**Pojam baze podataka**

**Informacioni sistem –** integrisani skup komponenti za sakupljanje, snimanje, čuvanje, obradu i prenošenje informacija. Osnovne komponente su: hardver, softver, podaci, procedure za rad i ljudski resursi. Važan deo **svakog** informacionog sistema je **baza podataka**.

**Baza podataka –** je skup međusobno povezanih podataka koji se čuvaju zajedno i među kojima ima samo onoliko ponavljanja koliko je neophodno za njihovo optimalno korišćenje, najčešće od strane većeg broja korisnika. Ovakva organizacija podataka omogućava brzo dodavanje, izdvajanje, brisanje, povezivanje podataka.

**Podatak –** činjenica koja se čuva u b.p.

Primeri:

Naziv knjige je Računarstvo i informatika, prezime autora udžbenika je Vuković , danas je petak...

**Informacija –** se dobija obradom podataka

Primeri:

Ukupan broj knjiga iz Informatike, prosečna starost profesora u skoli, prezimena svih učenika rođenih u maju...

Softver koji upravlja bazom podataka je **Sistem za upravljanje bazama podataka** **DBMS** (Database Management System). Pored organizacije podataka u bazi podataka, ovi softverski paketi nude mogućnosti jednostavnog unosa, obrade, pretraživanja, uklanjanja i zaštite podataka prilikom višekorisničkog rada i laku administraciju sistema. Neki od najpoznatijih su: Access (Microsoft), MySQL (Sun), dBase, 11g i 12c (Oracle), DB2(IBM), SQL Sever (Microsoft)

Baza podataka mora da zadovoljava sledeće kriterijume:

1. Treba čuvati **sve** potrebne podatke (npr. ne smemo dozvoliti da nemamo cenu proizvoda ili naziv proizvođača)
2. Podatke čuvati samo na jednom i to pravom mestu. Podaci koje čuvamo na više mesta omogućavaju nam da vidimo u kakvoj su **vezi** podaci
3. Ne čuvati kao podatke one informacije koje mogu da se izračunaju na osnovu drugih podataka

**Podela baza podataka**

Na osnovu logičke strukture podataka baze podataka se dele na:

1. Hijerarhijske - zasnivaju se na hijerarhijskim strukturama podataka koje imaju oblik stabla. Ova struktura ima nivoe, s tim da je na prvom nivou osnovni ili koreni segment. Postoji dva tipa slogova – roditelj i dete. Jedan slog dete može imati samo jednog roditelja.
2. Mrežne - Zasnivaju se na mreži podataka povezanih tako da ne postoje ni osnovni ni podređeni segmenti. Mrežna struktura složenija je od hijerarhijske i u praksi se baze podataka mogu prevesti iz hijerarhijskog oblika u mrežni i obrnuto.

**predmet**

**student**

predmet1

predmet2

predmet3

student1

student2

student3

student4

student5

1. Relacione -zasnivaju se na matematičkom pojmu relacije

UČENIK (lični\_broj, ime, prezime, odeljenje)

Naziv relacije atributi

**Arhitektura sistema za upravljanje bazama podataka**

Strukture podataka, procedure

procedure

Pogledi

Spoljašnji nivo

Model podataka

Logički nivo

Fizička baza podataka, unutrašnji nivo

ANSI arhtektura baze podataka

Pogledje izgled baze podataka za različite korisnike

Model podataka - logička struktura podataka u bazi i skup operacija koje korisnik može izvršiti. Ovako mi vidimo bazu kada sa njom radimo.

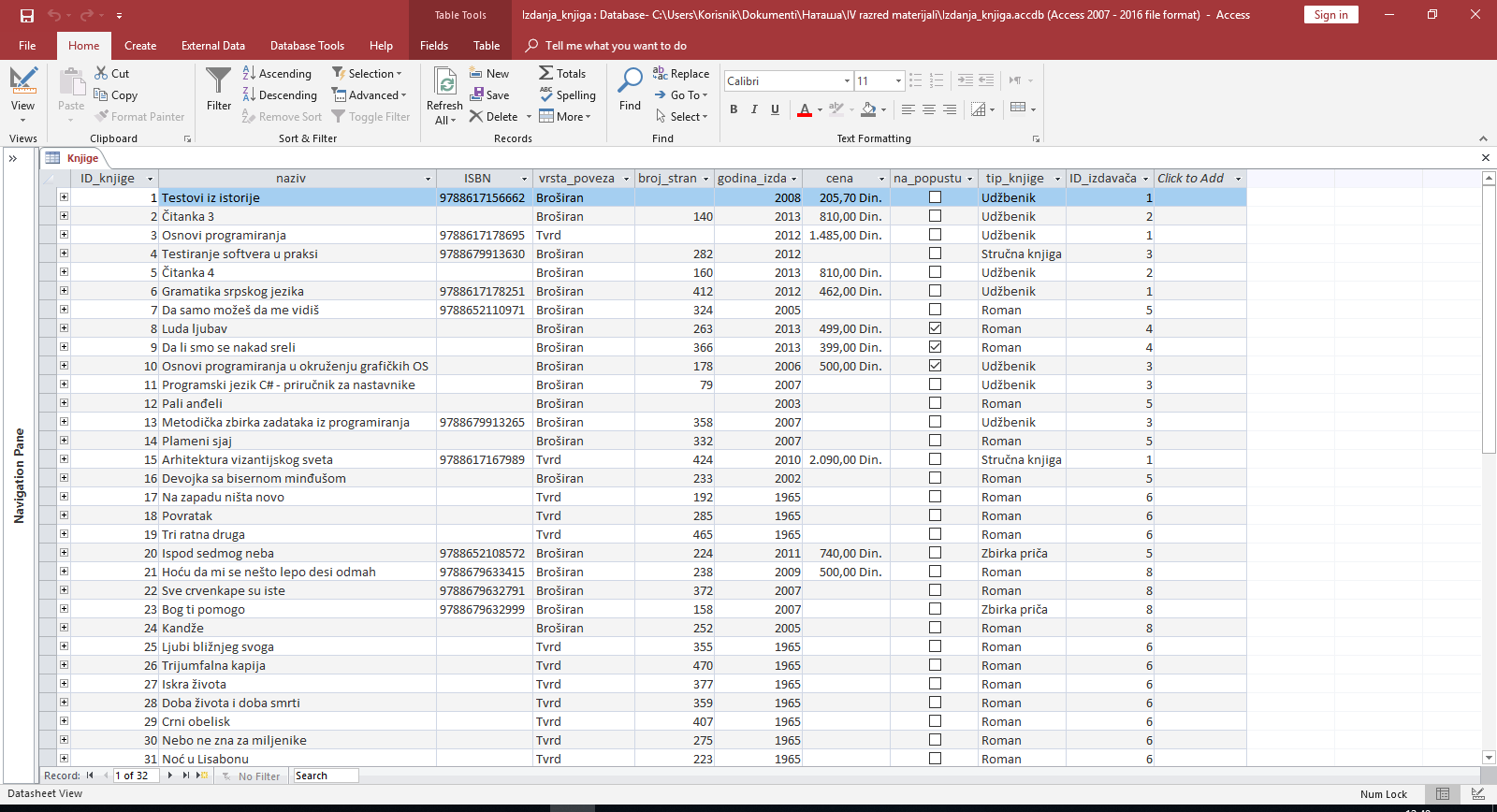
Čine ga:

1. Strukturni deo - u relacionoj bazi to su **entiteti**, logičke celine koje odgovaraju tabeli (KNJIGA, AUTOR)
2. Manipulativni deo – realizuje se postavljanjem upita
3. Pravila integriteta – garantuju da će stanje baze biti valjano (ne može se obrisati autor koji ima naslove u tabeli KNJIGA)

**Relaciona baza podataka**

Arhitektura najvećeg broja baza podataka odgovara predlogu američkog instituta za standarde (ANSI- American National Standards Institute)

U relacionoj bazi podataka entitet je predstavljen **tabelom, instanca entiteta -**redom tabele**,** a **atributi** su predstavljeni jednom kolonom tabele. Polje tabele koje čuva podatak nalazi se u preseku 1 reda i 1 kolone.



**polje Instanca entiteta =red**

**Entitet=tabela Kolona=atribut**

**Primarni ključ tabele –** vrednost ovog polja razlikuje se za svaki red tabele. Svaki red u tabeli mora da ima popunjenu vrednost primarnog ključa, vrednosti primarnog ključa ne smeju da se ponove, a ako više kolona čini primarni ključ, onda njihova kombinacija mora biti jedinstvena.

Tabela može da ima samo jedan primarni ključ i bitno je da se on pažljivo odabere (JMBG kod građana, PIB kod preduzeća ...). U praksi se često sreće rešenje da se uvodi veštački primarni ključ čije vrednosti automatski popunjava različitim vrednostima SUBP (tip podatka AutoNumber). Primarni ključevi služe i za povezivanje tabela

**Strani ključ –** kada jedna ili više kolona jedne tabele sadrži vrednosti primarnog ključa druge tabele, tu kolonu zovemo strani ključ.

Knjige Izdavač

ID\_KNJIGE ID\_IZDAVAČA *PRIMARNI KLJUČ*

NAZIV ADRESA

ISBN TELEFON

....

ID\_IZDAVAČA *STRANI KLJUČ*

**Veze između tabela**

Veza između tabela služi da se precizira u kakvom su odnosu podaci koji se čuvaju u tačno dve različite tabele. Ista tabela može da učestvuje u više veza.

Prema broju redova jedne tabele koji mogu biti u vezi sa redovima druge tabele, postoje tri različita tipa veze:

**1:1 - jedan prema jedan**

**1:n – jedan prema više**

**n:n – više prema više**

* Veza **1:1** – retka je, ali nam je nekada potrebna radi uštede memorije ili organizacije podataka po logičkim celinama**.** Na primer - jednom izdavaču u tabeli Izdavči odgovara jedan red u tabeli Kontakt\_izdavača sa njegovim kontakt informacijama.

Izdavači Kontakt\_izdavača

ID\_IZDAVAČA ID\_IZDAVAČA

NAZIV ADRESA

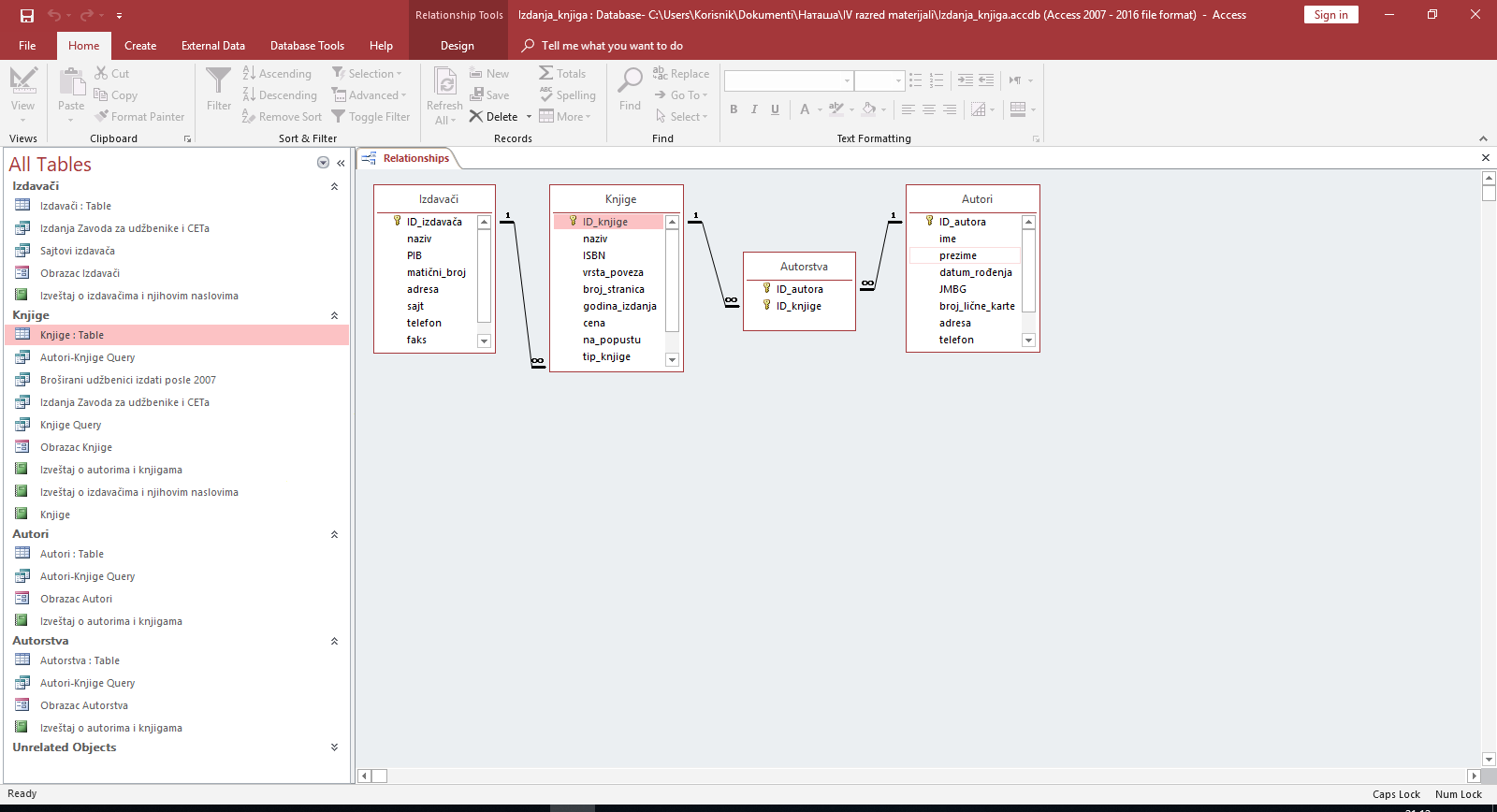
PIB TELEFON

MAT\_BR FAKS

MEJL

SAJT

* Veza **1:n** –najčešća veza**,** realizuje se preko stranog ključa. Jedan red jedne tabele može da bude u vezi sa više redova druge tabele dok obrnuto ne važi. Primer-veza između tabela Izdavači i Knjige. Jedan izdavač može da izda više knjiga, a jedna knjiga može imati samo jednog izdavača.
* Veza **n:n** – česta, ali nju nije moguće direktno realizovati. Ovakav tip veza zahteva da se kreira dodatna tabela. Veze ove dve tabele prema novoj su 1:n. Primer - tabela Knjige i Autori su u vezi n:n, jer jedan autor može napisati više knjiga, a jedna knjiga može imati više autora i zato formiramo novu tabelu Autorstva u kojoj ćemo imati složeni primarni ključ sastavljen od ID\_AUTORA i ID\_KNJIGE ili ćemo uvesti novu kolonu ID\_AUTORSTVA da bude primarni ključ ove tabele.



Veza n:n rešena uvođenjem nove tabele

Veza 1:n

**Referencijalni integritet** je pravilo koje nam garantuje ispravnost podataka stranog ključa. U koloni koja je strani ključ ne sme da postoji nijedna vrednost koja se ne nalazi u koloni koja je primarni ključ druge tabele. Na primer: ako pokušamo da unesemo proizvod sa šifrom prozvođača koja ne postoji u tabeli Proizvođači, sistem će to sprečiti jer je narušen referencijalni integritet.

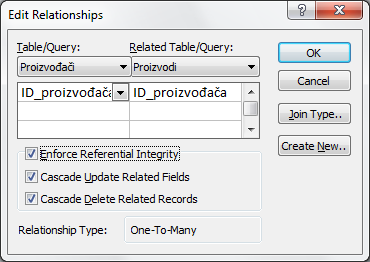
Pravila koja garantuju očuvanje referencijalnog integriteta podrazumevaju:

1. Pravila za brisanje

Kada kreiramo vezu između tabela i omogućimo referencijalni integritet, sistem nam nudi da potvrdimo opciju Cascade delete Related Fields, a to znači da će se pri brisanju vrednosti primarnog ključa jedne tabele obrisati i vrednosti stranog ključa povezane tabele

1. Pravila za izmene

Slično kao kod brisanja, u slučaju da potvrdimo opciju Cascade Update Related Fields, ako izmenimo šifru nekog proizvođača u tabeli Proizvođači, izmeniće se šifra proizvođača za sve proizvode tog proizvođača.



ZADATAK ZA VEŽBU

Kreirati bazu podataka **Bolnica** sa sledećim tabelama:

Bolnica Pacijenti Lekari Pregledi

ID\_bolnice ID\_pacijenta ID\_lekara ID\_pregleda

Naziv ID\_bolnice ID\_bolnice ID\_pacijenta

Adresa Ime Ime ID\_lekara

Telefon Prezime Prezime Datum pregleda

godine Specijalizacija Dijagnoza

(pedijatar, internista, hirurg) Uput (Yes/no)

Povezati tabele u bazi.