Baze podataka

Tanja Šević

20. januar 2026.

Metodika nastave informatike A prof. Sana Stojanović Đurđević

Sadržaj

1 Pojam baze podataka

Baza podataka je organizovana kolekcija podataka. Umesto da informacije čuvamo na papiru, baza podataka ih čuva u računaru.

Svrha baze podataka je da olakša pristup podacima, pretraživanje, menjanje sadržaja ili ažuriranje.

Šta je baza podataka?

Baza podataka je strukturirani skup podataka koji se čuva elektronski i kojim upravlja sistem za upravljanje bazama podataka, omogućavajući jednostavno čuvanje, pretragu, izmenu i upravljanje informacijama.

Zašto su baze podataka važne?

U savremenom svetu, zahvaljujući napretku tehnologije, količina informacija do kojih možemo doći i kojima možemo raspolagati je velika. Danas, u odnosu na generaciju naših baka i deka, ukoliko želimo da dođemo do određene informacije, na primer, koja hrana ima najviše vlakana, ne moramo tražiti knjigu iz biologije, pa lekciju u kojoj se ta tema obrađuje. Dovoljno je Google-u da postavimo pitanje i pretraživač će nam sam dati odgovor.

Ovako brzu pretragu nam omogućavaju upravo **baze podataka**. One omogućavaju da velika količina informacija bude dostupna u svakom trenutku.

1.1 Primena baze podataka

Baze podataka se koriste svuda oko nas:

- **U školama** (spisak učenika, predmeta, profesora, ocena, izostanaka),
- **U prodavnicama** (spisak proizvoda i cena),
- **U bolnicama** (kartoni pacijenata, datumi pregleda),
- **Na internetu** (profili na društvenim mrežama, onlajn prodavnica, aplikacije za plaćanje računa, sajtovi za puštanje muzike).

Uzmimo primer iz svakodnevnog života. Kada želimo da pustimo neku pesmu uključimo YouTube i on nam na *Glavnoj strani* da predlog pesama koje bismo mogli da pustimo. Da li smo se nekad zapitali na koji način on formira tu listu za nas? YouTube koristi baze podataka u kojima se čuvaju informacije o pesmama i izvođačima koje smo prethodno slušali. Na osnovu tih podataka, sistem nam predlaže sadržaj koji procenjuje da će nam se dopasti. Bez baza podataka, YouTube ne bi mogao da zapamti naše izbore i ne bi svaki korisnik imao prilagođenu pretragu.

U školi postoji spisak učenika sa njihovim podacima, ocenama, izostancima, to je takođe jedna baza podataka.

1.2 Sistemi za upravljanje bazama podataka

Najpoznatiji sistemi za upravljanje bazama podataka su:

- **SQLite** – Najkorišćeniji jer ne zahteva instalaciju (ugrađen je u aplikacije koje ga koriste) i celu bazu podataka čuva u jednom fajlu. Koriste ga mobilni telefoni.
- **Oracle Database** – Veoma moćan i skup sistem. Koriste ga velike institucije poput banaka i vojske jer je izuzetno siguran.
- **MySQL** – Jednostavan i besplatni softver. Koriste ga škole, ali i ogromni sajtovi poput Facebook-a i YouTube-a jer je brz i pouzdan.
- **Microsoft SQL Server** – Sistem koji je napravio Microsoft. Odlično sarađuje sa ostalim njihovim programima (Windows, Excel).
- **PostgreSQL** – Naprednija verzija besplatnih sistema. Često ga koriste naučnici i inženjeri jer može da čuva veoma složene podatke (npr. geografske mape).
- **IBM DB2** – Izuzetno izdržljiv sistem, napravljen da radi godinama bez ijedne sekunde prestanka. Koriste ga velike korporacije.
- **Redis** – Super-brzi sistem koji podatke čuva u radnoj memoriji (RAM). Koriste ga društvene mreže.

Ti sistemi korisnicima pružaju sve usluge rada sa podacima.

1.3 Razlika između podataka i informacije

Često se podatak i informacija koriste kao sinonimi, iako to nisu. Da bismo shvatili razliku hajde prvo da vidimo šta ovi pojmovi predstavljaju.

Šta je podatak?

Podatak je neka reč, broj ili simbol koji sam po sebi nema posebno značenje. Ako samo kažemo neki broj ili reč, on nam ne govori ništa korisno dok ga ne stavimo u određeni kontekst.

- **Primeri podataka:** 38°, Sunčano, 15:00, Beograd.

Šta je informacija?

Informacija nastaje kada podatke obradimo, organizujemo i damo im značenje. Informacija nam pomaže da donesemo neku odluku ili naučimo nešto novo.

- **Primer informacije:** „Temperatura" u **Beogradu** je **38** stepeni, biće **sunčano** danas u **15:00** „časova."

Ključna razlika

Glavna razlika se može prikazati kroz proces obrade:

PODACI $\xrightarrow{\text{Obrada}}$ **INFORMACIJA**

Karakteristika	**Podatak**	**Informacija**
Šta je to?	Neka reč, broj	Obrađen i smislen podatak
Značenje	Sam po sebi ga nema	Ima jasno značenje
Primer	„12"	„U mom odeljenju ima 12 „devojčica"

Tabela 1: Razlika između podatka i informacije

Zanimljivost: Računar je mašina koja neverovatno brzo pretvara milione podataka u korisne informacije koje mi svakodnevno koristimo (na primer, vremenska prognoza na telefonu).

1.4 Informacioni sistemi

Pojam **informacioni sistemi** najviše nas asocira da je to neka baza podataka koja nam pruža određene informacije. Međutim, informacioni sistemi su mnogo više od obične baze podataka. To je čitav proces u kom ljudi zahvaljujući određenim pravilima (pravila unosa podataka, bezbednosti), podatke pretvaraju u korisne informacije.

Čemu služe informacioni sistemi?

Glavni cilj informacionog sistema je da nam olakša rad i ubrza donošenje odluka. On to radi kroz četiri osnovna koraka:

- **Prikupljanje:** Unos podataka u sistem (npr. kada nastavnik upiše tvoju ocenu).
- **Čuvanje:** Smeštanje tih podataka u baze podataka da se ne bi izgubili.
- **Obrada:** Računanje i organizacija (npr. sistem sam izračuna tvoj prosek ocena).
- **Prikazivanje:** Davanje informacija korisniku (npr. ti vidiš svoju ocenu na ekranu telefona).

Gde se sve koriste?

Informacioni sistemi su danas svuda oko nas. Bez njih bi svet funkcionisao mnogo sporije i sa mnogo više grešaka. Najčešće ih srećemo:

- **U školama:** Za vođenje elektronskih dnevnika, evidenciju učenika i profesora, kao i za rad školske biblioteke.
- **U prodavnicama:** Za praćenje zaliha proizvoda na rafovima, očitavanje cena bar-kod skenerom i izdavanje računa.
- **U bolnicama:** Za čuvanje elektronskih kartona pacijenata, zakazivanje pregleda i praćenje rezultata analiza.
- **Na internetu:** Svaki profil na društvenim mrežama, onlajn prodavnica ili aplikacija za muziku i filmove je zapravo deo jednog velikog sistema.
- **U bankama:** Za sigurno čuvanje novca, plaćanje karticama i korišćenje bankomata.

Zašto su važni?

Bez informacionih sistema, naš svakodnevni život bi izgledao potpuno drugačije. Morali bismo da: podatke tražimo u knjigama, da idemo u banku kako bismo platili račun, ne bi postojali digitalni dnevnici, kasirke u prodavnici bi morale da znaju napamet cenu svakog artikla. Stvari koje čine našu svakodnevicu iziskivale bi više vremena.

Glavne prednosti korišćenja ovih sistema su:

- **Ušteda vremena** - lako dolazimo do informacija
- **Smanjenje grešaka** - računari su precizniji od ljudi za pamćenje hiljada cifara
- **Veća efikasnost** -za isto vreme nam omogućavaju da uradimo mnogo više stvari
- **Dostupnost** - podaci su nam uvek dostupni

Zaključak: Informacioni sistemi nisu samo tehnologija, već način na koji savremeno društvo funkcioniše, pomažući nam da budemo bolje povezani i informisani.

1.5 Veza baze podataka i informacionih sistema

Kao što smo ranije rekli, informacioni sistemi nisu isto što i baze podataka. Da bismo lakše razumeli, uporedimo ih sa stvarima iz svakodnevnog života:

- **Primer iz kuhinje:** Baza podataka je **frižider**, a informacioni sistem je **kuvar** koji uzima namirnice iz tog frižidera.

- **Primer sa muzičkom aplikacijom:** Baza podataka je **digitalna arhiva** sa milionima pesama. Informacioni sistem je **aplikacija (poput YouTube-a)** koja nam omogućava da pretražujemo tu bazu, slušamo pesme i dobijamo preporuke na osnovu prethodno slušanih pesama.

Zaključak: Bez baze podataka, informacioni sistem ne bi imao gde da čuva informacije. Sa druge strane, bez informacionog sistema, baza podataka bi bila samo gomila nabacanih podataka.

1.6 Modeli baza podataka

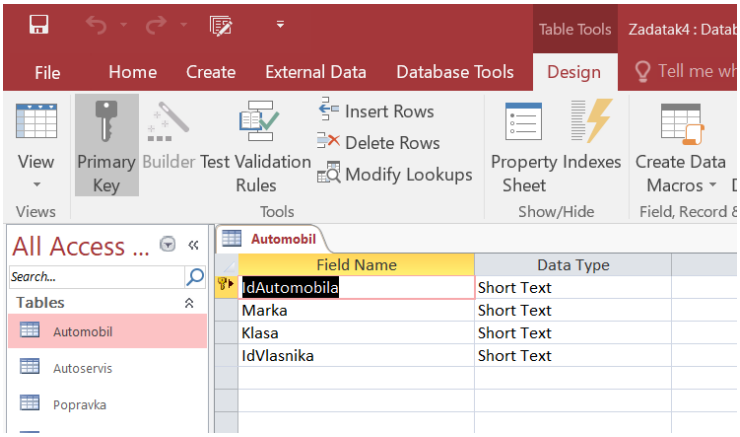
Postoji više načina za organizovanje podataka, ali najzastupljeniji je **relacioni model**.

Relacioni model

Osnovna ideja je da se podaci čuvaju u **tabelama** (relacijama). Kao što vidimo na primeru tabele `Automobil` (Slika 1), svaki red predstavlja jedan zapis, a svaka kolona jedan atribut tog zapisa.

Karakteristike ovog modela su:

- **Tabele:** Podaci su organizovani u dvodimenzionalne nizove.
- **Ključevi:** Svaki red je jedinstven zahvaljujući primarnom ključu.
- **Veze:** Tabele se povezuju putem zajedničkih kolona.



Slika 1: Prikaz sistema za upravljanje relacionom bazom podataka (Access)

Ostali modeli

Pored relacionog, značajni su i:

- **Hijerarhijski model** – podaci su u strukturi stabla.
- **Mrežni model** – omogućava složenije veze gde jedan podatak može pripadati većem broju vlasnika.

1.7 Relacione baze podataka

Relacione baze podataka su najkorišćeniji tip baza. Sastoje se od tabela koje su međusobno povezane.

Šta je relaciona baza podataka?

Relaciona baza podataka je skup digitalnih tabela u kojima su podaci organizovani tako da su međusobno povezani. Reč „relacija" sugeriše da je u pitanju neka veza.

Kako one funkcionišu?

Da bi tabela bila deo relacione baze, ona mora da ima tri osnovna dela:

- **Entitete (tabele)** - Objekti o kojima čuvamo podatke (na primer tabela *Učenici*, tabela *Nastavnici*, tabela *Ocene*).
- **Atribute (kolone)** - Osobine tih objekata (na primer, u tabeli *Učenici* kolone su: *Ime*, *Prezime*, *Razred*).
- **Primarni ključ (ID)** - Jedinstveni broj koji ima svaki red u tabeli (na primer JMBG).

Zašto su relacione baze najkorišćenije?

Relacione baze su kao *digitalni organizator*. Glavne prednosti su što:

- **Nema ponavljanja podataka:** Podaci o jednom učeniku se pišu samo jednom. Na primer, ako se učenik preseli, promenimo adresu samo na jednom mestu.
- **Laka pretraga:** Ukoliko unesemo upit: „Pokaži mi sve učenike iz VII-2 koji imaju peticu iz Informatike", sistem će nam sam napraviti spisak.
- **Tačnost:** Podaci su uvek tačni jer sistem ne dozvoljava da upišemo ocenu učeniku koji ne postoji u bazi.

1.8 Osnovni pojmovi relacionih baza podataka

Relacione baze podataka se zasnivaju na tabelama, a svaki element tabele ima svoj naziv i ulogu. Na osnovu Slike 1, možemo jasno videti osnovne elemente:

- **Tabela (Relacija):** Predstavlja skup podataka o istoj vrsti entiteta. Na slici 1 u levom panelu vidimo tabelu pod nazivom *Automobil*, koja služi za čuvanje podataka o vozilima.
- **Kolona (Atribut):** Određuje karakteristike podataka. Na slici su to polja: *IdAutomobila*, *Marka*, *Klasa* i *IdVlasnika*. Svaka kolona ima definisan tip podataka (*Data Type*); u ovom slučaju, za sva polja je izabran tip *Short Text*.
- **Red (Vrsta ili Torka):** Predstavlja jedan konkretan zapis u tabeli (npr. podaci o jednom specifičnom automobilu marke „Opel"). Iako se na slici vidi dizajn tabele (*Design View*), u realnom radu svaki popunjeni red bi predstavljao jedan jedinstveni objekat.
- **Polje (Ćelija):** To je mesto gde se seku red i kolona (npr. konkretna vrednost „Siva" u koloni *Marka*).
- **Primarni ključ (Primary Key):** Posebno važan pojam koji vidimo na slici je polje *IdAutomobila*, pored kojeg se nalazi simbol ključa. To znači da je ovo polje jedinstveni identifikator koji sprečava da se u bazi pojave dva identična zapisa.

1.9 Razlika matematičkog pojma relacije i relacije relacione baze podataka

Iako relaciona baza podataka svoje korene vuče iz matematičke teorije skupova, postoje suštinske razlike u njihovoj definiciji i primeni:

Matematički pojam relacije

U matematici, relacija se definiše kao podskup Dekartovog proizvoda dva ili više skupova. Njene ključne karakteristike su:

- **Apstraktnost:** Matematička relacija je isključivo skup uređenih n-torki.
- **Odsustvo naziva:** Kolone u matematičkoj relaciji nemaju imena (poput "Marka"ili "Model"), već se identifikuju isključivo na osnovu svog redosleda (prvi element, drugi element, itd.).
- **Beskonačnost:** Matematička relacija može biti beskonačna.

Relacija u relacionim bazama podataka (Tabela)

U kontekstu baza podataka, relacija je tabela sa strogo definisanom strukturom, što možemo videti na primeru slike 2:

- **Imenovane kolone (Atributi):** Svaka kolona ima svoje jedinstveno ime, kao što su `IdAutomobila` ili `Marka`. Ovo omogućava pristup podacima putem naziva, a ne pozicije.
- **Tipovi podataka:** Svaka kolona ima definisan tip (npr. *Short Text*), što osigurava da podaci budu uniformni. U čistoj matematici ovaj koncept ne postoji na ovaj način.
- **Konačnost:** Za razliku od matematičke relacije, svaka tabela u bazi podataka uvek sadrži konačan broj redova (zapisa), ograničen fizičkim resursima sistema.

1.10 Pojam i tipovi ključeva

U relacionim bazama, **ključevi** su *specijalne kolone* koje služe za identifikaciju i povezivanje podataka. Bez njih, tabele bi bile samo nepovezani spiskovi.

Postoji više tipova ključeva koji omogućavaju da baza bude organizovana i tačna:

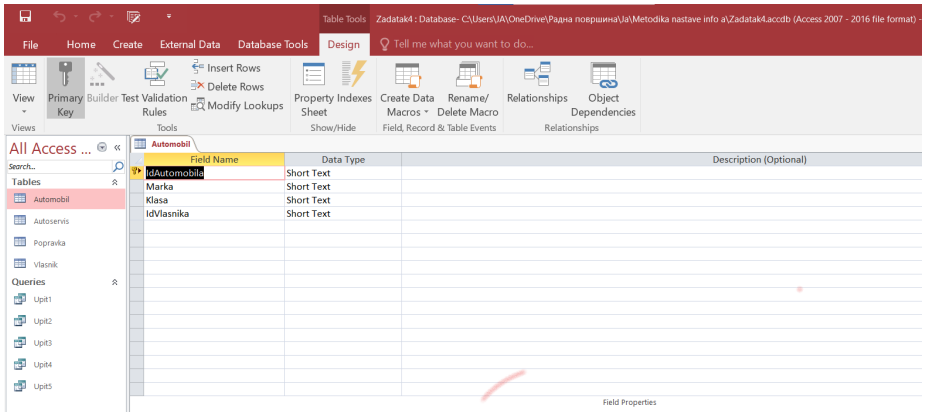
- **Primarni ključ (Primary Key):** jedinstveno određuje svaki red. Svaka tabela mora imati primarni ključ i on mora biti jedinstven.
  - *Primer:* U tabeli *Radnici*, *idRadnika* bi bio primarni ključ.
- **Strani ključ (Foreign Key):** Služi kao „most" za povezivanje dve tabele. On u jednoj tabeli pokazuje na primarni ključ u drugoj.
  - *Primer:* U tabeli *Plate*, koristimo *idRadnika* kao strani ključ da bismo povezali isplatu sa pravim radnikom.
- **Kandidat za ključ (Candidate Key):** Kolone koje bi mogle biti primarni ključ jer su jedinstvene (npr. JMBG ili broj lične karte).
  - *Primer:* U tabeli *Učenici*, pored ID broja, i *JMBG* i *Broj pasoša* su jedinstveni. Oni su kandidati za ključ.
- **Složeni ključ (Composite Key):** Ključ koji se dobija spajanjem dve ili više kolona kako bi se dobila jedinstvenost (npr. *idRadnika* + *idProjekta*).
  - *Primer:* U bioskopu, kolona *Broj sedišta* se ponavlja u svakoj sali. Tek kada spojimo *Broj sale* i *Broj sedišta*, dobijamo jedinstvenu kombinaciju koja tačno određuje jedno mesto.

1.11 Šema relacione baze podataka

U programu Microsoft Access, šema baze podataka se sastoji iz dva ključna dela (Slika 2):

1. **Logička struktura (Leva strana):** U panelu *All Access Objects* vidi se spisak entiteta: *Automobil*, *Autoservis*, *Popravka* i *Vlasnik*. Ovaj spisak predstavlja „kostur" šeme, odnosno skup svih objekata o kojima baza vodi računa.

2. **Definicija atributa (Centralni deo):** Tabela u sredini ekrana, gde se definišu *Field Name* (*IdAutomobila, Marka, Klasa, IdVlasnika*) i *Data Type* (*Short Text*), predstavlja detaljan arhitektonski plan šeme. Ovde se određuju pravila za svako pojedinačno polje pre nego što se krene sa unosom podataka.



Slika 2: Prikaz logičke strukture i definicije atributa u Access-u

Ovakva organizacija omogućava projektantu baze da jasno vidi kako su podaci strukturirani i da osigura integritet informacija unutar celog sistema.

2 Kreiranje baza podataka

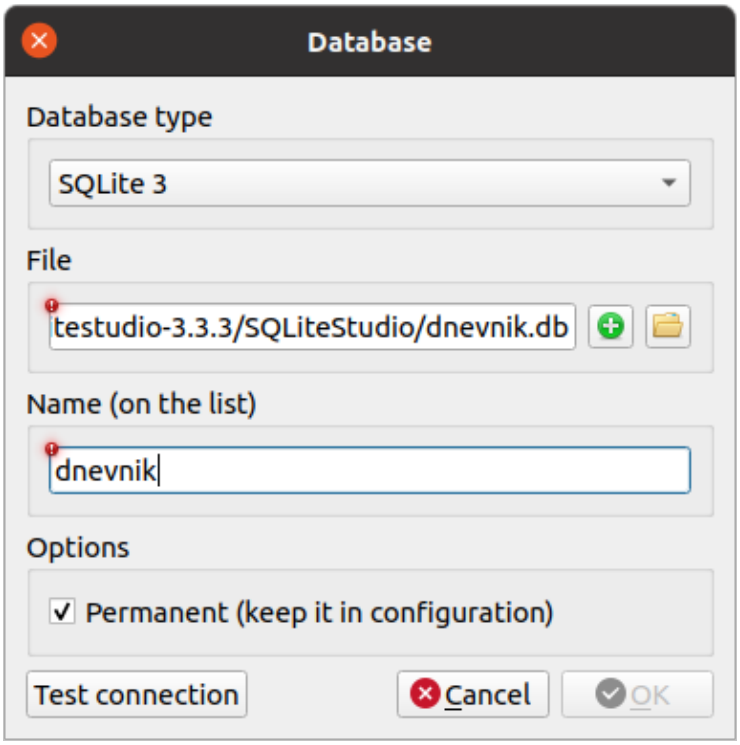
Da bismo objasnili **kreiranje baze podataka** potrebno je da navedemo koji sistem koristimo za upravljanje bazom podataka.

Za rad sa bazama podataka koristićemo program **SQLite Studio**, koji je intuitivan i pruža jasan grafički prikaz svih delova baze.

Novu bazu podataka kreiramo putem komande menija **Database** → **Add a database**. Postupak obuhvata dva ključna koraka:

1. **Izbor fajla:** Klikom na dugme sa znakom plus biramo lokaciju na računaru i definišemo naziv fajla u kojem će podaci biti fizički sačuvani (npr. `dnevnik.db`). Ovi fajlovi obično imaju ekstenziju *.db* ili *.sqlite*.

2. **Naziv u interfejsu:** Određujemo naziv pod kojim će se baza prikazivati unutar samog programa (npr. `dnevnik`).



Slika 3: Dodavanje baze u SQLite Studio

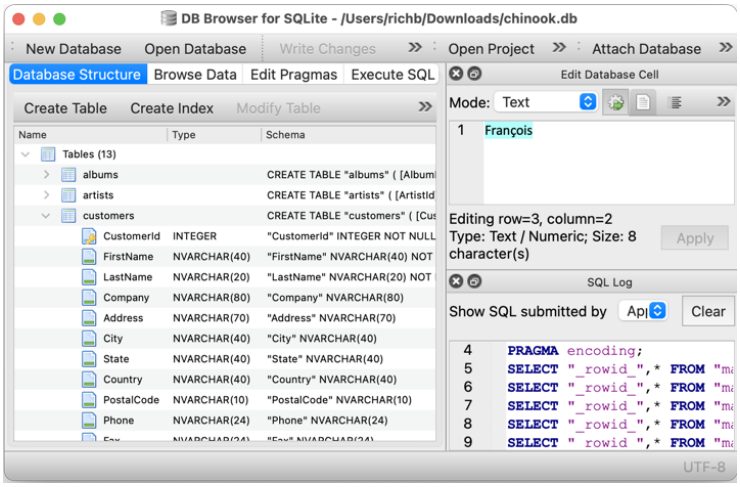
Ukoliko već imamo postojeću bazu, koristimo istu komandu (**Database** → **Add a database**), ali umesto kreiranja novog fajla, jednostavno pronađemo i izaberemo postojeću datoteku na disku. Na ovaj način sistem povezuje program sa već unetim podacima.

2.1 Upoznavanje konkretnog sistema za upravljanje bazama podataka

Kada uspešno dodamo bazu, na ekranu razlikujemo tri bitna dela:

- **Leva strana (Navigator):** Tu se nalazi stablo sa tvojim bazama. Klikom na bazu vidimo njene tabele.

- **Centralni deo (Editor):** Ovde se otvaraju tabele kada kliknemo na njih. Tu unosimo podatke ili menjamo strukturu kolona.
- **Gornja traka (Alatnica):** Sadrži važne dugmiće poput *Commit* (zeleni znak potvrde), koji služi da **sačuvamo** izmene koje smo napravili.
- **Statusna traka (Dno ekrana):** Veoma važan deo koji nam ispisuje poruke o greškama. Ako nešto ne uradimo kako treba (npr. zaboravimo primarni ključ), ovde će se pojaviti crveni tekst sa objašnjenjem.
- **Log prozor:** Prikazuje istoriju svih operacija koje je program izvršio nad bazom podataka.



Slika 4: Primer SQLite baze podataka

2.2 Planiranje jednostavnih baza podataka

Izgradnja baze podataka počinje od **planiranja** tabela, koje su nosioci informacija, i njihovog povezivanja.

Planiranje je najbitnija faza u kreiranju baze podataka. Proces planiranja se obično deli na nekoliko ključnih koraka:

1. **Analiza zahteva:** „Određivanje svrhe baze podataka. Potrebno je odgovoriti na pitanje šta sistem treba da podrži (npr. „Šta sve moj elektronski dnevnik treba da pamti?").
2. **Identifikacija entiteta:** Određivanje glavnih objekata (tema) o kojima čuvamo podatke. U školskom sistemu to su najčešće: *Učenik*, *Predmet*, *Ocena* ili *Izostanak*.

3. **Definisanje atributa:** Određivanje specifičnih osobina za svaki entitet. Na primer, za entitet *Učenik* atributi su: ime, prezime, JMBG i adresa.

4. **Uspostavljanje relacija:** Definisanje načina na koji su entiteti međusobno povezani. Na primer, uspostavlja se veza tipa jedan-prema-više, jer jedan učenik može imati više ocena.

5. **Normalizacija:** Završna provera strukture kako bi se osiguralo da su podaci pravilno podeljeni u tabele i da se nigde nepotrebno ne ponavljaju.

2.3 Korišćenje šablona za kreiranje jednostavnih baza podataka

Šabloni su unapred dizajnirane baze koje već sadrže tabele, upite, forme i izveštaje prilagođene određenoj nameni. Korišćenje šablona (templates) je najbrži način da se formira funkcionalna baza podataka. Pomažu nam da preskočimo tehničko podešavanje i odmah pređemo na unos podataka.

Uključivanje šablona

Objasnićemo kako se u Access-u uključuje gotov šablon. Umesto kreiranja prazne baze podataka (*Blank Database*), možete iskoristiti biblioteku gotovih rešenja u centralnom delu ekrana. Postupak uključivanja šablona podrazumeva sledeće korake:

1. **Pretraga (Search):** U polje *Search for online templates* na vrhu ekrana ukucajte ključnu reč na engleskom jeziku (npr. `Students`, `Inventory` ili `Tasks`) i pritisnite taster *Enter*.

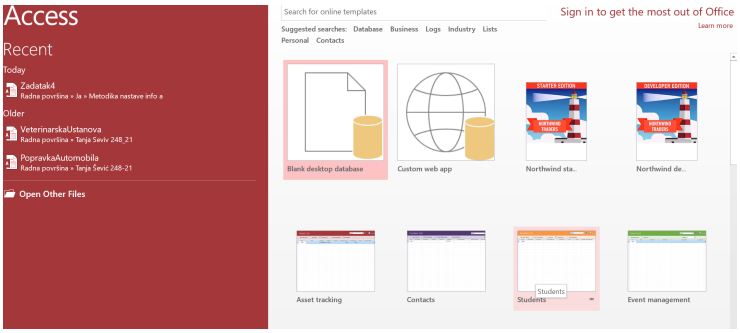
2. **Odabir šablona:** Iz ponuđene liste dostupnih rezultata, kliknite jednom na šablon koji najbolje odgovara vašim potrebama.

3. **Ime i lokacija:** U novom iskačućem prozoru, unutar polja *File Name*, upišite željeno ime svoje baze. Klikom na ikonicu fascikle pored ovog polja birate tačnu lokaciju na računaru gde će baza biti sačuvana.

4. **Kreiranje (Create):** Kliknite na dugme *Create*. Program će preuzeti šablon sa interneta, konfigurisati strukturu i otvoriti bazu.

5. **Omogućavanje sadržaja:** Po otvaranju, pojavljuje se žuta traka sa sigurnosnim upozorenjem (*Security Warning*). Neophodno je kliknuti na dugme **Enable Content** kako bi svi tasteri, makroi i automatizacije unutar šablona postali funkcionalni.

Iako šabloni značajno ubrzavaju proces izrade baze, korisnik treba da bude svestan i njihovih ograničenja.



Slika 5: Primer gotovih šablona u Access-u

Prednosti

- **Brzina izrade:** Baza je spremna za rad za svega nekoliko sekundi, bez potrebe za kreiranjem tabela.
- **Profesionalni dizajn:** Šablone su kreirali stručnjaci, što znači da su relacije između tabela ispravno postavljene, čime su izbegnute greške u logici.
- **Kompletnost:** Uz tabele se dobijaju i dizajnirane forme, upiti i izveštaji.
- **Edukacija:** Analizom šablona, početnik može naučiti kako se pravilno povezuju podaci u složenim sistemima.

Mane

- **Suvišni podaci:** Šabloni često sadrže previše polja koja vam nisu potrebna.
- **Teško prilagođavanje:** Ponekad je teže prepraviti postojeću kompliko-vanu strukturu nego napraviti novu.
- **Specifičnost potreba:** Šabloni su pravljeni za opšte slučajeve. Na primer, američki šablon za školu možda nema polja koja bi odgovarala našem obrazovnom sistemu.
- **Zavisnost od engleskog jezika:** Većina šablona je na engleskom jeziku.

3 Rad s tabelama

Tabele predstavljaju osnovnu komponentu u relacionim bazama podataka. One omogućavaju organizaciju i skladištenje podataka na način koji olakšava upravljanje podacima i dobijanje korisnih informacija.

3.1 Kreiranje tabele

Pokazaćemo kako se pravi tabela u Microsoft Access-u.

Tabela u Access programu može se prikazati i kreirati u dva osnovna oblika, od kojih svaki predstavlja različit pristup organizaciji podataka:

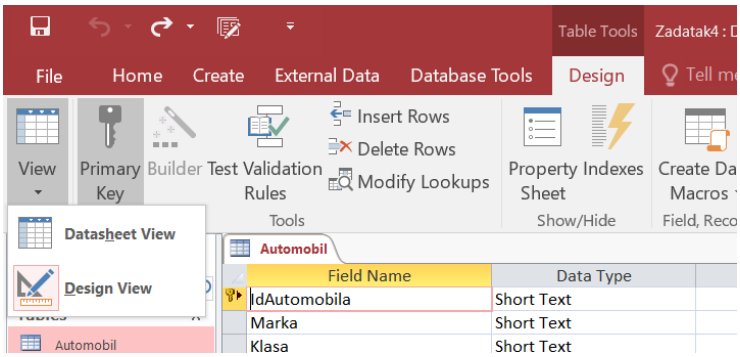
1. **Prikaz lista sa podacima (*Datasheet View*)**
2. **Dizajnerski prikaz (*Design View*)**

Kreiranje tabele započinje upotrebom taba **Create**, gde su dostupne dve ključne komande:

Kreiranje tabele unosom podataka (*Table*)

Ova komanda otvara *Datasheet* prikaz (prikazan na Slici 6). Nova tabela se kreira direktnim unošenjem vrednosti, pri čemu važe sledeća pravila:

- Polja se dodaju unošenjem podataka u kolonu „Dodaj novo polje“.
- Access automatski stvara polje pod nazivom *ID* koje služi kao primarni ključ.
- Ovako kreirani ključ naziva se **veštački primarni ključ**. On se generiše u vidu jedinstvenog broja koji raste (*auto-increment*) ili se dodeljuje nasumično (*random*) svakom zapisu.



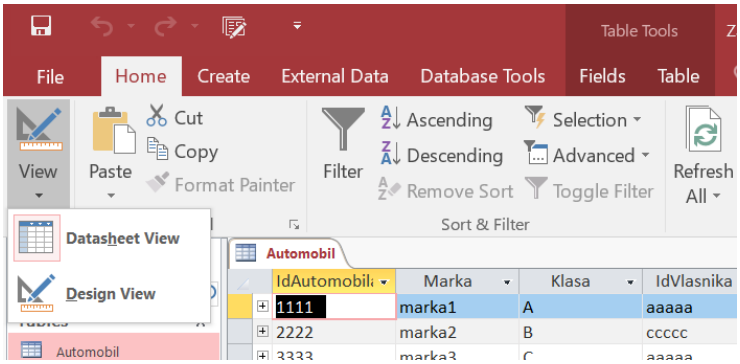
Slika 6: Prikaz lista sa podacima (Datasheet View) - Kreiranje definisanjem polja

Kreiranje tabele definisanjem polja (*Table Design*)

Ova komanda otvara dizajnerski prikaz (prikazan na Slici 7), što predstavlja profesionalniji pristup. Za svako novo polje potrebno je:

- Zadati naziv (*Field Name*), kao što su *Marka* ili *Klasa*.

- Odrediti tip podataka (*Data Type*) koji će se u njemu čuvati, npr. *Short Text*.



Slika 7: Dizajnerski prikaz (Table design) - Kreiranje definisanjem polja

Čuvanje tabele

Kada je tabela kreirana, neophodno je sačuvati strukturu pre unosa velike količine podataka. Čuvanje (*Save*) se može izvršiti na dva načina:

- Klikom na komandu **Save** na traci sa alatkama (Quick Access Toolbar).
- Desnim klikom na jezičak tabele (gde privremeno stoji naziv poput *Table1*) i odabirom opcije **Save**, nakon čega se unosi željeni naziv (npr. *Automobil*).

3.2 Izbor tipa podataka

Izbor tipa podataka *Data Type* je jedan od najvažnijih koraka pri kreiranju tabele, jer on određuje kakvu vrstu informacija polje može da prihvati, koliko će memorije zauzimati i koje operacije će moći da se vrše nad tim podacima.

Tip podataka	**Namena**	**Primer iz našeg zadatka**
Short Text	Tekstualni podaci do 255 karaktera	Polja `Marka` i `Klasa`
Number	Brojne vrednosti za proračune	Broj sedišta ili kubikaža
AutoNumber	Automatski generisan broj	Veštački primarni ključ (ID)
Date/Time	Datumi i vremenske odrednice	Datum poslednje popravke
Yes/No	Logička polja (Check-box)	Da li je automobil na servisu?

Tabela 2: Najčešće korišćeni tipovi podataka u Access-u

Kao što se vidi na Slici 7, definisanje tipa podataka se vrši odmah pored naziva polja, što omogućava Access-u da automatski kontroliše unos korisnika i spreči npr. unos teksta u polje predviđeno za brojeve.

3.3 Postavljanje primarnog ključa

1. **Otvorite tabelu u *Design View*.**
2. **Odaberite polje koje će biti primarni ključ** (tip primarnog ključa je najčešće *AutoNumber*).
3. **Postavite primarni ključ.** Postoje dva načina da to uradite:
   - **Preko izbornika:** Kliknite na željeno polje, a zatim na kartici *Table Design* kliknite na ikonu *Primary Key*.
   - **Desnim klikom:** Kliknite desnom tipkom miša na sivi kvadratić levo od naziva polja i odaberite *Primary Key*.

   Napomena: Pored naziva polja pojaviće se mala ikona ključa.

3.4 Unošenje podataka u tabelu

U Access-u postoje tri nivoa unosa podataka:

Direktan unos u tabelu (Datasheet View)

Ovo je najbrži način unosa, sličan radu u Excel-u, kada dizajn nije primaran.

1. U levoj koloni (*Navigation Pane*) kliknite dvoklikom na željenu tabelu. Tabela će se otvoriti u prikazu mreže.
2. Pronađite red na dnu koji ima ikonicu zvezdice (∗). To je prazan red spreman za novi unos.
3. **Važno:** Ako je prva kolona `ID` podešena na *AutoNumber*, nju preskačete. Access će automatski dodeliti broj.
4. Nakon unosa podataka u red, pritisnite **Enter** ili **Tab**. Podaci se automatski čuvaju.

Unos preko Forme

Forme omogućavaju pregledniji unos jednog po jednog zapisa, čime se smanjuje mogućnost greške.

1. Obeležite tabelu u listi sa leve strane jednim klikom.
2. U gornjem meniju izaberite tab **Create**.
3. Kliknite na dugme **Form**. Access će automatski generisati jednostavnu formu.
4. Za novi unos, koristite navigacionu traku na dnu i kliknite na ikonicu strelice sa žutom zvezdicom (▶ ∗) — *New (Blank) Record*.

Unos preko Lookup polja (Padajući meni)

Koristi se za povezivanje tabela (npr. biranje grada iz unapred definisane liste).

1. Otvorite tabelu u **Design View** (desni klik na tabelu).
2. Za željeno polje, u koloni *Data Type* izaberite **Lookup Wizard**.
3. Izaberite opciju: *„I want the lookup field to get the values from another „table”*.
4. Pratite čarobnjak da povežete polje sa izvornom tabelom.

Pravila pri unosu podataka

- **Tip podataka:** Access strogo kontroliše tipove. Ne možete uneti tekst u polje koje je definisano kao *Date* ili *Number*.
- **ID polja:** Greška *„Index or primary key cannot contain a Null „value”* znači da ste preskočili obavezan ključ koji nije *AutoNumber*.
- **Čuvanje:** Podaci se automatski čuvaju kad se promeni fokus.
- **Brisanje:** Ceo red brišete desnim klikom na sivu kockicu sa leve strane reda i odabirom opcije **Delete Record**. Moraju se popuniti sva polja reda koji želimo da obrišemo, da bismo mogli da obrišemo red.

4 Veza između tabela

Veza je *logička povezanost* između dve tabele koja se uspostavlja preko zajedničkih kolona. Možemo je posmatrati kao „most” koji omogućava bazi da spoji informaciju iz jedne tabele sa informacijom iz druge.

Svrha povezivanja tabela jeste stvaranje koherentne strukture (jedinstvene celine). Koherentna struktura nam omogućava da se tabele ne ponašaju kao gomila izolovanih fajlova, već kao celina. Za to nam pomaže **primarni ključ**, koji omogućava da podatke o subjektu (poput učenika) unosimo samo u jednu tabelu, dok ga u svim ostalim tabelama identifikujemo preko njegove jedinstvene vrednosti.

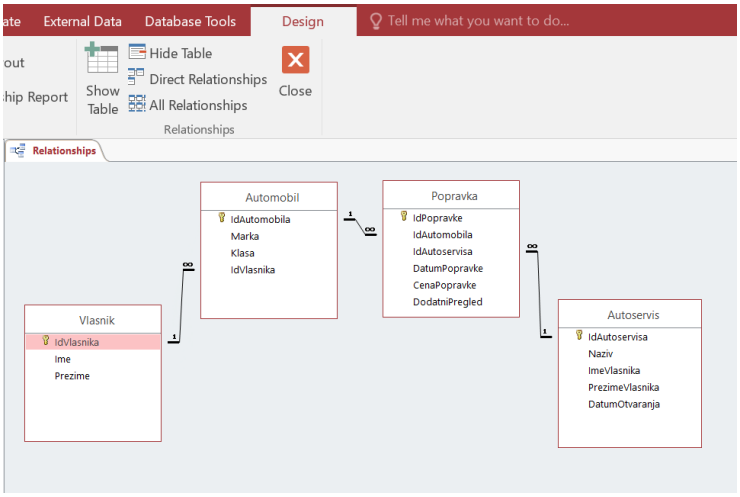
4.1 Pojam veze

Postoji više vrsta veze, u zavisnosti od broja objekata delimo ih na:

- **Jedan-prema-više ($1 : N$):** Ovo je najčešći tip veze. Primeri:
  - Veza između učenika i njegovih izostanaka (svaki izostanak pripada tačno jednom učeniku).
  - Veza između odeljenja i učenika (jedno odeljenje ima više učenika).

- **Više-prema-više ($M : N$):** Pojavljuje se kada više zapisa iz jedne tabele odgovara većem broju zapisa iz druge. Primeri:
  - Veza između učenika i predmeta (učenik ima više predmeta, predmet sluša više učenika).
  - Veza između nastavnika i predmeta (nastavnik predaje više predmeta, a isti predmet može predavati više nastavnika).
- **Jedan-prema-jedan ($1 : 1$):** Specifična veza gde se zapisi iz dve tabele uparuju po principu „jedan na jedan". Ovakve tabele se često mogu predstaviti kao jedinstvena tabela. Primer:
  - Svakom učeniku pripada jedan broj u dnevniku. Svaki broj u dnevniku određuje jednog učenika.

4.2 Kreiranje veze između tabela



Slika 8: Prikaz kreiranja veza između tabela

Da bismo kreirali vezu između tabela potrebni su nam:

- **Primarni ključ (Primary Key)** - osigurava da svaki zapis bude prepoznatljiv (npr. `id_ucenik`).
- **Strani ključ (Foreign Key)** - referencira na primarni ključ matične tabele, čime se fizički uspostavlja veza između podataka.
- **Zajednički tip podataka** - da bi veza funkcionisala, polja koja se povezuju moraju imati identičan tip podataka (npr. oba moraju biti *Number*).

- **Referencijalni integritet** - pravilo koje aktiviramo prilikom kreiranja veze kako bismo osigurali da baza ne dozvoli unos podataka koji nemaju svog „vlasnika” u matičnoj tabeli (npr. da se ne može uneti ocena za nepostojećeg učenika).

Nakon što obezbedimo potrebne elemente, u programu Access koristimo alat **Relationships** iz kartice **Design**, pomoću kog vizuelno prevlačenjem primarnog ključa preko stranog, definišemo međusobnu zavisnost tabela.

Literatura

[1] Školska informatika
Dostupno na: https://skolskainformatika.weebly.com/uploads/9/2/2/7/9227239/kreiranje_tabele_i_rad_sa_tabelama.pdf

[2] Portal Petlja,
Dostupno na: https://petlja.org/sr-Latn-RS/kurs/18676/12/6426

[3] Kako se pravi SQL tabela,
Dostupno na: https://www.doit.rs/kako-se-pravi-sql-tabela/

[4] Planiranje jednostavnih baza podataka,
Dostupno na: https://nastavaracunarstva.wordpress.com/2013/10/17/planiranje-jednostavnih-baza-podataka/