

Predstavljanje relacija i šema relacija

Relaciju možemo predstaviti tako da navedemo njeno **ime i skup atributa** koji je čine:

$R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$, ako domeni nisu bitni dovoljno je navesti samo atribute:

$R(A_1, A_2, \dots, A_n)$.

Primjer: Učenik(Razred, BrojDnevnik, Ime, Prezime)

Uobičajena konvencija za predstavljanje relacione šeme je **dijagram relacije**. Relaciju predstavljamo pravougaonom tabelom koja ima onoliko ćelija koliko je atributa u relaciji.

Ime relacije se ispisuje iznad tabele, a imena atributa u ćelijama .

Primjer: dijagram relacione šeme baze podataka Generacija2016.

L_Podaci

IMBG	Odjeljenje	Br_dnevnik	Prezime	Ime	Dat_rođ	Pol	Adresa	Grad
------	------------	------------	---------	-----	---------	-----	--------	------

Odjeljenja

Odjeljenje	Smjer
------------	-------

Primjer dijagrama relacija

Osnovne osobine koje treba da ima relacioni model

SUBP (DBMS) se naziva relacionim ako podržava relacione operacije.

Osim što podržava relacione operacije, da bi neki model mogao da se nazove relacionim, on treba da ima osnovne osobine kao što su:

- Sistemski tretman *NULL* vrijednosti.
- *NULL* je univerzalna vrijednost *nepoznato*. Koristi se kad iz bilo kog razloga u momentu unosa podataka nije poznata vrijednost nekog ulaznog podatka. *NULL* može biti vrijednost svake kolone bez obzira na njen tip, **osim primarnog ključa** i u slučaju kad se izričito zahtijeva da vrijednosti u nekoj koloni ne smiju biti nedefinisane (dodatno ograničenje **NOTNULL**).
- Neprekidan pristup dinamičkom katalogu relacionog modela.
- Katalog relacionog modela predstavlja zapis o strukturi baze podataka. Spada u tzv. metapodatke, koje smo spomenuli u uvodu. Zapisan je u istoj formi (relacije-tabele) kao i sami podaci. Pravo pristupa ovom katalogu imaju samo ovlašćeni korisnici – administratori. Naziva se dinamički, jer definicije podataka koje se mogu vidjeti odgovaraju upravo tekućem stanju u bazi podataka.
- Fizička nezavisnost znači da korisničke aplikacije ostaju neizmijenjene kada se promijeni fizička organizacija baze ili fizički metod pristupa podacima (vidi Arhitektura baze podataka). Praktično, fizička organizacija podataka ima uticaj samo na performanse (brzinu pristupa podacima, odziv sistema).
- Logička nezavisnost.
Ekvivalentno prethodnom uslovu, samo u obrnutom smjeru.
- Integritetska nezavisnost:
 - Integritet entiteta definiše da je primarni ključ jedinstven na nivou cijele relacije.
 - Integritet domena određuje skup dozvoljenih vrijednosti kolone.
 - Relacioni integritet.
Svaki atribut u relaciji vezuje se za određeni domen, pa povezivanjem atributa ne smije da se naruši (analiziraj kasnije kardinalnost).
 - Referencijalni integritet odražava definisane odnose (relacije) među tabelama kada se u njih dodaju ili se iz njih brišu zapisi (analiziraj kasnije anomalije), naglašava da izmjena bilo kog podatka ne smije da mijenja **karakter** drugog.

Zadatak: uvod u kreiranje relacionog modela

Želimo kreirati bazu podataka **Generacija2016** koja predstavlja generaciju učenika (npr. neke gimnazije, koju ćemo analizirati kao sistem iz realnog svijeta koji želimo modelirati i dizajnirati bazu koja ga predstavlja).

Izbor relacija i definisanje atributa

Prvo analiziramo problem, odredimo podatke koje želimo pamtiti (poželjno je da to uradimo u pisanoj formi), na osnovu toga definišemo objekte i podatke, prepoznamo entitete i njihova svojstva, koje želimo predstaviti relacijama; odnosno definišemo tabele i kolone.

Krenućemo od skupa učenika škole koji želimo predstaviti relacijom-tabelom Ucenik u kojoj ćemo smještati i čuvati podatke o razredu, odjeljenju, smjeru, broju u dnevniku, imenu, prezimenu, datumu rođenja, polu i adresi stanovanja.

Nedostaju predmeti i ocjene, njihovo uvođenje ćemo realizovati kasnije, kada budemo radili u MS Access-u.

Moguće rješenje je definisanje baze Generacija 2016 sa dvije tabele-relacije: L_Podaci i Odjeljenja i njihovih atributa (kolona).

L_Podaci

JMBG	Odjeljenje	Br_dnevnik	Prezime	Ime	Dat_rod	Pol	Adresa	Grad
0123456	I-1	18	Marković	Marko	10-01-01	M	Beogradska 7	Zvornik
0123456	I-1	3	Žarić	Ana	05-03-00	Ž	Lj. Bogdana 3	Višegrad
2345678	I-2	22	Petrović	Milan	02-01-01	M	Karađorđa 7	Teslić
3466677	I-2	23	Rodić	Ana	05-12-99	Ž	Savska 34	Kotor Varoš
4565553	I-3	22	Petrović	Petar	09-12-00	M	Čairska 3	Laktaši
3219871	I-3	1	Adžić	Zlatko	07.07-00	M	S. Đaka 19	Banja Luka

Odjeljenja

Odjeljenje	Smjer
I-1	opšti
I-2	opšti
I-3	društveno-jezički
I-4	matematički
I-5	računarski

Nastavak zadatka: Definirati domene atributa relacije L_podaci.

Rješenje:

Atribut	Domen	Značenje	Definicija domena
JMBG	Jedinstveni broj građanina	Skup mogućih matičnih brojeva radnika	Niz karaktera, dužine 12 ¹⁰
Odjeljenje	Oznaka i broj odjeljenja	Skup mogućih imena osoba	Niz karaktera, dužine do 5
Br_dnevnik	Broj u dnevniku	Skup mogućih brojeva u dnevniku	Cijeli brojevi, opseg 1–32
Prezime	Prezime učenika	Skup mogućih prezimena	Niz karaktera, dužine do 15.
Ime	Ime učenika	Skup mogućih imena	Niz karaktera, dužine do 15.
Dat_Rođ	Datum Rođenja	Moguće vrijednosti za datume rođenja učenika prvog razreda	Datum, opseg, od 01-JAN-99 nadalje
Pol	Pol	Pol učenika	Karakter (1), vrijednost M i Ž
Adresa	Adresa učenika	Moguće adrese učenika	Niz karaktera (25)
Mjesto	Mjesto stanovanja	Moguće mjesto stanovanja	Niz karaktera (15)

Kandidati za ključ i primarni ključ

Kandidat za ključ (*candidate key*) je atribut, ili skup atributa, čije vrijednosti jednoznačno određuju n-torku unutar relacije.

Na primjer, kandidat za ključ relacije Auto mogao bi biti atribut RegistarskiBroj.

Svaka relacija mora da ima barem jednog kandidata za ključ: to je skup svih atributa koji čine torku. Kandidat za ključ mora da ispuni još jedan dodatan uslov, a to je da se **ne može razbiti na prostije dijelove**; zbog toga skup svih atributa relacije ne mora *obavezno* biti kandidat za ključ.

Kandidat za ključ se može sastojati od samo jednog atributa – **prost ključ** (*simple key*), ili od više njih – **složen ključ** (*composite key*).

Neki autori smatraju da složeni ključevi nisu dobro rješenje i zato dodaju svojim tabelama **vještačke identifikatore** – npr. polja tipa *autonumber*; da bi izbjegli složene ključeve.

Ukoliko relacija ima više kandidata za ključ, tada **biramo jednog** od njih i **proglašavamo ga primarnim ključem**.

Sada, kad smo se upoznali sa primarnim ključem, relaciji možemo pridružiti i **osobinu adresabilnosti**:

Svaka n-torka relacije određena je vrijednošću primarnog ključa.

Svaki podatak je određen nazivom relacije, nazivom atributa i vrijednošću primarnog ključa.

Atributi koji čine primarni ključ zovu se **primarni atributi**, a svi ostali se zovu **sporedni atributi**.

Primarni ključ ima dvostruku ulogu: jednoznačno definiše n-torke (red u tabeli) i omogućava da se preko njega ostvari veza sa drugim relacijama (tabelama).

- Kako se relacija može predstaviti?
- Osnovne osobine koje treba da ima relacioni model?
- Šta je kandidat za ključ?
- Razlika između prostog i složenog ključa?
- Šta je određeno vrijednošću primarnog ključa?