

Konceptuální modelování (1)

Úvod

- **databáze** = logicky organizovaná kolekce souvisejících dat; data + schéma + integr. omezení...
- **database management system (DBMS)** = SW zajišťující přístup k databázi (bezpečnost, spolehlivost, integrita dat,...)
- **database system** = informační systém (DB, DBMS, HW, lidé, procesy,...)

Proč DB?

- sdílení dat a znovupoužitelnost
- jednotné rozhraní a jazyky (pro definici a manipulaci s daty)
- informační bezpečnost (autentizace, autorizace)
- administrace a údržba (replikace, zálohy, migrace,...)

Proces návrhu DB:

- porozumění a modelování reality
- organizace informací
- zvážení identifikovaných požadavků
- vytvoření příslušného DB schématu

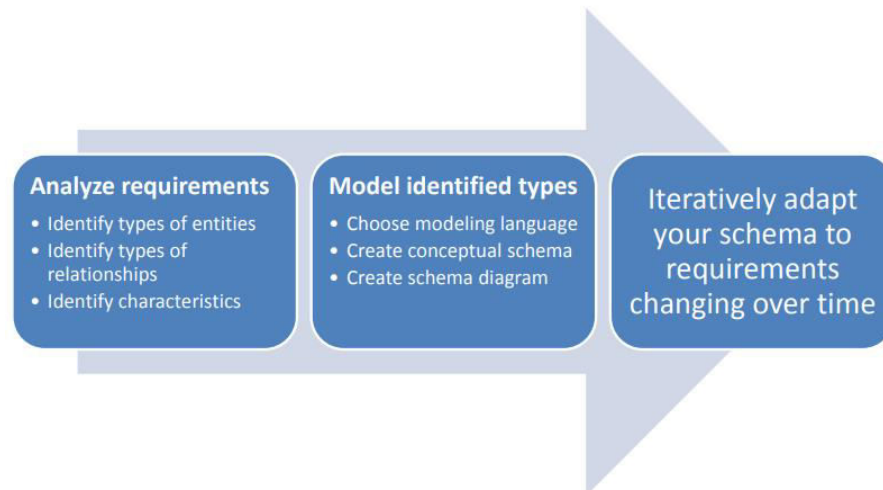
Vrstvy modelování DB:

- **konceptuální**
 - o modeluje část reality (problem domain) relevantní pro DB aplikaci
 - o identifikace a popis reálných entit a vztahů mezi nimi
 - o např. ER či UML diagramy
- **logická**
 - o specifikuje, jak jsou konceptuální komponenty reprezentovány v DB struktuře
 - o např. relační model, objektově-relační, graf,...
- **fyzická**
 - o specifikuje, jak jsou logické DB struktury implementovány v určitém techn. prostředí
 - o datové soubory, indexové struktury (B+ stromy,...) atd.

Konceptuální modelování DB

- proces vytváření konceptuálního schématu dané problémové domény
 - o ve zvoleném modelovacím jazyce a na základě daných požadavků
- často se potřebuje více konceptuálních schémat (různé pohledy)
- **model** = modelovací jazyk
 - o množina konstruktů, které můžeme použít k vyjádření něčeho
 - o UML model = {třídy, atributy, asociace}
 - o relační model = {relační schéma, atributy}

- **schéma** = vyjádření v modelovacím jazyce
 - o instance modelu
 - o relační schéma = {Person(name,email)}
- **diagram** = vizualizace schématu



Krok 1: Analýza požadavků

- začneme s požadavky různých stakeholderů
- identifikujeme důležité typy real-world entit, jejich charakteristiky, typy vztahů mezi nimi a jejich charakteristiky
- vypořádáme se s nejednoznačnostmi

• Example

Our environment consists of persons which may have other persons as their colleagues. A person can also be a member of several research teams. And, they can work on various research projects. A team consists of persons which mutually cooperate. Each team has a leader who must be an academic professor (assistant, associate or full). A team acts as an individual entity which can cooperate with other teams. Usually, it is formally part of an official institution, e.g., a university department. A project consists of persons working on a project but only as research team members.

• Identified entity types

- **Person**
- **Team**
- **Project**
- **Professor**
 - Assistant Professor
 - Associate Professor
 - Full Professor
- **Institution**
- **Department**

- Example

Our environment consists of **persons** which may have other **persons** as their **colleagues**. A **person** can also be a member of several research **teams**. And, they (**person**) can work on various research **projects**. A **team** consists of **persons** which mutually cooperate. Each **team** has a leader who must be an academic **professor** (**assistant**, **associate** or **full**). A **team** acts as an individual entity which can cooperate with other **teams**. Usually, it (**team**) is formally part of an official **institution**, e.g., a university **department**. A **project** consists of **persons** working on a project but only as research **team members**.



- Relationship types

- Person is colleague of Person
- Person is member of Team
- Person works on Project
- Team consists of Person
- Team has leader Professor
- Team cooperates with Team
- Team is part of Institution
- Project consists of Person who is a member of Team

- Example

Each person has a name and is identified by a personal number. A person can be called to their phone numbers. We need to know at least one phone number. We also need to send them emails.

- Person characteristics

- **Personal number**
- **Name**
- One or more **phone numbers**
- **Email**

- Example

We need to know when a person became a member of a project and when they finished their membership.

- Identified membership characteristics

- **From**
- **To**

Krok 2: Tvorba schématu

- namodelujeme identifikované typy a charakteristiky s pomocí vhodného konceptuálního datového modelu, vizualizujeme to pomocí diagramu

Krok 2.1: Výběr modelovacího jazyka

- např. UML (Unified Modeling Language) či ER (Entity-Relationship model)

Krok 2.2: Tvorba konceptuálního schématu

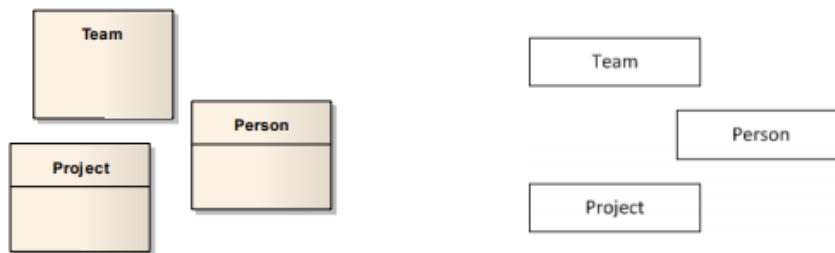
- vyjádříme identifikované typy entit, vztahy a jejich charakteristiky pomocí konstruktů, které nám nabízí daný modelovací jazyk
- UML: třídy, asociace, atributy
- ER: entitní typy, vztahové typy, atributy

ER, UML

- ER není standardizovaný, má různé notace a rozšíření (např. ISA hierarchie)
 - o více zaměřen na data design
- UML je z rodiny modelů jako class diagramy, use-case, stavové diagramy,...
 - o je standardizovaný
 - o více zaměřen na code design

Type of real-world entities

Persons, research teams and research projects.



UML	ER
Class Name	Entity type Name

Attributes of a type of real-world entities

A person is characterized by their personal number, name, optional email address and one or more phone numbers.

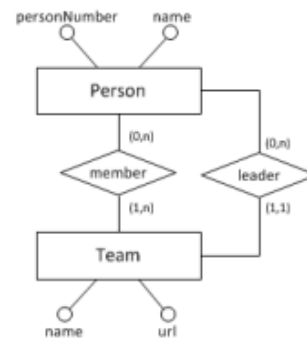
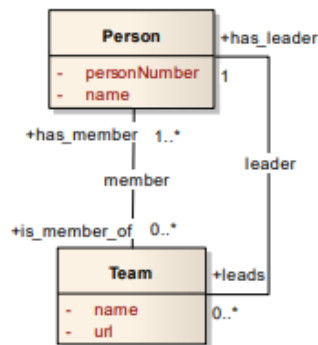


UML	ER
Attribute of a class Name and cardinality	Attribute of an entity type Name and cardinality

Type of a relationship between two real-world entities

A team has one or more members, a person can be a member of zero or more teams.

A team has exactly one leader, a person can be a leader of zero or more teams.



UML

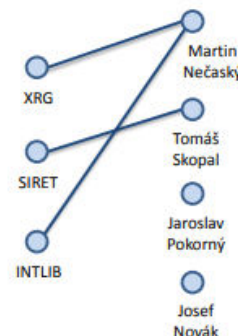
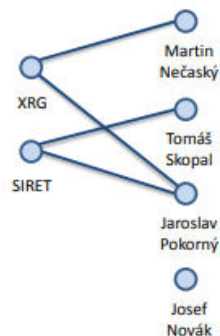
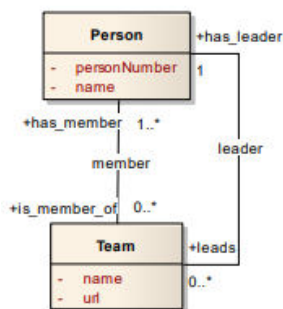
ER

Binary association: name and two participants with names and cardinalities

Binary relationship type: name and two participants with cardinalities

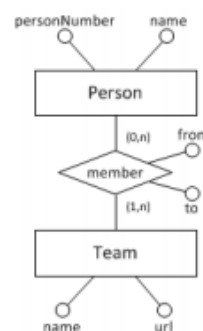
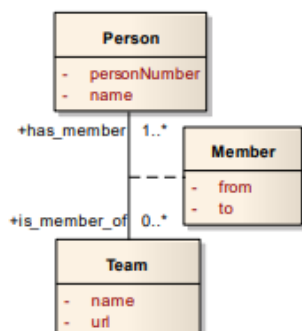
Relationship
member

Relationship
leader



Attributes of a type of relationship between real-world entities

A person is a team member within a given time interval



UML

ER

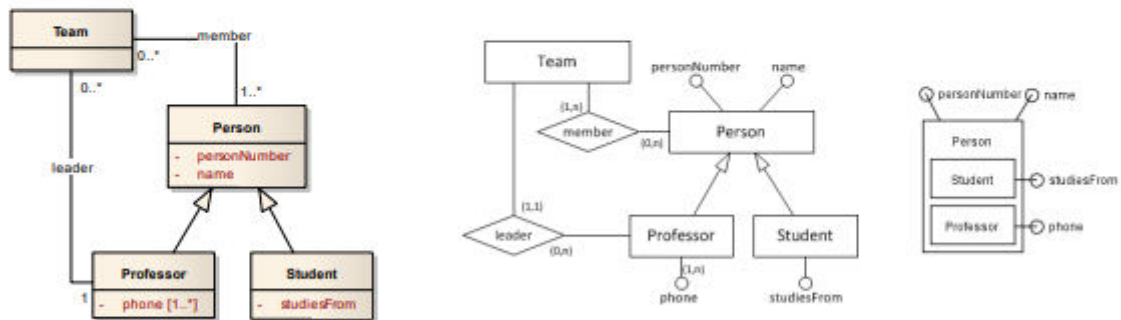
Attribute of a binary association class
Name and cardinality

Attribute of a relationship type
Name and cardinality

Generalization / Specialization

Type of entities which is a specialization of another type

Each person has a personal number and name. A professor is a person which also has one or more phones and can lead teams. A student is a person which also has a date of study beginning.



UML	ER
-----	----

Generalization: specific association with no name, roles and cardinalities

ISA hierarchy: specific relationship with no name and cardinalities

- entitní typy mohou být zdrojem vícera hierarchií
- každý entitní typ může mít nejvýše jednu generalizaci
- další omezení:
 - o **covering constraint (pokrytí)** – complete/partial
 - každá entita musí být alespoň jednoho určitého typu
 - např. každá Osoba musí být Profesor či Student (nebo oboje)
 - o **disjointness constraint (překrytí)** – exclusive/overlapping
 - každá entita musí být nejvýše jednoho určitého typu
 - např. neexistuje Student, který by byl zároveň Profesorem

Composite Attributes

Structured characteristics of real-world entity types

A person has one or two addresses comprising of a street, city and country.



UML	ER
-----	----

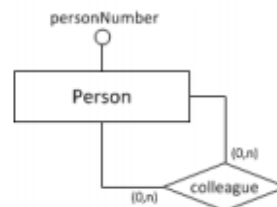
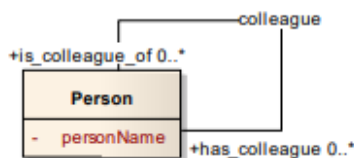
No specific construct
Auxiliary class

Composite attribute: name, cardinality and sub-attributes

Recursive Relationships

Type of a relationship between entities of the same type

A person has zero or more colleagues.

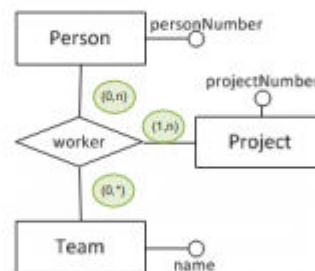
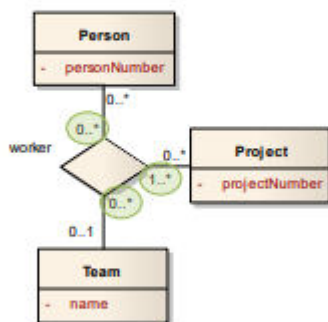


UML	ER
Normal association with the same participants	Normal relationship type with the same participants

N-ary Relationships

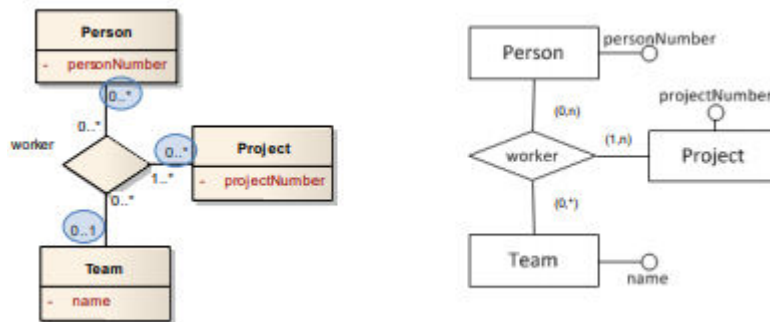
Type of a relationship between more than just two entities

A person works on a project but only as a team member.

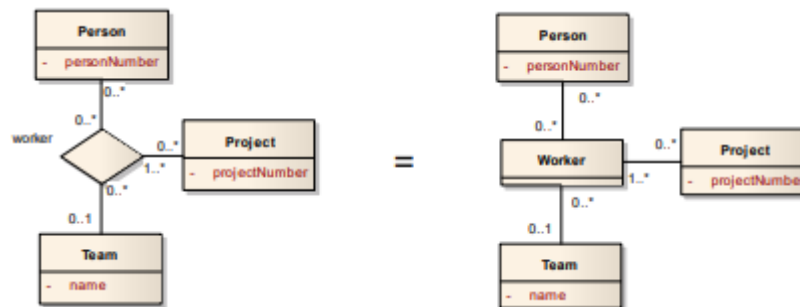


UML	ER
N-ary association Similar to a binary association but with three or more participants	N-ary relationship type Similar to a binary relationship type but with three or more participants

- n-ární vztahy mohou také mít atributy
- UML umožňuje expresivnější kardinality
 - o v příkladu může mít daná kombinace určité osoby a projektu spojení s 0 či více týmy přes danou asociaci



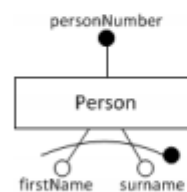
- n-ární vztahy mohou být nahrazeny binárními
 - o n-ární asociace = třída + separátní binární as. pro každého originálního účastníka



Identifiers

Full identification of real-world entities

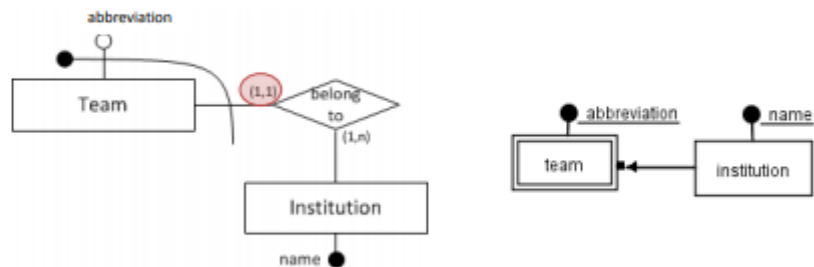
A person is identified either by their personal number or by a combination of their first name and surname.



UML	ER
N/A	Attribute or a group of attributes marked as an identifier

Partial identification of real-world entities

A team is identified by a combination of its name and a name of its institution.



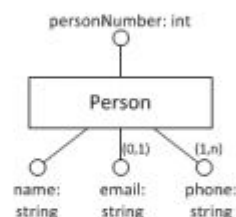
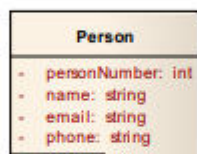
UML	ER
N/A	Attribute or a group of attributes marked as a partial identifier

- každý entitní typ musí být vždy identifikovatelný
 - o alespoň množinou všech svých atributů (není-li specifikován explicitně)
- částečné identifikátory vytváří identifikační závislosti
 - o je povolena jen kardinalita (1,1)
- entitní typy:
 - o **silný** – má alespoň jeden (plný) identifikátor
 - o **slabý** – nemá žádný (plný) identifikátor, alespoň jeden částečný
 - je tedy existenciálně i identifikačně závislý

Data Types

Data type of attributes

A person has a personal number which is an integer and name, email and phone which are all strings.



UML	ER
Attribute of a class may have a data type assigned	Attribute of entity type may have a data type assigned

- Note that...
 - Set of available data types is not specified strictly
 - Data types are actually not very important at the conceptual layer

Sample UML Diagram

