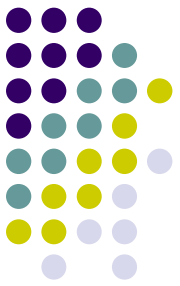


- **Modeliranje podataka  
(informaciono modeliranje)**
- **Alati za projektovanje informacionih sistema  
i SUBP (CASE alati)**
- **AllFusion ERwin Data Modeler**

# Modeliranje podataka (informaciono modeliranje)



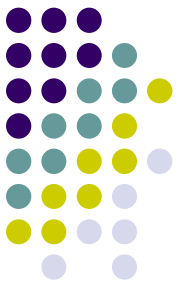
- Izvodi se na osnovu definisanih kritičnih funkcija za prethodno urađeni funkcionalni model (funkcionalnu dekompoziciju) sistema.
- Kritične funkcije predstavljaju, svaka za sebe, po jedan glavni projekat koji se razvija u okviru aktivnosti “Modeliranje podataka”.
- Model podataka prikazuje u kakvom su međusobnom odnosu podaci u nekom realnom sistemu.

# Modeliranje podataka (informaciono modeliranje)



- Informaciono modeliranje omogućuje:
  - definisanje systemske dokumentacije koja se moze koristiti za bazu podataka i za razvoj aplikacija;
  - bolju komunikaciju međusobno i sa krajnjim korisnicima;
  - jasnu sliku o poslovnim pravilima i
  - 'logičku' sliku baze podataka koju mogu koristiti automatski alati za generisanje sistema za upravljanje bazama podataka (DBMS).

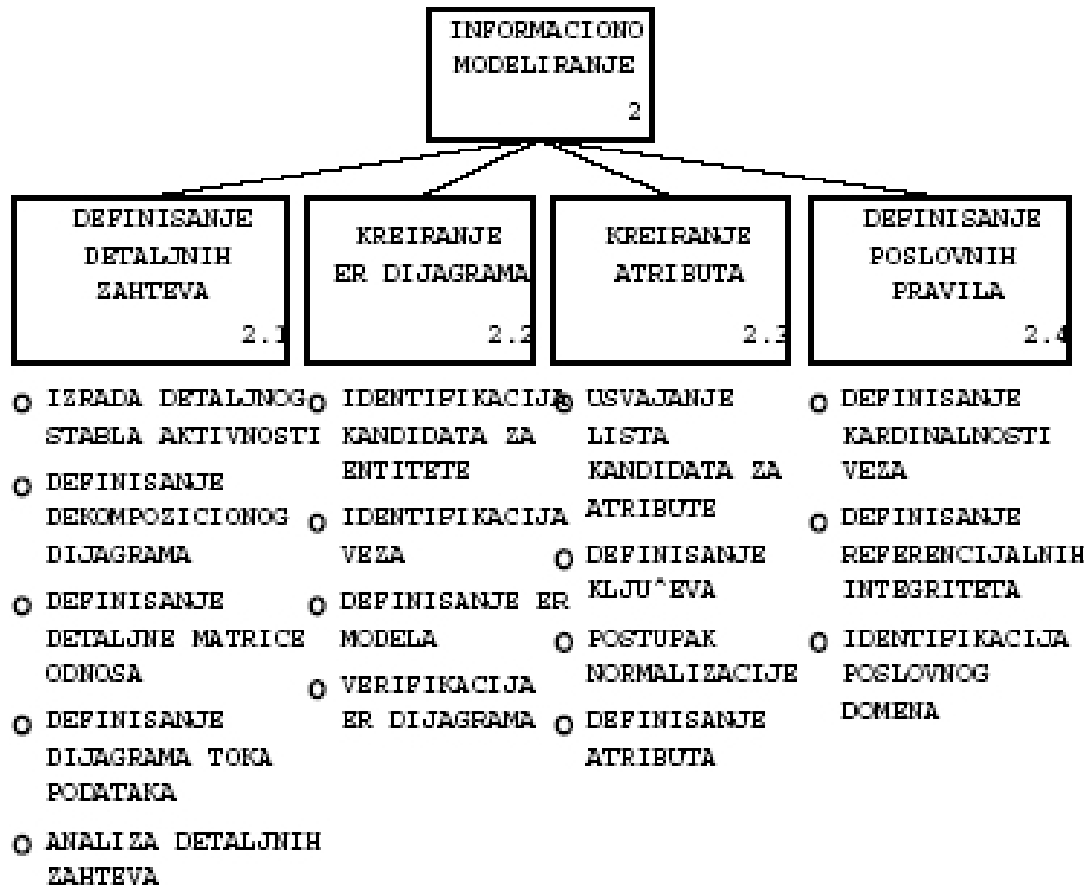
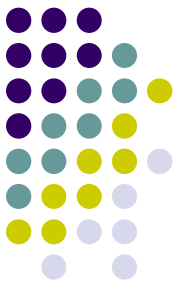
# Modeliranje podataka (informaciono modeliranje)



- Definisiranje detaljnih zahteva
- Kreiranje ER dijagrama (uz identifikaciju entiteta) prema IDEF1x standardu
  - sprovodi se uz pomoć top-down postupka
- Izbor atributa
  - Sprovodi se bottom-up postupkom (analiziranjem dokumentacije)
- Definisiranje poslovnih pravila

# Modeliranje podataka

## Stablo aktivnosti





# Model podataka

- Modeliranje podataka je intelektualno sredstvo pomoću koga se prikazuje kako su podaci u realnom sistemu međusobno povezani.
- Model podataka poseduje tri osnovne komponente:
  - **Strukturu modela**
    - Skup koncepata za opis objekata sistema, njihovih atributa i međusobnih veza
  - **Ograničenja**
    - Semantička ograničenja vrednosti podataka koja u svakom trenutku moraju biti zadovoljena. Ovo su **pravila integriteta modela podataka**.
  - **Operacije**
    - Nad konceptima modela, pod definisanim ograničenjima, preko kojih se može opisati dinamika sistema

# Model podataka



- Svaki model podataka treba da zadovolji dva bitna kriterijuma:
  - Da poseduje koncepte pogodne za modeliranje realnih sistema
  - Da se njegovi koncepti, strukture, ograničenja i operacije mogu jednostavno implementirati na računaru



# Model podataka

- Na osnovu toga kako zadovoljavaju ova dva kriterijuma, modeli podataka se mogu klasifikovati u sledeće “generacije”
  - Prvu generaciju, koju čine konvencionalni programski jezici
  - Drugu generaciju, koju čine baze podataka
  - Treću generaciju, koju čine tzv. semantički bogati modeli podataka (**model objekti-veze**)

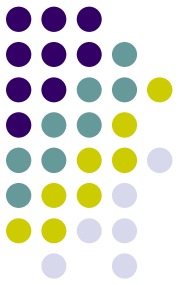




# Model podataka

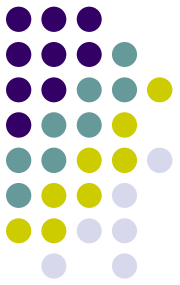
- Imajući u vidu karakteristike generacija modela podataka, zadovoljavajući metodološki pristup projektovanju baza podataka bio bi:
  - Koristeći neki model treće generacije modela podataka formirati semantički bogat model realnog sistema
  - Prevesti dobijeni model u neki od modela prve ili druge generacije

# Model objekti-veze (ukratko)



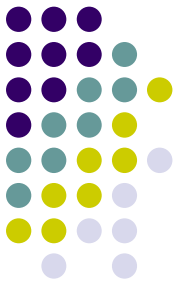
- Struktura modela objekti-veze (entity - relationship)
  - Predstavlja se ER dijagramima (DOV)
  - U modelu objekti-veze sistem se predstavlja kao skup objekata i njihovih veza
  - Objekat (entitet - entity) je koncept ("stvar") koji se po svojim osobinama razlikuje od ostalih objekata.
  - Objekat u modelu može da predstavlja:
    - Fizički objekat realnog sistema (konkretan radnik ili proizvod )
    - Vremenski trenutak ili period (semestar, obračunski period i sl.)
    - koncept (smer studija, klasa daktilografije i sl.)

# Model objekti-veze (ukratko)



- Pojedinačni objekti se klasifikuju u **tipove objekata**
- **Tip objekta** je opšti predstavnik neke klase
- Svaki pojedinačan objekat predstavlja **jedno pojavljivanje (primerak) datog tipa**
- Predstavljanje nekog konkretnog objekta njegovim tipom, naziva se **generalizacija**
- Obrnuti postupak, navođenje pojavljivanja datog tipa naziva se **specijalizacija.**

# Model objekti-veze (ukratko)



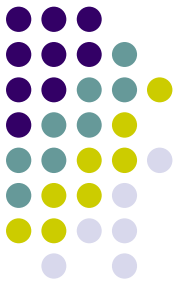
- Atribut i domen
  - Objekti se u sistemu opisuju preko svojih svojstava ili **atributa**.
  - Svaki atribut u u jednom trenutku ima neku vrednost. Atributi uzimaju vrednost iz skupa mogućih vrednosti. Ovaj skup se naziva **domen**.

Radnik

Sifra radnika

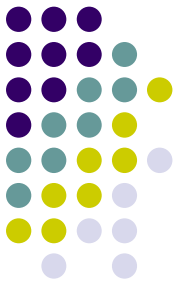
Prezime i ime  
datum rođenja

# Model objekti-veze (ukratko)



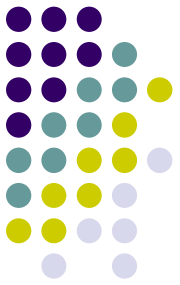
- **Koncept jakog i slabog objekta:**
  - Slab objekat ne može da postoji (egzistencijalno je zavisan) od nadređenog (jakog) objekta
  - Slab objekat ne može da se identifikuje (identifikaciono je zavisan) bez veze sa nadređenim objektom
  - Kardinalnost preslikavanja slab  $\rightarrow$  jak se podrazumeva (1,1)

# Model objekti-veze (ukratko)



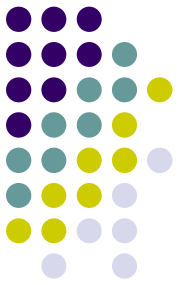
- Veze u modelu opisuju način povezivanja (uzajamnog dejstva) objekata
- Postupak generalizacije i specijalizacije se može primeniti i na veze
- Karakteristika veze je kardinalnost preslikavanja
  - (DG,GG) gde je DG – donja granica, najmanji mogući, a GG – gornja granica, najveći mogući broj pojavljivanja tipa objekta za jedno pojavljivanje povezanog tipa objekta

# Model objekti-veze (ukratko)



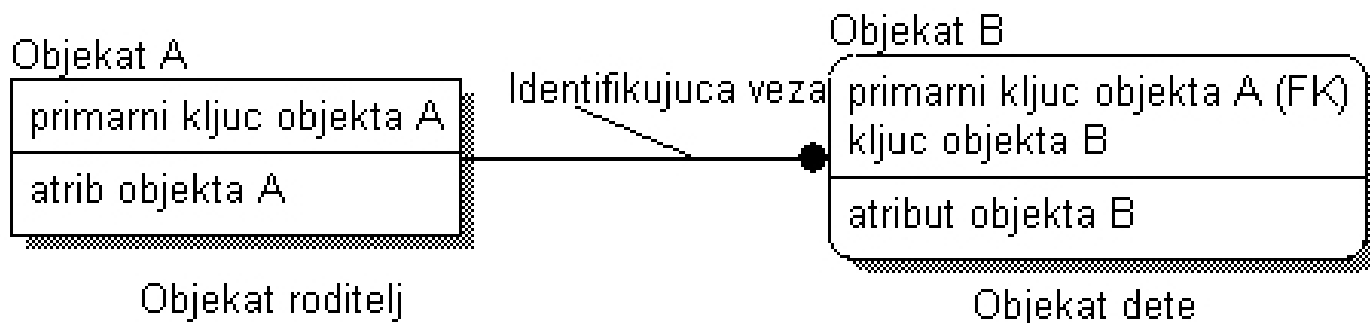
- Objekat od koga je uspostavljena veza se zove “roditelj” (parent), a objekat prema kome je uspostavljena veza se zove “dete” (child)
- Postoje sledeći tipovi veza:
  - Identifikujuće
  - Neidentifikujuće
  - Veza kategorije, tj. veza prema podtipovima
  - Neodređujuća veza – više prema više

# Model objekti-veze (ukratko)



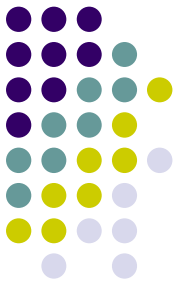
- **Identifikujuća veza**

- Ključevi objekta “roditelj” predstavljaju deo identiteta objekta “dete”, tj. objekat “dete” zavisi od objekta “roditelj” preko identifikatora.
- Svaki primerak objekta “dete” mora biti povezan sa najmanje jednim primerkom objekta “roditelj”.
- Pojavljivanja objekta “roditelj” se definišu nezavisno, a pojavljivanja objekta “dete” se ne mogu identifikovati bez identifikatora objekta “roditelj”



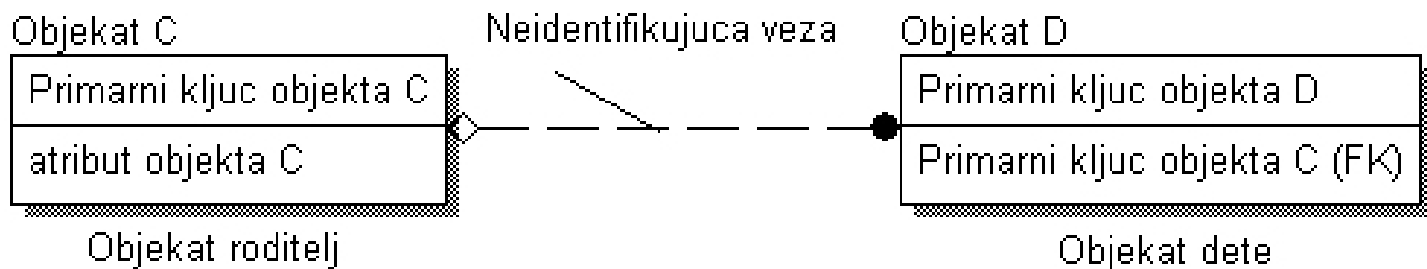


# Model objekti-veze (ukratko)

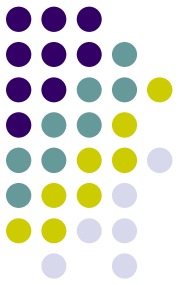


- **Neidentifikujuća veza**

- Ako se svaki primerak objekta “dete” može jedinstveno identifikovati, bez znanja veze sa primerkom objekta “roditelj”

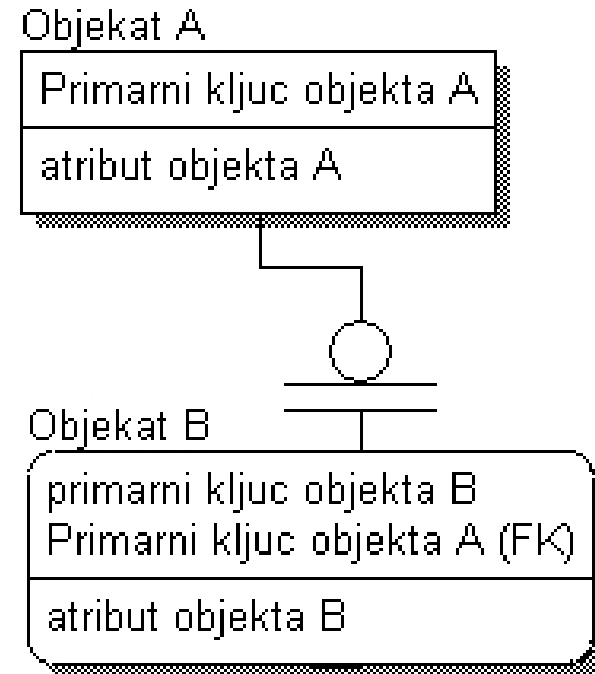


# Model objekti-veze (ukratko)

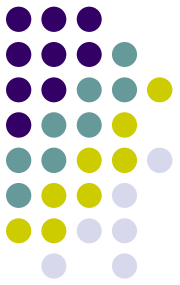


- **Veza kategorije**

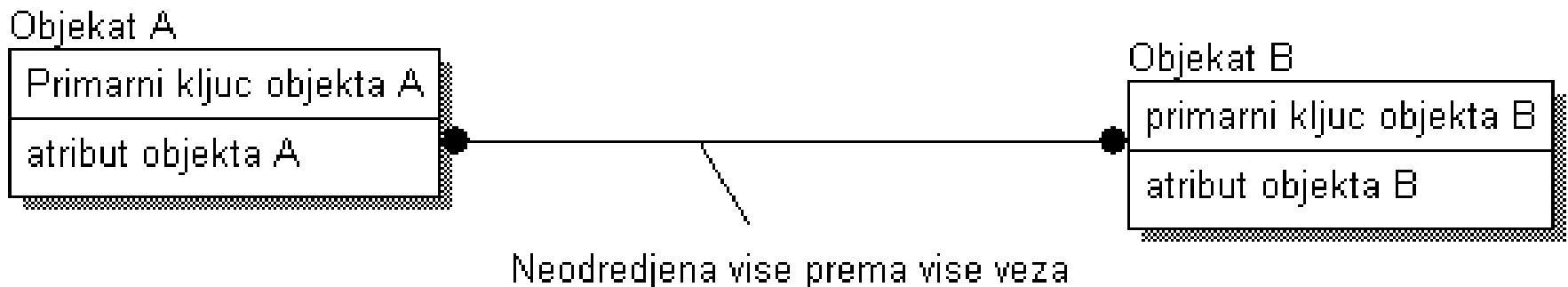
- Definisana je za hijerarhijsku vezu između nadređenog generalizovanog objekta koji sadrži zajedničke osobine i podređenih objekata



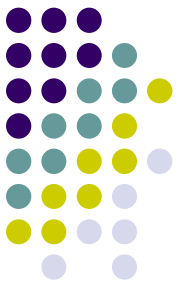
# Model objekti-veze (ukratko)



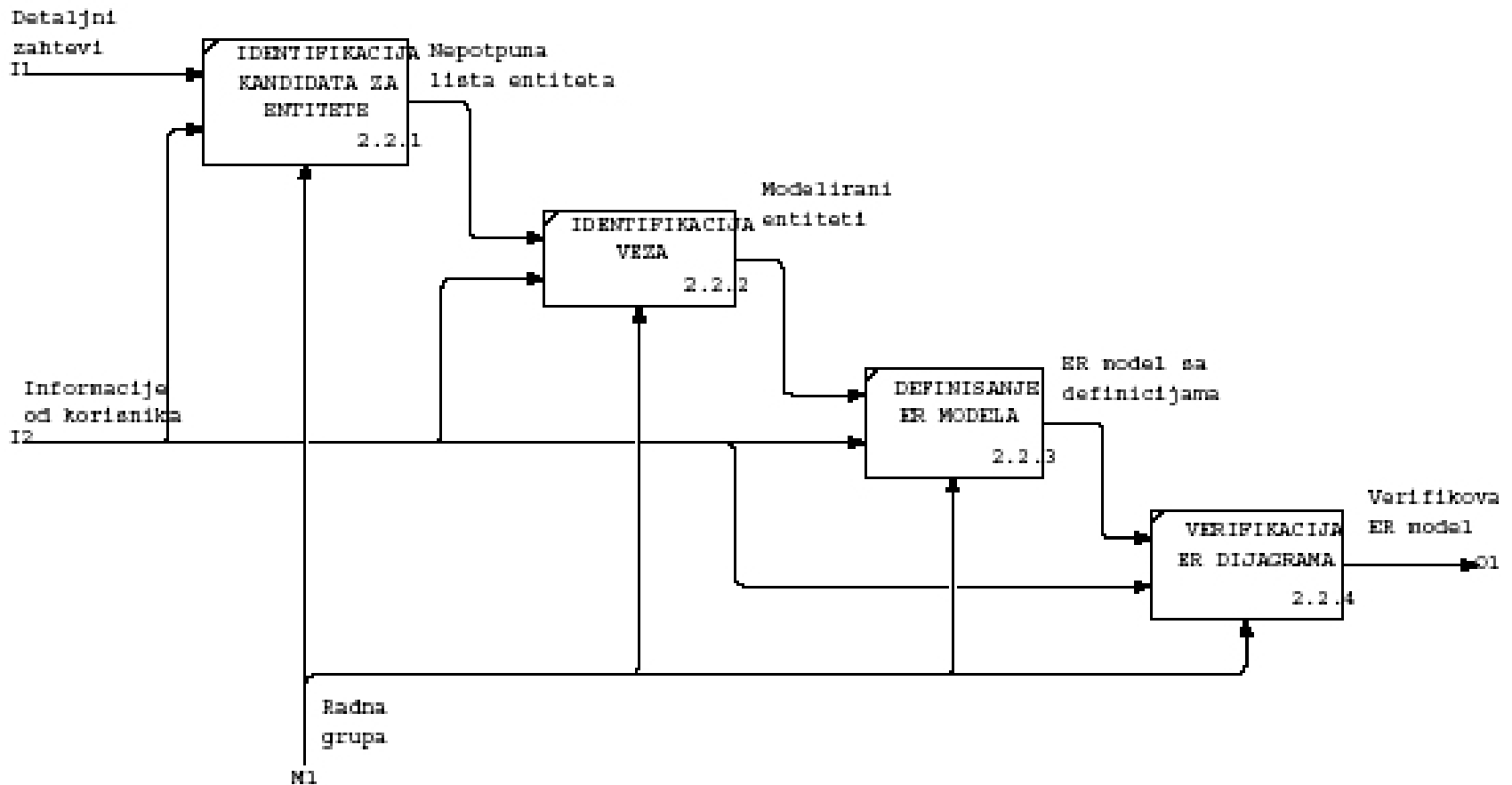
- **Neodređujuća veza tipa više prema više**
  - Primerak jednog objekta povezuje sa nijednim, jednim ili više primeraka drugog objekta, a svaki primerak drugog objekta povezuje sa nijednim, jednim ili više primeraka prvog objekta.



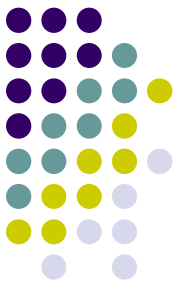
# Model objekti-veze (ukratko)



- Dekompozicioni dijagram za kreiranje ER dijagrama

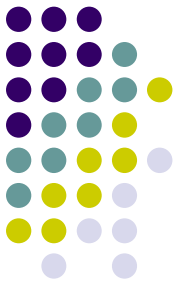


# Alati za projektovanje informacionih sistema i SUBP (CASE alati)



- Razlozi za uvođenje naprednih alata
  - Softverska kriza
    - Niska produktivnost u razvoju informacionih sistema
    - Visoki proizvodni troškovi
    - Zaostajanje softvera u odnosu na hardver
- Rešenje softverske krize je u iskorišćenju osobina inženjera proverenih u praksi i to pre svega:
  - metodičnosti
  - operativne discipline

# Alati za projektovanje informacionih sistema i SUBP (CASE alati)



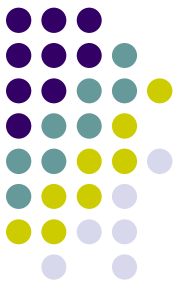
- Kao rezultat nastaje **softverski inženjering** koji u sebi sadrži sistematizovane i koordinirane aktivnosti potrebne pri
  - projektovanju
  - implementaciji
  - eksploataciji
  - održavanju softverskih proizvoda

# Alati za projektovanje informacionih sistema i SUBP (CASE alati)



- Automatizacija softverskog inženjeringa se izvodi posebnim alatom, čiji je naziv **CASE** (**C**omputer **A**ided **S**oftware **E**ngineering)
- Osnovni cilj CASE alata:
  - Da razvijanje softvera postane više inženjerska delatnost, a manje individualna umetnost i umeće.

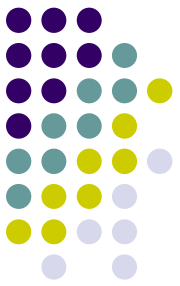
# Alati za projektovanje informacionih sistema i SUBP (CASE alati)



- Uspešnim korišćenjem pravilno izabranog CASE alata može se:
  - Minimizirati vreme i troškovi razvoja softvera
  - Povećati produktivnost u pisanju softvera
  - Podići nivo kvaliteta
  - Povećati pouzdanost
  - Standardizovati proizvedeni softver

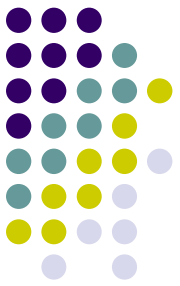


# Alati za projektovanje informacionih sistema i SUBP (CASE alati)



- Elementi CASE alata:
  - **Repozitorijum**
    - Aktivni rečnik podataka koji podržava definisanje različitih tipova objekata i njihovih veza
  - **Alati za grafičko predstavljanje**
    - Omogućavaju razvoj različitih tipova dijagrama na osnovu ranije definisanih standarda
  - **Softver za definisanje teksta**
    - Omogućava definisanje naziva, sadržaja i detalja stavki u repozitorijumu
  - **Analizator softvera**
    - Analizira ulaze u dijagram i repozitorijum i utvrđuje njihovu leksičku kompleksnost i proverava kompatibilnost sa ostalim objektima
  - **Softverski interfejs prema repozitorijumu**
    - Obezbeđuje vezu između dijagrama i repozitorijuma
  - **Korisnički interfejs**
    - Obezbeđuje interaktivnu i off-line obradu

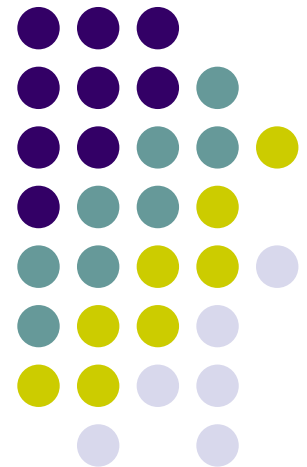
# Alati za projektovanje informacionih sistema i SUBP (CASE alati)

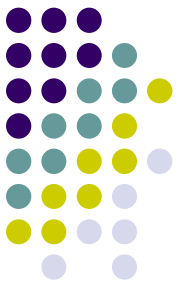


- **Najpoznatiji CASE alati:**
  - ORACLE Designer
  - ERwin
  - DBDesigner
  - Entity Relationship Diagrammer (ERD)
  - MagiCASE
  - EasyCASE System Designer
  - S-Designor
  - Application Development Workbench (ADW)

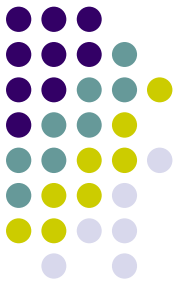


# CASE alat za modeliranje podataka



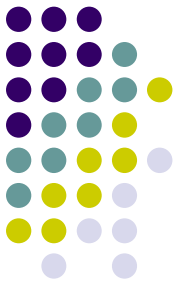


- Izgrađen na konceptima koji su postavljeni IDEF1X (**I**ntegration **D**efinition for information modeling) standardom
- Namenjen za modeliranje podatka **ER** (entity Relationship) metodom
- Mogu se ravnopravno koristiti **IDEF1X** i **IE** (**I**nformation **E**ngineering) notacije



- Modeliranje podataka obuhvata dva aspekta
  - Logičko definisanje modela
  - Fizički dizajn baze

Korišćenje ERwin-a podrazumeva istovremeni razvoj logičkog modela i fizičko definisanje baze



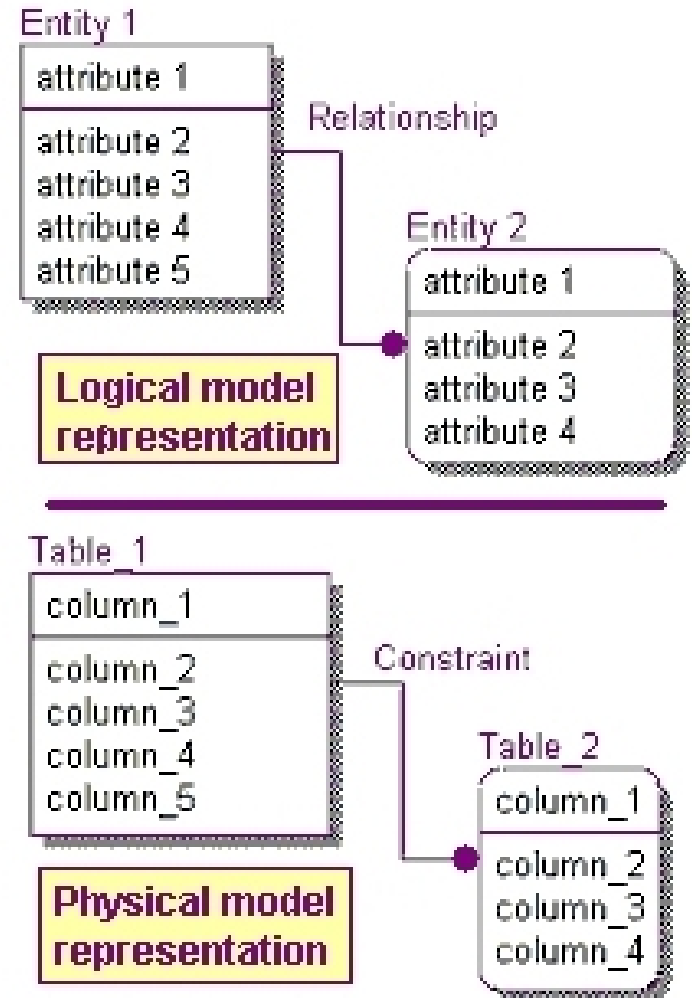
## • Osnovi objekti u modelima

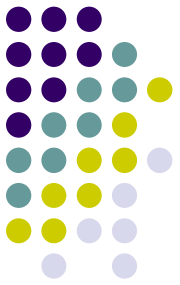
### • Logički model

- Objekti
- Atributi
- Veze (relacije)

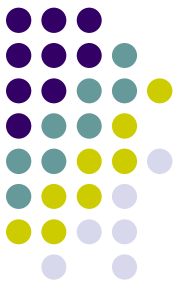
### • Fizički model

- Tabele
- Polja
- Ograničenja (constraints)
- Pogledi (views)



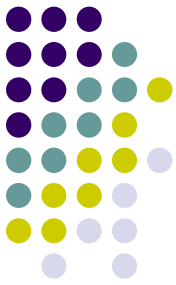


- U bilo kom trenutku definisanja logičkog nivoa baze, potrebno je odabrati **ciljni server** (DBMS na kome će baza biti realizovana)
- Izbor ciljnog servera **nije definitivan**, što znači da se on može menjati
- ERwin insistira na **nezavisnosti modela od DBMS platforme**



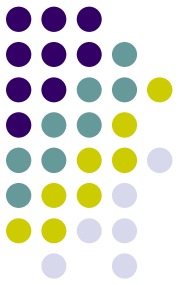
- U Erwin-u postoje **logički i fizički tipovi podataka**
- **Logički tip podataka** (logical datatype) je predefinisani skup karakteristika atributa, koji određuje dužinu polja, pripadnost ERwin domenu i naziv.
- ERwin poseduje listu predefinisanih logičkih tipova koje možemo dodeljivati atributima



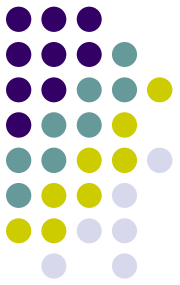


## Predefinisani logički tipovi

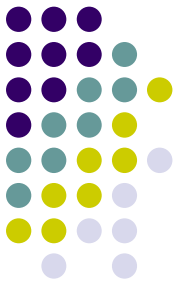
Datatype	ERwin Domain	Length	Precision
• AUDIO	Blob	None	None
• BINARY	Blob	Optional	None
• BOOLEAN	Number	None	None
• CHAR	String	Optional	None
• DATE	Datetime	None	None
• DECIMAL	Number	Optional	Optional
• FLOAT	Number	None	None
• HUGE	Number	None	None
• INTEGER	Number	None	None
• INTERVAL	Number	Optional	None
• LARGE BINARY	Blob	Optional	None
• LONG	Number	None	None
• LONG FLOAT	Number	None	None
• MONEY	Number	Optional	Optional
• NCHAR	String	Optional	None
• NUMBER	Number	Optional	Optional
• NVAR CHAR	String	Mandatory	None
• SHORT FLOAT	Number	None	None
• SMALLINT	Number	None	None
• TEXT	String	Optional	None
• TIME	Datetime	Optional	None
• TIMESTAMP	Datetime	Optional	None
• UNIQUEID	Number	None	None
• VARCHAR	String	Mandatory	None
• VIDEO	Blob	None	None



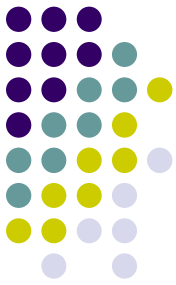
- Mapiranje logičkih u fizičke tipove podataka
  - Logički tipovi podataka su nezavisni od DBMS, ali oni odgovaraju postojećim fizičkim tipovima.
  - Kreiranjem fizičkog modela iz logičkog, ERwin automatski dodeljuje fizički tip podataka za svaki atribut na osnovu odgovarajućeg logičkog tipa



- Mapiranje fizičkih u fizičke tipove podataka
  - Kreiranjem novog fizičkog modela iz postojećeg fizičkog modela ERwin automatski dodeljuje odgovarajući fizički tip podataka za ciljni DBMS.
  - Prilikom prelaska sa jedne verzije DBMS na noviju, ERwin može da promeni tipove podataka za ciljni DBMS.
  - Prilikom kreiranja fizičkog modela putem inverznog inženjerstva, ERwin izdvaja informacije o tipovima podataka sa postojećeg DBMS i automatski dodeljuje odgovarajuće tipove svakom polju novih tabela



- Za svaki DBMS koji podržava, ERwin sadrži odgovarajuću šemu mapiranja logičkih u fizičke tipove, kao i fizičkih u fizičke tipove podataka
- Primer : za migraciju sa MS Access-a na Oracle koristi se sledeća predefinisana šema mapiranja:

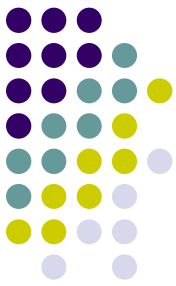


## Access

AutoNumber  
Byte  
Counter  
Currency  
Date/Time  
Double  
Integer  
Long Integer  
Memo  
OLE Object  
Replication ID  
Single  
Text  
Yes/No

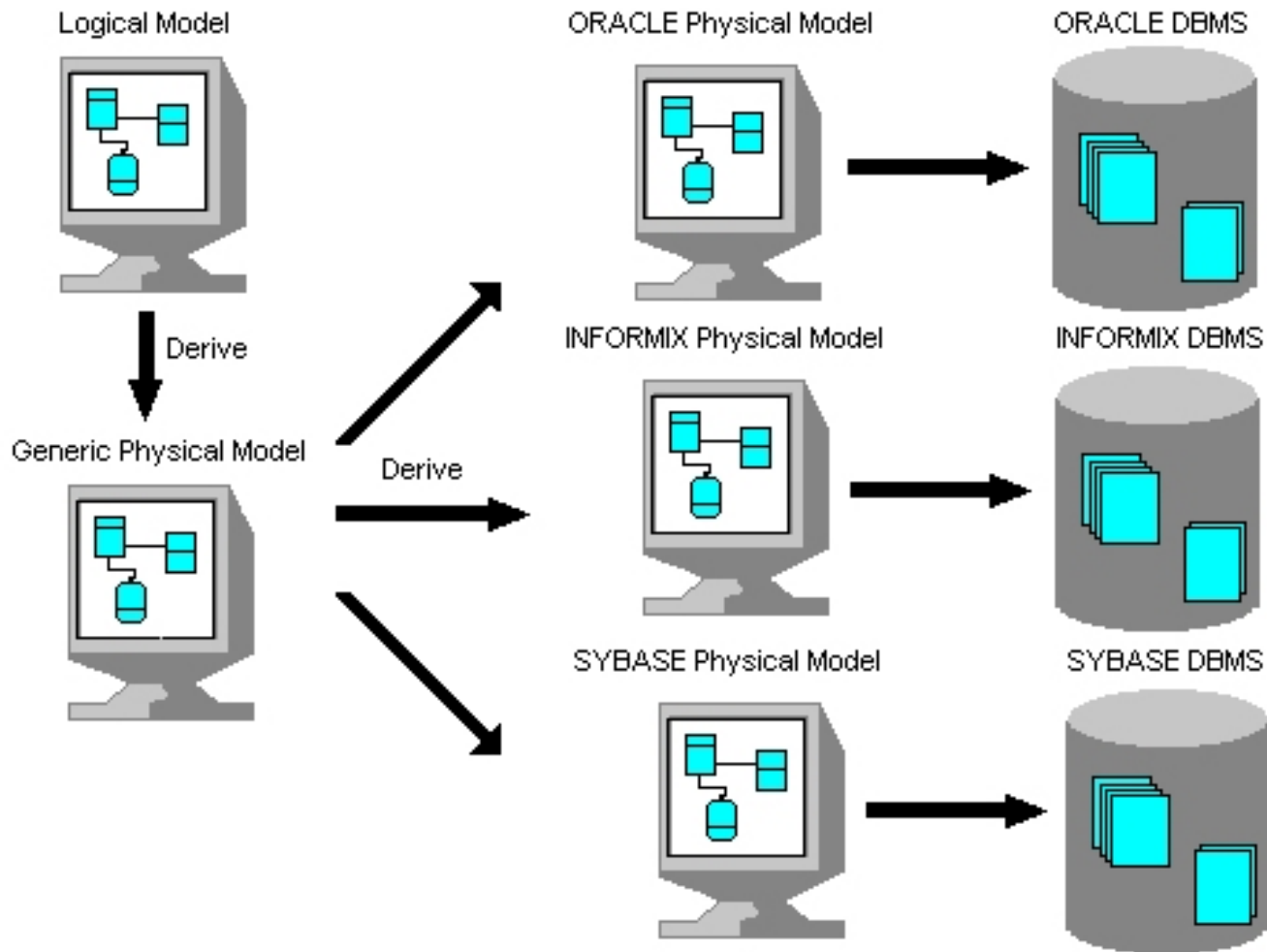
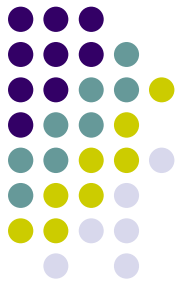
## ORACLE

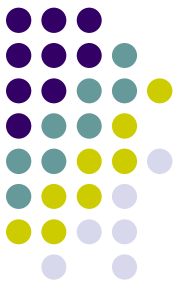
ROWID  
SMALLINT  
NUMBER  
NUMBER  
DATE  
FLOAT  
SMALLINT  
INTEGER  
LONG VARCHAR  
CLOB  
INTEGER  
FLOAT  
VARCHAR2  
SMALLINT



- Na osnovu logičkog modela može da se kreira **generički fizički model** koji sadrži rešenja nezavisna od konkretnog DBMS.
- Na osnovu generičkog modela može se kreirati više konkretnih fizičkih modela za različita DBMS okruženja
- Kada se kreira generički fizički model, odgovarajućim poljima se dodeljuju isti tipovi podataka kao kada se kao ciljni DBMS izabere ODBC (Open Database Connectivity).
- Kada preko ODBC drajvera pristupate bazi podataka, ERwin utvrđuje koji tipovi podataka su podržani i pruža mogućnost zamene bilo kog generičkog tipa podataka tipom podataka koji podržava konkretan DBMS.

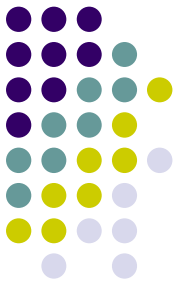
# AllFusion™ ERwin® Data Modeler



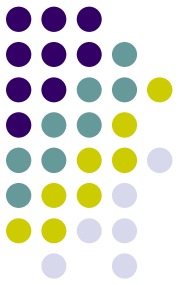


- Server **FRE** (Forward and Reverse Engineering) omogućava “živu” vezu ERwin modela i baze podataka
- **F**orward **E**ngineering
  - Konekcija se ostvaruje preko **ODBC** (Open Database Connectivity) drajvera
  - Na osnovu definisanog fizičkog modela može se kreirati baza na odabranoj RDBMS platformi

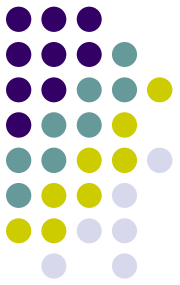




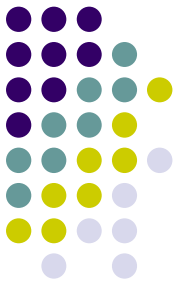
- Posедуje i **modul za inverzni inženjering (RE)**, koji na osnovu postojeće baze podataka kreira ER dijagram po metodologiji IDEFX1
  - To se postiže korišćenjem sistemskih kataloga postojeće baze podataka



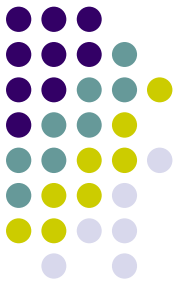
- Oba smeri se svode na:
  - sinhronizaciju modela podataka za bazom podataka i
  - Poređenje modela sa stanjem baze



Isti model podataka se može koristiti za generisanje više baza podataka na različitim DBMS platformama ili za konvertovanje aplikacija sa jedne DBMS platforme na drugu



- **ERwin-ModelMart** predstavlja alat koji omogućava višekorisnički rad na modelima podataka (Workgroup)
  - Vodi se računa o svim aspektima timskog rada:
    - zaključavanju modela
    - verzijama
    - ovlašćenjima korisnika



- Razvijen je set specijalizovanih verzija namenjenih različitim RDBMS ili razvojnim alatima kao što su:
  - Erwin for Oracle
  - Erwin for Sybase
  - Erwin for Visual Basic
  - Erwin for Power Builder
- Cilj je da se pojača sprega sa proizvodom za koji je verzija specijalizovana

# AllFusion™ ERwin® Data Modeler

