikim slovima na način da između stupaca dodate znak – (uključujući razmake), pritom je potrebno paziti da ako boja nije unesena treba za boju prikazati vrijednost NEPOZNATA. Svi stupci moraju biti odgovarajuće nazvani.

6. zadatak (3 boda)

Korištenjem podupita dohvatite sve stupce proizvoda čija je vrijednost cijene bez pdv-a barem dvostruko manja od prosječne vrijednosti cijene bez pdv-a svih proizvoda. Također, korištenjem podupita, unutar istog selecta, prikažite i stupac Zarada koji predstavlja ukupnu zaradu odgovarajućeg proizvoda, te ukoliko proizvod nije prodan niti jednom ispišite 0. Svi stupci moraju biti odgovarajuće nazvani.

7. zadatak (2 boda)

Deklarirajte varijablu ukProdano te u nju spremite ukupnu količinu svih stavaka u tablici Stavka. Na kraju ispišite rezultat.

8. zadatak (5 bodova)

- a) (2 boda) Kreirajte Windows Forms aplikaciju u kojoj se klikom na gumb pod nazivom Connect, aplikacija spaja na vašu lokalnu bazu podataka SimpleWebShop pomoću Windows autentifikacije (ako to ne radi može i SQL autentifikacija). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje 04.02.2021!*, inače u skočnom prozoru treba ispisati *Greška prilikom spajanja 04.02.2021!*. (pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient;*)
- b) (3 boda) U prethodno izrađenu aplikaciju dodajte 1 Label i 1 TextBox koji će služiti za unos niza znakova naziva proizvoda. Potrebno je dodati gumb naziva Proizvodi čijim klikom se unutar DataGridView kontrole (koju također treba dodati) prikazuju svi proizvodi čija je cijena bez pdv-a veća od 0 i čiji naziv sadrži niz znakova predan preko TextBoxa. Potrebno je koristiti konekciju stvorenu u prethodnom zadatku, te je stoga varijablu konekcije potrebno deklarirati na razini klase. Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno.

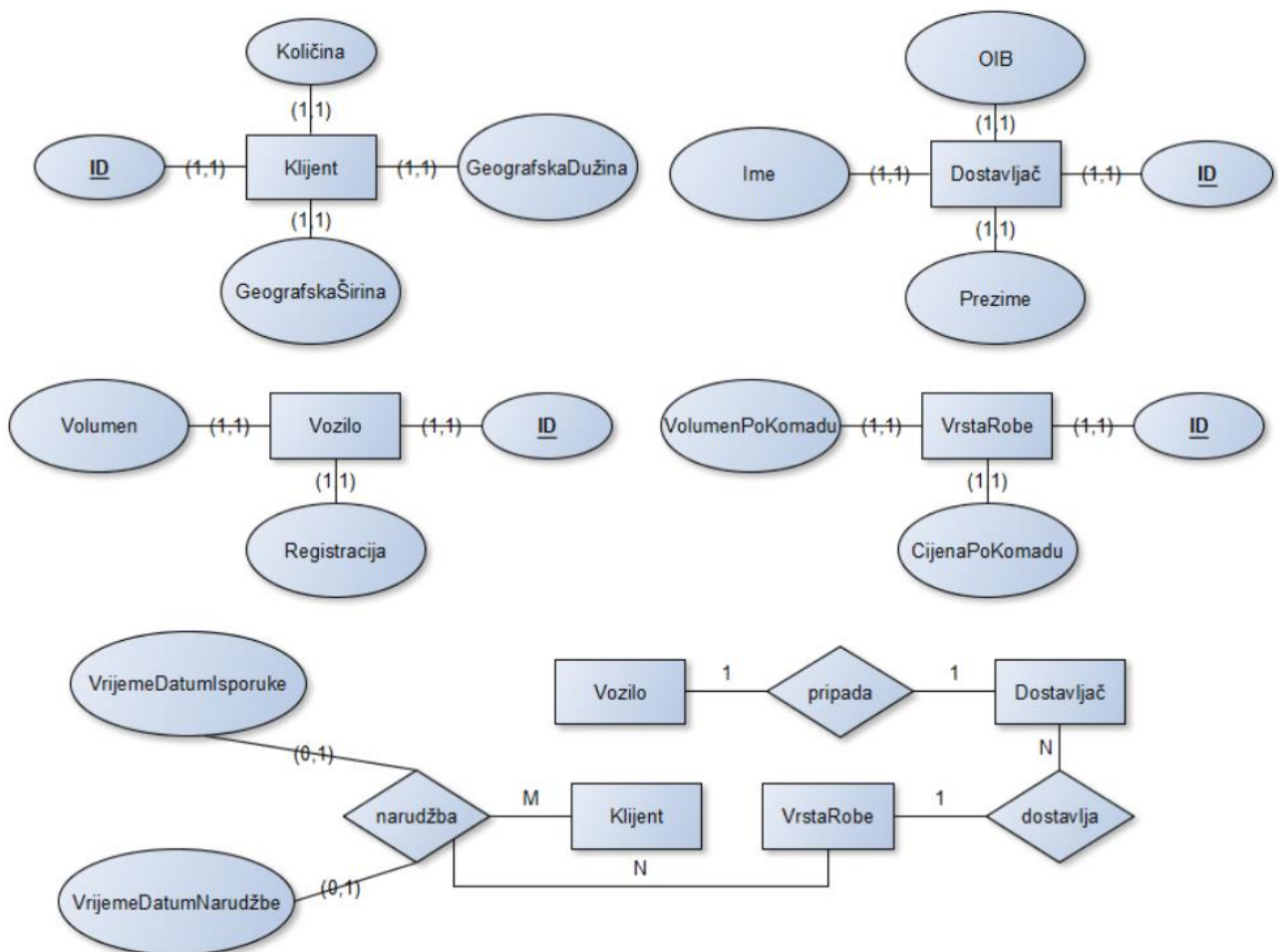


Pismeni ispit #10 – grupa A (18.02.2021.)

1. zadatak (7 bodova)

Na osnovu ER modela prikazanog na slici kreirajte relacijsku shemu baze podataka **Dostava** vlastitom skriptom u SQL jeziku. U skripti moraju biti vidljive naredbe za kreiranje baze podataka i tablica, te na kraju naredba za **brisanje** kreirane baze podataka. Potrebno je postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa te uzeti u obzir i sljedeće stavke:

- svaki dostavljač u bazi mora imati zabilježeno vozilo, ali postoje vozila u bazi koja nemaju dodijeljenog dostavljača
- VolumenPoKomadu - decimalan broj unutar intervala 0–100i koji se sastoji od 4 znamenke s 2 znamenke iza decimalnog zarez
- VrijemeDatumNarudžbe - predefinirana vrijednost je trenutni datum i vrijeme
- Registracija - niz od točno 8 znakova, jedinstvena



2. zadatak (3 boda)

Dohvatite one države koje ukupan broj stavaka koji pripadaju pojedinoj državi imaju manji od 2000, uz uvjet da se razmatraju samo kupci čiji prezime počinje znakom **a**. Potrebno je prikazati stupce **Naziv države** i **Broj stavaka**, te ukoliko neka država ne sadrži niti jednu stavku, potrebno je ispisati vrijednost 0. Svi stupci moraju biti odgovarajuće nazvani.

3. zadatak (3 boda)

Za svaku kombinaciju godine i mjeseca kada su izdani računi potrebno je dohvatiti ukupnu sumu količine proizvoda u pripadajućoj kombinaciji godine i mjeseca. Potrebno je dohvatiti prvih 5% redaka koji imaju najmanju vrijednost ukupne sume količine proizvoda te prikazati **stupce Godina, Mjesec i Ukupna količina proizvoda**.

4. zadatak (3 boda)

Za slučajno dohvaćeni ID grada u intervalu $<0, 15]$ prikažite sve stupce grada. Stupac naziv grada prikažite velikim slovima. Također, korištenjem podupita, unutar istog dohvata, dohvatite broj kupaca u gradu, te nazovite stupac *Broj kupaca*. Svi stupci moraju biti odgovarajuće nazvani.

5. zadatak (2 boda)

Dohvatite stavke čija cijena po komadu mora biti manja ili jednaka 1000 i račun ID mora biti veći od 46330. Potrebno je prikazati stupce **CijenaPoKomadu** i **IDStavka**. Potom postavite neklasterirani indeks da što više ubrzate navedeni upit prema broju logičkih čitanja. Potrebno je komentarom označiti odabir indeksa i broj logičkih čitanja, te napisati naredbu za brisanje indeksa. ˇ

6. zadatak (2 boda)

Izračunajte i dohvatite euklidsku udaljenost između proizvoljnog geometrijskog poligona i točke, uz uvjet da se točka ne nalazi unutar ili na poligonu.

7. zadatak (5 bodova)

a) **(2 boda)** Kreirajte *Windows Forms* aplikaciju u kojoj se klikom na gumb pod nazivom *Spajanje*, aplikacija spaja na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *Windows* autentifikacije (ako to ne radi može i *SQL* autentifikacija). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje 18.02.2021!*, inače u skočnom prozoru treba ispisati grešku koja se pojavila. (pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient;*)

b) **(3 boda)** U prethodno izrađenu aplikaciju dodajte 2 *Label*-a i 2 *TextBox*-a koji će služiti za unos tipa kreditne kartice i isteka godine kreditne kartice. Potrebno je dodati gumb naziva *DohvatiKK* čijim klikom se unutar *DataGridView* kontrole (koju također treba dodati) prikazuju sve kreditne kartice čiji je tip jednak unesenom tipu i čija je vrijednost isteka godine jednaka unesenom isteku godine. Potrebno je koristiti konekciju stvorenu u prethodnom zadatku, te je stoga varijablu konekcije potrebno deklarirati na razini klase.

Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno.



Pismeni ispit #11 – grupa A (17.04.2021.)

1. zadatak (8 bodova)

Na temelju korisničkih zahtjeva izradite **dijagram entiteta** i **ER dijagram** baze podataka *FrizerskiSaloni* koja će bilježiti podatke o salonima, uslugama i zaposlenicima. Rješenje se izrađuje pomoću *yEd Graph Editor*-a i predaje u datoteci pod nazivom **Zadatak1.graphml**.

Za salone je potrebno pamtit naziv, adresu, radno vrijeme (niz od maksimalno 40 dijakritičkih znakova, predefinirana vrijednost je 07:00-19:00) te zaposlenike koji rade u salonu.

Jedan zaposlenik može raditi u maksimalno jednom salonu, a u jednom salonu može raditi više zaposlenika. Postoje zaposlenici koji su trenutno na bolovanju, te oni nemaju unesen salon u kojem rade.

Za zaposlenike je potrebno bilježiti ime, prezime, OIB (točno 11 nedijakritičkih znakova), e-mail, datum zaposlenja (ne smije biti veći od današnjeg datuma), adresu stanovanja te telefone (niz od maksimalno 13 znakova). Jedan zaposlenik mora imati unesen minimalno jedan telefonski broj, ali može imati i više telefonskih brojeva, a jedan telefonski broj odgovara točno jednom zaposleniku.

Svaki salon može pružati više usluga, a jedna usluga može biti u ponudi u više salona. Za uslugu je potrebno pamtit naziv (točno 20 dijakritičkih znakova, jedinstven), težinu izvedbe (cijeli broj) te opis usluge (opcionalna za unos, maksimalan broj znakova).

Svaki salon određuje svoju: cijenu usluge (decimalan broj s maksimalno 4 znamenke i 0 znamenki iza decimalnog zareza), trajanje usluge (tip podataka za vrijeme), iznos popusta usluge (decimalan broj) i vrijeme trajanja popusta na uslugu – datum-vrijeme početka i datum-vrijeme kraja (oba opcionalna za unos).

Vrijednosti koje nisu navedene da su opcionalne za unos moraju se unijeti.

2. zadatak (6 bodova)

Na osnovu izrađenog ER modela kreirajte **relacijsku shemu** baze podataka **vlastitom skriptom**. U skripti moraju biti vidljive naredbe za **kreiranje** baze podataka i tablica, te na kraju naredba za **brisanje** kreirane baze podataka. Potrebno je postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa. Rješenje se predaje u datoteci pod nazivom **Zadatak2.sql**.

3. zadatak (3 boda)

Dohvatite ID kreditne kartice, broj, tip i ukupan plaćeni iznos onih kreditnih kartica koje ukupan plaćeni iznos imaju manji od 8000. U dohvat moraju biti vidljive i one kartice koje nikada nisu korištene te njima za ukupan plaćeni iznos mora biti upisana vrijednost 0. Svi stupci moraju biti odgovarajuće nazvani.

4. zadatak (3 boda)

Dohvatite naziv i broj proizvoda te ukupan broj računa na kojima se proizvod pojavljuje. Prikazuju se samo oni proizvodi koji se pojavljuju na više od 200 računa. Dohvat je potrebno poredati od najveće do najmanje vrijednosti ukupnog broja računa. Svi stupci moraju biti odgovarajuće nazvani.



5. zadatak (2 boda)

Kreirajte *Windows Forms* aplikaciju u kojoj se klikom na gumb pod nazivom *Spajanje*, aplikacija spaja na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *SQL autentifikacije*. Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje 17.04.2021!*, inače u skočnom prozoru treba ispisati „Pojavila se greška!”. (pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient;*)

6. zadatak (3 boda)

Dohvatite stavke čija je cijena po komadu veća od 10000 i račun ID manji od 46330. Potrebno je prikazati stupce *Kolicina* i *IDStavka*. Postavite neklasterirani indeks da što više ubrzate navedeni upit prema broju logičkih čitanja. Potrebno je komentarom označiti odabir indeksa i broj logičkih čitanja, te napisati naredbu za brisanje indeksa.



Pismeni ispit #12 – grupa A (28.06.2021.)

1. zadatak (4 boda)

Na osnovu izrađenog ER modela prikazanog na slici kreirajte **relacijsku shemu** baze podataka *StraniJezik* ili uz pomoć GUI-a (sučelja) u SQL Serveru ili pisanjem SQL koda.

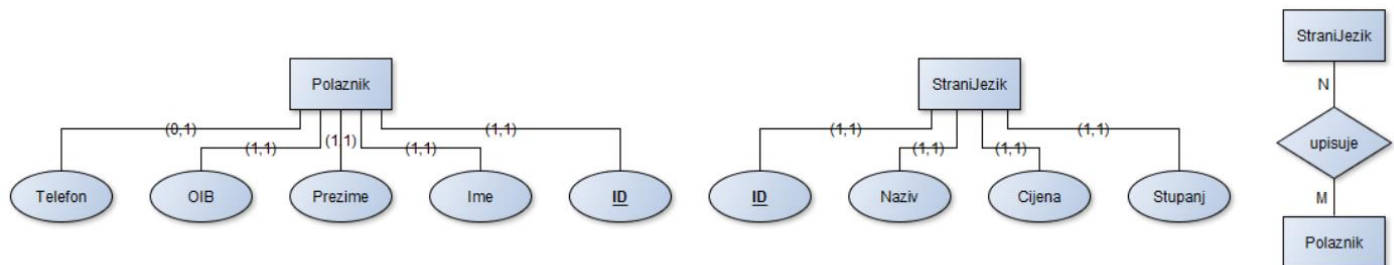
Potrebno je u svaku kreiranu tablicu **unijeti minimalno jedan redak s proizvoljnim vrijednostima** te postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa.

Kao rješenje predaje se **automatski generirana SQL skripta** koju DBMS sam generira (i podaci i shema).

Datoteku je potrebno spremiti pod nazivom **Zadatak1.sql** te na kraju obrisati kreiranu bazu podataka.

Potrebno je uzeti u obzir i sljedeće stavke:

- Stupanj – točno 4 znaka
- Cijena – tip podataka za novac



2. zadatak (1 bod)

Dohvatite sve kupce iz tablice *Kupac*, unutar komentara zabilježite koliko je redaka dohvaćeno, te potom dohvatite koliko ima različitih kupaca u tablici *Kupac* i ponovno unutar komentara zabilježite koliko je redaka dohvaćeno. Unutar komentara usporedite i objasnite rezultat.

3. zadatak (1 bod)

Od svih kupaca kojima ime počinje na *a* i koji imaju vrijednost grad ID-a jednaku 1 dohvatite prvih 10 kupaca abecedno (uzlazno) sortiranih po imenu. Potrebno je prikazati stupce *Ime*, *GradID* i *IDKupac*.

4. zadatak (1 bod)

Svim računima koji nemaju uneseni komentar i koji nisu izdani u 2002. ili 2004. godini, postavite komentar na vrijednost godine u kojoj je račun izdan.

Napomena: na ispitnom roku je napisano „koji su nisu izdani u 2002. ili 2004. godini“, tj. višak je ili riječ su ili riječ nisu. U ovom zadatku ostavljeno je da se dohvaćaju računi koji nisu izdani u 2002. ili 2004. godini jer je taj uvjet teži.

5. zadatak (4 boda)

Dohvatite one proizvode na kojima je zarađeno manje od 5000, pri čemu moraju biti vidljivi i oni na kojima nije ništa zarađeno (unutar stupca *Zarada*). Stupac *CijenaBezPDV* potrebno je kubirati, a stupac *Naziv* potrebno je prikazati s malim slovima i spojiti ga sa stupcem *Boja* prikazanim velikim slovima na način da između stupaca dodate znak – (bez razmaka). Pritom je potrebno paziti da ako boja nije unesena treba za boju prikazati vrijednost *NIJE_UNESENA*. Svi stupci moraju biti odgovarajuće nazvani.



6. zadatak (3 boda)

Za svaku državu prikažite ukupan iznos koji su njeni kupci potrošili u 2003. godini. Potrebno je prikazati stupce ID države, naziv države i ukupno potrošnju samo onih država koje su potrošile više od 17000000.

7. zadatak (1.5 bod)

Korištenjem podupita dohvatite gradove koji nemaju niti jednog kupca.

8. zadatak (5 bodova)

- a) **(2 boda)** Kreirajte *Windows Forms* aplikaciju u kojoj se klikom na gumb pod nazivom *Connect* aplikacija spaja na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *Windows* autentifikacije (ako to ne radi može i *SQL* autentifikacija). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje 28.06.2021!*, inače u skočnom prozoru treba ispisati *Greška prilikom spajanja 28.06.2021!*. (Pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient;*)
- b) **(3 boda)** U prethodno izrađenu aplikaciju dodajte *Label* i *TextBox* koji će služiti za unos niza znakova naziva proizvoda. Potrebno je dodati gumb naziva *Kupci* čijim klikom se unutar *DataGridView* kontrole (koju također treba dodati) prikazuju svi kupci čije prezime sadrži niz znakova predan preko *TextBoxa*. Potrebno je koristiti konekciju stvorenu u prethodnom zadatku, te je stoga varijablu konekcije potrebno deklarirati na razini kase.

Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno.

9. zadatak (1.5 bod)

Kreirajte dvije geografske točke: Zagreb (15.983,45.815) i Split (16.456,43.514), te potom dohvatite udaljenost između te dvije točke.

10. zadatak (3 boda)

S Merlina preuzmite skriptu *BazaFilmoteka* te kreirajte bazu podataka. Normalizirajte preuzetu bazu podataka do 3NF. Na kraju je potrebno pobrisati sve kreirane tablice i kreiranu bazu podataka. Zadatak riješite u već otvorenoj skripti.



4. zadatak (3 boda)

Dohvatite prvih 10 najskupljih stavki (cijena po komadu) čija je količina veća od 10. U rezultatu upita prikazuje se ID stavke, kvadratna vrijednost cijene po komadu, naziv odgovarajućeg proizvoda velikim slovima te broj proizvoda.

5. zadatak (3 boda)

Dohvatite naziv i broj proizvoda te ukupan broj računa na kojima se proizvod pojavljuje. Prikazuju se samo oni proizvodi koji se pojavljuju na više od 100 računa.

6. zadatak (3 boda)

Dohvatite poredane ID-eve proizvoda, od onih koji u sumi imaju najveću prodanu količinu do onih koji imaju najmanju. Potrebno je prikazati stupce ID proizvoda, naziv i ukupna količina. Ukoliko ukupna količina za neki proizvod nije poznata potrebno je ispisati 0. Svi stupci moraju biti odgovarajuće nazvani.

7. zadatak (2 boda)

Korištenjem podupita dohvatite nazive proizvoda koji nisu prodani niti jedanput.

8. zadatak (1 bod)

Unutar komentara u SQL-u nadopunite sljedeći C# kod tako da u tablicu kupac unesete novog kupca čije se ime i prezime unose preko *TextBoxova imeK* i *prezimeK*.

```
SqlCommand cmd = new SqlCommand(_____,sql);  
cmd._____;
```

9. zadatak (1.5 bod)

Pomoću naredbe *geography::STPointFromText('POINT(x y)',4326)* kreirajte dvije geografske točke: Zagreb (15.983,45.815) i Split (16.456,43.514), te potom pomoću funkcije *STDistance* dohvatite udaljenost između te dvije točke.

10. zadatak (3 boda)

Od svih računa koji su izdani prije današnjeg datuma i čija je vrijednost kupca ID veća od 16000, dohvatite prvih 10 koji imaju najmanju vrijednost kupca ID. Potrebno je prikazati stupce *DatumIzdavanja*, *IDRacun* i *KupacID*. Postavite neklasterirani indeks da što više ubrzate navedeni upit prema broju logičkih čitanja. Potrebno je komentarom označiti odabir indeksa i broj logičkih čitanja, te napisati naredbu za brisanje indeksa.



Pismeni ispit #14 – grupa A (02.09.2021.)

1. zadatak (6.5 bodova)

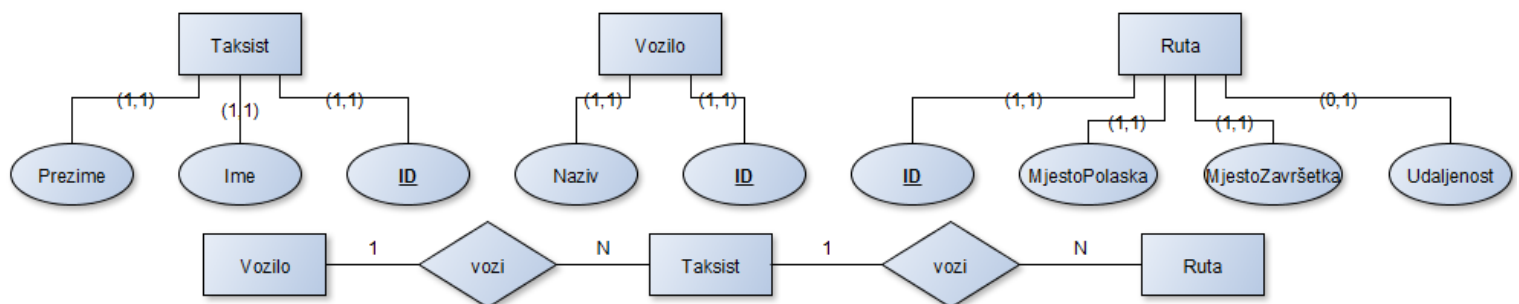
Na osnovu izrađenog ER modela prikazanog na slici kreirajte **relacijsku shemu** baze podataka *Taksi* ili uz pomoć GUI-a (sučelja) u SQL Serveru ili pisanjem SQL koda.

Potrebno je u svaku kreiranu tablicu **unijeti minimalno jedan redak s proizvoljnim vrijednostima** te postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa.

Kao rješenje predaje se **automatski generirana SQL skripta** koju DBMS sam generira (i podaci i shema). Datoteku je potrebno spremiti pod nazivom **Zadatak1.sql** te na kraju obrisati kreiranu bazu podataka.

Potrebno je uzeti u obzir i sljedeće stavke:

- BrojLicence – točno 9 znakova



2. zadatak (1 bod)

Dohvatite ime, prezime i email kupaca koji dolaze iz grada pod nazivom *Split*, ali tako da spojite sva tri navedena stupca u jedan stupac s razmakom između imena i prezimena, a dvotočkom između prezimena i emaila. Stupac nazovite *Podaci*.

3. zadatak (1 bod)

Dohvatite sve podatke kupaca koji se zovu *Kim*, a koji dolaze iz gradova pod ID-evima 3, 7, 11 ili 13.

4. zadatak (3 boda)

Za svaki grad prikažite ukupan iznos koji su njegovi kupci potrošili u 2001. godini. Potrebno je prikazati stupce za naziv grada i ukupnu potrošnju samo onih gradova koji su potrošili manje od 1000000. Svi stupci moraju biti odgovarajuće nazvani.

5. zadatak (3 boda)

Dohvatite prvih 10% kreditnih kartica s kojima je plaćeno najviše računa. Potrebno je prikazati stupce ID kreditne kartice, broj kartice, tip i broj računa (koliko ih ima, a NE stupac BrojRacuna u tablici Racun) koji je plaćen odgovarajućom karticom. Svi stupci moraju biti odgovarajuće nazvani.



6. zadatak (5 bodova)

- a) **(2 boda)** Kreirajte *Windows Forms* aplikaciju u koju se klikom na gumb pod nazivom *Spoji* aplikacija spaja na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *Windows* autentifikacije (ako to ne radi može i *SQL* autentifikacija). (Pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient;*)
- b) **(3 boda)** U prethodno izrađenu aplikaciju dodajte *TextBox* koji će služiti za unos ID-a kupca. Potrebno je dodati gumb naziva *Dohvati kupca* čijim klikom se unutar *DataGridView* kontrole (koju također treba dodati) prikazuju svi računi unesenog kupca (čiji je ID unesen preko *TextBox-a*). Potrebno je koristiti konekciju stvorenu u prethodnom zadatku, te je stoga varijablu konekcije potrebno deklarirati na razini klase.

Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno.

7. zadatak (2 boda)

Korištenjem podupita dohvatite one račune koji su plaćeni kreditnom karticom čija je godina isteka veća od prosječne vrijednosti godine isteka svih kreditnih kartica.

8. zadatak (3 boda)

Deklarirajte varijablu *ukPlacenilznos* koja odgovara tipu podataka za stupac *UkupnaCijena* tablice *Stavka*. U varijablu *ukPlacenilznos* spremite ukupan plaćeni iznos svih stavaka u tablici *Stavka*, te ispišite rezultat u *Messages*.

9. zadatak (3 boda)

Normalizirajte sljedeću tablicu do 3NF. Na kraju je potrebno pobrisati kreirane tablice!

Ime i prezime	OIB	Institucija 1	Institucija 2
Tomo Tomić	125125125	Muzej iluzija	Tehnički muzej
Miro Mirić	785785785	Tehnički muzej	Muzej suvremene umjetnosti



Pismeni ispit #15 – grupa A (23.03.2022.)

1. zadatak (4 boda)

Dohvatite sve različite tipove kreditnih kartica koje su korištene za kupnju proizvoda čija je boja proizvoda *crna* ili *crvena* i čiji naziv proizvoda sadrži na bilo kojem mjestu *znak znamenke*, te samo one tipove koji ukupnu količinu prodanih proizvoda (zbroy količina) koji pripadaju pojedinom tipu kreditne kartice imaju unutar intervala [15000, 25000]. Potrebno je prikazati stupce *Tip* i *Ukupna prodana količina proizvoda*.

2. zadatak (3 boda)

Korištenjem isključivo podupita, za sve kupce čija je vrijednost ID-a grada veća od prosječne vrijednosti ID-a grada u tablici *Grad*, dohvatite ukupan broj računa (koliko ih ima) koje je napravio pojedini kupac. U dohvat je potrebno prikazati sve stupce kupca, te dodatno stupac *Ukupan broj računa*.

3. zadatak (1 bod)

Dohvatite sve kupce iz tablice *Kupac*, unutar komentara zabilježite koliko je redaka dohvaćeno, te potom dohvatite koliko ima različitih imena i prezimena kupaca u tablici *Kupac* i ponovno unutar komentara zabilježite koliko je redaka dohvaćeno. Unutar komentara usporedite i objasnite rezultat.

4. zadatak (4 boda)

Kreirajte *Windows Forms* aplikaciju za rad s bazom *SimpleWebShop*. Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno kao **Zadatak4**.

- a) **(2 boda)** Aplikacija se treba spojiti na vašu lokalnu bazu podataka *SimpleWebShop* pomoću SQL autentifikacije. Potrebno je dodati dva *Label-a* s tekstovima: *Korisničko ime* i *Zaporka*; te dva *TextBox-a* naziva *txtKorisnickolme* i *txtZaporka*. Potrebno je unijeti korisničko ime u *TextBox* naziva *txtKorisnickolme* i zaporku u *TextBox* naziva *txtZaporka*, te pritisnuti gumb s tekstom *Spoji* (naziva *btnSpoji*). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje na bazu SimpleWebShop!*, inače u skočnom prozoru treba ispisati grešku koja se pojavila. (Pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient*;) Za potrebe
- b) **(2 boda)** dijela zadatka zgodno je instancu konekcije deklarirati na razini klase, ali nije nužno za rješenje ovog dijela zadatka.
- b) **(2 boda)** U prethodno izrađenu aplikaciju dodajte tri *Label-a* s tekstovima: *Naziv*, *Broj proizvoda* i *Cijena bez PDV-a*; i tri *TextBox-a* naziva: *txtNaziv*, *txtBrojProizvoda* i *txtCijenaBezPDV*. Potrebno je dodati gumb s tekstom *Unesi* naziva *btnUnesi* čijim klikom se u tablicu *Proizvod* unosi novi proizvod s vrijednostima iz *TextBox-ova* *txtNaziv*, *txtBrojProizvoda* i *txtCijenaBezPDV*. Ukoliko je unos bio uspješan potrebno je u skočnom prozoru ispisati poruku *Uspješno unesen proizvod!*. Potrebno je koristiti konekciju stvorenu u **a)** zadatku, odnosno varijablu konekcije potrebno je deklarirati na razini klase (ako to već nije prethodno napravljeno).



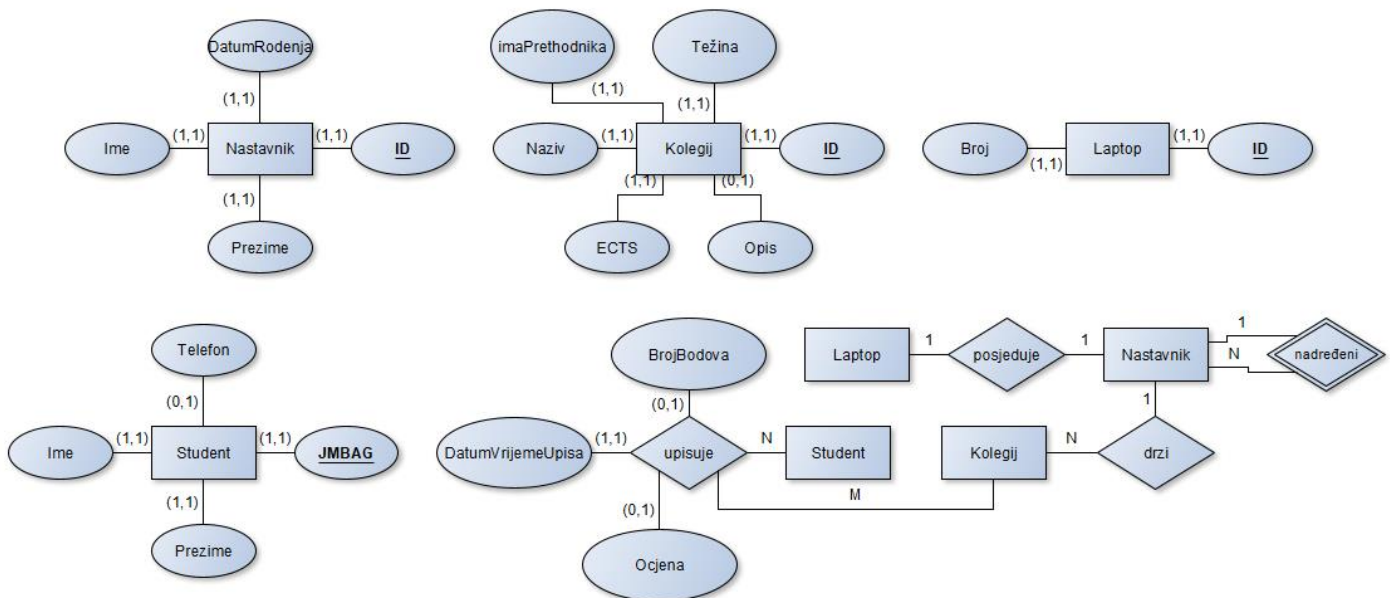
5. zadatak (2 boda)

Od svih računa koji su izdani prije današnjeg datuma i koji imaju vrijednost stupca *KupacID* veću od 0, dohvatite prvih 5% koji imaju najveću vrijednost kupca ID. Potrebno je prikazati stupce *IDRačun*, *DatumIzdavanja* i *KupacID*. Stupac *KupacID* potrebno je nazvati ID kupca. Unutar komentara u SQL-u naznačite redoslijed izvršavanja dohвата. Potrebno je naznačiti po koracima dijelove dohвата. Objasnite zašto se u dijelu filtriranja redaka ne može koristiti preimenovani novi naziv stupca *ID kupca*, a u dijelu sortiranja može.

6. zadatak (8 bodova)

Na osnovu izrađenog ER modela prikazanog na slici kreirajte **relacijsku shemu** baze podataka *Referada* pisanjem SQL koda. Potrebno je svaku kreiranu tablicu **unijeti minimalno jedan redak s proizvoljnim vrijednostima** te postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa. Kao rješenje predaje se **vlastito napisana SQL skripta**. Datoteku je potrebno spremiti pod nazivom **Zadatak6.sql** te na kraju obrisati kreiranu bazu podataka. Potrebno je uzeti u obzir i sljedeće uvjete:

- Telefon – točno 13 nedijakritičkih znakova
- Broj – jedinstven
- Naziv – predefinirana vrijednost *Proizvoljan kolegij*
- ImaPrethodnika – tip podataka za istinu ili laž
- Težina – cijeli broj unutar intervala [0, 255]
- BrojBodova – decimalan broj od ukupno 6 znakova, a maksimalno 4 iza decimalnog zareza
- Svaki kolegij ne mora imati pridruženog nastavnika koji ga drži
- Nastavnik mora imati laptop, dok svaki laptop ne mora nužno pripadati nastavniku



7. zadatak (3 boda)

Deklarirajte varijablu *brojGradova*. U varijablu *brojGradova* spremite koliko ima gradova koji se nalaze u državi čiji naziv države počinje znakom *a*. Rezultat ispišite u *Messages*.



Pismeni ispit #16 – grupa A (23.06.2022.)

1. zadatak (4 boda)

Dohvatite sve stavke čiji je popust u postotcima različit od nule i koje se odnose na proizvode čija je boja *crna*. Potrebno je prikazati stupce *Popust u postotcima* i *Ukupna cijena*. U stupcu *Ukupna cijena* prikazuje se zarada na svim prodanim stavkama koje imaju jednak popust u postotcima.

2. zadatak (3 boda)

Korištenjem isključivo podupita, dohvatite sve kreditne kartice čiji tip kreditne kartice odgovara tipu kreditne kartice koji se najčešće pojavljuje u tablici Kreditna kartica.

3. zadatak (1 bod)

Dohvatite sve nazive gradova iz tablice *Grad* te dodatno prikažite stupac *Broj kupaca* u kojem se za svaki grad prikazuje broj kupaca koji dolazi iz tog grada.

4. zadatak (4 boda)

Kreirajte *Windows Forms* aplikaciju za rad s bazom *SimpleWebShop*. Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno kao **Zadatak4**.

- a) **(2 boda)** Aplikacija se treba spojiti na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *SQL autentifikacije*. Potrebno je dodati dva *Label-a* s tekstovima: *Korisničko ime* i *Zaporka*; te dva *TextBox-a* naziva: *txtKorisnickolme* i *txtZaporka*. Potrebno je unijeti korisničko ime u *TextBox* naziva *txtKorisnickolme* i zaporku u *TextBox* naziva *txtZaporka*, te pritisnuti gumb s tekstom *Spoji* (naziva *btnSpoji*). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje na bazu SimpleWebShop!*, inače u skočnom prozoru treba ispisati grešku koja se pojavila. (Pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient*; Za potrebe **b**) dijela zadatka zgodno je instancu konekcije deklarirati na razini klase, ali nije nužno za rješenje ovog dijela zadatka.)
- b) **(2 boda)** U prethodno izrađenu aplikaciju dodajte tri *Label-a* s tekstovima: *Cijena od*, *Cijena do* i *Boja*; i tri *TextBox-a* naziva: *txtcijenaOd*, *txtcijenaDo* i *txtBoja*. Potrebno je dodati gumb s tekstom *Dohvati* (naziva *btnDohvati*) čijim klikom se iz tablice *Proizvod* dohvaćaju svi proizvodi čija je boja jednaka vrijednosti boje u *TextBox-u* *txtBoja* te je cijena proizvoda u rasponu vrijednosti zapisanih u *TextBox-ovima* *txtcijenaOd* i *txtcijenaDo*. Potrebno je koristiti konekciju stvorenu u **a**) zadatku, odnosno varijablu konekcije potrebno je deklarirati na razini klase (ako to već nije prethodno napravljeno).

5. zadatak (2 boda)

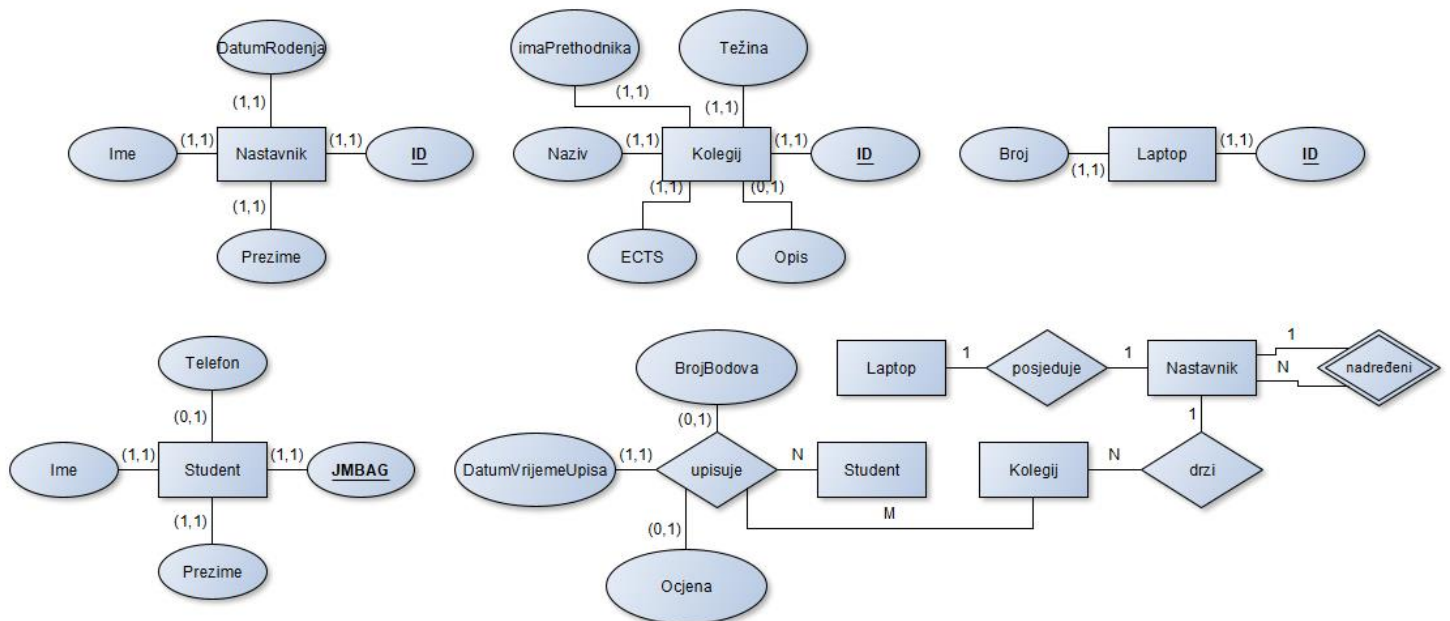
Dohvatite račune iz tablice *Račun* koji pripadaju kupcima čije ime sadrži znakove *me* te imaju zapisan telefon. Potrebno je prikazati prvih 10% računa sortiranih prema ID računa od najmanjeg prema najvećem koji pripadaju kupcima s različitim prezimenima. U dohvat u prikazuje stupac *Prezime* te svi stupci iz tablice *Račun*. Potrebno je u komentarima naznačiti po koracima redoslijed dohvata.



6. zadatak (8 bodova)

Na osnovu izrađenog ER modela prikazanog na sljedećoj slici kreirajte **relacijsku shemu** baze podataka *Referada* pisanjem SQL koda. Potrebno je u svaku kreiranu tablicu **unijeti minimalno jedan redak s proizvoljnim vrijednostima** te postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa. Kao rješenje predaje se **vlastito napisana SQL skripta**. Datoteku je potrebno spremiti pod nazivom **Zadatak6.sql** te na kraju obrisati kreiranu bazu podataka. Potrebno je u obzir uzeti i sljedeće uvijete:

- Telefon – točno 13 nedijakritičkih znakova
- Broj – jedinstven
- Naziv – predefinirana vrijednost *Proizvoljan kolegij*
- ImaPrethodnika – tip podataka za istinu ili laž
- Težina – cijeli broj unutar intervala [0, 255]
- BrojBodova – decimalan broj od ukupno 6 znakova, a maksimalno 4 iza decimalnog zareza
- svaki kolegij ne mora imati pridruženog nastavnika koji ga drži
- nastavnik mora imati laptop, dok svaki laptop ne mora nužno pripadati nastavniku



7. zadatak (3 boda)

Deklarirajte varijablu $p1$ i varijablu $p2$, obje tipa *geometry*, koje predstavljaju točke u koordinatnom sustavu. Pozicija točke $p1$ je (2, 10), a točke $p2$ (4, 14). Izračunajte i ispišite u *Messages* udaljenost između te dvije točke. Za izračun udaljenosti koristite metodu *STDistance*.



Pismeni ispit #17 – grupa A (07.07.2022.)

2. zadatak (1 bod)

Svim računima koji imaju uneseni komentar i koji su izdani u 2001. ili 2003. godini postavite komentar na *Izdani u 2001. ili 2003. godini!*.

3. zadatak (2 boda)

Dohvatite ime i prezime kupaca iz tablice *Kupac* te dodatno prikažite stupac *Broj racuna* u kojem se za svakog kupca prikazuje broj računa koje je taj kupac generirao.

4. zadatak (3 boda)

Korištenjem podupita dohvatite ID i naziv grada onih gradova koji imaju barem jednog kupca. Svi stupci moraju biti odgovarajuće nazvani.

5. zadatak (3 boda)

Dohvatite naziv države i ukupan broj kupaca koji dolazi iz te države. U ukupan broj kupaca iz neke države uključeni su samo oni kupci kojima je poznat *email* i *telefon*. Prikazuju se samo one države iz kojih dolazi više od 5000 kupaca. Svi stupci moraju biti odgovarajuće nazvani.

6. zadatak (2 boda)

Dohvatite različite nazive proizvoda i njihove ukupne cijene iz tablice *Stavka*. Dohvaćaju se oni proizvodi čija je količina jednaka 1, a u broju proizvoda sadrže znakove *so*. Potrebno je prikazati prvih 5 proizvoda sortiranih prema ukupnoj cijeni od najveće prema najmanjoj.

7. zadatak (3 boda)

Kreirajte *Windows Forms* aplikaciju za rad s bazom *SimpleWebShop*. Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno kao **Zadatak7**.

- a) **(2 boda)** Aplikacija se treba spojiti na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *SQL autentifikacije*. Potrebno je dodati dva *Label-a* s tekstovima: *Korisničko ime* i *Zaporka*; te dva *TextBox-a* naziva: *txtKorisnickolme* i *txtZaporka*. Potrebno je unijeti korisničko ime u *TextBox* naziva *txtKorisnickolme* i zaporku u *TextBox* naziva *txtZaporka*, te pritisnuti gumb s tekstom *Spoji* (naziva *btnSpoji*). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje na bazu SimpleWebShop!*, inače u skočnom prozoru treba ispisati grešku koja se pojavila. (Pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient*; . Za potrebe **b)** dijela zadatka zgodno je instancu konekcije deklarirati na razini klase, ali nije nužno za rješenje ovog dijela zadatka.)
- b) **(2 bod)** U prethodno izrađenu aplikaciju dodajte gumb s tekstom *Dohvati* (naziva *btnDohvati*) čijim klikom se iz tablice *Kreditna kartica* dohvaćaju sve kreditne kartice, na način da se unutar *DataGridView* kontrole naziva *tablica* prikažu svi dohvaćeni podaci. Potrebno je koristiti konekciju stvorenu u **a)** zadatku, odnosno varijablu konekcije potrebno je deklarirati na razini klase (ako to već nije prethodno napravljeno).



8. zadatak (3 boda)

Od svih računa koji su izdani prije današnjeg datuma i čija je vrijednost stupca *KupacID* veća od 16000, dohvatite prvih 10 koji imaju najmanju vrijednost stupca *KupacID*. Potrebno je prikazati stupce *DatumIzdavanja* i *IDRacun*. Postavite neklasterirani indeks da što više ubrzate navedeni upit prema broju logičkih čitanja. Potrebno je komentarom označiti odabir indeksa i broj logičkih čitanja, te napisati naredbu za brisanje indeksa.



Pismeni ispit #18 – grupa A (08.09.2022.)

1. zadatak (3 boda)

Deklarirajte varijablu *brojRacuna*. U varijablu *brojRacuna* spremite koliko ima računa koji nemaju zabilježen komentar, a koji su plaćeni kreditnom karticom tipa *Visa*. Rezultat ispišite u *Messages* u formatu *Broj računa bez komentara koji su plaćeni kreditnom karticom tipa Visa je X* (umjesto vrijednosti *X* treba ispisati vrijednost).

2. zadatak (2 boda)

Pomoću naredbe `geometry::STPolyFromText('POLYGON((x1 y1, x2 y2, ... ,xn yn, x1 y1))', 0)` kreirajte proizvoljni poligon od minimalno 7 točaka, te ga dohvatite.

3. zadatak (2 boda)

Od svih proizvoda čiji broj proizvoda počinje znakom *s* i čija je vrijednost stupca *CijenaBezPDV* veća od 0, dohvatite prvih 10% najskupljih proizvoda. Potrebno je prikazati stupce *IDProizvod*, *BrojProizvoda* i *CijenaBezPDV*. Stupac *CijenaBezPDV* potrebno je nazvati *Cijena* te vrijednosti u stupcu prikazati kao decimalne brojeve zaokružene na dvije decimale. Unutar komentara u SQL-u naznačite redoslijed izvršavanja dohvata. Potrebno je naznačiti po koracima dijelove dohvata. Objasnite zašto se u dijelu filtriranja redaka ne može koristiti naziv stupca *Cijena*, a u dijelu sortiranja može.

4. zadatak (3 boda)

Korištenjem podupita dohvatite ID i naziv grada onih gradova koji imaju barem jednog kupca. Svi stupci moraju biti odgovarajuće nazvani.

5. zadatak (3 boda)

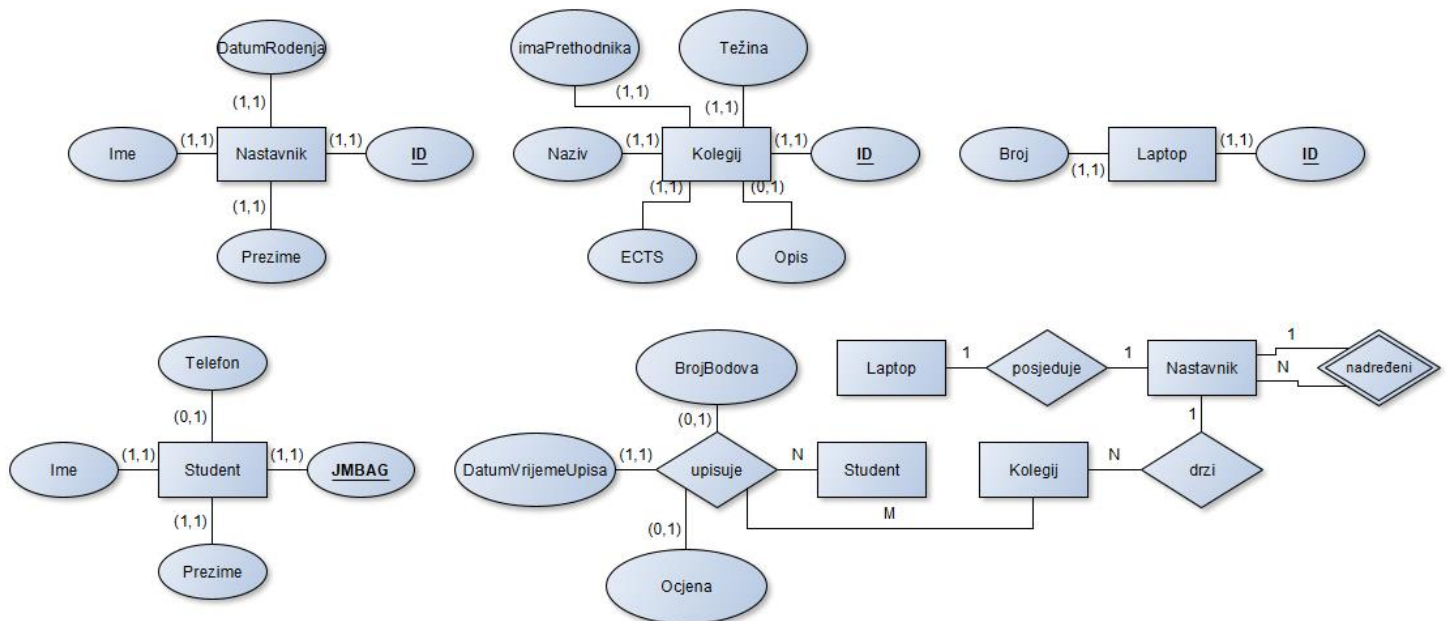
Dohvatite naziv države i ukupan broj kupaca koji dolazi iz te države. U ukupan broj kupaca iz neke države uključeni su samo oni kupci kojima je poznat *email* i *telefon*. Prikazuju se samo one države iz kojih dolazi više od 5000 kupaca. Svi stupci moraju biti odgovarajuće nazvani.



6. zadatak (8 bodova)

Na osnovu izrađenog ER modela prikazanog na sljedećoj slici kreirajte **relacijsku shemu** baze podataka *Referada* pisanjem SQL koda. Potrebno je u svaku kreiranu tablicu **unijeti minimalno jedan redak s proizvoljnim vrijednostima** te postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa. Kao rješenje predaje se **vlastito napisana SQL skripta**. Datoteku je potrebno spremiti pod nazivom **Zadatak6.sql** te na kraju obrisati kreiranu bazu podataka. Potrebno je u obzir uzeti i sljedeće uvijete:

- Telefon – točno 13 nedijakritičkih znakova
- Broj – jedinstven
- Naziv – predefinirana vrijednost *Proizvoljan kolegij*
- ImaPrethodnika – tip podataka za istinu ili laž
- Težina – cijeli broj unutar intervala [0, 255]
- BrojBodova – decimalan broj od ukupno 6 znakova, a maksimalno 4 iza decimalnog zareza
- svaki kolegij ne mora imati pridruženog nastavnika koji ga drži
- nastavnik mora imati laptop, dok svaki laptop ne mora nužno pripadati nastavniku



7. zadatak (3 boda)

Kreirajte *Windows Forms* aplikaciju za rad s bazom *SimpleWebShop*. Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno kao **Zadatak7**.

- a) **(1 bod)** Aplikacija se treba spojiti na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *SQL autentifikacije*. Potrebno je dodati dva *Label*-a s tekstovima: *Korisničko ime* i *Zaporka*; te dva *TextBox*-a naziva: *txtKorisnickoIme* i *txtZaporka*. Potrebno je unijeti korisničko ime u *TextBox* naziva *txtKorisnickoIme* i zaporku u *TextBox* naziva *txtZaporka*, te pritisnuti gumb s tekstom *Spoji* (naziva *btnSpoji*). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje na bazu SimpleWebShop!*, inače u skočnom prozoru treba ispisati grešku koja se pojavila. (Pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient*; . Za potrebe **b)** dijela zadatka zgodno je instancu konekcije deklarirati na razini klase, ali nije nužno za rješenje ovog dijela zadatka.)
- b) **(2 boda)** U prethodno izrađenu aplikaciju dodajte gumb s tekstom *Dohvati* (naziva *btnDohvati*) čijim klikom se iz tablice *Kreditna kartica* dohvaćaju sve kreditne kartice, na način da se unutar *DataGridView* kontrole naziva *tablica* prikažu svi dohvaćeni podaci. Potrebno je koristiti konekciju stvorenu u **a)** zadatku, odnosno varijablu konekcije potrebno je deklarirati na razini klase (ako to već nije prethodno napravljeno).



Pismeni ispit #19 (11.07.2024.)

1. zadatak (2 boda)

U svijetu gdje je biti kupac norma, postoji tajno društvo gradova koji nemaju kupca. Koristeći podupit, zadatak je pronaći te prazne gradove te dohvatiti njihov ID i Naziv.

2. zadatak (2 boda)

Kreirajte dvije proizvoljne geometrijske linije te im dohvatite vrijednost presjeka. Dohvaćeni stupac nazovite 'Presjek'.

3. zadatak (2 boda)

Dohvatite stupac 'Ukupna cijena stavki' koji sadrži vrijednosti ukupne ostvarene cijene svih stavki čiji je popust u postocima veći ili jednak 20% te čija je vrijednost količine strogo veća od prosječne vrijednosti stupca Količina svih n-torki u tablici. Dodatno, ako se ukupna potrošnja ne može iskazati numerički, potrebno je ispisati broj nula (0).

4. zadatak (3 boda)

Dohvatite Tip, Istek godinu, Broj zapisa te Ukupnu potrošnju svih kreditnih kartici čiji tip nije ni Visa ni Diners te čija je vrijednost kupca IstekGodina između 2005. i 2007. godine. Razmatrajte samo one n-torke čiji je zbroj ostvarenih zapisa veći od 5000 te čija je ukupna ostvarena potrošnja veća od 100000. Dohvaćene n-torke poredajte silazno prema ukupnoj ostvarenoj potrošnji. Dodatno, ako se ukupna potrošnja ne može iskazati numerički, potrebno je ispisati broj nula (0).

5. zadatak (3 boda)

Jedan korisnik može imati samo jednu adresu (zip) stanovanja te jednu firmu u kojoj radi. Na istoj adresi može biti više firmi i korisnika. Potrebno je tablicu NF_X pretvoriti u 3NF. Na kraju je potrebno pobrisati stvorene tablice.

```
CREATE TABLE NF_X
(
    custID int not null,
    custName nvarchar(50) not null,
    custAge date,
    custZip nvarchar(100),
    custCity nvarchar(100),
    company nvarchar(100),
    companyName nvarchar(100),
    companyCity nvarchar(100),
    companyZip nvarchar(100)
)
```

6. zadatak (3 boda)

U zadatku je potrebno koristiti barem jednu lokalnu varijablu. Koristeći podupit povećajte popust u postocima za 5% svim stavkama čija je količina strogo veća od 29 te čija trenutna ukupna cijena **nije** unutar intervala [donja, gornja]. Donja granica predstavlja 50 %-tnu vrijednost prosječne ukupne cijene svih stavki, a gornja granica predstavlja 150 %-tnu vrijednost prosječne ukupne cijene svih stavku. Prosječne ukupne cijene svih stavki zaokružite na dva decimalna mjesta.



7. zadatak (3 boda)

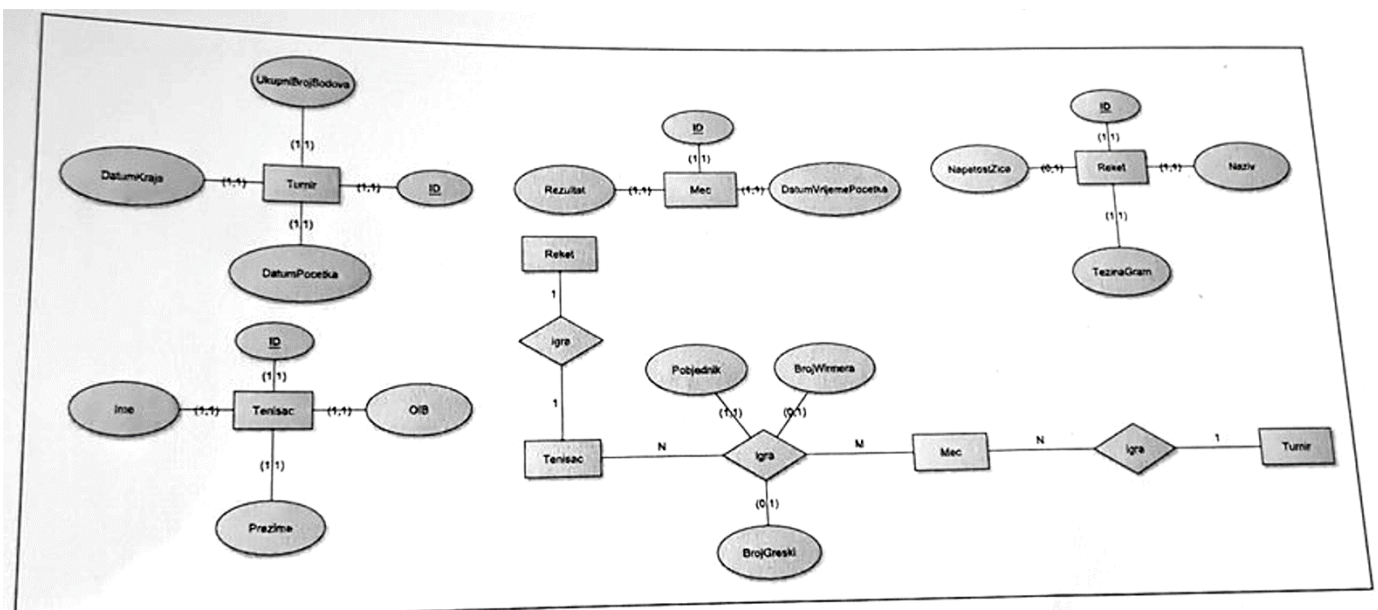
Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno. Pomoć: dodajte `using System.Data.SqlClient;` te deklarirajte varijablu konekcije na razini klase.

- (1 bod)** Kreirajte *Windows Forms* aplikaciju u kojoj se klikom na gumb s tekстом *Spoji* (naziv: *btnSpoji*) aplikacija spaja na bazu podataka *SimpleWebShop* pomoću *SQL autentifikacije*. Pri čemu su *korisničko ime* („sa“) i *lozinka* („Baze123“) već unaprijed definirani te ih korisnik ne treba unositi. Ako je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spojeno!*, inače u skočnom prozoru treba ispisati grešku generiranu od strane programa (*Exception*). Metodu za spajanje na bazu napišite unutar *try-catch* bloka.
- (2 boda)** U prethodno izrađenu aplikaciju dodajte *TextBox* (naziv: *txtPocetniZnak*) te gumb s tekстом *Dohvati* (naziv: *btnDohvati*), čijim klikom se unutar *DataGridView* kontrole (naziv: *dgvTablica*) prikazuju svi stupci proizvoda čiji naziv počinje znakom ili skupom znakova unesenim u *txtPocetniZnak*. Metodu za dohvat te ispis rezultata napišite unutar *try-catch* bloka.

8. zadatak (7 bodova)

Na osnovu ER modela prikazanog na slici kreirajte **relacijsku shemu** baze podataka **vlastitom skriptom**. U skripti moraju biti vidljive naredbe za **kreiranje** baze podataka i tablica, te na kraju naredba za **brisanje** kreirane baze podataka. Potrebno je postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa, te uzeti u obzir sljedeće stavke:

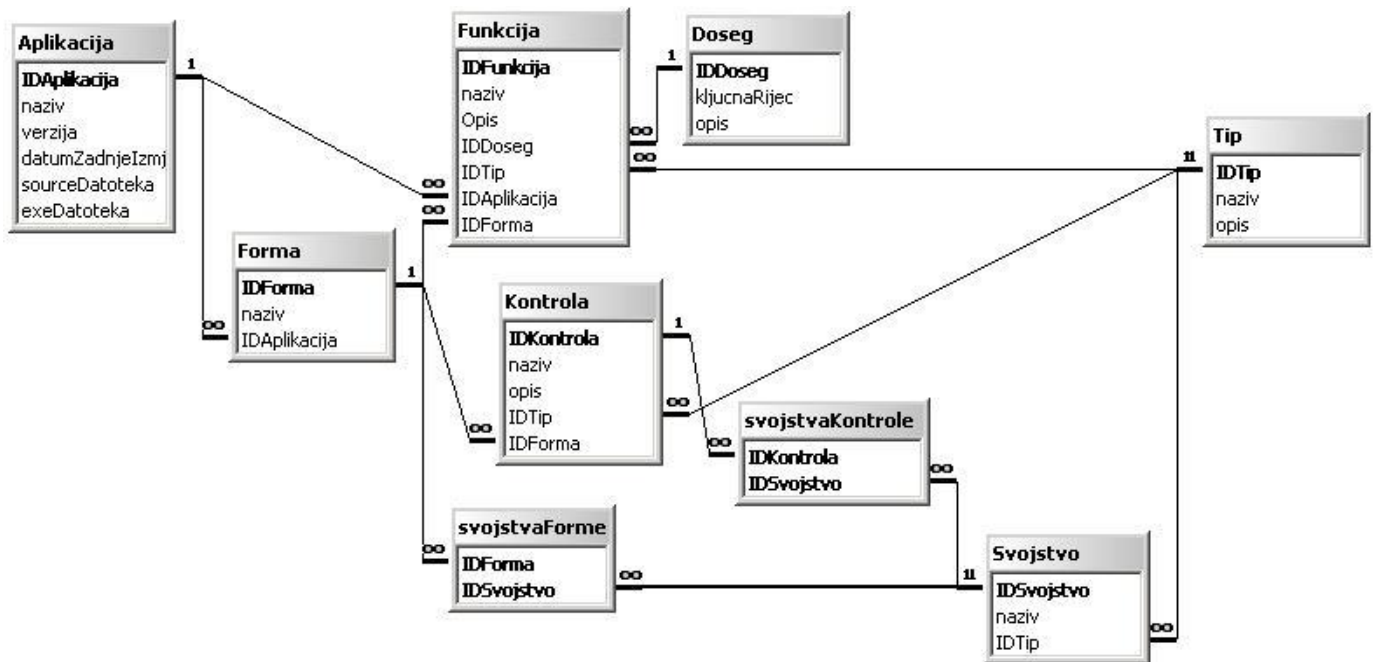
- OIB* – točno **13** nedijakritičkih znakova, jedinstven
- TežinaGram* – broj s plivajućim zarezom unutar intervala **<100, 450]**, predefinirane vrijednosti **120**
- NapetostŽica* – broj s plivajućim zarezom od ukupno **3** znamenke, **1** iza plivajućeg zareza
- DatumVrijemePočetka* meča – predefinirano trenutno datum-vrijeme
- Pobjednik* – tip podataka istina ili laž
- svaki tenisač ne mora imati pridružen reket te isto tako svaki reket ne mora biti pridružen tenisaču



Prvi kolokviji

Prvi kolokvij #1

Zadana je baza kao na slici:



Nad bazom trebate napraviti SQL upite:

- Napisat naredbu kojom se tablici *Aplikacija* dodaje atribut *datumKreiranja*. Atribut nikada ne može biti prazan.
- Napisati naredbu kojom bi se postavilo da se ne zna doseg svih funkcija kojima je ključna riječ za doseg Protected.
- Ispisati sve ID-eve i nazive aplikacije koje nisu mijenjane više od dvije i pol godine, a *sourceDatoteka* im je nepoznata (nije upisana). Upit mora biti neovisan o vremenu izvođenja.
- Ispisati sve forme zajedno s funkcijama koje im pripadaju, te odgovarajući tip funkcije. Na popisu trebaju biti sve forme (i one koje nemaju definiranu funkciju). Ispišite naziv forme, naziv funkcije i naziv odgovarajućeg tipa.
- Naći 10% najvećih aplikacija čini naziv ne počinje riječju *Evidencija*. Najveća je aplikacija ona koja ima najviše formi.
- Za svaku formu aplikacije naziva *Studentska služba* ispišite koliko ima kontrola.
- Napišite naredbe kojima bi se iz tablice *svojstvo* obrisala svojstva koja se ne koriste kao svojstva formi.
- Za svaki mjesec tekuće godine ispišite koliko je kreiranih aplikacija. Ako niste riješili a) zadatak pretpostavite da u tablici *aplikacija* postoji atribut *datumKreiranja*.
- Ispišite naziv tipa kontrole koja se najčešće koristi.



Prvi kolokvij #2 – grupa A (01. 12. 2010.)

Napraviti ER model baze podataka skladišta novih automobila. Osim ER dijagrama treba se definirati svaki entitet, kardinalitet atributa i primarne ključeve. Imena entiteta i veza možete odabrati prema želji. Za svaku tablicu i vezu koja postaje tablica ispisati zaglavlje tablice i podcrtati primarne ključeve. Model se treba sastojati od sljedećih atributa (svi se potrebni atributi nalaze ovdje, nije potrebno dodavati nove). Svi automobili koji postoje u narudžbi nalaze se u skladištu u dostatnim količinama. Jedna narudžba odnosi se samo na jedan automobil.

broj_sasije, opis_automobila, id_narudzba, datum_narudzba, id_izdatnica, datum_izdatnice, id_kupca, ime_kupca, prezime_kupca, adresa_kupca, id_skladistar, ime_skladistar, prezime_skladistar



Prvi kolokvij #3 – grupa A (18. 11. 2014.)

Korisnički zahtjevi

Kreirate bazu podataka „Životinjsko carstvo“. Životinjske vrste (tigar, majmun, slon i sl.) se dijele na različite podskupine, pa unutar podskupina na porodice.

Za svaku životinju je potrebno pamtit i njeni naziv, latinski naziv (opcionalan za unos) te porodicu kojoj pripada. Porodice se opisuju nazivom i opisom isto kao i podskupine životinja.

U jednoj podskupini životinja može biti više porodica dok porodica pripada točno jednoj životinjskoj podskupini.

Svaka životinja je pridružena točno jednoj porodici, a jedna porodica može sadržavati više životinja.

Osim podjele životinja potrebno je pamtit i njihova staništa gdje obitavaju na razini kontinenta.

Kontinent je opisan nazivom i površinom, te opisom koji je opcionalan za unos. Na jednom kontinentu može živjeti više životinjskih vrsta, a jedna životinjska vrsta može živjeti na više kontinenata. Primjerice tigar živi i u Africi i u Aziji, a u oba spomenuta kontinenta žive i druge životinje.

Zadaci

1. Izradite dijagram entiteta uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „Zadatak1.graphml“.
2. Izradite ER dijagram uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „Zadatak2.graphml“.
3. Na osnovu izrađenog ER modela kreirajte relacijsku shemu baze uz pomoć SQL servera. Rješenje je potrebno predati u obliku SQL skripte koju DBMS sam generira, a naziv skripte treba biti „Zadatak3.sql“.
4. Na osnovu izrađenog ER modela kreirajte relacijsku shemu baze vlastitom skriptom. Rješenje je potrebno predati u obliku SQL skripte pod nazivom „Zadatak4.sql“.
5. Na testnoj bazi „SimpeWebShop“ dohvatite sljedeće podatke:
 - a) dohvatite različite nazive svih proizvoda
 - b) dohvatite sve kupce čije je treće slovo imena 'a', a drugo slovo prezimena 'b' dok poredak ostalih slova nije bitan
 - c) dohvatite prvih 5 proizvoda
 - d) dohvatite sve proizvode čija je boja 'Crvena'



Prvi kolokvij #4 – grupa A (17. 11. 2015.)

Korisnički zahtjevi

Kreirajte bazu podataka „Biciklist“ koja će pamtit i podatke o biciklima i njihovim vlasnicima te odvoženim utrka. Za bicikl je potrebno pamtit njegov naziv, serijski broj od točno 12 znakova, boju, veličinu kotača te vlasnika bicikla (biciklista). Biciklist može posjedovati više bicikala dok svaki bicikl pripada samo jednom biciklistu. Za biciklista je potrebno pamtit njegovo ime, prezime, OIB, e-mail koji je opcionalan za unos te utrke koje je biciklist odvezio. Za svaku utrku potrebno je bilježiti naziv utrke, kilometražu, datum održavanja i tip utrke. Za tip utrke potrebno je pamtit naziv i opis (do maksimalno 100 znakova). Svaka utrka može biti samo jednog tipa, ali istom tipu može pripadati više utrka.

Zadaci:

1. Izradite dijagram entiteta uz pomoć yEd Graph Editor – rješenje se predaje u datoteci pod nazivom „Zadatak1.graphml“.
2. Izradite ER dijagram uz pomoć yEd Graph Editor – rješenje se predaje u datoteci pod nazivom „Zadatak2.graphml“.
3. Na osnovu izrađenog ER modela kreirajte relacijsku shemu baze uz pomoć SQL servera. Rješenje je potrebno predati u obliku SQL skripte koju DBMS sam generira, a naziv skripte treba biti „Zadatak3.sql“.
4. Na osnovu izrađenog ER modela kreirajte relacijsku shemu baze vlastitom skriptom. Rješenje je potrebno predati u obliku SQL skripte pod nazivom „Zadatak4.sql“.
5. Na testnoj bazi „SimpleWebShop“ dohvatite sljedeće podatke:
 - a) iz tablice Proizvod dohvatite sve proizvode koji su crne boje
 - b) dohvatite sve podatke iz tablice KreditnaKartica sortirane padajuće po godini isteka kartice
 - c) dohvatite sve kupce čije ime ima drugo slobo u rasponu od 'g' do 'm' (prvo slovo nije bitno)
 - d) dohvatite prvih 1% država.

Rješenje se predaje u obliku SQL skripte pod nazivom „Zadatak5.sql“, a zadaci su odvojeni komentarima. Primjerice: --a)... --b) itd.



Prvi kolokvij #5 – grupa B (17. 11. 2015.)

Korisnički zahtjevi

Kreirajte bazu podataka „Restoran“ koja će pamtit i podatke o kuharima, gostima i jelima koja su u ponudi restorana. Za gosta je potrebno pamtit i ime, prezime, JMBAG, kontakt telefon koji je opcionalan za unos te naručeno jelo s jelovnika. Svaki gost mora naručiti neko jelo. Za pojedino jelo potrebno je bilježiti naziv, cijenu i kalorijsku vrijednost (broj s decimalnim zarezom). Uprava restorana provodi politiku da gost može naručiti samo jedno jelo, dok jedno jelo može naručiti više gostiju. Jedno jelo pripravlja samo jedan kuhar za kojeg se pamti ime, prezime, broj licence i datum rođenja. Kuhar zna i može spremati više jela. Također potrebno je pamtit i od kojih se namirnica sastoji jelo. Namirnice se mogu pojavljivati u više jela. Za namirnice je potrebno pamtit i naziv i cijenu po kilogramu.

Zadaci:

1. Izradite dijagram entiteta uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „Zadatak1.graphml“.
2. Izradite ER dijagram uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „Zadatak2.graphml“.
3. Na osnovu izrađenog ER modela kreirajte relacijski shemu baze uz pomoć SQL servera. Rješenje je potrebno predati u obliku SQL skripte koju DBMS sam generira, a naziv skripte treba biti „Zadatak3.sql“.
4. Na osnovu izrađenog ER modela kreirajte relacijsku shemu baze vlastitom skriptom. Rješenje je potrebno predati u obliku SQL skripte pod nazivom „Zadatak4.sql“.
5. Na testnoj bazi „SimpleWebShop“ dohvatite sljedeće podatke:
 - a) iz tablice Stavka dohvatite sve stavke čija je količina jednaka 2
 - b) dohvatite sve podatke iz tablice Stavka sortirane rastuće prema količini
 - c) dohvatite sve kupce čije ime ima prvo slovo jednako 'z', a treće slovo jednako 'c'
 - d) dohvatite prvu državu u tablici Država.

Rješenje se predaje u obliku SQL skripte pod nazivom „Zadatak5.sql“, a zadaci su odvojeni komentarima. Primjerice: --a)... --b) itd.



Prvi kolokvij #6 – grupa B (06. 12. 2016.)

Korisnički zahtjevi

Kreirajte bazu podataka „Lingvist“ koja će pamtit i podatke o jezicima, predavačima i polaznicima škole stranih jezika. Za polaznike je potrebno bilježiti ime, prezime, OIB(11 znakova), telefon opcionalan za unos i strane jezike koje je polaznik upisao. Jedan strani jezik može upisati više polaznika. Za strane jezike potrebno je bilježiti naziv, stupanj, cijenu (realan broj) i predavača koji ga predaje. Za predavača je potrebno bilježiti ime, prezime i datum rođenja. Jedan predavač može predavati više stranih jezika. Satovi stranih jezika održavaju se u učionicama za koje je potrebno bilježiti naziv i broj sjedećih mjesta s predefiniranom vrijednošću 10. Jedan strani jezik može se održavati u samo jednoj učionici, dok se u jednoj učionici može održavati više stranih jezika. Strani jezik ne mora nužno imati unesenu učionicu u kojoj se održava.

Zadaci:

1. Izradite dijagram entiteta uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „Zadatak1.graphml“.
2. Izradite ER dijagram uz pomoć yEd Graph Editora – rješenje se predaje u datoteci pod nazivom „Zadatak2.graphml“.
3. Na osnovu izrađenog ER modela kreirajte relacijski shemu baze uz pomoć SQL servera. Rješenje je potrebno predati u obliku SQL skripte koju DBMS sam generira, a naziv skripte treba biti „Zadatak3.sql“.
4. Na osnovu izrađenog ER modela kreirajte relacijsku shemu baze vlastitom skriptom. Rješenje je potrebno predati u obliku SQL skripte pod nazivom „Zadatak4.sql“.
5. Na testnoj bazi „SimpleWebShop“ dohvatite sljedeće podatke:
 - a) dohvatite popust i ukupnu cijenu stavke iz tablice Stavka, te preimenujte stupce u 'Popust' i 'Cijena'
 - b) dohvatite sve podatke računa čija je vrijednost KupacID između 200 i 300 (uključujuće)
 - c) dohvatite sve različite proizvode koji su crne boje
 - d) dohvatite sve podatke iz tablice Proizvod sortirane silazno padajuće) po cijeni, te uzlazno po nazivu proizvoda
 - e) dohvatite sve podatke kupaca čije ime počinje slovom 'e' ili 'l', a drugo slovo od kraja imena je slovo 'i'
 - f) dohvatite prvih 10 najjeftinijih proizvoda

Rješenje se predaje u obliku SQL skripte pod nazivom „Zadatak5.sql“, a zadaci su odvojeni komentarima. Primjerice: --a)... --b) itd.



Prvi kolokvij #7 – grupa A (21. 11. 2017.)

Korisnički zahtjevi

Kreirajte bazu tvrtke FPZ_RoomEscape koja će voditi evidenciju o sobama, zaposlenicima i timovima koji su odigrali pojedinu sobu. Koncept RoomEscape-a je pobjeći iz sobe rješavanjem logičkih zadataka unutar određenog vremena uz mogućnost traženja pomoći (hintova). Za pojedinu sobu potrebno je bilježiti naziv (jedinstven), cijenu (decimalan broj s ukupno 4 znamenke, a 1 znamenkom iza decimalnog zareza), težinu (prirodan broj unutar intervala [1, 5], opcionalno za unos), detaljan opis sobe (maksimalan broj znakova), maksimalno dozvoljeno vrijeme unutar kojeg se mora izaći iz sobe (tip podatka za vrijeme) te najbolje ostvareno vrijeme koje je nepoznato prilikom unosa sobe u bazu podataka (tip podatka za vrijeme). Za svaku sobu mora se znati koji zaposlenik je zadužen za nju (točno jedan) tzv. game master.

Za zaposlenike koji rade u tvrtki potrebno je bilježiti ime, prezime, datum rođenja (opcionalno za unos), telefon (maksimalno 12 ne-dijakritičkih znakova) i nadređenog zaposlenika, odnosno voditelja. Pojedini zaposlenik može biti zadužen za vođenje/nadgledanje više soba. Zaposlenik može imati samo jednog voditelja, a voditelj ima grupu zaposlenika za koji je zadužen. Voditelj nema voditelja iznad sebe.

Svaku sobu može odigrati više timova (ne istovremeno), ali isto tako svaki tim može odigrati više soba (ne istovremeno). Za pojedini tim potrebno je bilježiti naziv tima, broj članova u timu te sobe koje su odigrali. Za timove koji su odigrali neku sobu potrebno je bilježiti također i vrijeme unutar kojeg su riješili sobu (realan broj), koliko pomoći (hintova) su zatražili, datum igranja (predefinirana vrijednost je trenutni datum) i koliko su ukupno s mogućim popustom platili sobu (tip podatka za novac). Vrijednosti koji nisu navedene da su opcionalne za unos, moraju se unijeti.

1. zadatak (8 bodova)

Na temelju korisničkih zahtjeva izradite dijagram entiteta i ER dijagram uz pomoć yEd Graph Editora. Rješenje se predaje u datoteci pod nazivom zadatak1.graphml.

2. zadatak (6 bodova)

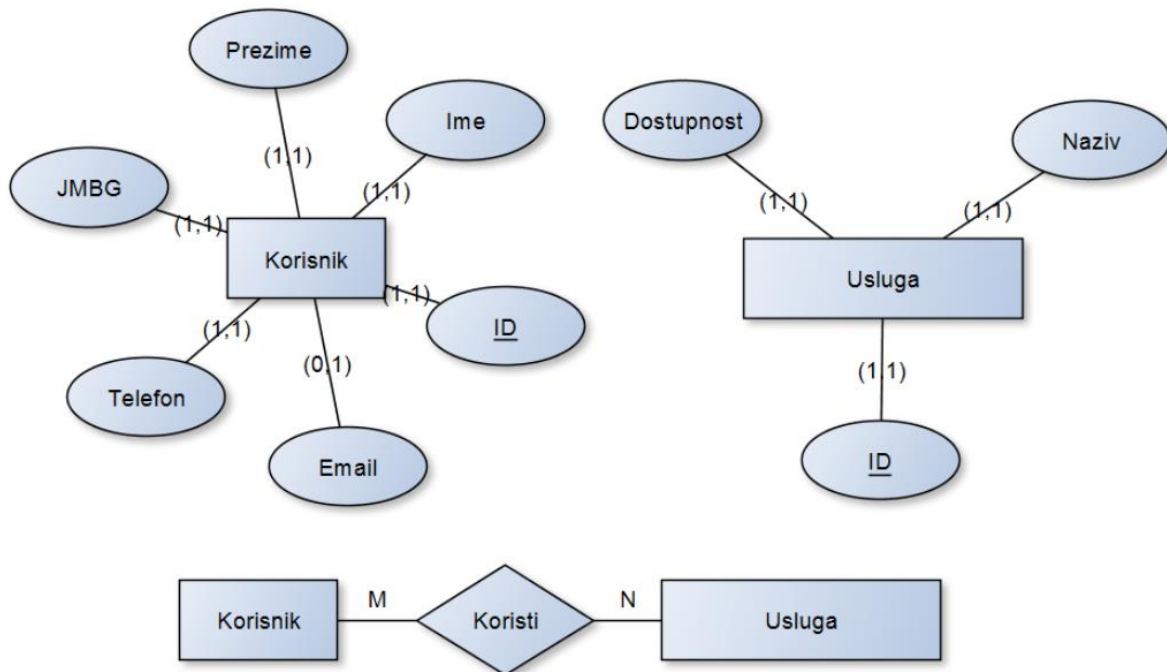
Na osnovu izrađenog ER modela kreirajte relacijsku shemu baze podataka vlastitom skriptom. Na kraju je potrebno napisati naredbu za brisanje baze podataka. Rješenje se predaje u datoteci pod nazivom Zadatak2.sql.



3. zadatak (4 boda)

Na osnovi izrađenog ER modela prikazanog na slici 1 kreirajte relacijsku shemu baze podataka Zadatak3 uz pomoć GUI-a (sučelja) u SQL Serveru. Potrebno je napraviti print screen dijagrama baze podataka u kojem se vide svi nazivi stupaca tablica, tipovi podataka, obavezno/neobavezno za unos i veze između tablica. Rješenje se predaje u datoteci pod nazivom Zadatak3.png. Potrebno je uzeti u obzir i sljedeće stavke:

- Dostupnost – bool vrijednost
- JMBG – 13 znakova



Slika 1. Dijagram entiteta i ER dijagram

4. zadatak (7 bodova)

A testnoj bazi SimpleWebShop riješite zadatke u nastavku. Rješenje se predaje u datoteci pod nazivom Zadatak4.sql, a zadaci su odvojeni i naznačeni komentarima.

- dohvatite tip i broj kreditne kartice svih kreditnih kartica, te spojite stupac u jedan s razmakom između njih i preimenujte ga u „Tip i broj“
- dohvatite sve podatke kreditnih kartica koje ističu u 11 ili 12 mjesecu 2006. godine
- dohvatite ID kreditne kartice, mjesec isteka i godinu isteka svih kartica, ali tako da poredate od onih koje najranije ističu do onih koje najkasnije ističu (i po godini i po mjesecu)
- dohvatite sve podatke kreditnih kartica čiji broj na drugome mjestu ne sadrži znakove 3, 5 ili 4, te između drugog znaka i kraja sadrži znakovni niz 11.
- dohvatite prvih 15% računa uzlazno poredanih po ID-u kupca. Potrebno je prikazati stupce broja računa i ID-a kupca.
- dohvatite sve podatke računa koji imaju unesenu kreditnu karticu s kojom je račun plaćen
- dohvatite sve različite mjesece i godine isteka kreditnih kartica



Korisnički zahtjevi

Kreirajte bazu podataka Veterina tvrtke FPZ_Veterina koja želi bilježiti podatke o veterinarima koji rade kod njih, kućnim ljubimcima koji su došli na pregled i njihovim vlasnicima. Za veterinara koji radi u tvrtki potrebno je bilježiti ime, prezime, broj licence (cijeli broj, jedinstven), telefon (opcionalan za unos), veterinara koji je mu je nadređen (voditelj veterinarske stanice) i kućne ljubimce koje je veterinar pregledao. Nadređeni veterinar vodi sve veterinare u veterinarskoj stanici i nema nadređenu osobu, a svaki veterinar može imati samo jednog nadređenog veterinara. Jedan veterinar može pregledati više kućnih ljubimaca (ne istovremeno), dok jedan kućni ljubimac može doći više puta na pregled kod različitih veterinara. Za kućnog ljubimca potrebno je bilježiti ime, vrstu (pas, mačka, papiga itd.), broj čipa (npr. MP26, jedinstven), godine starosti (prirodan broj unutar intervala [0, 25]) i njegovog vlasnika. Prilikom pregleda kućnog ljubimca potrebno je bilježiti veterinara koji ga je pregledao, datum i vrijeme prijama (tip podatka za datum i vrijeme, predefiniрана vrijednost je trenutni datum i vrijeme), trajanje pregleda (u minutama, realan broj, opcionalno za unos), cijenu pregleda (tip podataka za novac) i opis dijagnoze (maksimalan broj znakova). Za vlasnike kućnih ljubimaca potrebno je bilježiti ime, prezime, OIB (11 ne-dijakritičkih znakova) i telefon. Jedan vlasnik može imati više kućnih ljubimaca, dok jedan ljubimac ima samo jednog vlasnika. Vrijednosti koji nisu navedene da su opcionalne za unos, moraju se unijeti.

1. zadatak (8 bodova)

Na temelju korisničkih zahtjeva izradite dijagram entiteta i ER dijagram uz pomoć yEd Graph Editora. Rješenje se predaje u datoteci pod nazivom zadatak1.graphml.

2. zadatak (6 bodova)

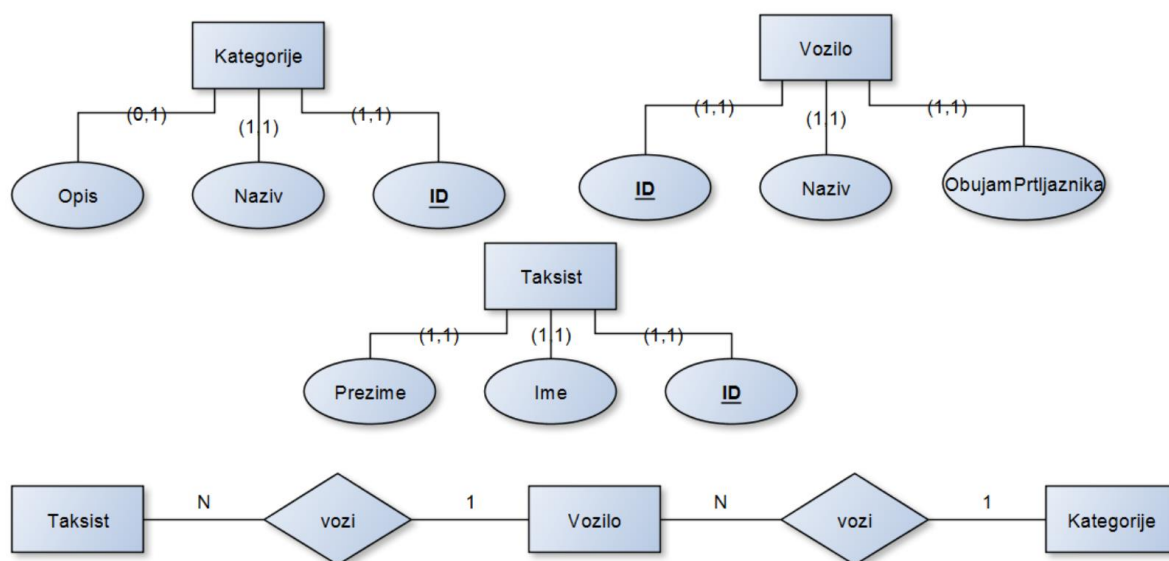
Na osnovu izrađenog ER modela kreirajte relacijsku shemu baze podataka vlastitom skriptom. Na kraju je potrebno napisati naredbu za brisanje baze podataka. Rješenje se predaje u datoteci pod nazivom Zadatak2.sql.



3. zadatak (4 boda)

Na osnovi izrađenog ER modela prikazanog na slici 1 kreirajte relacijsku shemu baze podataka Zadatak3 uz pomoć GUI-a (sučelja) u SQL Serveru. Potrebno je napraviti print screen dijagrama baze podataka u kojem se vide svi nazivi stupaca tablica, tipovi podataka, obavezno/neobavezno za unos i veze između tablica. Rješenje se predaje u datoteci pod nazivom Zadatak3.png. Potrebno je uzeti u obzir i sljedeće stavke:

- ObujamPrtljavnika – decimalan broj s ukupno 5 znamenki od čega su 2 iza decimalnog zareza
- svaka kategorija ima barem jedno vozilo, no svako vozilo ne mora pripadati u neku od kategorija



Slika 1. Dijagram entiteta i ER dijagram

4. zadatak (7 bodova)

Na testnoj bazi SimpleWebShop riješite zadatke u nastavku. Rješenje se predaje u datoteci pod nazivom Zadatak4.sql, a zadaci su odvojeni i naznačeni komentarima.

- dohvatite naziv, boju i cijenu bez PDV-a svih proizvoda. Stupac cijenu bez PDV-a preimenujte u Cijena
- dohvatite sve podatke proizvoda koji imaju ID unutar intervala [300, 500] u kojima je boja crna ili crvena
- dohvatite broj računa, datum izdavanja računa i ID kupca svih računa. Dohvat poredajte od najnovije izdanih računa do najstarije izdanih računa te za račune koji su izdani na isti datum potrebno je i uzlazno sortiranje po ID-u kupca
- dohvatite sve podatke proizvoda čiji naziv ne sadrži znakovni niz 'set', a predzadnje slovo mora biti iz skupa a, b ili c
- dohvatite prvih 20 stavki s najvećim iznosom popusta. Potrebno je prikazati stupce ID stavke i popust
- dohvatite sve podatke kupaca koji nemaju unesen pripadajući grad
- dohvatite sva različita prezimena i ID-eve gradova kupaca.



Prvi kolokvij #9 – grupa B (20. 11. 2018.)

Korisnički zahtjevi

Kreirajte bazu podataka IToprema tvrtke FPZ_ITProdaja koja želi bilježiti podatke o prodavačima koji rade u tvrtki, uređajima koje prodaje i kupcima koji su kupili uređaje. Za prodavače koji rade u tvrtki potrebno je bilježiti ime, prezime, datum zaposlenja, ocjenu kvalitete rada (prirodan broj unutar intervala $[0, 5]$, početna vrijednost je 2) i uređaje koje prodaje. Jedan prodavač može prodavati više uređaja, no jedan uređaj prodaje točno jedan prodavač, no može se desiti da postoji uređaj u bazi podataka koji nema dodijeljenog prodavača. Za uređaj koji je u ponudi tvrtke (npr. Samsung Galaxy S7) potrebno je bilježiti naziv, vrstu uređaja (npr. pametni telefon), dostupnu količinu (cijeli broj), cijenu bez popusta (decimalan broj s ukupno 6 znamenki, od čaga su 2 iza decimalnog zareza) i specifikacije (maksimalan broj znakova). Za kupca je potrebno bilježiti ime, prezime, email (opcionalan za unos, bez dijakritičkih znakova), šifru kupca (10 znakova, jedinstven) i uređuje koje je kupio. Jedan kupac može kupiti više uređaja, dok jedan uređaj može kupiti više kupaca (ne isti uređaj već tog tipa npr. Samsung Galaxy S7). Za kupca koji je kupio određeni uređaj potrebno je bilježiti datum kupnje (predefinirana vrijednost je trenutni datum, opcionalno za unos), kupljenu količinu (cijeli broj), popust (realan broj) i ukupnu cijenu (tip podataka za novac, opcionalno za unos) za određeni uređaj.

Vrijednosti koje nisu navedene da su opcionalne za unos moraju se unijeti.

1. zadatak (8 bodova)

Na temelju korisničkih zahtjeva izradite dijagram entiteta i ER dijagram uz pomoć yEd Graph Editora. Rješenje se predaje u datoteci pod nazivom zadatak1.graphml.

2. zadatak (6 bodova)

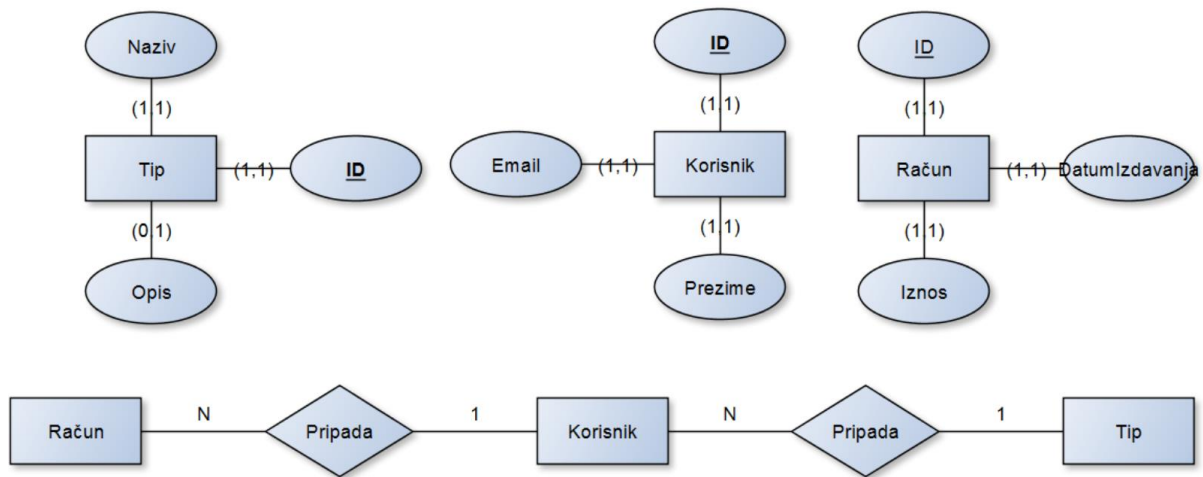
Na osnovu izrađenog ER modela kreirajte relacijsku shemu baze podataka vlastitom skriptom. Na kraju je potrebno napisati naredbu za brisanje baze podataka. Rješenje se predaje u datoteci pod nazivom Zadatak2.sql.



3. zadatak (4 boda)

Na osnovi izrađenog ER modela prikazanog na slici 1 kreirajte relacijsku shemu baze podataka Zadatak3 uz pomoć GUI-a (sučelja) u SQL Serveru. Potrebno je napraviti print screen dijagrama baze podataka u kojem se vide svi nazivi stupaca tablica, tipovi podataka, obavezno/neobavezno za unos i veze između tablica. Rješenje se predaje u datoteci pod nazivom Zadatak3.png. Potrebno je uzeti u obzir i sljedeće stavke:

- Opis – maksimalan broj znakova



Slika 1. Dijagram entiteta i ER dijagram

4. zadatak (7 bodova)

Na testnoj bazi SimpleWebShop riješite zadatke u nastavku. Rješenje se predaje u datoteci pod nazivom Zadatak4.sql, a zadaci su odvojeni i naznačeni komentarima.

- dohvatite broj računa, komentar i ID kupca svih računa. Stupac ID-a kupca preimenujte u Kupac, a stupac broj računa u Broj.
- dohvatite sve podatke računa koji imaju ID unutar intervala [50000, 55000] i kojima je ID kupca jednak 6877 ili 7635
- dohvatite ID stavke, popust i ukupnu cijenu svih stavaka, ali poredajte po iznosu popusta silazno. Za one stavke koje imaju jednak iznos popusta potrebno je i sortiranje po ukupnoj cijeni uzlazno
- dohvatite sve podatke proizvoda čiji naziv završava znakom g, drugi znak je u rasponu od b do h, a treći znak u nazivu nije a.
- dohvatite prvih 20 stavki s najmanjim iznosom popusta. Potrebno je prikazati stupce ID stavke i popust
- dohvatite sve podatke računa koji nemaju unesenu kreditnu karticu s kojom je račun plaćen
- koliko ima različitih ID-eva računa i količina stavaka iz tablice Stavka. Rezultat navedite u komentaru, a mora biti vidljiva i SELECT naredba



Drugi kolokviji

Drugi kolokvij #9 – grupa A (12. 01. 2016.)

Ubacivanje, brisanje i ažuriranje podataka (3 boda)

1. U tablicu KreditnaKartica ubacite novu karticu čije je tip 'Visa', broj 2562, mjesec isteka je rujan, a godina isteka 2018. (1 bod)
2. Obrišite sve kupce koji nemaju upisan email. (1 bod)
3. Svim računima koji nemaju upisanu vrijednost komentara dodajte komentar 'Nema komentara'. (1 bod)

Dohvaćanje podataka iz više tablica (14 bodova)

4. Za svakog kupca dohvatite ime, prezime i naziv grada iz kojeg dolazi. (3 boda)
5. Za svaki proizvod dohvatite njegov naziv, te ukupnu prodanu količinu tog proizvoda (tablica stavke). (3 boda)
6. Za svakog kupca dohvatite koliko ima zabilježenih računa. Potrebno je dohvatiti i one kupce koji nemaju niti jedan račun. (3 boda)
7. Za svaki popust u postocima dohvatite koliko stavaka ima isti popust. (3 boda)
8. Dohvatite najskuplji proizvod crne boje. (2 boda)

Normalizacija baze podataka (4 boda)

9. Napišite skriptu (kreiranje novih tablica, te njihovo punjenje) koja normalizira sljedeće podatke (4 boda)

Skijalište	Staza 1	Staza 2
Sljeme	Bijeli spust	Zeleni spust
Platak	Tešnje	Radeševo 1
Bjelolasica	Vrelo	Livade

Postavljanje indeksa (4 boda)

10. Postavite neklasterirani indeks da ubrzate upit ispod prema broju logičkih čitanja (Uključivanje statistike: SET STATISTICS IO ON) (4 boda)

```
SELECT Kolicina, UkupnaCijena  
FROM Stavka  
WHERE Kolicina < 30 AND UkupnaCijena > 2000
```



Drugi kolokvij #10 – grupa E (12. 01. 2016.)

Ubacivanje, brisanje i ažuriranje podataka (3 boda)

1. U tablicu KreditnaKartica ubacite novu karticu čije je tip 'Visa', broj '12082205041216', mjesec isteka siječanj i godina isteka 2020. (1 bod)
2. Obrišite kupca čiji je email 'michael7@adventure-works.com' i prezime počinje slovom 'z'. (1 bod)
3. Svim proizvodima čija je boja nepoznata (NULL) promijenite vrijednost boje u 'Nepoznata'. (1 bod)

Dohvaćanje podataka iz više tablica (14 bodova)

4. Za svaki račun plaćen karticom dohvatite broj računa te tip i broj kreditne kartice. (3 boda)
5. Za svakog kupca dohvatite ime i prezime te koliko je računa napravio. (3 boda)
6. Za svakog kupca dohvatite prezime i email te broj i tip svih kartica koje je koristio za kupovinu. (3 boda)
7. Za sve proizvode plave boje ispišite naziv proizvoda i ukupnu prodanu količinu (stupac nazovite 'BrojProdanihProizvoda') tih proizvoda (tablica stavka). (3 boda)
8. Dohvatite nazive svih proizvoda čija je cijena bez PDV-a manja ili jednaka prosječnoj cijeni bez PDV-a svih proizvoda. (2 boda)

Normalizacija baze podataka (4 boda)

9. Napišite skriptu (kreiranje novih tablica, te njihovo punjenje) koja normalizira sljedeće podatke (4 boda)

Grad2	Država2	PovršinaDržave
Zagreb	Hrvatska	56594
Gold Coast	Queensland	1852642
Osijek	Hrvatska	56594
Sunshine Coast	Queensland	1852642

Postavljanje indeksa (4 boda)

10. Postavite neklasterirani indeks da ubrzate upit ispod prema broju logičkih čitanja (Uključivanje statistike: SET STATISTICS IO ON) (4 boda)

```
SELECT IDStavka, UkupnaCijena
FROM Stavka
WHERE CijenaPoKomadu < 100 AND Kolicina >= 2
```



Drugi kolokvij #11 – Grupa F (12. 01. 2016)

Ubacivanje, brisanje i ažuriranje podatka (3 boda)

1. U tablicu Proizvod ubacite novi proizvod čiji je naziv 'Server', broj proizvoda 'SE-4321', boja 'Siva' i čija je cijena 20000. (1 bod)
2. Obrišite kupca čiji je telefon '119-555-0192' i prezime počinje slovom 'd'. (1 bod)
3. Svim kupcima čiji je telefon nepoznat (NULL) promijenite vrijednost telefona u 'Nepoznat'. (1 bod)

Dohvaćanje podataka iz više tablica (14 bodova)

4. Za svaki račun dohvatite broj računa, datum izdavanja te ime i prezime kupca. (3 boda)
5. Za svaku boju ispišite ukupnu cijenu proizvoda te boje. (3 boda)
6. Za svaki račun dohvatite broj računa i datum izdavanja te naziv i cijenu bez PDV-a svih proizvoda koji se nalaze na tom računu. (3 boda)
7. Za svaki tip kreditne kartice ispišite broj kupaca (tablica račun) koji koriste taj tip kreditne kartice. (3 boda)
8. Dohvatite ID svih stavki čija je cijena po komadu veća ili jednaka od prosječne cijene po komadu svih stavki. (2 boda)

Normalizacija baze podataka (4 boda)

9. Napišite skriptu (kreiranje novih tablica, te njihovo punjenje) koja normalizira sljedeće podatke. (4 boda)

Država2	Kontinent	PovršinaKontinenta
New South Wales	Australija	8500000
Hrvatska	Europa	10500000
Irska	Europa	10500000
Queensland	Australija	8500000

Postavljanje indeksa (4 boda)

10. Postavite neklasterirani indeks da ubrzate upit ispod prema broju logičkih čitanja (uključivanje statistike: SET STATISTICS IO ON) (4 boda)

```
SELECT IDKupac, Ime, Prezime
FROM Kupac
WHERE Ime LIKE 'A%' AND Prezime LIKE 'M%'
```



Pismeni ispiti - rješenja

Pismeni ispit #12 – grupa A (28.06.2021.)

7. zadatak (1.5 bod)

Korištenjem podupita dohvatite gradove koji nemaju niti jednog kupca

SELECT

*

FROM

Grad g

WHERE

g.IDGrad NOT IN (SELECT DISTINCT k.GradID FROM Kupac k)



7. zadatak (2 boda)

Korištenjem podupita dohvatite nazive proizvoda koji nisu prodani niti jedanput.

```
SELECT  
p.Naziv  
FROM  
Proizvod p  
WHERE  
IDProizvod NOT IN (SELECT DISTINCT ProizvodID FROM Stavka)
```

9. zadatak (1.5 bod)

Pomoću naredbe *geography::STPointFromText('POINT(x y)',4326)* kreirajte dvije geografske točke: Zagreb (15.983, 45.815) i Split (16.456, 43.514), te potom pomoću funkcije *STDistance* dohvatite udaljenost između te dvije točke.

```
DECLARE @Zagreb geography;  
SET @Zagreb = geography::STPointFromText('POINT(15.983 45.815)',4326);  
  
DECLARE @Split geography;  
SET @Split = geography::STPointFromText('POINT(16.456 43.514)',4326);  
  
SELECT @Zagreb.STDistance(@Split) AS 'Udaljenost';
```



7. zadatak (2 boda)

Korištenjem podupita dohvatite one račune koji su plaćeni kreditnom karticom čija je godina isteka veća od prosječne vrijednosti godine isteka svih kreditnih kartica.

```
SELECT
*
FROM
Racun r
INNER JOIN
KreditnaKartica k
ON
r.KreditnaKarticaID=k.IDKreditnaKartica
WHERE
k.IstekGodina > (SELECT AVG(k.IstekGodina) FROM KreditnaKartica k)
```

8. zadatak (3 boda)

Deklarirajte varijablu *ukPlaceniIznos* koja odgovara tipu podataka za stupac *UkupnaCijena* tablice *Stavka*. U varijablu *ukPlaceniIznos* spremite ukupan plaćeni iznos svih stavaka u tablici *Stavka*, te ispišite rezultat u *Messages*.

```
DECLARE @ukPlaceniIznos AS numeric;
SET @ukPlaceniIznos = (SELECT SUM(s.UkupnaCijena) FROM Stavka s);

PRINT CAST(@ukPlaceniIznos AS VARCHAR);
```



Pismeni ispit #15 – grupa A (23.03.2022.)

2. zadatak (3 boda)

Korištenjem isključivo podupita, za sve kupce čija je vrijednost ID-a grada veća od prosječne vrijednosti ID-a grada u tablici *Grad*, dohvatite ukupan broj računa (koliko ih ima) koje je napravio pojedini kupac. U dohvat je potrebno prikazati sve stupce kupca, te dodatno stupac *Ukupan broj računa*.

```
SELECT
k.IDKupac AS 'ID kupca',
k.Ime AS 'Ime kupca',
k.Prezime AS 'Prezime kupca',
k.Email AS 'Email kupca',
k.Telefon AS 'Telefon kupca',
k.GradID AS 'ID grada kupca',
(SELECT COUNT(IDRacun) FROM Racun WHERE KupacID=k.IDKupac) AS 'Ukupan broj
računa'
FROM
Kupac k
WHERE
k.GradID > (SELECT AVG(IDGrad) FROM Grad)
```

7. zadatak (3 boda)

Deklarirajte varijablu *brojGradova*. U varijablu *brojGradova* spremite koliko ima gradova koji se nalaze u državi čiji naziv države počinje znakom *a*. Rezultat ispišite u *Messages*.

```
DECLARE @brojGradova AS int;
SET @brojGradova = (SELECT COUNT(g.IDGrad) FROM Grad g INNER JOIN Drzava d
ON g.DrzavaID=d.IDDrzava WHERE g.Naziv LIKE 'a%');

PRINT CAST(@brojGradova AS VARCHAR);
```



Pismeni ispit #16 – grupa A (23.06.2022.)

1. zadatak (4 boda)

Dohvatite sve stavke čiji je popust u postotcima različit od nule i koje se odnose na proizvode čija je boja *crna*. Potrebno je prikazati stupce *Popust u postotcima* i *Ukupna cijena*. U stupcu *Ukupna cijena* prikazuje se zarada na svim prodanim stavkama koje imaju jednak popust u postotcima.

```
SELECT
s.PopustUPostocima AS 'Popust u postotcima',
SUM(s.UkupnaCijena) AS 'Ukupna cijena'
FROM Stavka s
INNER JOIN Proizvod p ON s.ProizvodID=p.IDProizvod
WHERE
s.PopustUPostocima <> 0
AND
p.Boja = 'Crna'
GROUP BY s.PopustUPostocima;
```

2. zadatak (3 boda)

Korištenjem isključivo podupita, dohvatite sve kreditne kartice čiji tip kreditne kartice odgovara tipu kreditne kartice koji se najčešće pojavljuje u tablici Kreditna kartica.

```
SELECT
*
FROM KreditnaKartica k
WHERE
k.Tip = (SELECT TOP 1 k.Tip FROM KreditnaKartica k GROUP BY k.Tip ORDER BY
COUNT(*) DESC);
```

3. zadatak (1 bod)

Dohvatite sve nazive gradova iz tablice *Grad* te dodatno prikažite stupac *Broj kupaca* u kojem se za svaki grad prikazuje broj kupaca koji dolazi iz tog grada.

```
SELECT
g.Naziv AS 'Naziv grada',
ISNULL(COUNT(k.IDKupac), 0) AS 'Broj kupaca'
FROM Grad g
LEFT JOIN Kupac k ON g.IDGrad=k.GradID
GROUP BY g.Naziv;
```



4. zadatak (4 boda)

Kreirajte *Windows Forms* aplikaciju za rad s bazom *SimpleWebShop*. Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno kao **Zadatak4**.

- c) **(2 boda)** Aplikacija se treba spojiti na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *SQL autentifikacije*. Potrebno je dodati dva *Label*-a s tekstovima: *Korisničko ime* i *Zaporka*; te dva *TextBox*-a naziva: *txtKorisnickolme* i *txtZaporka*. Potrebno je unijeti korisničko ime u *TextBox* naziva *txtKorisnickolme* i zaporku u *TextBox* naziva *txtZaporka*, te pritisnuti gumb s tekstom *Spoji* (naziva *btnSpoji*). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje na bazu SimpleWebShop!*, inače u skočnom prozoru treba ispisati grešku koja se pojavila. (Pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient*; . Za potrebe **b)** dijela zadatka zgodno je instancu konekcije deklarirati na razini klase, ali nije nužno za rješenje ovog dijela zadatka.)
- d) **(2 boda)** U prethodno izrađenu aplikaciju dodajte tri *Label*-a s tekstovima: *Cijena od*, *Cijena do* i *Boja*; i tri *TextBox*-a naziva: *txtcijenaOd*, *txtcijenaDo* i *txtBoja*. Potrebno je dodati gumb s tekstom *Dohvati* (naziva *btnDohvati*) čijim klikom se iz tablice *Proizvod* dohvaćaju svi proizvodi čija je boja jednaka vrijednosti boje u *TextBox*-u *txtBoja* te je cijena proizvoda u rasponu vrijednosti zapisanih u *TextBox*-ovima *txtcijenaOd* i *txtcijenaDo*. Potrebno je koristiti konekciju stvorenu u **a)** zadatku, odnosno varijablu konekcije potrebno je deklarirati na razini klase (ako to već nije prethodno napravljeno).

Izgled sučelja:

The screenshot shows a Windows Forms application titled 'Zadatak 4'. On the left, there is a login section with labels 'Korisničko ime' and 'Zaporka', corresponding text boxes containing 'sa' and 'Baze123', and a 'Spoji' button. Below this is a filter section with labels 'Cijena od', 'Cijena do', and 'Boja', corresponding text boxes containing '10', '100', and 'Cma', and a 'Dohvati' button. On the right, there is a table with 6 columns: IDProizvod, Naziv, BrojProizvoda, Boja, and CijenaBezPDV. The table contains 15 rows of product data, with the first row (ID 708) highlighted in blue.

IDProizvod	Naziv	BrojProizvoda	Boja	CijenaBezPDV
708	Sport-100 Helmet...	HL-U509	Cma	42.3379
815	LL Mountain Fron...	FW-M423	Cma	73.5015
841	Men's Sports Sho...	SH-M897-S	Cma	72.5879
849	Men's Sports Sho...	SH-M897-M	Cma	72.5879
850	Men's Sports Sho...	SH-M897-L	Cma	72.5879
851	Men's Sports Sho...	SH-M897-X	Cma	72.5879
852	Women's Tights, S	TG-W091-S	Cma	90.7379
853	Women's Tights, M	TG-W091-M	Cma	90.7379
854	Women's Tights, L	TG-W091-L	Cma	90.7379
858	Half-Finger Glove...	GL-H102-S	Cma	29.6329
859	Half-Finger Glove...	GL-H102-M	Cma	29.6329
860	Half-Finger Glove...	GL-H102-L	Cma	29.6329
861	Full-Finger Glove...	GL-F110-S	Cma	45.9679
862	Full-Finger Glove...	GL-F110-M	Cma	45.9679



Programski kod:

```
using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Zadatak4
{
    public partial class Form1 : Form
    {
        SqlConnection sql;

        public Form1()
        {
            InitializeComponent();
        }

        private void btnSpoji_Click(object sender, EventArgs e)
        {
            try
            {
                sql = new SqlConnection("Server= DESKTOP-PC\\SQLEXPRESS;
                Database=SimpleWebShop; User ID=" + txtKorisnickoIme.Text
                + "; Password=" + txtZaporka.Text);
                sql.Open();

                MessageBox.Show("Uspješno spajanje na bazu SimpleWebShop!");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void btnDohvati_Click(object sender, EventArgs e)
        {
            if (sql != null)
            {
                SqlCommand cmd = new SqlCommand("SELECT * FROM Proizvod WHERE
                CijenaBezPDV BETWEEN " + txtcijenaOd.Text + " AND "
                + txtcijenaDo.Text + " AND Boja='" + txtBoja.Text + "'", sql);
                SqlDataReader reader = cmd.ExecuteReader();
                DataTable dt = new DataTable();
                dt.Load(reader);
                dataGridView1.DataSource = dt;
            }
            else
            {
                MessageBox.Show("Prije dohvata podataka morate se spojiti na
                bazu!");
            }
        }
    }
}
```



5. zadatak (2 boda)

Dohvatite račune iz tablice *Račun* koji pripadaju kupcima čije ime sadrži znakove *me* te imaju zapisan telefon. Potrebno je prikazati prvih 10% računa sortiranih prema ID računa od najmanjeg prema najvećem koji pripadaju kupcima s različitim prezimenima. U dohvat se prikazuje stupac *Prezime* te svi stupci iz tablice *Račun*. Potrebno je u komentarima naznačiti po koracima redoslijed dohvata.

```
SELECT DISTINCT TOP 10 PERCENT
k.Prezime AS 'Prezime kupca',
r.IDRačun,
r.DatumIzdavanja,
r.BrojRačuna,
r.KupacID,
r.KreditnaKarticaID,
r.Komentar
FROM Račun r
INNER JOIN
Kupac k ON r.KupacID=k.IDKupac
WHERE
k.Ime LIKE '%me%'
AND k.Telefon IS NOT NULL
ORDER BY r.IDRačun ASC
```

--REDOSLIJED DOHVATA:

```
--1.
--FROM Račun r
--INNER JOIN
--Kupac k ON r.KupacID=k.IDKupac

--2.
--WHERE
--k.Ime LIKE '%me%'
--AND k.Telefon IS NOT NULL

--3.
--k.Prezime AS 'Prezime kupca',
--r.IDRačun,
--r.DatumIzdavanja,
--r.BrojRačuna,
--r.KupacID,
--r.KreditnaKarticaID,
--r.Komentar

--4.
--DISTINCT

--5.
--ORDER BY r.IDRačun DESC

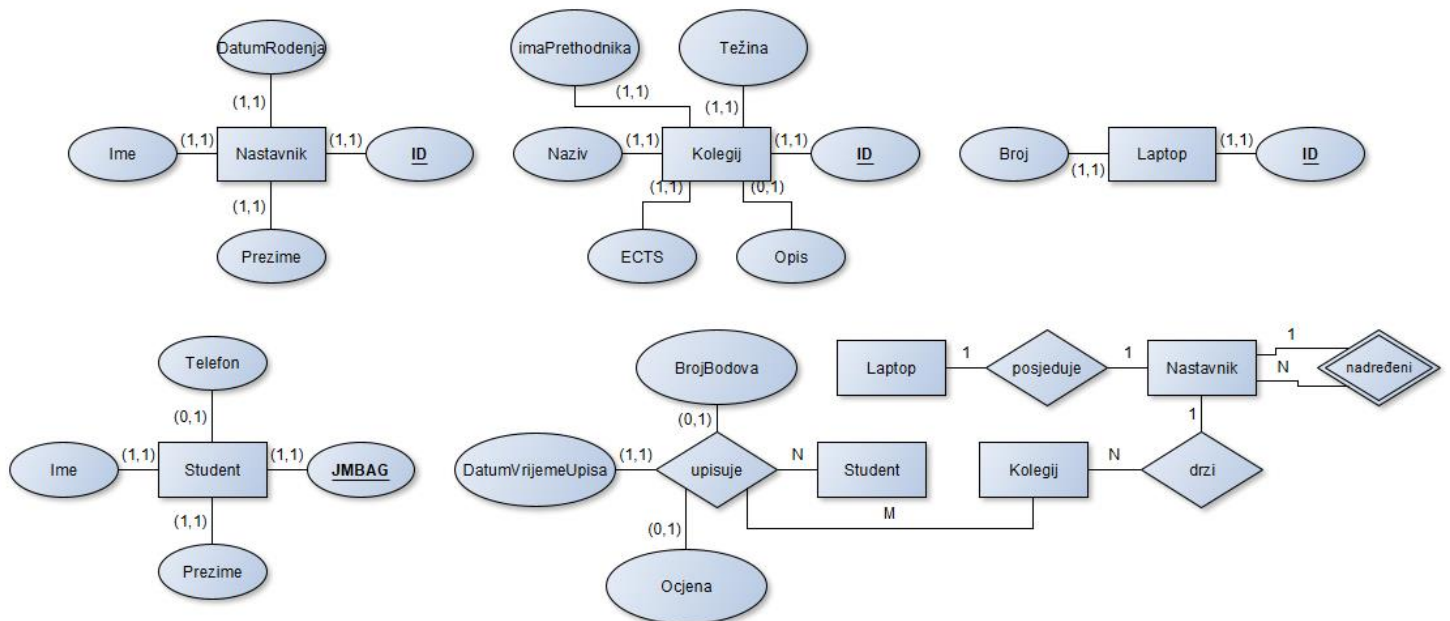
--6.
--TOP 10 PERCENT
```



6. zadatak (8 bodova)

Na osnovu izrađenog ER modela prikazanog na sljedećoj slici kreirajte **relacijsku shemu** baze podataka *Referada* pisanjem SQL koda. Potrebno je u svaku kreiranu tablicu **unijeti minimalno jedan redak s proizvoljnim vrijednostima** te postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa. Kao rješenje predaje se **vlastito napisana SQL skripta**. Datoteku je potrebno spremiti pod nazivom **Zadatak6.sql** te na kraju obrisati kreiranu bazu podataka. Potrebno je u obzir uzeti i sljedeće uvijete:

- Telefon – točno 13 nedijakritičkih znakova
- Broj – jedinstven
- Naziv – predefinirana vrijednost *Proizvoljan kolegij*
- ImaPrethodnika – tip podataka za istinu ili laž
- Težina – cijeli broj unutar intervala [0, 255]
- BrojBodova – decimalan broj od ukupno 6 znakova, a maksimalno 4 iza decimalnog zareza
- svaki kolegij ne mora imati pridruženog nastavnika koji ga drži
- nastavnik mora imati laptop, dok svaki laptop ne mora nužno pripadati nastavniku



7. zadatak (3 boda)

Deklarirajte varijablu *p1* i varijablu *p2*, obje tipa *geometry*, koje predstavljaju točke u koordinatnom sustavu. Pozicija točke *p1* je (2, 10), a točke *p2* (4, 14). Izračunajte i ispišite u *Messages* udaljenost između te dvije točke. Za izračun udaljenosti koristite metodu *STDistance*.

```
DECLARE @p1 geometry;  
SET @p1 = geometry::STPointFromText('POINT(2 10)', 0);  
  
DECLARE @p2 geometry;  
SET @p2 = geometry::STPointFromText('POINT(4 14)', 0);  
  
PRINT 'Udaljenost između točaka: ' + CAST(@p1.STDistance(@p2) AS VARCHAR);
```



Pismeni ispit #17 – grupa A (07.07.2022.)

2. zadatak (1 bod)

Svim računima koji imaju uneseni komentar i koji su izdani u 2001. ili 2003. godini postavite komentar na *Izdani u 2001. ili 2003. godini!*.

UPDATE

Racun

SET

Komentar = 'Izdani u 2001. ili 2003. godini!'

WHERE

Komentar IS NOT NULL

AND YEAR(DatumIzdavanja) IN (2001, 2003);

3. zadatak (2 boda)

Dohvatite ime i prezime kupaca iz tablice *Kupac* te dodatno prikažite stupac *Broj racuna* u kojem se za svakog kupca prikazuje broj računa koje je taj kupac generirao.

SELECT

k.Ime,

k.Prezime,

COUNT(r.KupacID) AS 'Broj racuna'

FROM Kupac k

INNER JOIN Racun r ON k.IDKupac=r.KupacID

GROUP BY k.Ime, k.Prezime

4. zadatak (3 boda)

Korištenjem podupita dohvatite ID i naziv grada onih gradova koji imaju barem jednog kupca. Svi stupci moraju biti odgovarajuće nazvani.

SELECT

g.IDGrad AS 'ID grada',

g.Naziv AS 'Naziv grada'

FROM Grad g

WHERE

g.IDGrad IN

(SELECT DISTINCT k.GradID FROM Kupac k)



5. zadatak (3 boda)

Dohvatite naziv države i ukupan broj kupaca koji dolazi iz te države. U ukupan broj kupaca iz neke države uključeni su samo oni kupci kojima je poznat *email* i *telefon*. Prikazuju se samo one države iz kojih dolazi više od 5000 kupaca. Svi stupci moraju biti odgovarajuće nazvani.

```
SELECT
d.Naziv AS 'Država',
COUNT(k.IDKupac) AS 'Broj kupaca'
FROM Drzava d
INNER JOIN Grad g ON d.IDDrzava=g.DrzavaID
INNER JOIN Kupac k ON g.IDGrad=k.GradID
WHERE
k.Email IS NOT NULL
AND
k.Telefon IS NOT NULL
GROUP BY d.Naziv
HAVING
COUNT(k.IDKupac) > 5000
```

6. zadatak (2 boda)

Dohvatite različite nazive proizvoda i njihove ukupne cijene iz tablice *Stavka*. Dohvaćaju se oni proizvodi čija je količina jednaka 1, a u broju proizvoda sadrže znakove *so*. Potrebno je prikazati prvih 5 proizvoda sortiranih prema ukupnoj cijeni od najveće prema najmanjoj.

```
SELECT DISTINCT TOP 5
p.Naziv AS 'Naziv proizvoda',
s.UkupnaCijena AS 'Ukupna cijena'
FROM Proizvod p
INNER JOIN Stavka s ON p.IDProizvod=s.ProizvodID
WHERE
s.Kolicina = 1
AND
p.BrojProizvoda LIKE '%so%'
ORDER BY s.UkupnaCijena DESC
```



7. zadatak (3 boda)

Kreirajte *Windows Forms* aplikaciju za rad s bazom *SimpleWebShop*. Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno kao **Zadatak7**.

- a) **(2 boda)** Aplikacija se treba spojiti na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *SQL autentifikacije*. Potrebno je dodati dva *Label*-a s tekstovima: *Korisničko ime* i *Zaporka*; te dva *TextBox*-a naziva: *txtKorisnickoIme* i *txtZaporka*. Potrebno je unijeti korisničko ime u *TextBox* naziva *txtKorisnickoIme* i zaporku u *TextBox* naziva *txtZaporka*, te pritisnuti gumb s tekstom *Spoji* (naziva *btnSpoji*). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje na bazu SimpleWebShop!*, inače u skočnom prozoru treba ispisati grešku koja se pojavila. (Pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient*;). Za potrebe b) dijela zadatka zgodno je instancu konekcije deklarirati na razini klase, ali nije nužno za rješenje ovog dijela zadatka.)
- b) **(2 bod)** U prethodno izrađenu aplikaciju dodajte gumb s tekstom *Dohvati* (naziva *btnDohvati*) čijim klikom se iz tablice *Kreditna kartica* dohvaćaju sve kreditne kartice, na način da se unutar *DataGridView* kontrole naziva *tablica* prikažu svi dohvaćeni podaci. Potrebno je koristiti konekciju stvorenu u a) zadatku, odnosno varijablu konekcije potrebno je deklarirati na razini klase (ako to već nije prethodno napravljeno).

Izgled sučelja:

IDKreditnaKartica	Tip	Broj	IstekMjesec	IstekGodina
1	American Express	33332664695310	11	2006
2	MasterCard	55552127249722	8	2005
3	Visa	77778344838353	7	2005
4	Visa	77774915718248	7	2006
5	Diners	11114404600042	4	2005
6	MasterCard	55557132036181	9	2006
7	MasterCard	55553635401028	6	2007
8	American Express	33336081193101	7	2007
9	MasterCard	55553465625901	2	2005
10	American Express	33332126386493	8	2008
11	American Express	33335352517363	10	2008
12	American Express	33334316194519	4	2008



Programski kod:

```
using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Zadatak_7
{
    public partial class Form1 : Form
    {
        SqlConnection sql;

        public Form1()
        {
            InitializeComponent();
        }

        private void btnSpoji_Click(object sender, EventArgs e)
        {
            try
            {
                sql = new SqlConnection("Server= DESKTOP-PC\\SQLEXPRESS;
                Database=SimpleWebShop; User ID=" + txtKorisnickoIme.Text
                + "; Password=" + txtZaporka.Text);
                sql.Open();

                MessageBox.Show("Uspješno spajanje na bazu SimpleWebShop!");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void btnDohvati_Click(object sender, EventArgs e)
        {
            if (sql != null)
            {
                SqlCommand cmd = new SqlCommand
                ("SELECT * FROM KreditnaKartica", sql);
                SqlDataReader reader = cmd.ExecuteReader();
                DataTable dt = new DataTable();
                dt.Load(reader);
                tablica.DataSource = dt;
            }
            else
            {
                MessageBox.Show("Prije dohvata podataka morate se spojiti na
                bazu!");
            }
        }
    }
}
```



8. zadatak (3 boda)

Od svih računa koji su izdani prije današnjeg datuma i čija je vrijednost stupca *KupacID* veća od 16000, dohvatite prvih 10 koji imaju najmanju vrijednost stupca *KupacID*. Potrebno je prikazati stupce *DatumIzdavanja* i *IDRacun*. Postavite neklasterirani indeks da što više ubrzate navedeni upit prema broju logičkih čitanja. Potrebno je komentarom označiti odabir indeksa i broj logičkih čitanja, te napisati naredbu za brisanje indeksa.

```
SET STATISTICS IO ON;
```

```
SELECT TOP 10  
r.DatumIzdavanja,  
r.IDRacun  
FROM Racun r  
WHERE  
r.DatumIzdavanja < GETDATE()  
AND  
r.KupacID > 16000  
ORDER BY KupacID ASC
```

```
--broj logičkih čitanja bez postavljanja indeksa: 185
```

```
--u obzir za neklasterirani indeks dolaze DatumIzdavanja i KupacID
```

```
CREATE NONCLUSTERED INDEX i1 ON Racun(DatumIzdavanja) INCLUDE (KupacID)  
--broj logičkih čitanja sa i1: 88  
DROP INDEX Racun.i1;
```

```
CREATE NONCLUSTERED INDEX i2 ON Racun(KupacID) INCLUDE (DatumIzdavanja)  
--broj logičkih čitanja sa i2: 2  
DROP INDEX Racun.i2
```

```
--bolji odabir je indeks i2 jer ima manje logičkih čitanja
```

```
SET STATISTICS IO OFF
```



Pismeni ispit #18 – grupa A (08.09.2022.)

1. zadatak (3 boda)

Deklarirajte varijablu *brojRacuna*. U varijablu *brojRacuna* spremite koliko ima računa koji nemaju zabilježen komentar, a koji su plaćeni kreditnom karticom tipa *Visa*. Rezultat ispišite u *Messages* u formatu *Broj računa bez komentara koji su plaćeni kreditnom karticom tipa Visa je X* (umjesto vrijednosti *X* treba ispisati vrijednost).

```
DECLARE @brojRacuna AS int;
SET @brojRacuna =
(SELECT COUNT(r.IDRacun) FROM Racun r
INNER JOIN KreditnaKartica k ON r.KreditnaKarticaID=k.IDKreditnaKartica
WHERE r.Komentar IS NULL AND k.Tip='Visa')

PRINT 'Broj računa bez komentara koji su plaćeni kreditnom karticom tipa
Visa je ' + CAST(@brojRacuna AS VARCHAR)
```

2. zadatak (2 boda)

Pomoću naredbe *geometry::STPolyFromText('POLYGON((x1 y1, x2 y2, ... ,xn yn, x1 y1))', 0)* kreirajte proizvoljni poligon od minimalno 7 točaka, te ga dohvatite.

```
DECLARE @poligon geometry

SET @poligon = geometry::STPolyFromText('POLYGON((0 0, 0 2, 2 6, 4 2, 4 0,
2 2, 0 0))',0)

SELECT @poligon
```



3. zadatak (2 boda)

Od svih proizvoda čiji broj proizvoda počinje znakom s i čija je vrijednost stupca *CijenaBezPDV* veća od 0, dohvatite prvih 10% najskupljih proizvoda. Potrebno je prikazati stupce *IDProizvod*, *BrojProizvoda* i *CijenaBezPDV*. Stupac *CijenaBezPDV* potrebno je nazvati *Cijena* te vrijednosti u stupcu prikazati kao decimalne brojeve zaokružene na dvije decimale. Unutar komentara u SQL-u naznačite redoslijed izvršavanja dohvata. Potrebno je naznačiti po koracima dijelove dohvata. Objasnite zašto se u dijelu filtriranja redaka ne može koristiti naziv stupca *Cijena*, a u dijelu sortiranja može.

```
SELECT TOP 10 PERCENT
p.IDProizvod,
p.BrojProizvoda,
ROUND(p.CijenaBezPDV, 2) AS 'Cijena'
FROM
Proizvod p
WHERE
p.BrojProizvoda LIKE 's%'
AND p.CijenaBezPDV > 0
ORDER BY 'Cijena' DESC

--REDOSLIJED DOHVATA:

--1.
--FROM
--Proizvod p

--2.
--WHERE
--p.BrojProizvoda LIKE 's%'
--AND p.CijenaBezPDV > 0

--3.
--p.IDProizvod,
--p.BrojProizvoda,
--ROUND(p.CijenaBezPDV, 2) AS 'Cijena'

--4.
--ORDER BY p.CijenaBezPDV DESC

--5.
--TOP 10 PERCENT

--u dijelu filtriranja redaka (WHERE) ne može se koristiti naziv stupca Cijena
--za razliku od dijela sortiranja (ORDER BY)
--jer se naziv stupca dodjeljuje u koraku 3,
--tj. nakon filtriranja koje se izvršava u koraku 2

--dio sortiranja radi se tek u koraku 4,
--a tada će već biti poznat naziv stupca
```



4. zadatak (3 boda)

Korištenjem podupita dohvatite ID i naziv grada onih gradova koji imaju barem jednog kupca. Svi stupci moraju biti odgovarajuće nazvani.

```
SELECT
g.IDGrad AS 'ID grada',
g.Naziv AS 'Naziv grada'
FROM Grad g
WHERE
g.IDGrad IN
(SELECT DISTINCT k.GradID FROM Kupac k)
```

5. zadatak (3 boda)

Dohvatite naziv države i ukupan broj kupaca koji dolazi iz te države. U ukupan broj kupaca iz neke države uključeni su samo oni kupci kojima je poznat *email* i *telefon*. Prikazuju se samo one države iz kojih dolazi više od 5000 kupaca. Svi stupci moraju biti odgovarajuće nazvani.

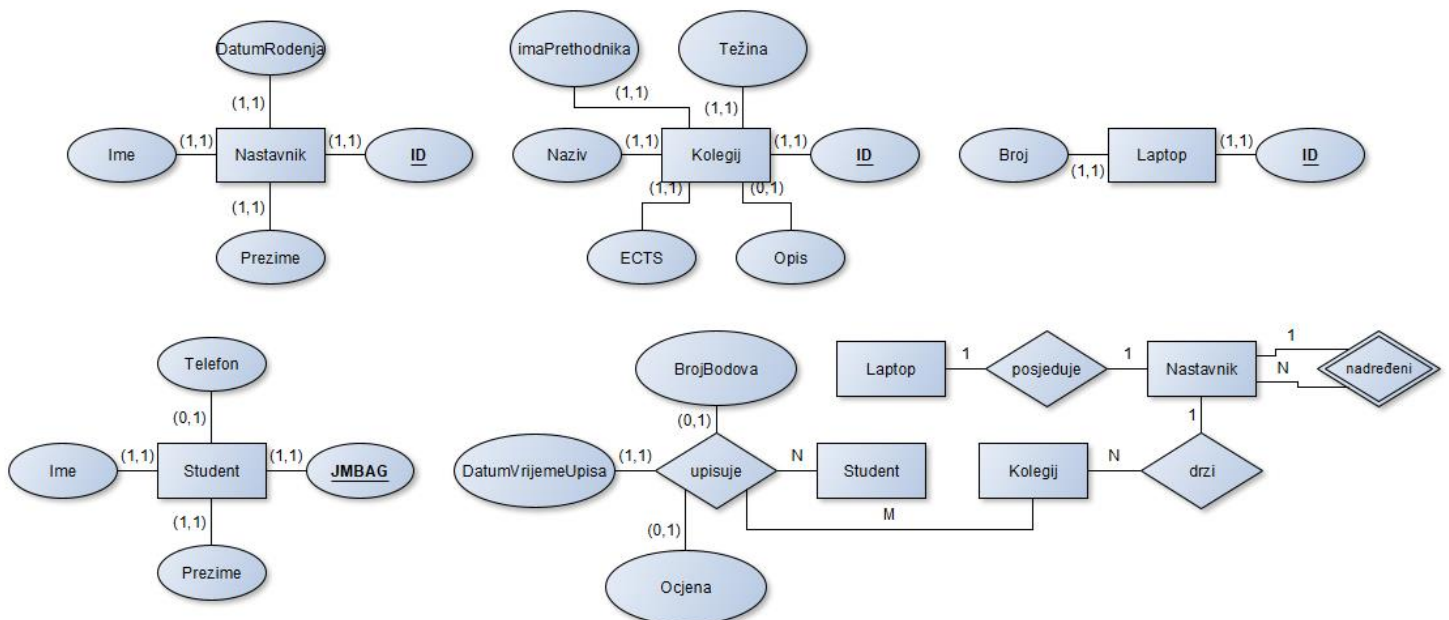
```
SELECT
d.Naziv AS 'Država',
COUNT(k.IDKupac) AS 'Broj kupaca'
FROM Drzava d
INNER JOIN Grad g ON d.IDDrzava=g.DrzavaID
INNER JOIN Kupac k ON g.IDGrad=k.GradID
WHERE
k.Email IS NOT NULL
AND k.Telefon IS NOT NULL
GROUP BY d.Naziv
HAVING COUNT(k.IDKupac) > 5000
```



6. zadatak (8 bodova)

Na osnovu izrađenog ER modela prikazanog na sljedećoj slici kreirajte **relacijsku shemu** baze podataka *Referada* pisanjem SQL koda. Potrebno je u svaku kreiranu tablicu **unijeti minimalno jedan redak s proizvoljnim vrijednostima** te postaviti da se **DBMS sam brine** oko jedinstvenih vrijednosti primarnog ključa. Kao rješenje predaje se **vlastito napisana SQL skripta**. Datoteku je potrebno spremiti pod nazivom **Zadatak6.sql** te na kraju obrisati kreiranu bazu podataka. Potrebno je u obzir uzeti i sljedeće uvijete:

- Telefon – točno 13 nedijakritičkih znakova
- Broj – jedinstven
- Naziv – predefinirana vrijednost *Proizvoljan kolegij*
- ImaPrethodnika – tip podataka za istinu ili laž
- Težina – cijeli broj unutar intervala [0, 255]
- BrojBodova – decimalan broj od ukupno 6 znakova, a maksimalno 4 iza decimalnog zareza
- svaki kolegij ne mora imati pridruženog nastavnika koji ga drži
- nastavnik mora imati laptop, dok svaki laptop ne mora nužno pripadati nastavniku



```

CREATE DATABASE Referada
GO
USE Referada
GO

CREATE TABLE Laptop(
IDLaptop int NOT NULL PRIMARY KEY IDENTITY(1,1),
Broj int NOT NULL UNIQUE
)

CREATE TABLE Kolegij(
IDKolegij int NOT NULL PRIMARY KEY IDENTITY(1,1),
Tezina tinyint NOT NULL,
ImaPrethodnika bit NOT NULL,
Naziv varchar(50) DEFAULT 'Proizvoljan kolegij',
ECTS int NOT NULL,
Opis varchar(50) NULL
)

CREATE TABLE Nastavnik(
IDNastavnik int NOT NULL PRIMARY KEY IDENTITY(1,1),
Ime nvarchar(50) NOT NULL,
DatumRodenja date NOT NULL,
LaptopID int NOT NULL FOREIGN KEY REFERENCES Laptop(IDLaptop),
KolegijID int NOT NULL FOREIGN KEY REFERENCES Kolegij(IDKolegij),
NastavnikID int NOT NULL FOREIGN KEY REFERENCES Nastavnik(IDNastavnik)
)

CREATE TABLE Student(
JMBAG int NOT NULL PRIMARY KEY IDENTITY(1,1),
Telefon char(13) NULL,
Ime varchar(50) NOT NULL,
Prezime varchar(50) NOT NULL
)

CREATE TABLE Upisuje(
JMBAG int NOT NULL FOREIGN KEY REFERENCES Student(JMBAG),
KolegijID int NOT NULL FOREIGN KEY REFERENCES Kolegij(IDKolegij),
BrojBodova decimal(6,4) NULL,
DatumVrijemeUpisa datetime NOT NULL,
Ocjena int NULL,
CONSTRAINT PK_Upisuje PRIMARY KEY (JMBAG, KolegijID)
)

INSERT INTO Laptop (Broj) VALUES (1234)
INSERT INTO Kolegij (Tezina, ImaPrethodnika, Naziv, ECTS) VALUES (2, 0, 'Baze
podataka', 6)
INSERT INTO Nastavnik (Ime, DatumRodenja, LaptopID, KolegijID, NastavnikID) VALUES
('Ivan', '1990-01-23', 1, 1, 1)
INSERT INTO Student (Ime, Prezime) VALUES ('Miro', 'Horvat')
INSERT INTO Upisuje(JMBAG, KolegijID, DatumVrijemeUpisa) VALUES (1, 1, GETDATE())

GO
USE SimpleWebShop
GO
DROP DATABASE Referada

```



7. zadatak (3 boda)

Kreirajte *Windows Forms* aplikaciju za rad s bazom *SimpleWebShop*. Kao rješenje kopirajte C# kod unutar komentara u SQL-u te dodatno predajte direktorij projekta zasebno kao **Zadatak7**.

- a) **(2 boda)** Aplikacija se treba spojiti na vašu lokalnu bazu podataka *SimpleWebShop* pomoću *SQL autentifikacije*. Potrebno je dodati dva *Label*-a s tekstovima: *Korisničko ime* i *Zaporka*; te dva *TextBox*-a naziva: *txtKorisnickoIme* i *txtZaporka*. Potrebno je unijeti korisničko ime u *TextBox* naziva *txtKorisnickoIme* i zaporku u *TextBox* naziva *txtZaporka*, te pritisnuti gumb s tekstom *Spoji* (naziva *btnSpoji*). Ukoliko je spajanje uspješno u skočnom prozoru treba ispisati *Uspješno spajanje na bazu SimpleWebShop!*, inače u skočnom prozoru treba ispisati grešku koja se pojavila. (Pomoć: nemojte zaboraviti dodati *using System.Data.SqlClient*;). Za potrebe b) dijela zadatka zgodno je instancu konekcije deklarirati na razini klase, ali nije nužno za rješenje ovog dijela zadatka.)
- b) **(2 bod)** U prethodno izrađenu aplikaciju dodajte gumb s tekstom *Dohvati* (naziva *btnDohvati*) čijim klikom se iz tablice *Kreditna kartica* dohvaćaju sve kreditne kartice, na način da se unutar *DataGridView* kontrole naziva *tablica* prikažu svi dohvaćeni podaci. Potrebno je koristiti konekciju stvorenu u a) zadatku, odnosno varijablu konekcije potrebno je deklarirati na razini klase (ako to već nije prethodno napravljeno).

Izgled sučelja:

IDKreditnaKartica	Tip	Broj	IstekMjesec	IstekGodina
1	American Express	33332664695310	11	2006
2	MasterCard	55552127249722	8	2005
3	Visa	77778344838353	7	2005
4	Visa	77774915718248	7	2006
5	Diners	11114404600042	4	2005
6	MasterCard	55557132036181	9	2006
7	MasterCard	55553635401028	6	2007
8	American Express	33336081193101	7	2007
9	MasterCard	55553465625901	2	2005
10	American Express	33332126386493	8	2008
11	American Express	33335352517363	10	2008
12	American Express	33334216194519	4	2008



Programski kod:

```
using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Zadatak_7
{
    public partial class Form1 : Form
    {
        SqlConnection sql;

        public Form1()
        {
            InitializeComponent();
        }

        private void btnSpoji_Click(object sender, EventArgs e)
        {
            try
            {
                sql = new SqlConnection("Server= DESKTOP-PC\\SQLEXPRESS;
                Database=SimpleWebShop; User ID=" + txtKorisnickoIme.Text
                + "; Password=" + txtZaporka.Text);
                sql.Open();

                MessageBox.Show("Uspješno spajanje na bazu SimpleWebShop!");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void btnDohvati_Click(object sender, EventArgs e)
        {
            if (sql != null)
            {
                SqlCommand cmd = new SqlCommand
                ("SELECT * FROM KreditnaKartica", sql);
                SqlDataReader reader = cmd.ExecuteReader();
                DataTable dt = new DataTable();
                dt.Load(reader);
                tablica.DataSource = dt;
            }
            else
            {
                MessageBox.Show("Prije dohvata podataka morate se spojiti na
                bazu!");
            }
        }
    }
}
```

