

# OSNOVNI POJMOVI

Podatak; entitet; model podataka;  
DBMS; hijerarhijski, mrežni i  
relacioni model podataka

# Pojam podatka

- **Podatak** je iskaz definisan jednom prostom izjavnom rečenicom
- **Informacija** je novi podatak koji poseduje neku relevantnu novinu, a rezultat je obrade poznatih podataka.

# Entitet

- **Entitet** je digitalna slika, opisana ograničenim brojem diskretnih vrednosti (atributima), nekog kontinualnog procesa koji se odvija u realnom svetu (nekog dela realnog sveta).
- Svojstva entiteta opisuju se atributima.

# Baza podataka

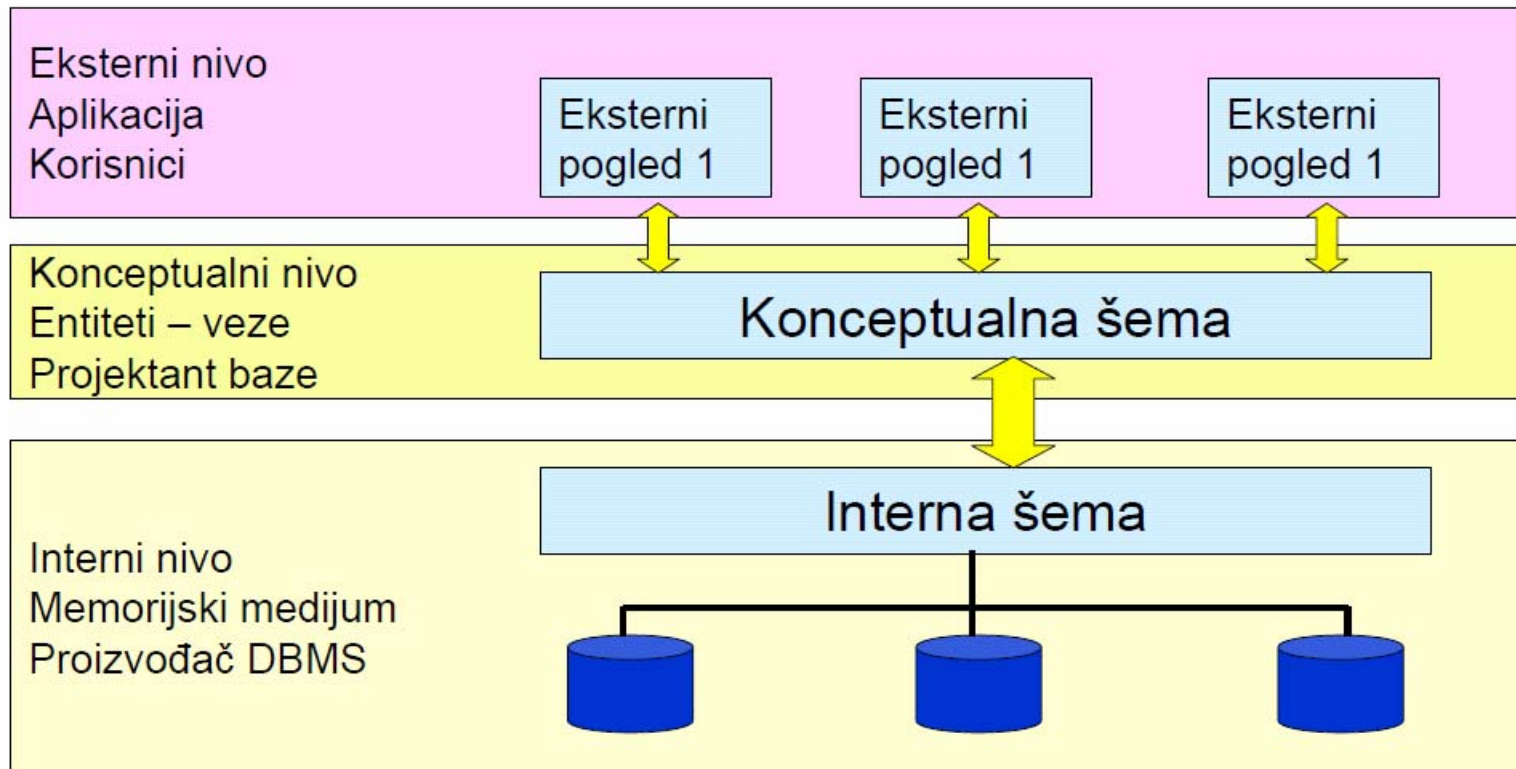
- Kolekcija međusobno logički povezanih podataka sa određenim značenjem

**DMBS** – (Data Base Management System)

Sistem za upravljanje bazama podataka

- Skup programa i procedura koji omogućavaju korisnicima da kreiraju, čuvaju, organizuju i koriste baze podataka

# ARHITEKTURA DB





# ARHITEKTURA SISTEMA ZA UPRAVLJANJE BAZAMA PODATAKA

Pri realizaciji celokupnog sistema baze podataka razlikujemo tri nivoa:

- **Interni nivo**

Nivo koji vodi računa o načinima fizičkog skladištenja podataka i njihovoj manipulaciji. Orijehtisan je prema računaru. Na internom nivou baza se opisuje internom shemom (*internal schema*) kojom se definiše način čuvanja podataka, tipovi podataka, definicija indeksa, veličina uskladištenih podataka itd.

- **Koncepcijski nivo**

Predstavlja opšti pogled na bazu, sagledavanje problema. Podrazumeva koncept realizacije baze podataka, na bazi saznanja o projektovanju baze podataka.

- **Eksterni nivo**

Orijehtisan prema korisniku, bavi se realizacijom programskog sistema, koji krajnjem korisniku omogućava pristup i obradu podataka. Podrazumeva izradu front-end korisničke aplikacije za rad sa bazom. Izbor programskog jezika zavisi od mogućnosti pojedinih alata kao i potreba krajnje aplikacije.

# SISTEM ZA UPRAVLJANJE BAZOM PODATAKA

Sistem za upravljanje bazom podataka (DBMS – Database Management System) je programski sistem koji obezbeđuje osnovne funkcije odabranog modela podataka u radu sa bazom podataka. Osnovni zahtevi koji se postavljaju pred savremeni DBMS:

- Opis i manipulacija podacima pomoću posebnog jezika (SQL).
- Zaštita integriteta podataka
- Visok nivo **interfejsa prema korisniku**, bez obzira na strukturu podataka
- Skup programskih alata za jednostavno razumevanje i korišćenje podataka u bazi.

DBMS se može podeliti na dva dela:

- Backend

Omogućava funkcije definisanja podataka, integritet i zaštitu podataka, manipulaciju itd.

- Frontend

Aplikacije, bilo da ih je napisao korisnik ili su sastavni deo DBMS-a (databasetools).

# Funkcije DBMS

- Čuvanje veoma velike količine podataka tokom dužeg vremenskog perioda
- Zaštita podataka od slučajnog ili neautorizovanog korišćenja
- Obezbeđivanje efikasnog i brzog pristupa podacima tokom upita ili modifikacije
- Kontroliše pristup podacima od strane više korisnika u istom trenutku



## **Administrator baze podataka**

- Administrator baze podataka (DBA – Database Administrator) je osoba zadužena za instalaciju, konfiguraciju i održavanje baze podataka. Prilikom izrade baze podataka, osoba koja stvara bazu najčešće postaje njen DBA. Administrator ima najviši nivo korisničkih prava, što se tiče pristupa bazi i manipulisanja podacima. DBA dodaje ostale korisnike, u njegovoj je nadležnosti da određenim korisnicima dozvoli ili zabrani pristup pojedinim podacima itd. Isto tako DBA je zadužen za održavanje baze (backup).

# IMPLEMENTACIONI MODELI BAZA PODATAKA

## MODEL

- Relacioni
- Mrežni
- Hijerarhijski
- Objektno orijentisani



## OSNOVNA STRUKTURA

- Relacija
- Graf
- Stablo
- Pokazivači

# TIPOVI I STRUKTURE BAZE PODATAKA

## **Centralizovana baza podataka – terminalski pristup**

- Podrazumeva smeštaj podataka na jednom mestu (serveru) i terminalski pristup od strane korisnika. Ovakav pristup znači da se svi zahtevi i obrade podataka vrše na serveru, na kojem su smešteni i podaci. Korisnik preko terminala, jedino unosi svoje zahteve, te dobija prikaz rezultata željenih operacija. Ovakav način organizacije baze postavlja velike zahteve za server, koje osim smeštaja svih podataka, vrši i sve operacije obrade podataka, njihovog formatiranja i prikaza. Stoga server mora imati vrlo veliku procesorsku snagu i visoke performanse.

## **Client- server pristup**

- Client – server struktura podrazumeva smeštaj podataka na server koji izvršava većinu zahteva za manipulisanje i obradu podataka. Korisnici koji pristupaju bazi, taj pristup ostvaruju preko svojih PC računara koji su mrežom povezana sa serverom. Za razliku od terminalskog pristupa, kod kojeg terminalske stanice nemaju nikakve mogućnosti učestvovanja u obradi podataka, PC računari na strani klijenta delimično učestvuju u obradi podataka. Korisnik preko programskog interfejsa formira zahtev za određenim podacima, koji se prosleđuje serveru. Server prihvata zahtev, obrađuje ga, a rezultate obrade vraća klijentu. Klijentski računar prihvata tako obrađene podatke i formira ih u obliku koji je definisan korisničkim interfejsom.



## **Paralelna struktura baze podataka**

Paralelna struktura podrazumeva formiranje baze u okruženju računara međusobno povezanih u lokalnu mrežu. Između računara postoji brza veza, ostvarena preko mrežnih kartica 10/100MBs. Podaci su najčešće smešteni na samo jednom računaru, ali se pri obradi podataka može koristiti procesorska snaga svih računara u mreži. Ovakvim konceptom baze dobija se mogućnost istovremenog korišćenja resursa više računara, pa pojedini računara mogu imati i nešto slabije karakteristike.

## **Distribuirana baza podataka**

Podrazumeva strukturu baze podataka u kojoj su podaci rašireni na više računara koji su mrežno povezani. Jednostavno rečeno, distribuirana baza podrazumeva više lokalnih, međusobno povezanih baza. Pred samim korisnikom je ta "raspršenost" podataka skrivena, te on ima osećaj da pristupa jednoj centralnoj bazi. U današnjim uslovima postoji sve veća potreba za realizacijom distribuiranih baza podataka. Razlozi za to su brojni:

- Mnogi korisnici za koje se rade baze podataka po svojoj prirodi su distribuirani na više lokacija. Uzmimo primer mnogih multinacionalnih kompanija, koje imaju predstavništva širom sveta. Svako predstavništvo u mestu u kojem se nalazi formira svoju lokalnu bazu podataka, a sve te baze zatim se povezuju u distribuiranu bazu podataka, koja objedinjava sve lokalne baze. Rukovodstvo kompanije ima mogućnost pristupa u sve baze i nadzora podataka koji se nalaze u bilo kojoj lokalnoj bazi.
- Distribucijom baze podataka povećava se raspoloživost i pouzdanost sistema. U slučaju otkaza bilo kojeg računara u mreži, podaci na tom mestu postaju nedostupni korisnicima, ali su podaci na svim ostalim mestima sačuvani i dostupni.
- U distribuiranim sistemima se koristi tehnika replikacije istih podataka na više lokacija u mreži. Zbog obrade manjih baza podataka, brzina obrade na pojedinim mestima je veća u odnosu na brzinu koju bi imao sistem koji obrađuje veliku centralizovanu bazu.



Da bi se u potpunosti iskoristile prednosti koje pružaju distribuirani sistemi, osim zadataka koji su zajednički sa zadacima centralizovanih sistema, distribuirani sistemi moraju omogućiti:

- pristup udaljenim računarima u mreži te prenos upita i podataka između računara
- postojanje sistemskog kataloga s podacima o distribuciji podataka u mreži
- održavanje konzistentnosti podataka koji se repliciraju
- izradu strategije za izvođenje pretraživanja i obrada koje dobijaju podatke iz više lokalnih baza.
- oporavak sistema u slučaju otkaza pojedinog računara iz mreže.

# Model podatka

- **Model podatka** služi za prikazivanje objekata sistema, njihovih atributa i njihovih međusobnih veza.
- U zavisnosti od karakteristika tabela, atributa a naročito veza među tabelama razlikujemo 3 osnovna modela projektovanja: ***hijerarhiski, mrežni i relacioni model.***
- Svaki model mora imati mogućnost: definisanja podatka, definisanja pravila za očuvanje podatka, definisanje pravila manipulacije podacima.

# Hijerarhiski model

- Sve veze među tabelama (atributima) moraju biti precizno određene.
- Zahteva stručnost svih korisnika (ne samo administratora)
- Ne kontrolisan gubitak informacija (brisanjem „roditelja“ gube se i tabele „naslednici“)
- Redundanca podataka – potreba za memorisanjem istih podataka na više mesta.

# Mrežni model

- U odnosu na hijerarhiski model omogućena je veza tipa M:N (jedan naslednik može imati više roditelja).



# Relacioni model

- Najpopularniji model, danas najdominantniji.
- Struktura veoma jednostavna, baza predstavlja skup tabela.
- Moguća formalno matematička interpretacija tabela.
- Pošto i relacija(veza) između tabela može da se predstavi kao tabela, u relacionom modelu sve je definisano samo tabelama.
- Relaciona baza podataka se sastoji od vremenski promenljivih tabela (relacija) koje mogu biti bazne i izvedene.
- Bazne tabele se još nazivaju fizičke tabele zato što su zapisane u memoriji.
- Izvedene se izvode iz fizičkih upita.
- U relacionim bazama postoji ne zavisnost aplikacije od podataka tako da se podaci mogu obrađivati pomoću raznih aplikacija što ih čini dostupnijim.
- SUBP – sistemi za upravljanje bazama podataka moraju imati razvijene sisteme za zaštitu podataka.

# Pojam relacije

- Kompletno zasnovan na relacionoj algebri – omogućen razvoj specifičnog softvera i obrada uz garantovanu konzistentnost podataka.
- U relacionim bazama podataka pojam relacija je u stvari sinonim za tabelu uz određena ograničenja.
  - Među atributima tabele(relacije) ne sme postojati funkcionalna zavisnost
  - Unutar jedne relacije ne smeju postojat dve identične n-torke
  - Redosled n-torki u relaciji je ne bitan
  - Svi atributi unutar jedne tabele moraju imati različita imena

- Entitet je deo realnog sveta opisan ograničenim brojem atributa.
- Atribut je jedno od svojstava entiteta.
- Domen je skup vrednosti koje atribut može da poprimi.
- Primarni ključ je atribut (ili više njih kada je to potrebno) čija vrednost jednoznačno određuje samo jednu n-torku u tabeli.  
Dve n-torke ne smeju imati istu vrednos ključa.
- Spoljni ključ je običan atribut u jednoj tabeli koji je u drugoj tabeli primarni ključ. Koristi se za uspostavljanje veze u tabelama.

# Coddova pravila

E.F. Codd je u svojim radovima definisao 12 pravila, od kojih najmanje 6 mora biti ispunjeno da bi se informacioni sistem smatrao relacionim.



- Svaki sistem za upravljanje bazom podataka koji se smatra relacionim mora imati mogućnost upravljanja bazom podataka na relacioni način i relacionim metodama.
- Sve informacije u Relacionim bazama moraju logički biti predstavljene na isti način (vrednostima podataka u tabelama)
- Svaki podatak korisniku mora biti dostupan uz pomoć imena relacije u kojoj se nalazi, vrednosti primarnog ključa i imena atributa.
- Mogućnost prikazivanja nepostojeće informacije (null)
- Dinamički katalog – autorizovanim korisnicima mora biti omogućeno da primenom nekog relacionog jezika menjaju bazu.
- Softverski paketi za obradu (manipulaciju) data bazom mora, precizno definisanom sintaksom i na korisniku blizak način omogućiti: definisanje tabela, definisanje atributa, unošenje podataka, definisanje proizvoljnog „pogleda“ na bazu podataka, manipulaciju podacima, postavljanje ograničenja korisnicima, autorizaciju korisnika, upravljanje proizvoljnim transakcijama.
- Nezavisnost integriteta baze.
  - Integritet entiteta: nijedan atribut koji je primarni ključ ili deo primarnog ključa neke bazne relacije ne može da uzme null vrednost. (primarnog ključa nema da nema!)
  - Referencijalni integritet: skup vrednosti spoljnog ključa relacije R1 mora biti podskup skupa vrednosti primarnog ključa relacije R2 sa kojom se povezuje R1. Relacije R1 i R2 ne moraju biti različite.
- Jezikom treće generacije se ne mogu zaobići pravila integriteta definisana relacionim jezikom.
- Mogućnost kreiranja pogleda i definisanja operacija nad njima.
- Nad pogledima se mogu izvršavati i operacije održavanje baze podataka.
- Aplikacioni programi ostaju ne promenjeni ako se promeni fizička organizacija baze ili fizični metod pristupa.
- Aplikacioni programi ostaju ne promenjeni ako se promene bazne tabele (sem u promeni struktura tabela).
- Sve navedene karakteristike su ne zavisne od distribucije baze podataka.

# PRIMER ENTITETA

Entitet:

- Osoba
- Proizvod
- Institucija
- Proces

# PRIMER ATRIBUTA

- **ATRIBUTI:**

- Ime
- Prezime
- Matični broj
- Godina rođenja
- Pol

- **OGRANIČENJA  
ATRIBUTA**

- Ime ne može biti izostavljeno
- Matični broj mora da ima 13 cifara
- Pol može biti M ili Ž

# PRIMER VEZA

## **VEZE**

- Jedna osoba može da ima samo jedan matični broj
- Jedan menadžer može da upravlja sa više osoba

## **OGRANIČENJA VEZA**

- Menadžer može da upravlja maksimalno sa 15 osoba



# ELEMENTI BAZE PODATAKA

- Baza
- Tabele: Radnici, Odeljenja, Radna mesta

- Tabele

Maticni broj	Ime	Prezime	Pol
120295460789	Dragan	Stojanovic	muski
051196138921	Dragan	Stojanovic	muski

- Zapisi
  - Polje
  - Karakter
- 120295460789 Dragan Stojanović muski
  - Dragan
  - D

# KLJUČ

- Ključ je atribut pomoću kojeg se entiteti međusobno razlikuju
- Svaki entitet mora da ima ključ
- Primer: matični broj

Maticni broj	Ime	Prezime	Pol
120295460789	Dragan	Stojanovic	muski
051196138921	Dragan	Stojanovic	muski

# FUNKCIJA KLJUČA

Primarni ključ ima dve funkcije:

- Da jednoznačno identifikuje svaki zapis u tabeli
- Da uspostavi vezu sa drugim entitetima

# RELACIJE

1:1 IDENTIFIKACIONI BROJ: proizvod

Jedan proizvod ima samo jedan ident broj

Jedan ident broj pripada samo jednom proizvodu

1:N FABRIKA : proizvodi

Jedna fabrika proizvodi više proizvoda

Više proizvoda se proizvodi u jednoj fabrici

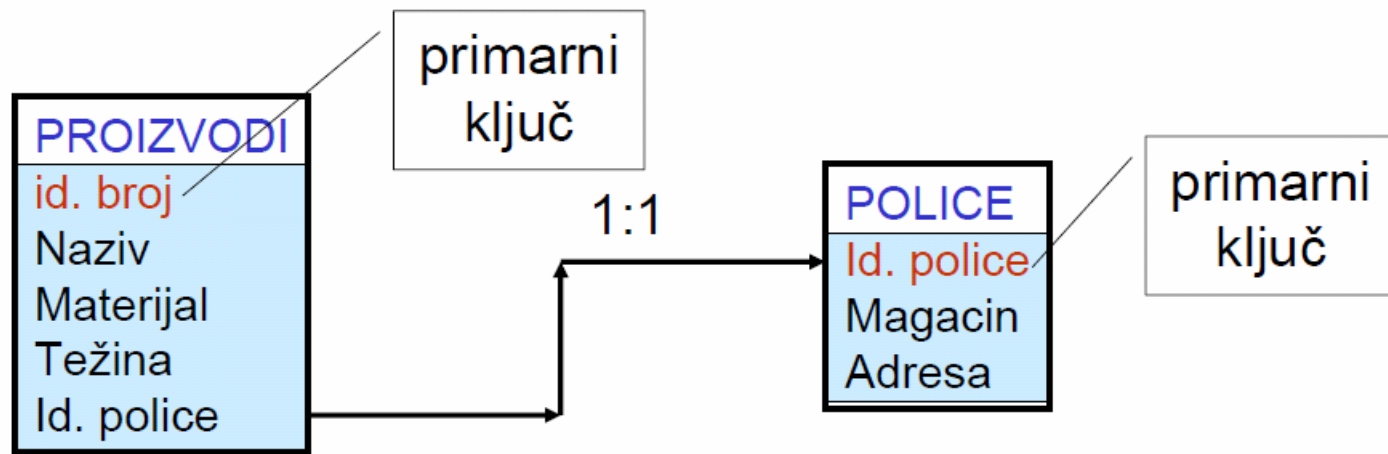
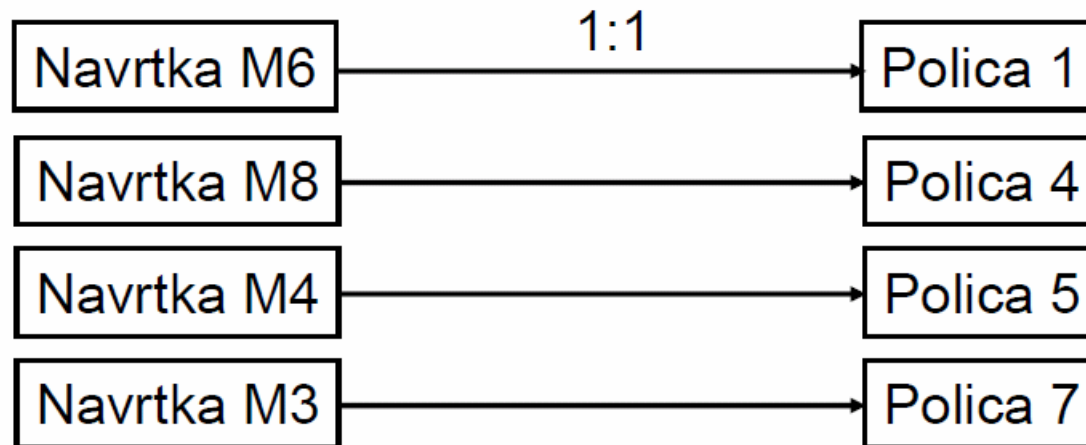
N:M FABRIKE : proizvodi

Više fabrika proizvodi više proizvoda

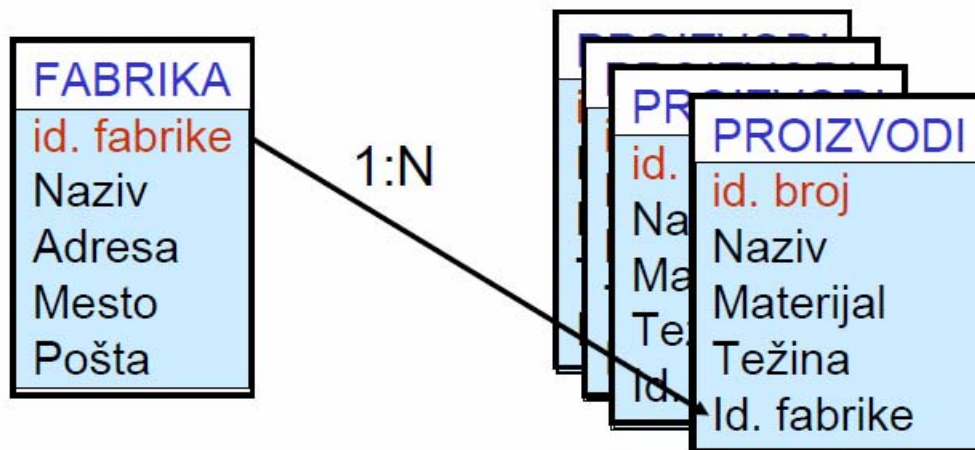
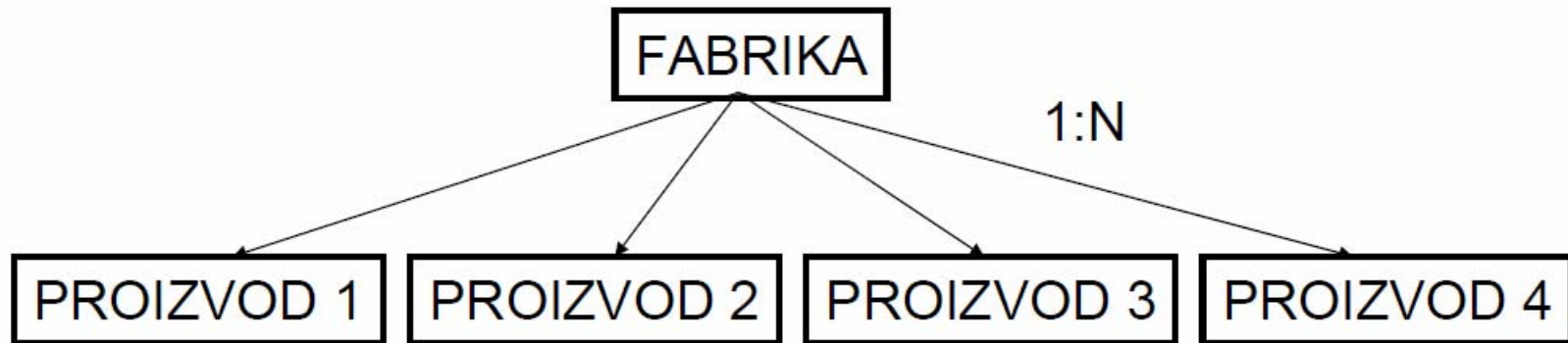
Svaki proizvod se proizvodi u više proizvoda



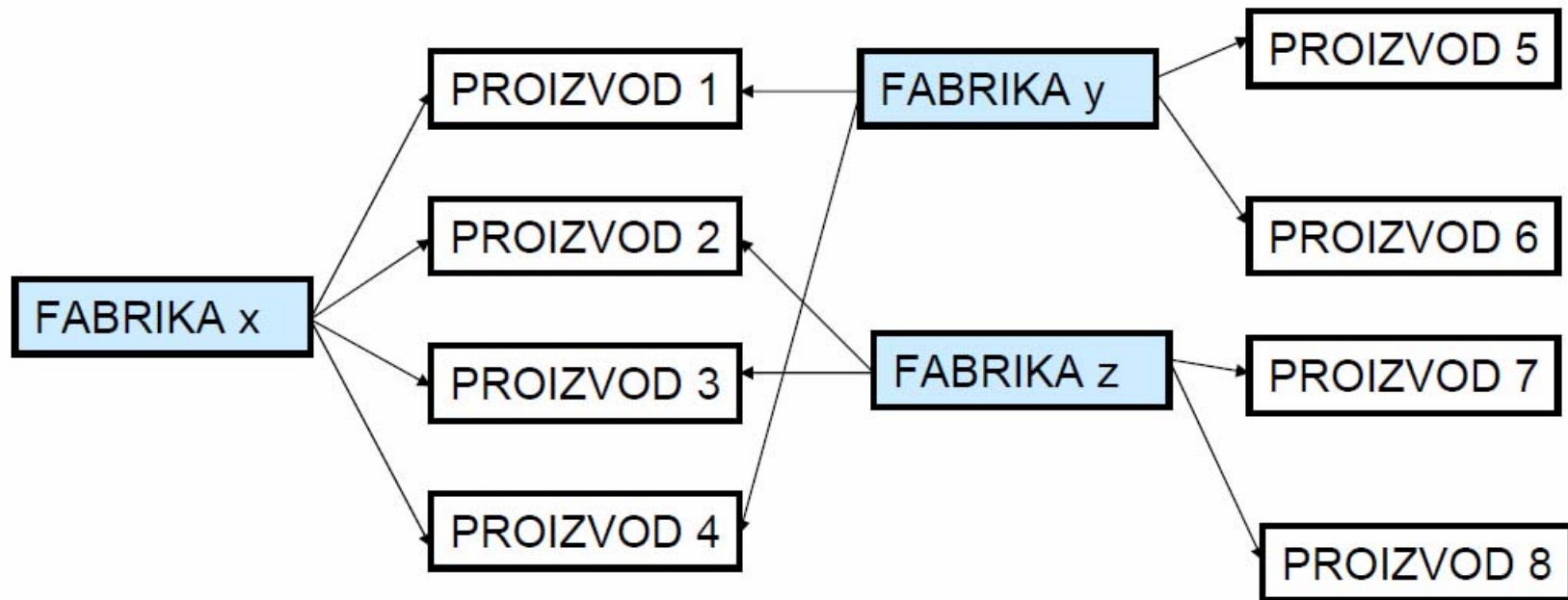
# RELACIJE 1:1



# RELACIJA 1:N



# RELACIJA N:N



- Svaka fabrika proizvodi više proizvoda
- Svaki proizvod se proizvodi u više fabrika