

# Načrtovanje podatkovnih baz

# Kako načrtujemo podatkovne baze?

- ▶ Pogovorimo se s končnimi uporabniki in zberemo zahteve.
- ▶ Narišemo konceptualni E-R Model (ali UML model).
- ▶ Pretvorimo E-R model v množico relacij.
- ▶ Izvedemo normalizacijo relacij (način eliminiranja podvajanja podatkov).
- ▶ Implementiramo podatkovno bazo v sistemu RDBMS.

## Grafični prikaz modelov

- ▶ Podatkovno bazo, tabele in sklice lahko načrtujemo s pomočjo grafično podprtih orodij.
- ▶ Vizualizacija nam omogoča preglednejši pogled na tabele v podatkovni bazi.
- ▶ Na voljo je več orodij.
- ▶ Odprtokodno orodje Dia
- ▶ Googlov draw.io
- ▶ OPOZORILO: sklice med tabelami v angleščini imenujemo tudi “relationship”. V slovenščino to po tradiciji malce “nerodno” prevedemo v “relacija” (v relacijski algebri pa velja relacija = tabela)
- ▶ Pazimo na kontekst uporabe pojma “relacija”.

# E-R model

## Definitions:

**entity** something about which data is collected, stored, and maintained

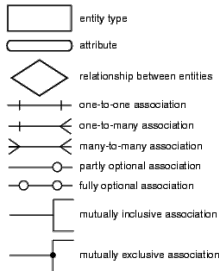
**attribute** a characteristic of an entity

**relationship** an association between entities

**entity type** a class of entities that have the same set of attributes

**record** an ordered set of attribute values that describe an instance of an entity type

## Symbols:



## Examples:

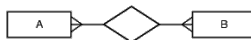
One A is associated with one B:



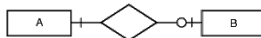
One A is associated with one or more B's:



One or more A's are associated with one or more B's:



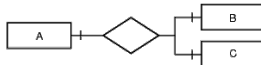
One A is associated with zero or one B:



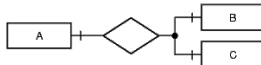
One A is associated with zero or more B's:



One A is associated with one B and one C:

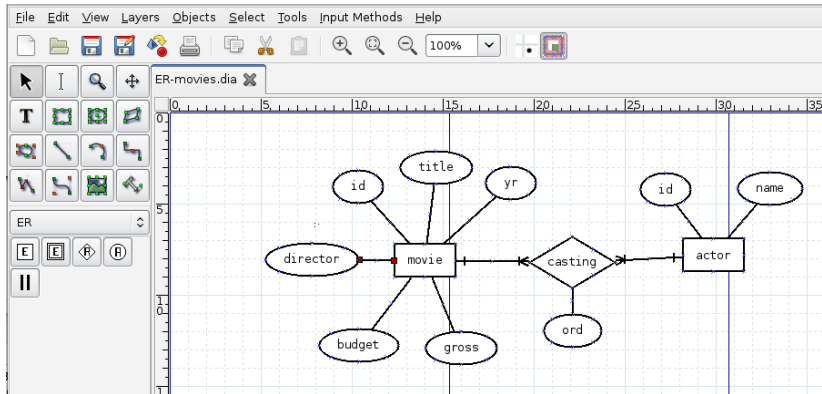


One A is associated with one B or one C (but not both):

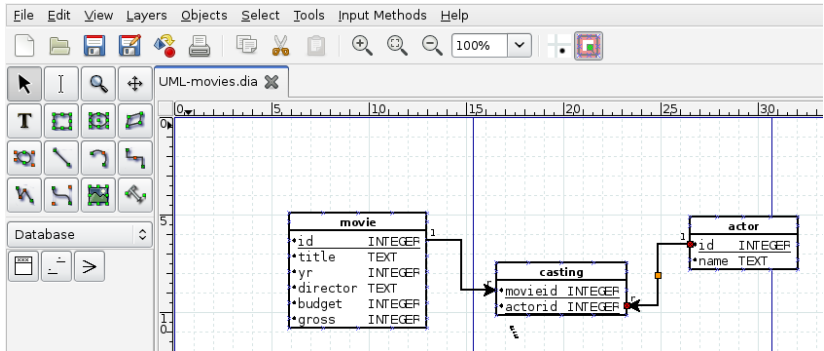


Vir: Oracle

# E-R model in Dia



# UML model in Dia



# Normalizacija

- ▶ *Normalizacija* je proces pri katerem sistematično pregledamo relacije (tabele) in anomalije. Ko identificiramo anomalijo, relacijo razbijemo na dve novi.
- ▶ Med procesom normalizacije ponavadi dobimo še globlji vpogled, kakšna bo interakcija med podatki v podatkovni bazi.
- ▶ Bolje najti probleme v podatkovni bazi v času načrtovanja kot v času operacij.
- ▶ Normalizacija nam pomaga odstraniti redundantnost zapisa podatkov.
- ▶ Ampak zato moramo morda delati več join-ov.
- ▶ Včasih se zaradi učinkovitosti namerno odločimo, da ne izvedemo nekega koraka normalizacije (npr. v podatkovnih skladiščih).

## Predpostavke za relacije (tabele)

- ▶ Vsaka vrstica ima za določen stolpec samo eno vrednost.
- ▶ Podatki v stolpcu so istega tipa.
- ▶ Isto ime stolpca se lahko v relaciji pojavi le enkrat.
- ▶ Vrstni red stolpcev ni važen.
- ▶ Nobeni dve vrstici ne smeta biti enaki.
- ▶ Vrstni red vrstic ni važen.



# Funkcijska odvisnost

- ▶ *Funkcijska odvisnost* opisuje odnos med stolpci znotraj iste relacije (tabele).
- ▶ Stolpec A je funkcijsko odvisen od (enega ali več) stolpcev  $A_1, \dots, A_n$ , če lahko s pomočjo vrednosti teh stolpcev v neki vrstici impliciramo vrednost stolpca A v tej vrstici.
- ▶ Primer: Številka študenta implicira študij študenta.
- ▶ Za nakazovanje funkcijske odvisnosti uporabimo simbol  $\rightarrow$ .
- ▶ Stolpec je lahko funkcijsko odvisen od kombinacije večih stolpcev.
- ▶ Primer: `Solsko_leto`, `Predmet`  $\rightarrow$  `Predavatelj`.
- ▶ Funkcijske odvisnosti so pogojene tako s strukturo tabel kot z naravo podatkov.

# Ključ

- ▶ Ključ: eden ali več stolpcev, ki enolično določajo vrstico.
- ▶ Izbor ključev temelji na konkretni aplikaciji baze. Kaj je ključ izvemo velikokrat iz konteksta in od uporabnikov.
- ▶ Za ključ vedno velja: Ključ -> vsi ostali stolpci.
- ▶ Obstajajo lahko funkcijske odvisnosti, ki na levi strani nimajo (samo) ključev.
- ▶ Kaj z njimi?

# Vrste normalizacij

- ▶ Prva normalizirana oblika (1NF)
- ▶ Druga normalizirana oblika (2NF)
- ▶ Tretja normalizirana oblika (3NF)
- ▶ Boyce-Codd normalizirana (BCNF)
- ▶ Četrta normalizirana oblika (4NF)
- ▶ Peta normalizirana oblika (5NF)
- ▶ Vsaka naslednja oblika vsebuje prejšnjo.

# Normalizacija

- ▶ Normalizacija v ustrezno obliko poteka na naslednji način:
  - ▶ Določimo ključne vsake tabele.
  - ▶ Določimo funkcijske odvisnosti.
  - ▶ Preverimo, ali so kršene zahteve ustrezne definicije.
  - ▶ Če pride do kršitve v neki tabeli, potem to tabelo razdelimo na dve tabeli.
  - ▶ Ponovno preverimo pogoje za izbrano obliko.
  - ▶ Če za nobeno tabelo ni kršena nobena zahteva, zaključimo.

## 1NF = predpostavke za relacijo

- ▶ Vsaka vrstica ima za določen stolpec samo eno vrednost.
- ▶ Podatki v stolpcu so istega tipa.
- ▶ Isto ime stolpca se lahko pojavi v tabeli le enkrat.
- ▶ Vrstni red stolpcev ni važen.
- ▶ Nobeni dve vrstici ne smeta biti enaki.
- ▶ Vrstni red vrstic ni važen.
- ▶ Primer: če se identični vrstici ponovita v neki tabeli, potem tabela že ni v 1NF.

## 2NF

- ▶ Ključ (ang. superkey): katera koli skupina stolpcev, za katere ne obstajata dve vrstici z istima vrednostima v teh stolpcih. Vsi drugi stolpci so funkcijsko odvisni od stolpcev, ki določajo ključ.
- ▶ 1NF - vsi stolpci skupaj določajo nek ključ.
- ▶ Minimalni ključ (ang. candidate key): ključ, za katerega nobena stroga podmnožica ne predstavlja ključa.
- ▶ Primarni ključ: izbrani minimalni ključ.
- ▶ Neključni stolpec: stolpec, ki ni v nobenem minimalnem ključu.

## 2NF

- 1NF + nobena stroga podmnožica kakega minimalnega ključa ne funkcijsko določa kak neključni stolpec.

Employees' Skills

Employee	Skill	Current Work Location
Brown	Light Cleaning	73 Industrial Way
Brown	Typing	73 Industrial Way
Harrison	Light Cleaning	73 Industrial Way
Jones	Shorthand	114 Main Street
Jones	Typing	114 Main Street
Jones	Whittling	114 Main Street

Employees

Employee	Current Work Location
Brown	73 Industrial Way
Harrison	73 Industrial Way
Jones	114 Main Street

Employees' Skills

Employee	Skill
Brown	Light Cleaning
Brown	Typing
Harrison	Light Cleaning
Jones	Shorthand
Jones	Typing
Jones	Whittling

# 3NF

- ▶ Pogoji: relacija je v 2NF in nimamo tranzitivnih funkcijskih odvisnosti.
- ▶ Tranzitivne funkcijske odvisnosti:
  - ▶ iz  $A \rightarrow B$ ,  $B \rightarrow C$  sledi  $A \rightarrow C$ .

Tournament Winners			
Tournament	Year	Winner	Winner Date of Birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

Tournament Winners		
Tournament	Year	Winner
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

Winner Dates of Birth	
Winner	Date of Birth
Chip Masterson	14 March 1977
Al Fredrickson	21 July 1975
Bob Albertson	28 September 1968



# BCNF

- ▶ Boyce-Codd-ova normalizirana oblika.
- ▶ Pogoji: relacija je v 3NF in za vsako funkcijsko odvisnost oblike  $A_1, \dots, A_n \rightarrow B$  velja, da stolpci  $A_1, \dots, A_n$  predstavljajo ključ.
- ▶ Torej: ne obstajajo nobene druge funkcijske odvisnosti razen od ključev.
- ▶ Ponavadi normaliziramo vsaj do te oblike.

# Multifunkcijska odvisnost

- ▶ Multifunkcijsko odvisnost zapišemo kot  $A \rightarrow B$ .
- ▶ Naj  $C$  predstavlja vse stolpce razen stolpcev določenih v  $A$  in  $B$ . Zapis  $(x, y, z)$  predstavlja “bločno” zapisano vrstico po skupinah stolpcev  $A, B, C$ .
- ▶ Če se pojavita vrstici  $(a, b, c)$  in  $(a, d, e)$ , potem morata obstajati vrstici  $(a, b, e)$  in  $(a, d, c)$ .

## 4NF

- ▶ Pogoj: relacija je v BCNF in nima *multifunkcijske odvisnosti*.
- ▶ Primer: vsak predmet (Course) ima predpisan nabor knjig (Book) in nabor predavateljev (Lecturer)

University courses

Course	Book	Lecturer
AHA	Silberschatz	John D
AHA	Nederpelt	John D
AHA	Silberschatz	William M
AHA	Nederpelt	William M
AHA	Silberschatz	Christian G
AHA	Nederpelt	Christian G
OSO	Silberschatz	John D
OSO	Silberschatz	William M

- ▶ Izbira knjig za predmet je neodvisna od izbir predavatelja, zato dodajanje knjige (ali predavatelja), zahteva dodajanje ustreznih kombinacij.
- ▶ Course ->> Book, Course ->> Lecturer

## Ostale oblike

- ▶ Ostalih višjih normaliziranih oblik si ne bomo ogledali.
- ▶ Slike tabel v angleščini so pridobljene iz Wikipedie (CC licenca)