

Питања за понављање (одговоре наћи у свескама):

- Шта је податак, а шта информација?
- Шта је ентитет?
- Шта је атрибут?
- Навести најчешће типове атрибута.

Domen atributa

Domen (domain) je skup svih prihvatljivih vrijednosti koje atribut može imati.

Domen atributa definiše se: tipom podataka, dužinom podataka i opsegom vrijednosti.

Definicija domena zahtijeva potpun opis važećih podataka.

To **može biti lista** sa svim prihvatljivim vrijednostima

(npr. domen atributa Ocjena: {1,2,3,4,5} ili

StručnaSprema: {niža, srednja, viša, visoka, master, doktorat}).

Ako se atributu ne definiše domen, onda on uzima sve vrijednosti tipa koji ga određuje, što ponekad dovodi do zabune. **Domen i tip nisu jedno te isto, izuzev u slučajevima kad zadržimo predefinisane sistemsku vrijednost domena (INTEGER, CHAR itd.).**

Domen je skup vrijednosti istog tipa, npr. skup imena rijeka nekog sliva, skup imena igrača nekog tima, skup imena mjesnih zajednica nekog grada, skup naslova knjiga neke biblioteke, itd.

Relacioni model

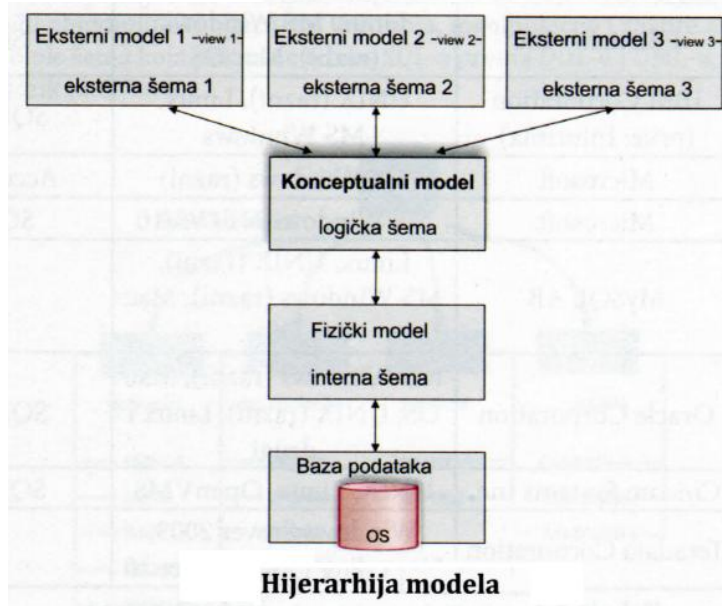
Model je reprezentacija skupa entiteta (objekata) i njihovih međusobnih veza. Izbor i definisanje entiteta i veza između njih je suština procesa modeliranja.

Postupak definisanja relacija – tabela i veza među njima znači kreiranje modela relacione baze podataka.

Modeli baze podataka

Na konceptualnom nivou definiše se konceptualni model sa odgovarajućom logičkom šemom koja sadrži opis svih entiteta i veza, atributa, domena i integritetska ograničenja.

Hijerarhijsko mjesto konceptualnog modela vidi se na slici (između korisnika i fizičkog nivoa).



Eksterni modeli (1–3) nastaju na osnovu konceptualnog modela i predstavljaju samo pogled (view) koji je nastao prema potrebama korisnika.

Na slici se može primijetiti da fizički model omogućava predstavljanje konceptualnog modela u formi zapisa podataka na neki fizički medij (najčešće je to disk, a *posrednik* između medija i fizičkog modela je operativni sistem – OS).

Kad se govori o modelima baze podataka, **najčešće se misli na konceptualni model**, pa ćemo mi ubuduće **kad kažemo model smatrati da je to konceptualni model** (a kad to ne bude slučaj, to ćemo posebno naglasiti).

Definisanje logičke šeme, odnosno pravljenje konceptualnog modela (uopšte, pa i kod baza podataka), podrazumijeva tri koraka:

- **detekciju i selekciju objekata** (entiteta) i **veza** između objekata,
- **imenovanje** objekata, njihovih atributa i veza,
- **klasifikaciju**, gdje se objekti svrstavaju u klase i tipove.

Vrste modela baza podataka

Model baze podataka je **formalni sistem** koji se sastoji od:

- skupa objekata – osnovnih elemenata (konceptata) baze podataka,
- skupa operacija koje se provode nad objektima,
- skupa integritetskih ograničenja (*integrity constraints*) kojima se definiše skup važećih pravila.

Razlikujemo nekoliko vrsta modela.

Jedan od njih je npr.

Ravni (tabelarni) model

Ravni model poznat i kao model ravnih datoteka (*flat files*) jeste model gdje se svi podaci čuvaju u samo jednoj tabeli. Glavni nedostatak jeste velika redundancija, a primjer koji bi odgovarao ovom modelu bila bi Excel tabela.

Ostali su: hijerarhijski model podataka, mrežni model podataka, relacioni model podataka.

Mi ćemo se baviti relacionim modelom koji je trenutno najrasprostranjeniji.

Osnivač relacionog modela Kod (engleski matematičar i naučnik) je osnovne pojmove relacionog modela preuzeo iz matematičke teorije.

Relacija, kao osnovni koncept relacionog modela je matematička relacija (često se može sresti pogrešna tvrdnja da se relacioni model tako zove zbog veza – relationships; Relacioni model je dobio ime po relacijama na kojim se model zasniva).

Relacija je imenovani podskup Dekartovog (kartezijevog) proizvoda dva ili više domena. Kombinacije vrijednosti domena su n-torke.

Nešto preciznija definicija relacije:

Neka su D_1, D_2, \dots, D_n ($n \equiv N$) domeni (ne obavezno različiti). Dekartov proizvod tih domena, u oznaci $D_1 \times D_2 \times \dots \times D_n$, jeste skup n-torki (v_1, v_2, \dots, v_n) takvih da je $v_i \in D_i$ za sve $i=1, 2, \dots, n$. Relacija R stepena n , definisana na domenima D_1, D_2, \dots, D_n , je proizvoljni konačni podskup navedenog Dekartovog proizvoda.

Prema tome, **relacija je skup n-torki**.

Relaciji bi u standardnoj računarskoj terminologiji odgovarala jedna datoteka, a n-torci jedan slog te datoteke.

Relacija ima jednostavnu reprezentaciju u obliku dvodimenzionalne tabele sa podacima, prikazanu na slici

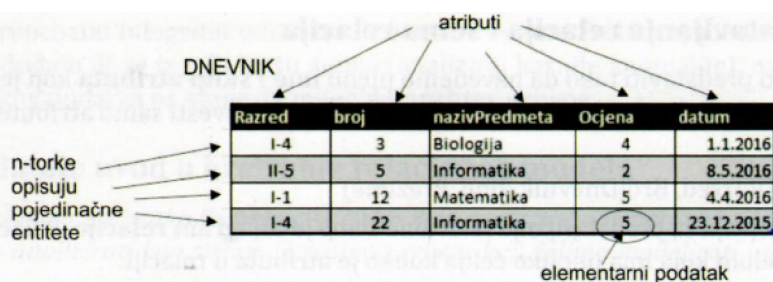


Tabela kao reprezentacija relacije

Nakon možemo da definišemo relacionu bazu: **Relaciona baza podataka je skup vremenski promjenljivih relacija (n-torki u relacijama).**

Sistemi relacionih baza podataka imaju sljedeće odlike:

- Svi podaci se predstavljaju preko **relacija** (*relation*) u obliku dvodimenzionalne tabele sa podacima. Osim atributa svakog entiteta, model podataka mora da definiše **i veze** ili odnose koje postoje između entiteta. **Veza** ili **odnos** (*relationship*) je udruživanje – asocijacija između entiteta.
- Sve vrijednosti su **skalarne**.
- Sve operacije obavljaju se nad cijelom relacijom, a rezultat je takođe cijela relacija. Taj koncept je poznat kao **cjelovitost** (*closure*).

Skup operacija koje se provode na relacijama, odnosno tabelama naziva se relacionom algebrom.

Šema relacije: Relaciona tabela i Šema relacione baze

Šema relacije predstavlja opis strukture tabele. Relaciona šema se još naziva i relaciona tabela, a nju karakterišu sljedeća svojstva:

- U relacionoj šemi može postojati samo jedan tip slogova, odnosno n-torki;
- Svaki red, odnosno slog ili n-torka (*tuple*) uključuje tačno određen broj polja podataka, tj. atributa i svaki od njih je eksplicitno imenovan;
- Relaciona šema ne sadrži dva jednaka naziva atributa, odnosno relacija ne sadrži dvije jednake kolone;
- **Redoslijed kolona, odnosno atributa je nebitan;**
- **Redoslijed n-torki je nebitan;**
- **Relacija nije uređena.** Relaciju možete zamisliti kao košaru u kojoj su torke nagomilane bez ikakvog posebnog redoslijeda. **Redni brojevi zapisa**, uobičajen mehanizam za pristupanje zapisima u nerelacionim bazama podataka, **ne postoje u relacijama;**
- Nove se tabele mogu stvarati povezivanjem preko vrijednosti polja podataka iz istog domena iz dvije postojeće tabele.

Naglasimo: relacija i tabela nisu isto. Kod tabela je bitan redoslijed redova i kolona, dok kod relacija nije bitan redoslijed atributa i n-torki.

Šemu relacije možemo shvatiti dvojako: kao definiciju strukture neke datoteke (kako smo naveli u definiciji), ali i kao predstavu svojstava klase objekata i veza unutar nekog sistema.

Šema relacione baze podataka je skup šema njenih relacija.

- Tri koraka definisanja logičke šeme?
- Od čega se sastoji model baze podataka?
- Vrste modela baza podataka?
- Nabrojati svojstva relacione šeme.