



**ALGEBRA  
BERNAYS**  
SVEUČILIŠTE

**UVOD U BAZE  
PODATAKA**

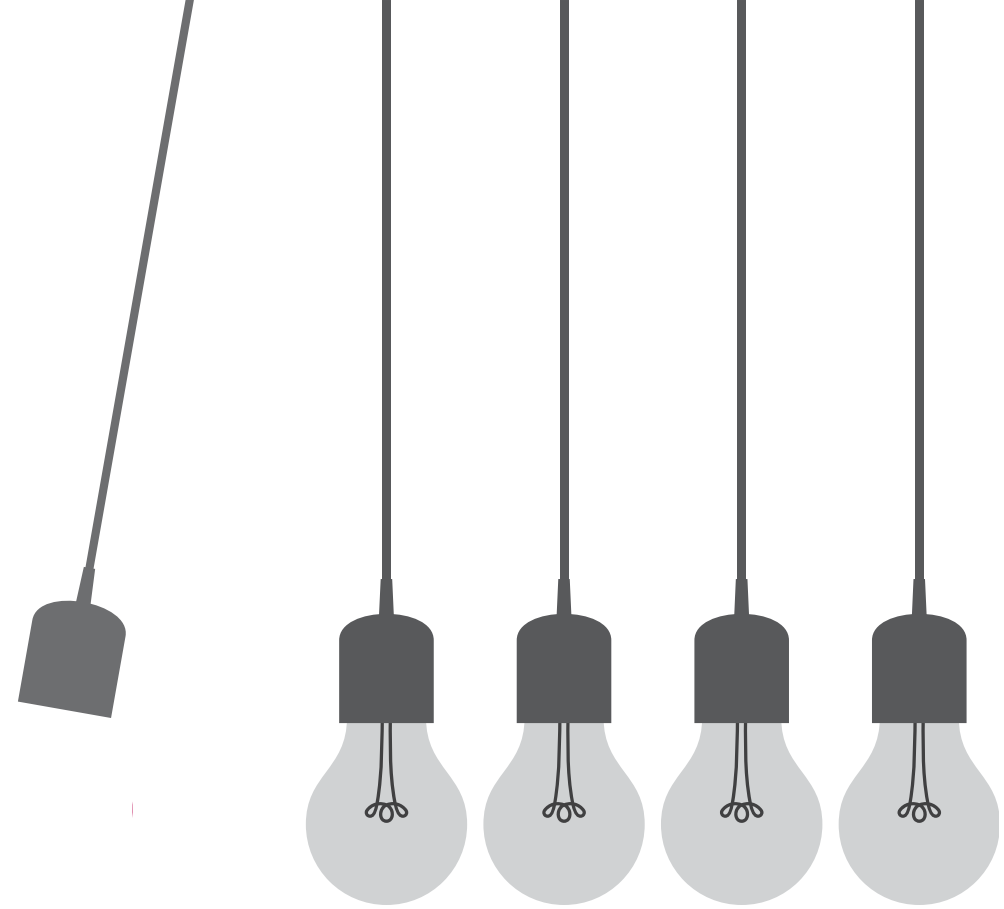
Upute za pohađanje i polaganje  
kolegija  
akademska godina 24/25

# Organizacija predavanja i vježbi

<b>Nositelji kolegija:</b>	Mario Fabijanić	<a href="mailto:mario.fabijanovic@algebra.hr">mario.fabijanovic@algebra.hr</a>
	Mateo Sokač	<a href="mailto:mateo.sokac@algebra.hr">mateo.sokac@algebra.hr</a>
<b>Asistenti:</b>	Aron Thompson	<a href="mailto:aron.thompson@algebra.hr">aron.thompson@algebra.hr</a>
	Matija Kolarić	<a href="mailto:mkolaric@algebra.hr">mkolaric@algebra.hr</a>
	Silvije Davila	<a href="mailto:silvije.davila@algebra.hr">silvije.davila@algebra.hr</a>
<b>Izvođenje nastave:</b>	<b>Predavanja</b>	2 sata tjedno - ukupno 30 sati
	<b>Vježbe</b>	2 sata tjedno - ukupno 30 sati
	<b>Konzultacije</b>	Prilikom vježbi ili e-mailom

# Informacije o kolegiju

- 5 ECTS bodova = 150 sati rada studenta
  - 30 sati predavanja
  - 30 sati vježbi
  - 90 sati rada kod kuće  
(15 tjedana po 6 sati)



# Cilj kolegija

Naučiti dizajnirati jednostavne relacijske baze podataka.

Naučiti koristiti programski jezik SQL za dohvaćanje, umetanje, ažuriranje i brisanje podataka iz relacijske baze podataka.

# Ishodi učenja

Ishod	MINIMALNI ISHODI UČENJA (po uspješnom završetku kolegija, student će moći)	ŽELJENI ISHODI UČENJA (uspješan student bi trebao moći)
I1	Konstruirati relacijski model podataka počevši od korisničkih zahtjeva.	Konstruirati relacijski model podataka počevši od korisničkih zahtjeva, uključujući podskupove i involuirane odnose.
I2	Izraditi bazu podataka koristeći DDL naredbe na temelju relacijskog modela.	Razlikovati učinke ograničenja na relacijskom modelu.
I3	Predložiti izmjene relacijskog modela kako bi se dosegla viša normalna forma.	Predložiti izmjene kompleksnog relacijskog modela kako bi se dosegla viša normalna forma.
I4	Koristiti osnovne naredbe za upravljanje podacima u relacijskoj bazi podataka.	Riješiti problem korištenjem spajanja.
I5	Konstruirati rješenje korištenjem sistemskih i agregatnih funkcija te grupiranja.	Konstruirati kompleksno rješenje korištenjem sistemskih i agregatnih funkcija te grupiranja.
I6	Primijeniti podupite kod izrade kompleksnijeg upita.	Riješiti problem kod izrade kompleksnijeg upita podupitom, uključujući podupit u FROM dijelu upita.

# Tematske cjeline

Tjedan nastave	Cjelina	Tjedan nastave	Cjelina
1. Tjedan	Entiteti i atributi Modeli podataka	9. Tjedan	Unutarnje spajanje tablica
2. Tjedan	Odnosi među entitetima Dijagram entiteta i ER model	10. Tjedan	Vanjska spajanja i unakrsno spajanje tablica
3. Tjedan	Relacijski model i ključevi	11. Tjedan	Funkcije
4. Tjedan	Pretvaranje ER sheme u relacijsku i normalizacija	12. Tjedan	Grupiranje i agregatne funkcije
5. Tjedan	Implementacija relacijskog modela	13. Tjedan	Podupiti
6. Tjedan	SELECT naredba	14. Tjedan	Podupiti
7. Tjedan	INSERT, UPDATE i DELETE naredbe	15. Tjedan	Ponavljjanje
8. Tjedan	Operatori		

# Literatura

## OBAVEZNA LITERATURA

- Harrington, J. L. (2016) *Relational Database Design and Implementation*. 4th edn. Amsterdam: Elsevier Inc.

## PREPORUČENA LITERATURA

- Churcher, C. (2012) *Beginning Database Design: From Novice to Professional*. New York City: Apress.

# Za potpis treba?

Za stjecanje prava na potpis potrebno je prisustvovati nastavi u postotku propisanom Pravilnikom o studijima i studiranju

DOLAZNOST NA PREDAVANJA I VJEŽBE	
Najmanje 50% fizičke prisutnosti na predavanjima	Najmanje 60% fizičke prisutnosti na vježbama

Tko ne dobije potpis, mora sljedeće godine ponovno upisati kolegij, platiti upis kolegija te nema pravo polaganja ispita.



# Polaganje kolegija

- Kolegij ima definiranih 6 ishoda učenja
- **Da bi student položio kolegij mora po svakom ishodu učenja ostvariti minimalno 50% bodova raspoloživih za taj ishod učenja**
- **Ako student ne ostvari 50% bodova iz nekog ishoda učenja, na sljedećem roku može opet polagati taj ishod učenja.**
- Metode provjeravanja skupova ishoda učenja:
  - Međuispiti
  - Školske zadaće

# Kako je to raspoređeno po ishodima učenja

Ishod	M1	M2	Školska zadaća	MAX
I1	21		2	23
I2	9		2	11
I3	5		2	7
I4		30	2	32
I5		12	2	14
I6		13		13
Ukupno	35	55	10	100

# Ocjenjivanje

Broj osvojenih bodova	Ocjena
0,00 – 50,00	1 (nedovoljan)
50,01 – 58,00	2 (dovoljan)
58,01 – 75,00	3 (dobar)
75,01 – 92,00	4 (vrlo dobar)
92,01 – 100,00	5 (izvrstan)

# Ispiti

- Na svakom kolegiju vrijedi **pravilo 3 + 1**
  - To znači da student mora položiti ispit iz najviše 4 izlaska
    - 3 redovna izlaska – Uključena u cijenu školarine
    - 1 izvanredni izlazak – Odlukom o naknadi troškova 4. prijava ispita se naplaćuje
  - Vremenski rok za položiti ispit je **12 mjeseci** od dana upisa kolegija
  - Ako student u 12 mjeseci ne položi kolegij, **mora ponovno upisati kolegij te ponovno polagati sve ishoda učenja kako je definirano kolegijem**
- **Vodite računa o rokovima prijave i odjave ispita na IE**
  - Ako niste prijavili ispit na vrijeme, ne možete pristupiti ni pismenom niti usmenom dijelu
  - Ako je student prijavio više ispitnih rokova iz istog kolegija, pri dobivanju ocjene kojom je zadovoljan, dužan je odjaviti svaki sljedeći rok koji je iz tog kolegija prijavio. U suprotnom, studentu se u Infoeduku unosi nedovoljan (1)

# Školske zadaće

- Školska zadaća provodi se na nastavi putem blic testova, a sastoje se od 1-2 kratkih pitanja
- Blic testovi se pišu na vježbama
- Na blic testovima piše se do tada obrađeno gradivo
- **Blic testovi se ne ponavljaju, nije ih moguće pisati naknadno i nije ih moguće ispravljati (ili popravljati)**
- Blic testovi se *neće najavljivati*

# Akademski standard ponašanja

- U komunikaciji (pisanoj i usmenoj) pridržavati se pravila poslovne komunikacije primjerene akademskoj razini
- Potrebno je držati se jasno definiranih rokova za predaju zadataka (školske zadaće, ispiti)
  - Svaki zadatak poslan nakon definiranog roka neće se ocjenjivati
- Samo oni studenti koji mogu potvrditi svoje pohađanje, smatrat će se prisutnima
  - Potpisivanje drugih studenata ili registracija njihovom karticom nije dopušteno i može biti predmet stegovnog postupka. Nastavnik će obrisati prisustvo ako utvrdi da je student prijavljen, a da nije prisutan na nastavi

# Pravila ponašanja na nastavi – fizička prisutnost

- Na nastavu se dolazi na vrijeme
- Pri ulasku u učionicu student prilazi do stola i prijavljuje se na nastavu karticom te sjeda na dostupno mjesto za rad
- Ometanje nastave i neaktivno sudjelovanje na nastavi nije dozvoljeno
  - Repetitivno kršenje ovog pravila sankcionira se prijavom Stegovnom povjerenstvu

**Pitanja?**





**ALGEBRA  
BERNAYS**  
SVEUČILIŠTE

**UVOD U BAZE  
PODATAKA**

Vježbe 01

# SADRŽAJ VJEŽBI

Tri su nam osnovna cilja:

- **upoznavanje s pojmovima** vezanim uz baze podataka
- **upoznavanje s jednom konkretnom izvedbom baze podataka** (radit ćemo s Microsoft SQL Server verzijom, tj. njezinom Express inačicom)
- **upoznavanje osnovnih elemenata SQL jezika** (jezika za „pričanje“ s bazom podataka)

# SADRŽAJ VJEŽBI

Cjeline koje ćemo obraditi su, u kratkoj formi, sljedeće:

- Entiteti i veze, ER model
- Tipovi podataka
- Kreiranje tablica i rad s tablicama: veze, primarni i strani ključ, ograničenja
- Relacijska algebra
- Normalne forme
- Jednostavni upiti: INSERT, UPDATE, DELETE, SELECT
- Složeniji upiti: spajanje tablica (JOIN)
- Grupiranje i agregatne funkcije
- Podupiti

# Uvod

Baza podataka je **skup međusobno povezanih podataka** pohranjenih na vanjskoj memoriji, koji su istovremeno dostupni raznim korisnicima i programima.

**Sustav za upravljanje bazom podataka** (DBMS) je poslužitelj baze podataka koji se nalazi između korisnika baze podataka i same baze. On obavlja sve operacije s podacima.

Najpoznatije baze podataka su:

- Microsoft SQL Server
- Oracle database
- IBM DB2
- MySQL
- Microsoft Access
- ...

# Od korisničkih zahtjeva do baze podataka

Kreiranje "dobre" baze podataka zahtijeva nekoliko koraka. Cijeli taj proces nazivamo modeliranje.

**Modeliranje se sastoji od sakupljanja zahtjeva, njihove analize te dizajniranja modela (po nekim pravilima).**

**Zahtjeve dobijamo od korisnika koji opisuje što želi.**

Te zahtjeve analiziramo i kreiramo (dizajniramo) model baze. Na osnovu tog modela zaista i kreiramo bazu podataka.

Pogledajmo primjer korisničkih zahtjeva (namjerno pojednostavljenih) te **kreirajmo jedan jednostavan model baze podataka.**

Naravno, taj model neće biti "pravi" nego samo jednostavna skica. Kasnije ćemo napraviti i "pravi" model.

# Jednostavni primjer: Baza d.o.o.

## Korisnički zahtjevi:

Poduzeće „Baza d.o.o.“ za potrebe svog poslovanja želi kreirati **bazu poslovnih partnera**. Ta baza mora sadržavati naziv poslovnog partnera (naziv poduzeća ili obrta), matični broj/OIB partnera, brojeve telefona i faks uređaja i podatke za kontakt zaposlenika kod partnera (vlasnik, predstavnik, voditelj marketinga, voditelj prodaje, ...).

**Kratka analiza:** kako su korisnički zahtjevi uobičajeno neprecizni (koji su to točno podaci za kontakt i koje sve zaposlenike uzeti u obzir?), u prvoj fazi se pokušava napraviti apstraktni model. To znači da nećemo detaljno istraživati koji su to podaci za kontakt i koje zaposlenike ćemo uzeti u obzir, nego ćemo pretpostaviti da su to neki jednostavni podaci pa ako će kasnije trebati više informacija popravljat ćemo model.

# Jednostavni primjer: Baza d.o.o.

Tražimo **važnije informacije** u tekstu zahtjeva **koji čine jednu logičku cjelinu**. To bi npr. mogli biti podaci o poslovnom partneru (naziv, adresa, matični broj, brojevi telefona i fax uređaja) i podaci o zaposlenicima.

Kratko zapisano, bitni elementi su:

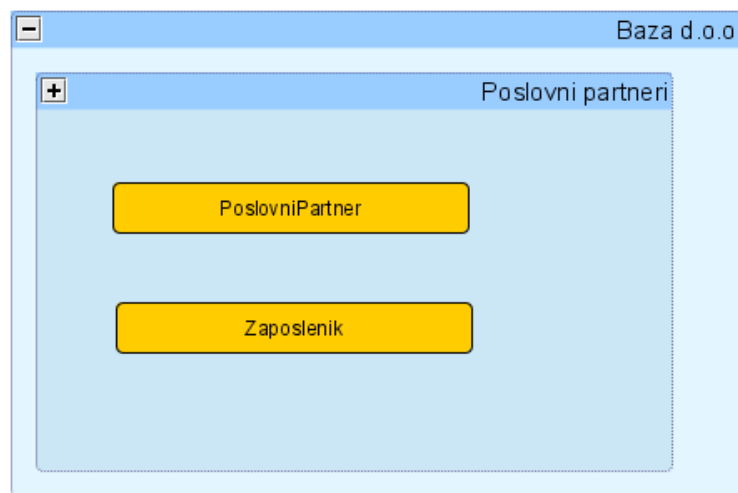
- **PoslovniPartner** (Naziv, Adresa, MB, BrojTelefona, Fax)
- **Zaposlenik** (ImePrezime, BrojTelefona, MailAdresa)

Daljnjom analizom dolazimo do toga da je za zaposlenika potrebno znati na kojem radnom mjestu radi pa tu informaciju dodajemo elementu Zaposlenik:

- **Zaposlenik** (ImePrezime, BrojTelefona, MailAdresa, ***RadnoMjesto***)

# Jednostavni primjer: Baza d.o.o.

Sada kada imamo popisane bitnije elemente, model možemo prikazati na nekoliko načina. Najjednostavniji model je skica:



Takvo nešto bi nacrtali na papiru dok razmišljamo o bazi. Mogli bi u skicu dodati i dijelove koji pripadaju bitnim elementima, ali to je obično previše pisanja po papiru. Te dijelove možemo dodati u prikaz pomoću tablica, koji je sljedeći način kako prikazati model.



# Jednostavni primjer: Baza d.o.o.

Nakon skice, **model organiziramo u tablice**, gdje po jedna tablica opisuje bitan element:

PoslovniPartner	
	Naziv Adresa MB BrojTelefona Fax

Zaposlenik	
	ImePrezime BrojTelefona MailAdresa RadnoMjesto

Ovdje odmah imamo popisane dijelove koji čine bitan element.

**Za početak, ovo je dovoljno dobro.** Kasnije ćemo otići korak dalje i pretvoriti ovakav dizajn u "pravi" dizajn.

Kod "pravog" dizajna ćemo koristiti ER dijagrame. U svakom slučaju, ovo su početni koraci koje je potrebno poduzeti kako bi napravili dizajn neke baze podataka.

# Zadatak za vježbu

## Korisnički zahtjevi:

Videoteka "Loš film" želi unaprijediti svoje poslovanje i prijeći s ručno ispunjavanih kartica na računalni program. Do sada su koristili kartice za svakog člana za svaki film.

Na karticama članova je pisao broj člana, ime i prezime, adresa te kontakt telefon. Sa zadnje strane zapisivane su posudbe dotičnog člana.

Na karticama filmova je pisao naziv filma i tip medija (VHS, Beta, CD, DVD). Sa zadnje strane zapisivane su informacije o dotičnom članu koji je napravio posudbu te vrijeme posuđivanja i vraćanja.

**Videoteka bi dodatno željela pamtit i više detalja: glavni glumac, trajanje, sporedni glumci, režiser, žanr i kratak opis sadržaja.**

**Zadatak:** Napravite kratku analizu korisničkih zahtjeva, generirajte neki model za bazu podataka te skicirajte taj model i napravite tablični prikaz modela.

# Zadatak za vježbu

## Korisnički zahtjevi:

Auto kuća želi uspostaviti centralno praćenje svih podataka važnih za svoje poslovanje.

Auto kuća se bavi:

- prodajom auta
- prodajom rezervnih dijelova
- servisiranjem auta.

**Zadatak:** Napravite kratku analizu korisničkih zahtjeva te skicirajte model baze podataka tabličnim prikazom.