

Sémantika – význam; valid – platný;  
integrita – celistvosť a konzistentnosť – dôslednosť bez protirečení, anomálií;

## 2.1 DATA MODELS, SCHEMAS, AND INSTANCES

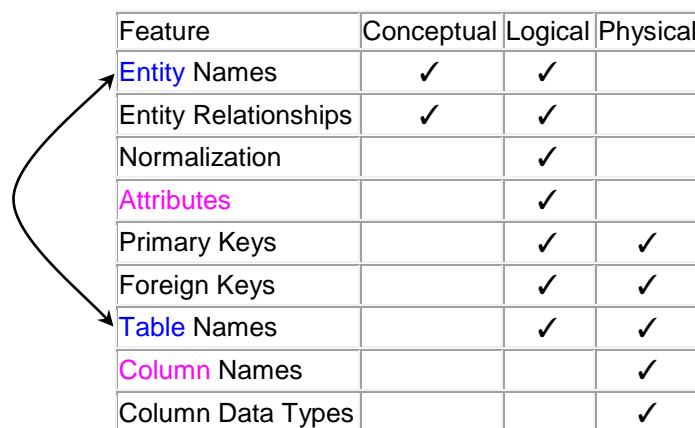
### A data model

- is the description of the **structure** of data, their **types, relationships** and **constraints** by providing the definition and format of data
- provides the necessary instruments to achieve the data **abstraction** - how the symbols and concepts relate to the real world hiding the physical storage structure and parameters
- most data models also include a set of **basic operations** for specifying retrievals and updates on the database.

**Conceptual, logical** and **physical** data models are **phases** in **designing** a successful database application using different levels of abstraction.

Conceptual and logical data models use concepts such as entities, attributes, and relationships:

- an **entity** represents a real-world object or concept, such as an employee or a project, that is described in the database
- an **attribute** represents some property of interest that further describes an entity, such as the employee's name or salary
- a **relationship** among two or more entities represents an association among two or more entities, for example, a works-on relationship between an employee and a project
- a **domain** - the combination of type and constraint that defines the set of possible values of an attribute.



Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Normalization		✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names		✓	✓
Column Names			✓
Column Data Types			✓

- **Conceptual DM** – is a series of assertions about the nature of an organization that describes its **semantics** using entity names and relationships based on requirements and analysis. It (should be) **independent** of any particular data management technology
- **Logical DM** – it enriches the CM with some **domain** information. It includes all attributes for each entity, the primary and foreign **keys**. **Normalization** occurs at this level.
- **Physical DM** – it contains all table structures, including column names, column data types and column constraints. It can be thought as **mapping** the LM to the database model for a given DBMS.

Data model is:

1. representation/description of data – structure	}	DDL
e.g.: relational model – tables		
semi structural model – trees, XML		
2. constraints		
3. operations over data		DML

### 2.1.2 Schemas, instances and database state

In any data model, it is important to distinguish between the **description** of the database and the database **itself**. The description of a database is called the **database schema** that contains **table schemas** (**entities**), which are specified during database design and is not expected to change frequently. A displayed schema is called a **schema diagram**.

The data in the database at a particular moment in time is called a **DB state** or **schema** (respectively **entity**) **instance**.

We mention that

**entity and entity instance**

are known also as

**entity type and entity**.

The DBMS stores the descriptions of the schema constructs and constraints as **meta-data**. The **DBMS** is partly **responsible** for ensuring

that every state of the database is a **valid state** - that is, a state that satisfies the structure and constraints specified in the schema.

## **Relational model and data integrity**

### **Relational model**

The base construct for data representation in relational model is the relation that consists of

- relational schema  $R(f_1:D_1, \dots, f_n:D_n)$ 
  - name of relation  $R$
  - names of attributes  $f_i$
  - domains of attributes  $D_i$

### **Relational schema corresponds to the entity.**

- schema instance
  - table as a set of tuples (**n-tica**)
  - table with columns and rows (in SQL)

**Data integrity** refers to the validity of data, meaning data in DB is consistent and correct. It is one of the most important aspects of the logical design of database. There are three types of integrity:

**entity integrity, domain integrity and referential integrity.**

### **Valid Values**

Only allowed values are permitted in the database. For example, if a column can only have positive integers, a value of '-1' cannot be allowed.

### **Constraints of data integrity**

- are rules about data that correspond to the requirements
- presents the semantic correctness.

It means that constraints of data integrity are conditions that the data defined in relational scheme should fulfil – a database can contain only such data.

3 examples based on 2 tables *Osoba* and *Projekt* that disturb

- the entity/horizontal (inserting row into *Osoba* with existing idO),
- domain/vertical (updating a value of *Cena* in *Projekt*) and
- referential/cross integrity (deleting a needed row from *Osoba*).

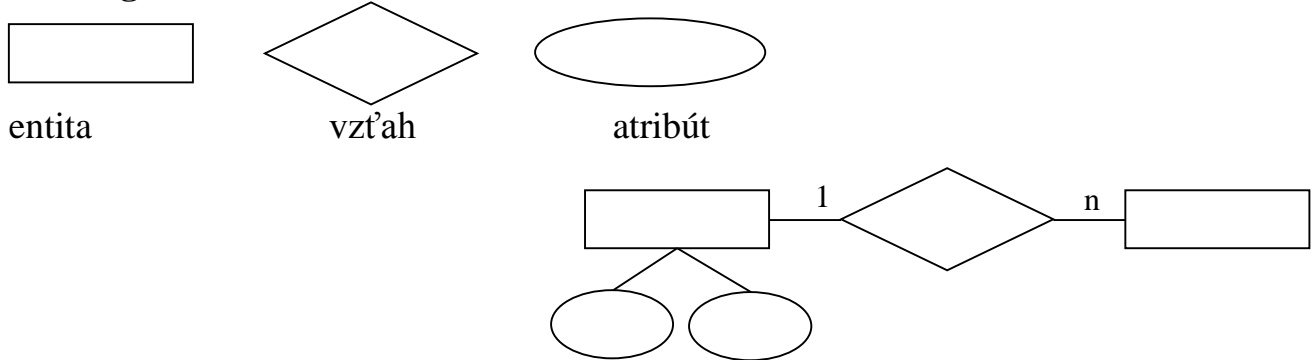
Common ways of enforcing data integrity include different constraints:

- entity integrity (**row**) – the primary key/unique **constraint** ensures that the DB is without data duplicity, there are no two equivalent rows
  - NOT NULL **constraint** - columns identified as NOT NULL may not have a NULL value.
- domain integrity (**column**) – the domain **constraint** ensures that the values in columns are valid
- referential integrity (**relationship**) – the foreign key **constraint** ensures the relationship reliability and validity. The relationship between the primary key of the base table and the foreign key of another table must be maintained – e.g. a primary key that is referred by a foreign key cannot be deleted
  - **cardinality constraint** of entity in a relationship expresses the existence or necessity/absence of one type of entity in relation to the existence of another type, i.e. the admissible **maximal** (1,...,N) count of entities in the relationship.  
Let A, B be two entity types. The three base relationships are:
    - 1:1 - an entity from A is related to **at most** one entity from B
    - 1:N - an entity from A can be related **to more than one** entities from B
    - M:N - several entities A can be related to one B and several entities B can be related to one A
  - **participation constraint** of an entity in a relationship expresses the necessity/existence of one type of entity in relation to the existence of another type, i.e. the possible **minimal** (0, 1) count of entities in the relationship.
    - 0 indicates partial participation (there may be some entities that do not participate in the relationship)
    - 1 indicates total participation (each entity must participate in the relationship)

## DB design and ER (Entity Relation) Diagrams

- získať požiadavky
  - analyzovať úlohu
    - identifikovať entity
    - identifikovať vzťahy a určiť, či sú typu 1:1, 1:n alebo m:n.
    - určiť domény a atribúty pre entity a vzťahy
    - určiť primárne kľúče pre množinu entít
  - načrtnúť ERD (entity-relationship diagram)
  - kontrolovať, či ERD zodpovedá požiadavkám úlohy
- (- transformácia ERD do databázového modelu pomocou DBMS)
- Entita** je *vecný* objekt, osoba, miesto alebo udalosť, ktorú je možné jednoznačne identifikovať v rámci úlohy.
- ER model** interpretuje DB/úlohu ako množinu entít a vzťahy medzi nimi.

### ER diagram



### Pojmy

**Entita** (Typ entity) je množina atribútov, určujúcich objekt.

**Arita** je počet atribútov

**Kľúč** (kľúčový atribút) je súčasťou jedinec. ident.

**Vzťah** - binárny

- rekurzívny
- ternárny
- viacnásobný

**Dva typy obmedzení vzťahu:** kardinalita a účasť/členstvo.

## Dizajn relačných databáz a ER model-diagram

Pri modelovaní relačných DB dôležitú úlohu zohráva početnostná charakterizácia vzťahu, ktorá vyjadruje, že jednému riadku tabuľky koľko partnerských riadkov inej tabuľky zodpovedá. V ER diagramoch **početnosť** môžeme popísať kardinalitou a účasťou. Kardinalita vyjadruje maximálny počet partnerských inštancií a účasť minimálny. **Kardinalita** sa zadáva hodnotami 1, n a účasť hodnotami 0, 1. Preto početnostná charakterizácia vzťahu pri ER modelovaní je hrubá. Pripomíname, že presnejšie vyjadrenie početnosti môžeme dosiahnuť napr. v menej rozšírenom Object Role Modeling pomocou frekvencie a v SQL pomocou CONSTRAINT.

Tri základné typy vzťahov sú založené na kardinalite

- **vzťah 1:1** – hovoríme, že medzi dvomi entitami je vzťah 1:1, ak **inštancie** jednej entity zodpovedá jedna partnerská inštancia druhej entity (v oboch smeroch). Príkladom je vzťah Osoba – Občiansky preukaz: jedna osoba má (maximálne) jeden OP a jeden OP patrí (práve) jednej osobe. Ďalší príklad je napr. Letenka – Sedadlo.
- **vzťah 1:n** - hovoríme, že medzi dvomi entitami je vzťah 1:n, ak inštancie jednej entity zodpovedá viac inštancií druhej entity, ale v druhom smere inštancie druhej entity zodpovedá jedna inštancia prvej entity. Príklad Matka – Dieťa pozri nižšie.
- **vzťah m:n** – v oboch smeroch ide o vzťah 1:n. Príkladom je Učiteľ – Predmet: jeden učiteľ môže učiť viac predmetov, daný predmet môže byť zabezpečený viacerými učiteľmi.

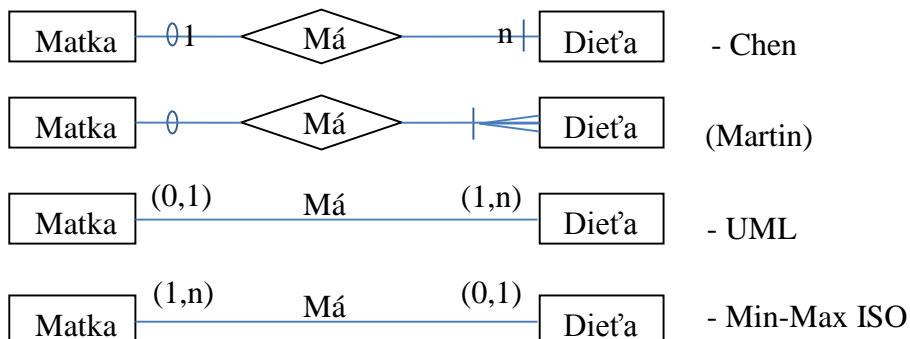
Daný typ môžeme upresniť **účasťou**, ktorá má hodnotu **1** ak účasť je **nutná**. Žiaľ grafické diagramy kardinalitu a účasť neoznačujú jednotne. Vzťah Matka – Dieťa typu 1:n



ilustrujeme dvomi tabuľkami a štyrmi rôznymi ale ekvivalentnými diagramami, ktoré kardinalitu upresňujú pomocou účasti. Všimnime si, že prvá matka má dve deti a matka tretieho dieťa v evidencii je neznáma.

Matka		
idM	Meno	...
1		
2		
...		

Dieťa			
idD	Meno	idM	...
1		1	
2		2	
3		null	
4		1	
...			



Rôzne označenia kardinality a účasti [https://www.wikiwand.com/wiki/Entity-Relationship\\_Modeling#/media/File:ERmodeling-modell.png](https://www.wikiwand.com/wiki/Entity-Relationship_Modeling#/media/File:ERmodeling-modell.png)

## Slabé entity

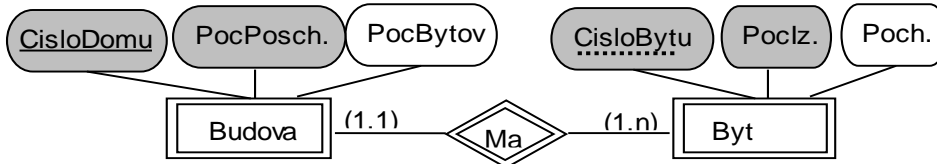
**Slabá entita** nemôže byť jednoznačne identifikovaná vlastnými atribútmi – potrebuje k tomu aj primárny kľúč *rodiča, základnej entity*.

Budova môže mať byty s rovnakými atribútmi a s rovnakými číslami bytu.

Preto jednoznačná identifikácia bytu nie je možné bez toho, aby sme ju nespojili s budovou.

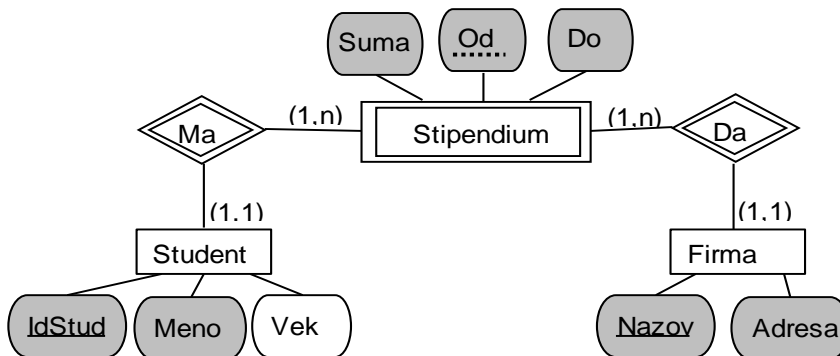
Budova je základná entita s jedinečným identifikátorom *CisloDomu*.

Byt je slabá entita, nemá vlastný jedinečný identifikátor.



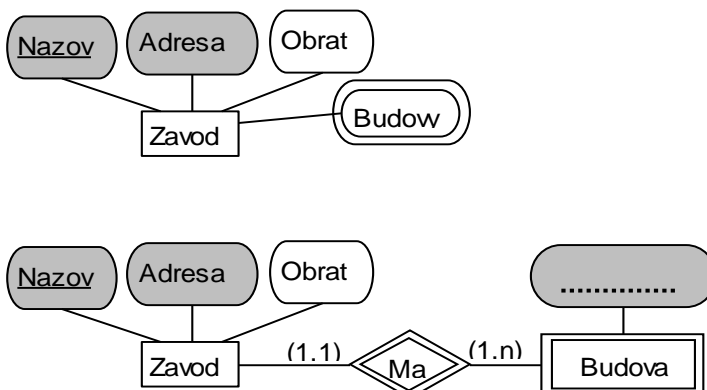
Byty môžu mať rovnaké čísla v rôznych budovách - teda *CisloBytu* neidentifikuje Byt jednoznačne.

Štipendium má dvoch identifikujúcich *rodičov*.



## Rozhodovania pri modelovaní

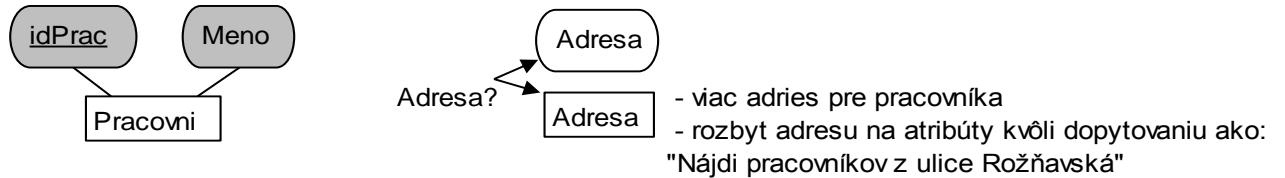
1) Náhrada (resolution) viachodnotových atribútov



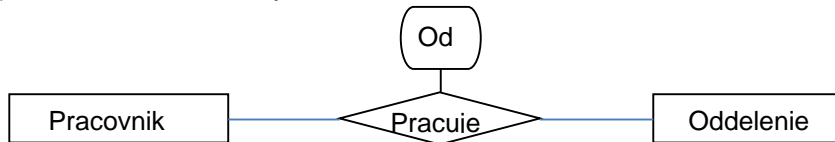
2) Náhrada vzťahu typu m:n



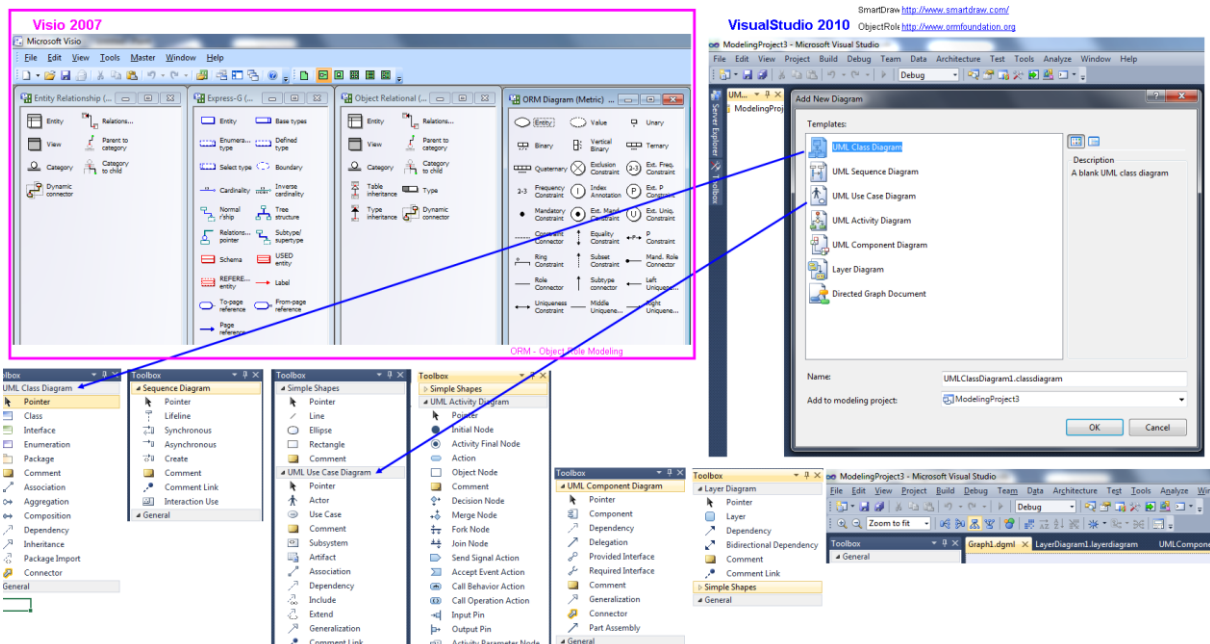
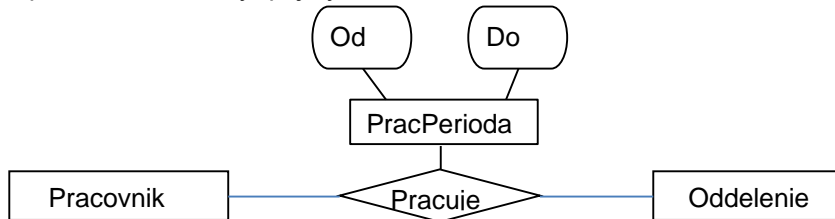
3) Môže sa stať, že sa má rozhodnúť, ako treba vlastnosť modelovať: ako atribút alebo ako entita



4) Atribút vzťahu – v ER je to dovolené



Ak pracovník môže byť prijatý viackrát:



Niektoré systémy na kreslenie diagramov



## Funkčná závislosť a uzávery

- Funkčná závislosť (FZ)
- Kľúč a nadkľúč
- Funkčný a atribútový uzáver

Uvažujme relačnú schému  $R(\mathcal{A})$  resp.  $R(\mathcal{A}, \mathcal{F})$

$\text{NazovTab}(A_1, A_2, \dots, A_n) \equiv \text{NazovTab}(A_1:T_1, A_2:T_2, \dots, A_n:T_n)$

- Relačná schéma = názov relácie, množina atribútov a ich typy (resp. + FZi).

## Funkčná závislosť (FZ)

### Definícia

Nech  $X$  a  $Y$  sú množiny atribútov. Hovoríme že,

$Y$  funkčne závisí od  $X$  ( $X$  funkčne určuje  $Y$ ;  $X$  je determinant)  $X \rightarrow Y$ , ak platí, že dva riadky/tuples, ktoré sa rovnajú vo všetkých atribútoch  $X$  rovnajú sa aj vo všetkých atribútoch  $Y$

$$t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y].$$

Poznámky:

- každej hodnote  $X$  zodpovedá pravé jedna hodnota  $Y$
- funkčná závislosť  $X \rightarrow Y$  je triviálna, ak  $Y$  je podmnožinou  $X$ .

A	B	C
1	A	a
1	B	b
2	A	d
3	D	c
4	A	a
5	D	c

- tabuľka reprezentuje súčasný stav relácie  $R(A,B,C)$ . Ktoré z nasledujúcich FZí platia v  $R$  na základe jej súčasného stavu:  
a)  $A \rightarrow B$ , b)  $B \rightarrow A$ , c)  $A \rightarrow C$ , d)  $C \rightarrow A$  e)  $B \rightarrow C$ , f)  $C \rightarrow B$

Riešenie: platí iba f).

### Vlastnosti

$A \rightarrow B_1B_2\dots B_k \Leftrightarrow A \rightarrow B_i, i=1,2,\dots,k$

Dokážme to v oboch smeroch

rozštiepenie  $\Rightarrow$  spojovanie  $\Leftarrow$

Tranzitívnosť:

$A_1A_2\dots A_n \rightarrow B_1B_2\dots B_k \ \& \ B_1B_2\dots B_k \rightarrow C_1C_2\dots C_m \Rightarrow A_1A_2\dots A_n \rightarrow C_1C_2\dots C_m$

## Kandidátny, nad a primárny kľúč

**Def. 1** Nech  $X$  je množina atribútov.  $X$  je (kandidátny) kľúč pre  $R$ , ak:

- 1)  $X$  funkčne určuje všetky ostatné (neklúčové) atribúty  $R$ .
- 2) Žiadna podmnožina  $X$  neurčuje funkčne ostatné atribúty.

Poznámky:

- z 1) vyplýva, že dva rozdielne riadky  $R$  sa nemôžu rovnať na  $X$ . Skutočne:
- (Kandidátny) kľúč pre reláciu  $R$  je minimálna množina atribútov  $X$ , pre ktorú platí  $X \rightarrow R$

Relácia môže mať viac kandidátnych kľúčov – z nich sa vyberie jeden – primárny kľúč.

**Def. 2** Množina atribútov  $X^*$  je nadkľúč, ak obsahuje kľúč  $X$ .

Poznámky:

- nadkľúč (superkľúč) je ľubovoľná nadmnožina kľúča.
- (Nech je daná relácia  $R(A)$ ). Potom množinu atribútov  $X \subseteq A$  takú, že  $X \rightarrow A$  nazývame nadkľúč.)
- Pre nadkľúč  $X$  platí: relácia nemá dva rozdielne riadky s rovnakými hodnotami pre atribúty z  $X$

**Príklad.** Je  $\{A,C\}$  nadkľúč?

A	B	C	D
1	Z	10	Aa
1	Z	20	Bb
	L	10	Cc
2	L	30	Dd
3	L	10	Cc

- $A \rightarrow B$
- $A, C \rightarrow B$
- $A, C \rightarrow D$

**Príklad.** Návrh kompozitného kľúča.

Relácia Dielo

Autor	NazovDiela	Rok	Vydavatel	PocetStran	Zaner
J.Austin	Sense and Sensibility	1811	T.Egerton	320	Romance
A.C.Clark	Cradle	1988	Warner Books	293	Scifi
G.Lee	Cradle	1988	Warner Books	293	Scifi
A.Hejlsberg	The C# Programming Language	2003	Addison Wesley	672	Software
S.Wiltamuth	The C# Programming Language	2003	Addison Wesley	672	Software
P.Golde	The C# Programming Language	2003	Addison Wesley	672	Software
...		1988			
...	Cradle				

- Autor môže napísať viac\* diel s tým istým názvom (ale nie v tom istom roku!)
- Nové vydanie (dotlač) sa uvažuje za to isté dielo
- (Dielo bez autora – Biblia?)

Uvažujme za determinanta najprv **dvojice**.

**NazovDiela, Rok** jednoznačne určia aj Vydavateľa, aj PocetStran, aj Zaner (spojovanie):

- NazovDiela, Rok  $\rightarrow$  Vydavatel
- NazovDiela, Rok  $\rightarrow$  PocetStran  $\Leftrightarrow$  NazovDiela, Rok  $\rightarrow$  Vydavatel, PocetStran, Zaner
- NazovDiela, Rok  $\rightarrow$  Zaner

Lenže:

- NazovDiela, Rok, **neurčuje** Autor, lebo niektoré diela majú viac autorov

Podobne:

- Rok, Autor **neurčuje** NazovDiela, lebo autor v danom roku môže napísať viac diel
- NazovDiela, Autor **neurčuje** Rok, lebo podľa predpokladu Autor s tým istým názvom môže napísať viac diel (v rôznych rokoch).

Riešením je **trojica**:

**NazovDiela, Rok, Autor**

1) najprv ukážeme, že trojica určuje všetky ostatné. Nech dva riadky na trojici sú rovnaké  $\Rightarrow$  sú rovnaké aj na dvojici Rok, Autor – lenže ona, ako sme videli vyššie, určí Vydavateľa, PocetStran, Zaner. ... a takto stanovujeme, že dva riadky sú rovnaké (vo všetkých atribútoch).

2) treba ešte ukázať, že žiadna podmnožina trojice neurčuje funkčne ostatné atribúty (to sme urobili vyššie).

Existuje aj iný kľúč?

## Pravidlá pre funkčné závislosti

Pomocou nasledujúcich pravidiel môžeme vytvárať nové funkčné závislosti alebo dokázať, že ide o funkčnú závislosť.

Vlastnosti funkčných závislostí

### Armstrongove pravidlá

A1)  $x \subseteq y \Rightarrow y \rightarrow x$  reflexívnosť (triv.závislosť)

A2)  $x \rightarrow y \Rightarrow \forall z: xz \rightarrow yz$  augmentácia

A3)  $x \rightarrow y \ \& \ y \rightarrow z \Rightarrow x \rightarrow z$  tranzitívnosť

Pr. Nech  $\{AB \rightarrow C, CD \rightarrow E\} \Rightarrow ABD \rightarrow E$

Riešenie:

$AB \rightarrow C$  - dané

$ABD \rightarrow CD$  - A2)

$CD \rightarrow E$  - dané

$ABD \rightarrow E$  - A3)

4)  $(x \rightarrow y) \ \& \ (x \rightarrow z) \Rightarrow x \rightarrow yz$

5)  $(x \rightarrow y) \ \& \ (wy \rightarrow z) \Rightarrow wx \rightarrow wz$

6)  $(x \rightarrow y) \ \& \ (z \subseteq y) \Rightarrow x \rightarrow z$  - dekompozícia

**Veta.** Armstrongove pravidlá

- sú **korektné** – funkčné závislosti  $F_i$

odvodené z  $F$  platia pre každú inštanciu  $R$

- sú **úplné** – ľubovoľnú funkčnú závislosť  $F_i$  je možné odvodiť z  $F$  pre každú inštanciu  $R$

- vlastnosti A1, A2, A3 sú **nezávislé** a bez hociktorej z nich úplnosť je narušená.

Dôkaz: na cvičení dokázať 5), 6)

4)

a)  $x \rightarrow y \Rightarrow x \rightarrow xy$  podľa A1), A2)

b)  $x \rightarrow z \Rightarrow xy \rightarrow yz$  podľa A2)

c)  $x \rightarrow yz$  podľa a), b), A3)

**Príklad**  $\Leftrightarrow$  kontra-príklad, že v A2)  $x \rightarrow y \Rightarrow \forall z: xz \rightarrow yz$

z ľavej strany **vyplýva** pravá ale z pravej strany **nevyplýva** ľavá strana.

X	Z	Y
1	A	$\gamma$
1	B	$\alpha$
1	A	$\gamma$
1	B	$\alpha$
2	B	$\alpha$

## Funkčný uzáver $F^+$

Uvažujme reláciu  $R(\mathcal{A}, \mathcal{F})$  a  $F \subseteq \mathcal{F}$ . **Funkčný uzáver**  $F^+$  je množina všetkých FZí, ktoré vyplývajú z  $F$  pomocou Armstrongových pravidiel.

## Atribútový uzáver $A^+ \equiv A^+ | \mathcal{F}$

Uvažujme reláciu  $R(\mathcal{A}, \mathcal{F})$  a množinu atribútov  $A \subseteq \mathcal{A}$ .

**Uzáver**  $A^+ \equiv A^+ | \mathcal{F}$  **atribútu**  $A$  vzhľadom na množinu FZí  $\mathcal{F}$  je množina všetkých atribútov  $X \subseteq \mathcal{A}$ , pre ktoré FZ  $A \rightarrow X$  logicky vyplýva (teda napr. na základe Armstrongových pravidiel) z  $\mathcal{F}$ .

### Algoritmus atribútového uzáveru:

a) Nech je  $X = \{A\}$ .

b) Cyklus:

hľadáme FZ  $B_1, \dots, B_m \rightarrow C$  takú, že  $B_i \in X, i = \overline{1, m}, C \in \mathcal{A}$  ale  $C \notin X$

$\Rightarrow$  dodáme  $C$  do  $X$ , tzn.  $X = X \cup C, X = \{A, C\}$ .

c) Na konci  $A^+ \equiv X$ .

**Príklad 1a).** Nech sú dané  $R(A,B,C,D,E,F)$  a FZi  $\mathcal{F} = \{AB \rightarrow C, BC \rightarrow AD, D \rightarrow E, CF \rightarrow B\}$ . Určme  $\{A,B\}^+$ .

Riešenie.

a) Namiesto  $X = \{AB\}$  môže byť definícia  $X = \{A,B\}$ , lebo z  $AB \rightarrow A$  a z  $AB \rightarrow B$ .

b) Pretože ľavá strana  $AB \rightarrow C$  je v  $X$ , preto pravú stranu dodáme do  $X$ ,  $X = \{A,B,C\}$ .

Pretože ľavá strana  $BC \rightarrow AD \rightarrow D$  je v  $X$ , preto časť pravej strany dodáme do  $X$ ,  $X = \{A,B,C,D\}$ .

$D \rightarrow E \Rightarrow X = \{A,B,C,D,E\}$ .

c)  $CF \rightarrow B$  nic.  $\Rightarrow \{A,B\}^+ = \{A,B,C,D,E\}$ .

Všeobecný **princíp**, pomocou ktorého môžeme stanoviť, či z  $FZ_1$  vyplýva  $FZ_2$ :

Uvažujme reláciu  $R(\mathcal{A}, \mathcal{F})$ ,  $F_1 \subseteq \mathcal{F}$  a  $F_2 : A \rightarrow B$ , kde  $A \in \mathcal{A}$ .

Ak  $B \in A^+$ , potom z  $F_1$  vyplýva  $F_2$ .

Teda

$B \in A^+$ $\Rightarrow A \rightarrow B$
---

Platí aj silnejšia ekvivalencia:  $B \in A^+ \Leftrightarrow F_1$  vyplýva  $F_2$ .

$\Leftrightarrow$

**Lema.**  $x \rightarrow y$  sa dá odvodiť z  $F$  pomocou Armstrongových axióm **práve vtedy** keď  $y \subseteq x^+$  vzhľadom na  $F$ .

Pre každý atribút  $a \in y \subseteq x^+$ . Platí  $x \rightarrow a$  podľa definície uzáveru  $x^+$ . Podľa vlastnosti 4) platí aj  $x \rightarrow y$ .

Naopak nech  $x \rightarrow y$  sa dá odvodiť. Potom pre každé  $a \in y$  platí  $x \rightarrow a$  podľa 6) a  $a \in x^+$ .

**Príklad 1b).** Pokračovanie predchádzajúceho príkladu. Vyplývajú z  $\mathcal{F}$  FZi:  $D \rightarrow A$  a  $AB \rightarrow D$ ?

Riešenie.

$AB \rightarrow D$  vyplýva, lebo  $D$  je z  $\{A,B\}^+ = \{A,B,C,D,E\}$ .

$D \rightarrow A$  nevyplýva, lebo  $A$  nie je z  $\{D\}^+ = \{D,E\}$ .