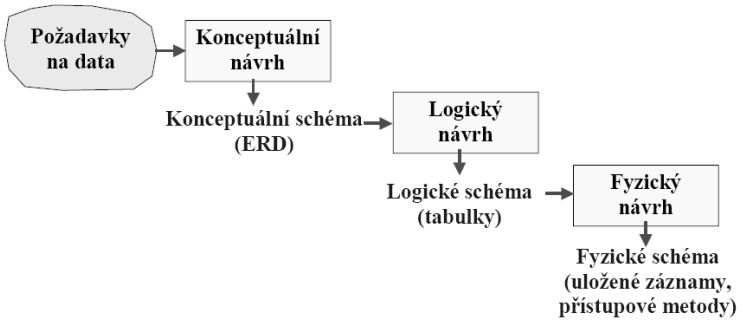
Konceptuální datové modely používané při analýze

Abstraktní datové typy jako definice domén atributů

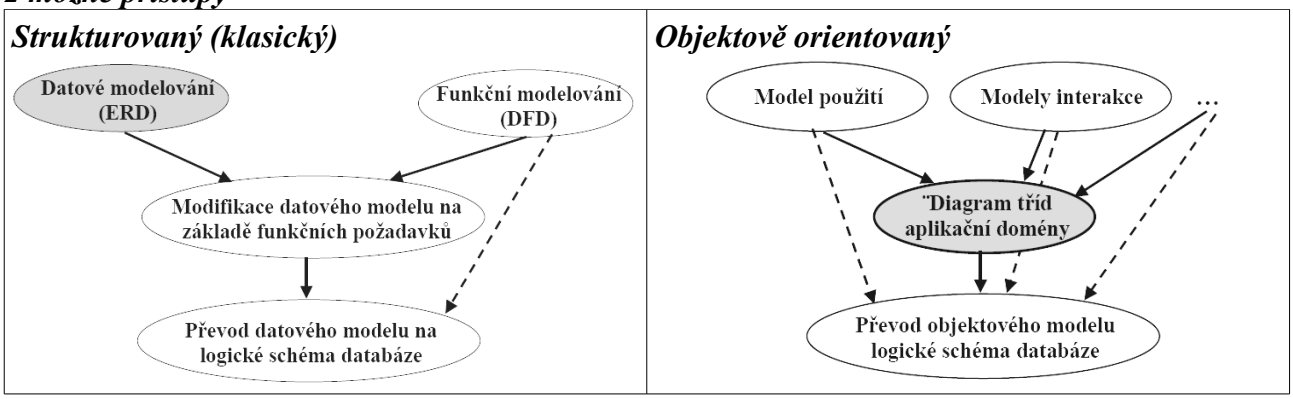
ADT (*Abstraktní datový typ*)

- zapouzdření datového typu
- lidský mozek je schopen řešit úlohy jen do určité míry složitosti
- umožnuje skrýt implementaci
- a způsob fyzického uložení informace v proměnné daného datového typu
- cílem je zjednodušit a zpřehlednit
- umožnuje vytvářet i složitější datové typy
  - zásobník, fronta a pole.
- Jednoduchým příkladem abstrakce dat budiž datový typ bod, který se skládá ze dvou čísel (souřadnic) a který může programátor používat při programování grafiky.
-

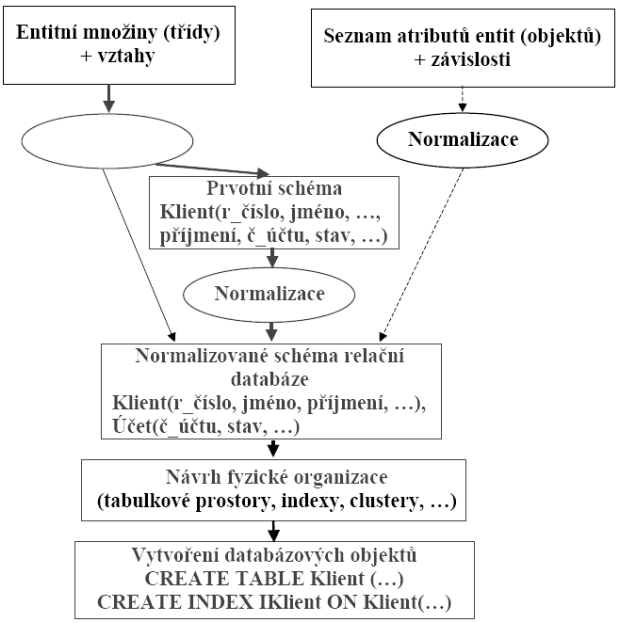
Fáze návrhu databáze



2 možné přístupy

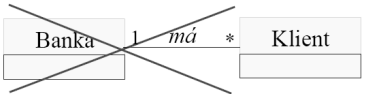


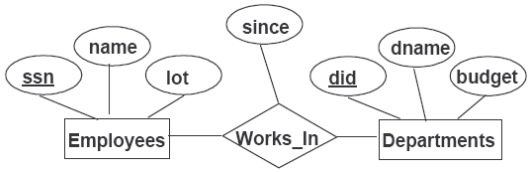
Přístup k návrhu



ER-model

- nejčastější datový model
- neboli ERD (*Entity Relationship Diagram*)
- někdy označovaný ERA (*Entity Relationship Attributes*)
- celkový systém by neměl být zahrnut do ERD
-





Kosočtverec – vztah
Obdélník – entita
Ovál - atribut

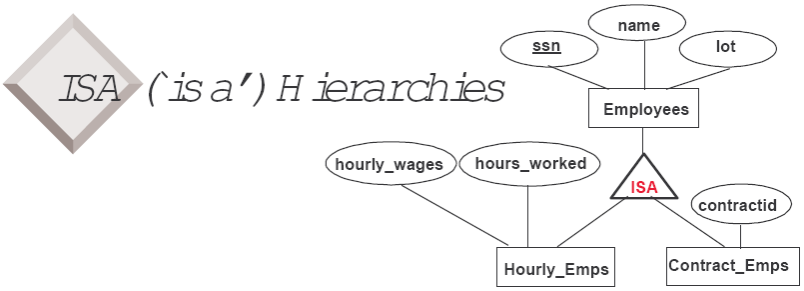
atribut

- nejmenší datová jednotka v bázi dat
- má definovaný typ (INT, VARCHAR(20), BIT) – také závisí na „vendorovi“ :) (ORACLE, MSSQL, MySQL, Postgre …)
- jednoduchý (rodné_číslo)
- složený (jméno = křestní_jméno + příjmení)
  - obvykle stromová struktura
- jednohodnotové / vícehodnotové / libovolné hodnoty (checkbox / dropdown / textfield)
- povolující / nepovolující prázdnou hodnotu (NOT NULL)
- odvozené (lze jí odvodit od jiného atributu nebo entity (věk od rodného čísla)
- Je-li hodnota atributu důležitá, i když neexistuje žádná entita s touto hodnotou jako vlastností, pak bychom ji měli modelovat jako entitu.

entita

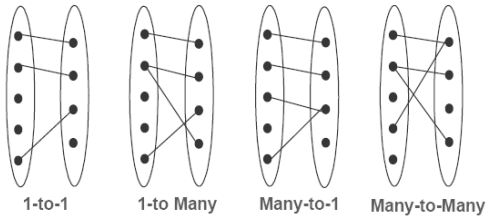
- objekt reálného světa
- je schopen nezávislé existence
- má vlastní identifikaci (primární – identifikační klíč)
- silná / slabá entitní množina (má / nemá klíčový identifikátor mezi vlastními atributy – dominance)
  - Pravidlo 1: Jako slabou modelovat tehdy, kdy entita kompletně zmizí při odstranění

odpovídající identifikující entity.
  - Pravidlo 2: Cokoliv s atributem, který je jednoznačný, by nemělo být modelováno jako slabá entitní množina.
  - Pravidlo 3: Jsme-li na pochybách, modelujeme jako silnou entitní množinu.
- ISA hierarchie
  - jestliže deklarujeme **A** ISA **B**, každá A entita je považovaná rovněž za entitu B



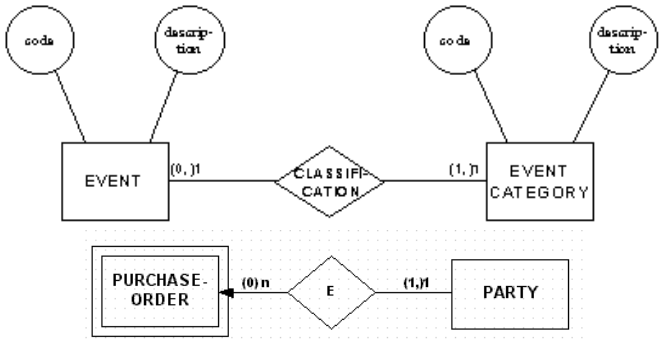
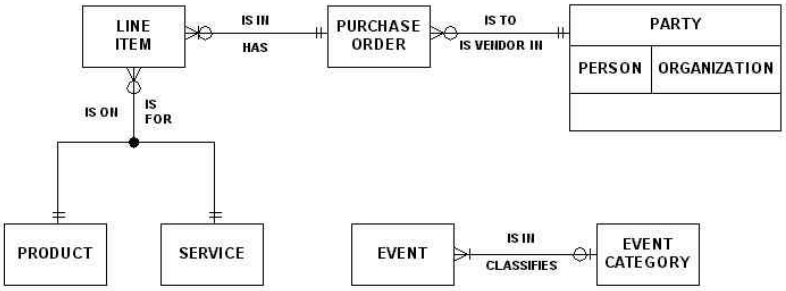
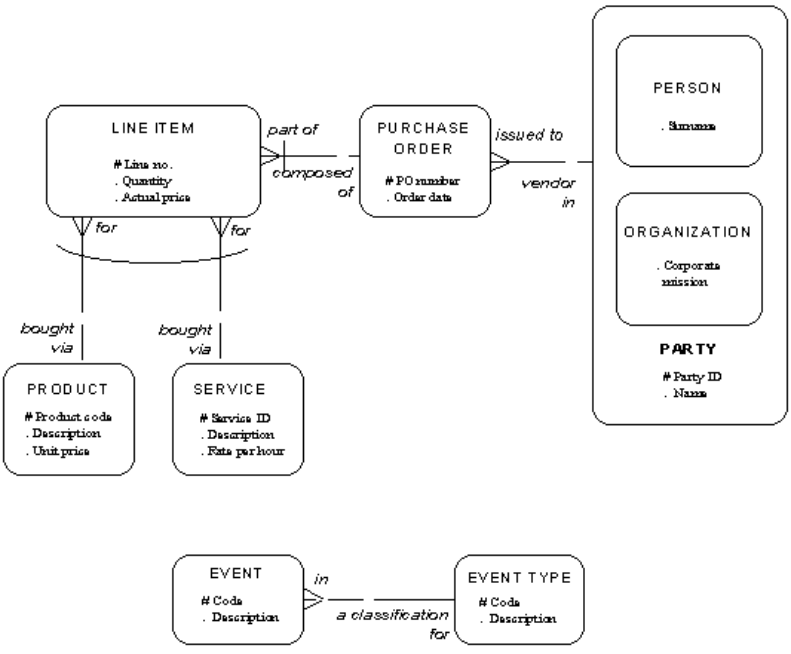
  - překlad buď na dvě relace (Hourly_Emps, Contract_Emps) nebo na tři relace (ještě Epmloyees)
  -

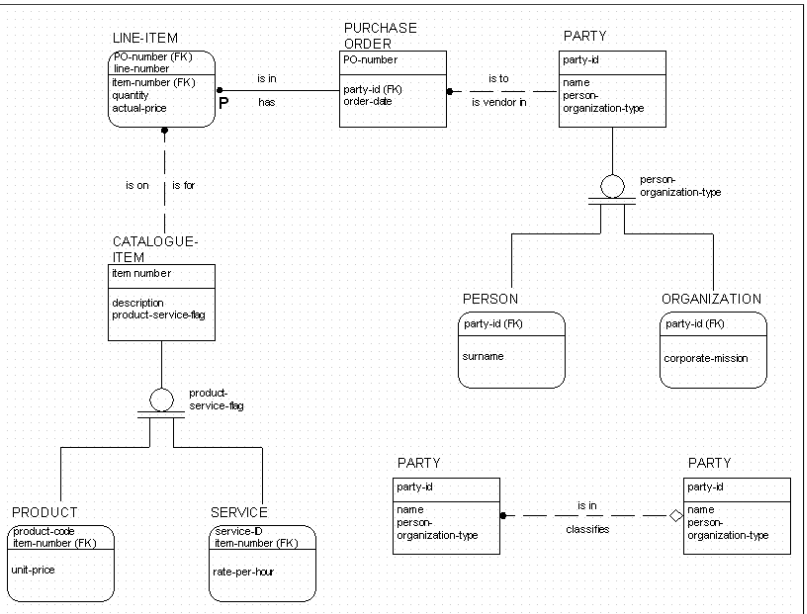
relace
- přiřazení 2 nebo více entit
- např. Novák **pracuje na** Katedře počítačů
- může mít atributy (např. od kdy tam pracuje)
- při překladu relace na entity, každá vztažená entita musí mít
  - klíčový atribut (foreign key)
  - popisné atributy
- existují 4 druhy kardinalit
  - one to one – jeden zaměstnanec může být ředitelem max na jednom oddělení
  - many to one – mnoho zaměstnanců pracuje na jednom oddělení
  - one to many – jeden zaměstnanec pracuje na více projektech
  - many to many – více zaměstnanců může nevim co..
    v zásadě se převádí na 2 vztahy 1:N



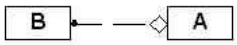
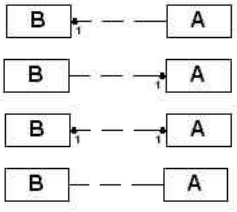
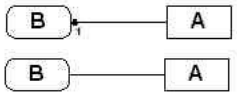
- závislosti
  - relace může být existenčně závislá na entitě
    např. Každá `did` hodnota od `Departmens` se musí vyskytovat v řádku tabulky `Works_In` s nenulovým `ssn`
    v SQL to řekneme takto:
    `FOREIGN KEY (ssn) REFERENCES Eployees`, kde ssn je rovněž definováno jako součást vytvářené tabulky
- integritní omezení
  - doménová – definuje množinu prvků, z kterých vybíráme atributy, například jestli se jedná o číslo (int) nebo varchar(20) PK, integer not null
  - entitní – zajišťuje že entita je pouze jedna, definice identifikačního klíče (PK,FK)
  - referenční – zajišťuje správnost vztahu mezi daty
- 3 způsoby záchováni ref.integrity:
  obvykle vázáno na operace ON_DELETE nebo ON_UPDATE (při mazání / editování)
  - restrict – operaci nepovolí, pokud by to mohlo narušit
  - cascade – dovolí aktualizace, ale upraví všechny záznamy v podřízené tabulce
  - set null – do FK (cizí klíč) nastaví nulu, ale pokud je to PK tak to nejde

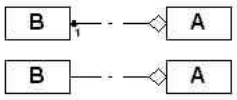
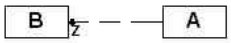
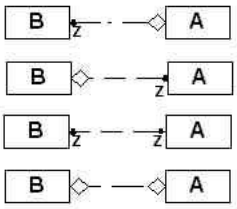
Grafická notace ER-diagramů:

Peter Chen model • nepovinnost ve vztahu naznačena číslem • relace inklinují k tomu nemít jméno • jestliže entita je slabá, používá se vztah se jmenovkou „E“	
Information Engineering • také nazývaný Crows-Foot • nepovinnost = prázdné kolečko • one or more = krátká čárka • one and only one = další čárka • more = crow's foot • vztahy jsou obvykle pojmenovány slovesy • unikátní identifikátory nejsou vyznačeny • plné kolečko značí „exclusive or“	
Richard Barker • vztahy reprezentují čáry rozdělené na polovinu • která polovina je „solid“ tam je povinost ve vztahu povinná • crow's foot značí „many“ • čteme: `PURCHASE_ORDER` může mít více `LINE_ITEM`ů • vztah je pojmenován zpravidla v obou směrech • unikátní identifikátor má „#“ • subtypes: `PERSON` a `ORGANISATION` jsou podtypy `PARTY` • výlučnost: každý `LINE_ITEM` musí být pro `PRODUCT` nebo pro `SERVICE` • užívá např. Oracle Corp.	

IDEF1X • zakulacené entity jsou závislé na jiné entitě • poměrně vyumělkované • málo používané viz. následující tabulka převodů	
UML	Viz class diagramy / object diagramy
XML	Datová struktura popsaná textově v XML notaci

Převody mezi Barkerovo a IDEF1X notacemi

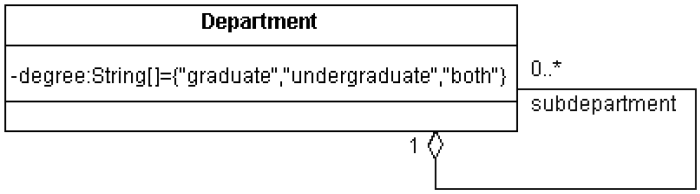
1			One to zero or more (dependent)
2			One to zero or more
3			Zero or one to zero or more
4			One to one or many
5			One to one or many (dependent)
6			One to zero or many
7			Zero or many to zero or many
8			One or many to one or many
9			Zero or many to one or many
10			One or many to zero or many
11			One to one
12			One to one (dependent)

13			Zero or one to one
14			One to zero or one
15			One to zero or one (dependent)
16			Zero or one to zero or one

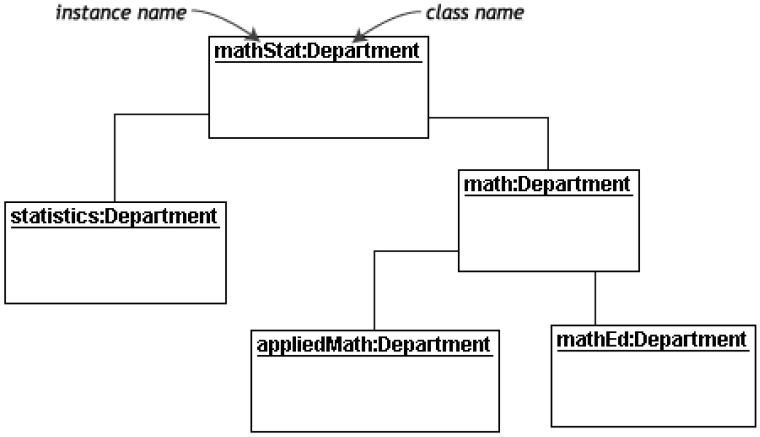
Objektový model

Object diagrams show instances instead of classes. They are useful for explaining small pieces with complicated relationships, especially recursive relationships.

This small class diagram shows that a university **Department** can contain lots of other **Departments**.

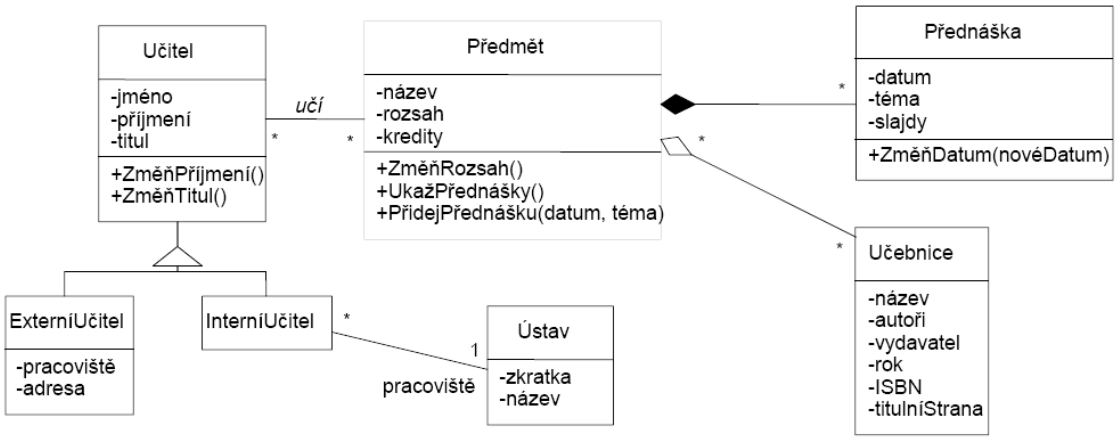


The object diagram below instantiates the class diagram, replacing it by a concrete example.



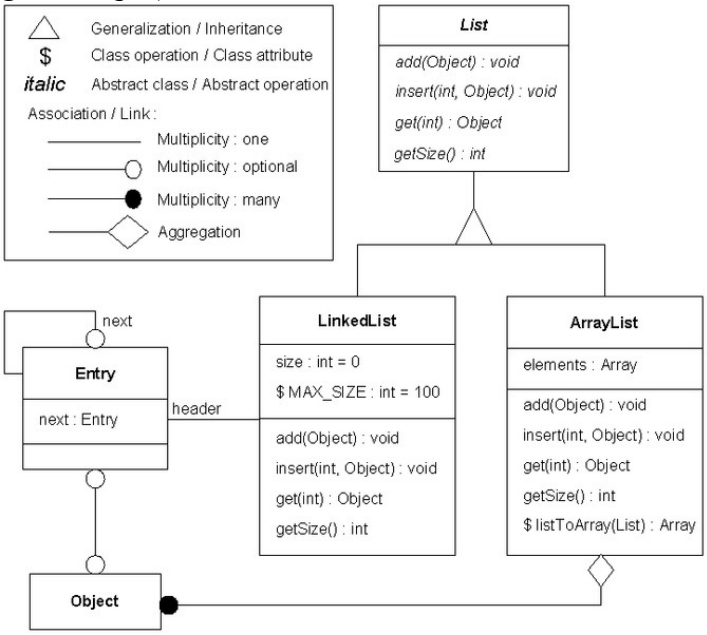
Each rectangle in the object diagram corresponds to a single instance. Instance names are underlined in UML diagrams. Class or instance names may be omitted from object diagrams as long as the diagram meaning is still clear.

- není tvořen entitami a relacemi, ale objekty a vazbami mezi nimi
- častěji než *object diagram* se užívá ***class diagram***



- objekt
  - uchovává informace
  - zprostředkovává služby
  - atributy (uchovává stav)
- hierarchie objektů (generalizace a specializace)
- agregace objektů
  - (více také viz: otázka 33. Unifikovaný modelovací jazyk UML)

OMT (*Object Modeling Technique*)



- objektový model
- dynamický model
- funkční model

- dědičnost, agregace, násobnost asociací (*multiplicity*), instance, závislosti (*dependences*)... viz. otázka 33 – UML.

1) Analýza (popis problému, objektový model = objektový diagram + datový slovník, dynamický model = stavové diagramy + DFD, funkční model)
2) Systémový návrh (možnosti paralelizace, subprocesy, podmínky)
3) Návrh objektů – metody: optimalizace, dědičnost, atributy...
4) Implementace