

2. přednáška

2. října 2012
10:32

Souborově orientované uchování dat

- Slabý HW
- Není možné uchovávat "velká data" - maximálně řádově jednotky MB
- Na každou úlohu samostatná aplikace, která má samostatná data
 - Další aplikace musejí mít svá vlastní data, i když jsou stejná - problém s duplikací dat
 - Každá aplikace musela napsat vlastní algoritmus pro manipulaci s daty
 - Nutno naprogramovat přístupové mechanismy
- Vzniká základní struktura - soubor, záznam, položka
- Program a data jsou na sobě závislé
- Vhodné pro úlohy, které mají výpočtový charakter (vědecko-tech. Výpočty), kde není vhodné použít SŘBD

Databázový přístup k datům (SŘBD)

- Možnost počítání v dekadické aritmetice - potřeba přesných výpočtů, např. s měnou

Databáze (báze dat)

- ☞ Množina souborů a popisu jejich dat, které jsou navzájem v logickém vztahu a jsou spravované systémem řízení báze dat

Na databáze jsou kladeny následující požadavky:

- **Neredundantnost** - databáze by neměla obsahovat zbytečné duplicity
- **Vícenásobná využitelnost** - přístup k datům všem oprávněným uživatelům
- **Integrita dat** - hodnoty uložených dat nesmí být mezi sebou ve sporu
- **Nezávislost dat** - změna struktury uložení dat nevyvolá změny v aplikačních programech
- **Možnost implementovat libovolný datový model**

Systém řízení báze dat (SŘBD)

- ☞ Programový systém, který zabezpečuje
 - definování struktury dat,
 - ukládání dat,
 - výběr dat,
 - ochranu dat a
 - komunikaci systému s uživatelem.

Požadavky na SŘBD:

- **DDL (Data Definition Language)** - Jazyk, který definuje strukturu dat a omezující podmínky (integritní omezení), např. CREATE TABLE, ALTER TABLE, ...
- **DML (Data Manipulation Language)** - Prostředky pro ukládání, výběr a obecně manipulaci s daty
 - Procedurální jazyky - určuje, jak data získat, pracují se záznamem
 - Neprocedurální jazyky - říká, jaká data se mají najít - popis vlastností, které hledáme - jazyk SQL
- ? Rychlík nerad používá zkratky (raději to člení do těch předchozích kategorií):
- **DCL (Data Control Language)** - např. příkazy pro řízení uživatelských práv: GRANT, REVOKE, ...
- **TCL (Transaction Control Language)** - příkazy pro řízení transakcí - BEGIN TRANSACTION, COMMIT
- **DQL (Data Query Language)** - jazyk pro dotazování: SELECT
- 4GL - jazyky 4. generace - Access

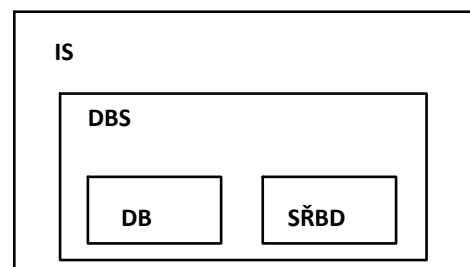
- Možnost definovat **přístupová práva**
- Přístup **více uživatelů**
- **Obnova systému po chybě**
- **Katalog dat** (= popis dat) přístupný uživateli

Příklady:

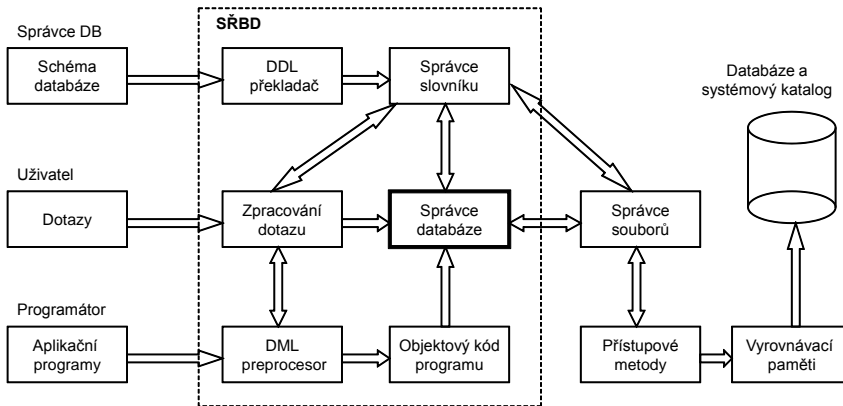
- Oracle, DB2, Access, PostgreSQL, MySQL, Firebird, MySQL, MSSQL, ...

Databázové okolí

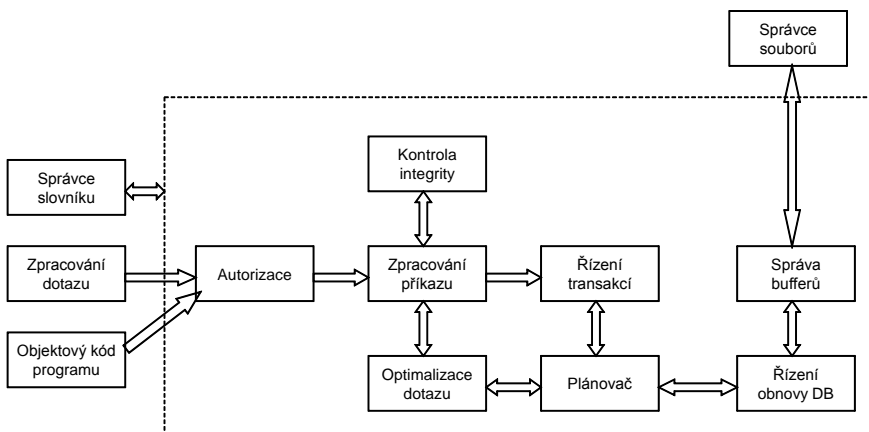
- **Databázový administrátor** - instaluje databázovou aplikaci, udržuje ji, sleduje, konfiguruje, ...
- **Gestor dat (administrátor dat)** - rozumí aplikaci, vidí souvislosti dat, hodnot, ...
- **Vývojový programátor** - navrhuje model databáze - analýzy, popis problémů, ...
- **Aplikační programátor**
- **Koncový uživatel**
 - **Programátor** - užívá dotazovací jazyk - procedury jazyka, nebo SQL
 - **Neprogramátor** - specialisté pro daný proces, pro ně se ta aplikace dělá



Komponenty SŘBD



Správce databáze



Koncepční modelování

- zabývá se problémem z logiky věci, ale ne z pohledu implementace
- nezajímají mě aplikační programy, programovací jazyk, distribuovaná databáze, fyzická omezení...

Objektově orientovaný model

- Entity a relace mezi nimi (entita = model jednoho objektu z reálného světa)
- Entity jsou popsány nejen atributy (stav), ale také jejím chováním

Datový model orientovaný na záznamy

Relační model dat

- Tabulky

Síťový model dat

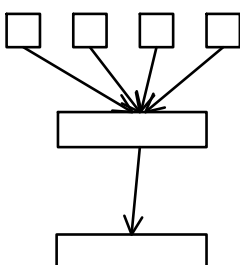
- Soubor, set (př. student a jeho známky), spojky - ukazatel na související záznam

Hierarchický model dat

- Speciální případ síťového modelu, grafem je strom

ANSI-SPARC (1975)

Tříúrovňová abstrakce při pohledu na data:



External level - pohled na data jednoho konkrétního uživatele

Conceptual level - jaká data se v modelu objeví, včetně všech omezení, přístupu k řádkům, ...

Internal level - fyzické uložení dat

3. přednáška

9. října 2012
10:14

ERA model

1976 - Chen

E - entita

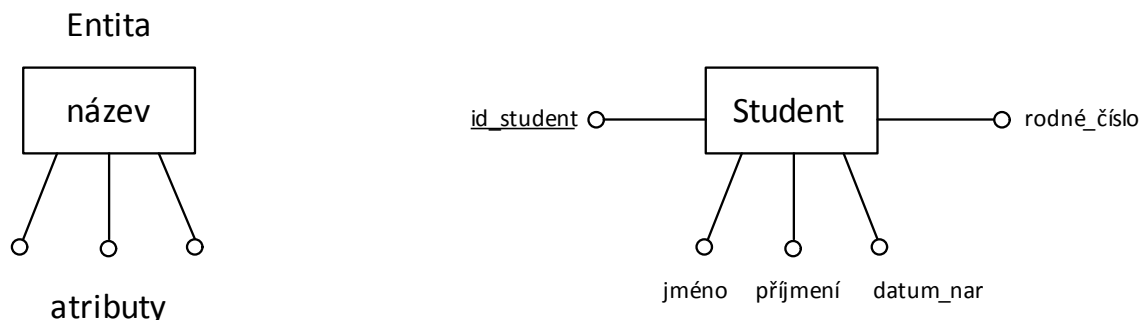
R - relace

A - atribut

Pohledy na zpracování dat:

souborově orientované	datové modely	datová struktura
soubor	entitní množina	tabulka
záznam	entita	sloupec
položka	atribut	sloupec

Příklady



Entita

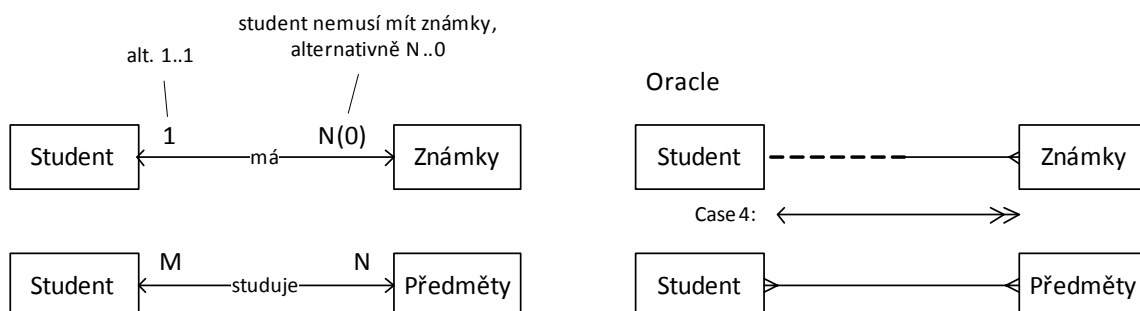
- model jednoho objektu z reálného světa
- popsána **atributy** - např. jméno, datum narození, adresa, ...
- musejí být navrženy podle očekávaných funkcí, např. váha studenta nemá ve studijní agendě smysl
- jeden nebo více atributů se musí určit jako **klíč** - jeho hodnota jednoznačně určí entitu,

Klíč

- pokud jej nemůžeme určit, vytvoříme si umělý klíč: `id_<entita>` (např. `id_student`)
- vlastnosti klíče: unikátní, nenulový (UNIQUE, NOT NULL)
- neměl by nést další informace (např. rok nástupu studia a fakulta je špatně)

Atribut

- určuje se typ: číslo, text, datum, ...
- nemůže být žádnou další datovou strukturou, musí být "jednoduchý" (primitivní datový typ)
- jako text uvádět ty hodnoty, se kterými se počítá



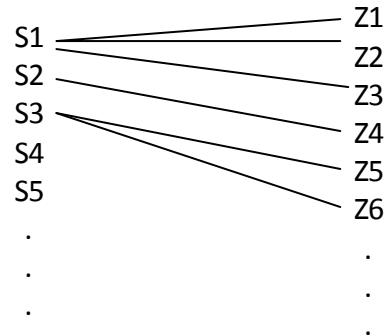
Relace (vztah)

- vyjádření vztahu mezi dvěma entitami
- určujeme
 - kardinalitu
 - 1:1 - vzácná, pokud se použije, musí mít důvod a musím být schopen odpovědět
 - 1:N - jednomu studentovi přísluší N známek a jedné známce přísluší jeden student
 - M:N - jeden student studuje N předmětů a jeden předmět studuje M studentů
 - povinnost výskytu
- nejčastějším modelem je vazba 1:N, kde N je nepovinné (student nemusí mít žádnou známku)

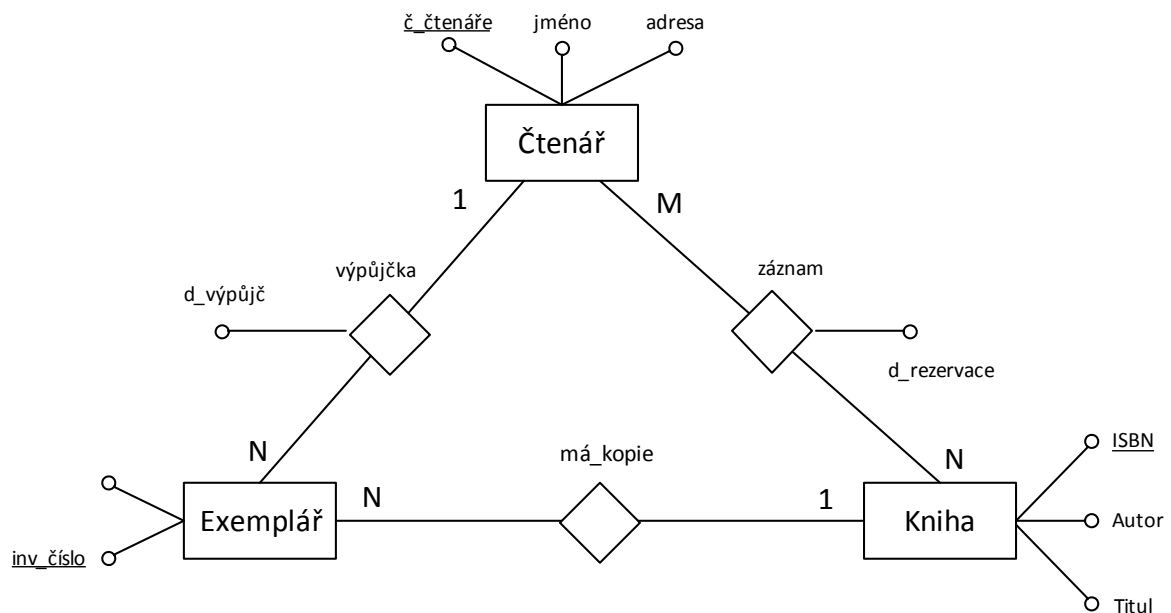


Pozn.: oprávněnost značení 1:N

z pohledu vazby je N konců u studentů, jeden konec u známky



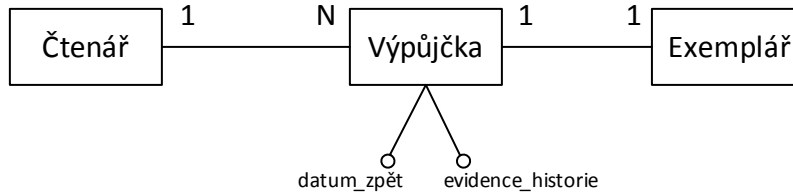
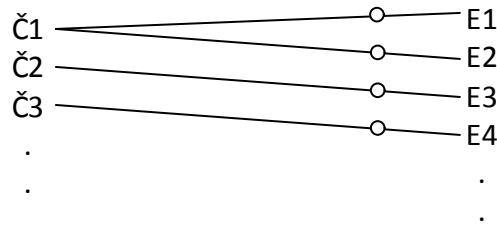
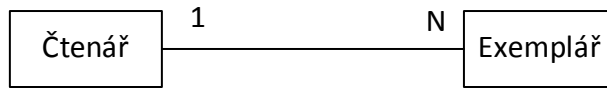
Chen - u vazby mohl být atribut, už se to však nepoužívá



Nedostatky tohoto modelu vůči reálnému světu:

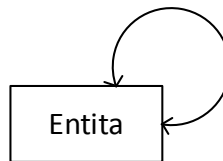
- není možné udělat rozumný rejstřík (autorů, titulů, ...) - nejsou strukturovaní autoři, jsou zapsaní v jedné položce
- nemá historii výpůjček - jeden exemplář má jen jednoho čtenáře

Zavedení historie:



Obecně vazba nemusí být binární, ale i ternární, n-nární. Nahrazují se entitou.

Unární vazba: entita ukazuje na jinou entitu ze stejné entitní množiny, např. rodokmen, jedna osoba má jednu matku a jednoho otce.



Mezi dvěma entitami může být více vazeb:

Oddělení - 1 - [zaměstnanec] - N -> Zaměstnanec

Oddělení - 1 - [vede] - 1 -> Zaměstnanec

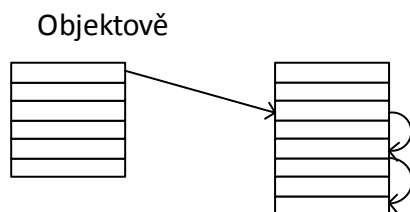
(zde vazba 1:1 má smysl)

Entita někdy nemůže existovat bez existence nějaké nadřazené entity - slabá entitní množina, např. dítě musí mít rodiče.

🗨️ Ne vše lze ERA modelem nakreslit. EER (Enhanced ER) model může - je silnější, ale není podporován databázemi.

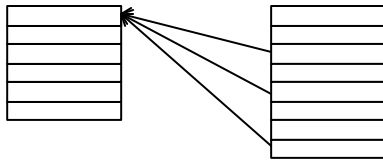
Realizace datového modelu (převod do datové struktury)

- entita a její atributy - tabulka a sloupce
- vazba
 - 1:N



Realizace pomocí cizích klíčů

Relačně



id_student	...
S1	
S2	
S3	
...	...

id_znam	...	id_student
Z1		S1
Z2		S1
Z3		S2
...

- o vazba M:N nejde pomocí cizího klíče zrealizovat, musí se zavést další pomocná množina

ID	Student
S1	...
S2	...

PK_Student	PK_Predmet
S1	P2
S1	P4

ID	Předmět
P1	...
P2	...
P3	...
P4	...
P5	...

Cizí klíče (ID_Student, ID_Predmet) jsou v pomocné entitní množině vždy a jsou primárním klíčem této pomocné entity.

4. přednáška

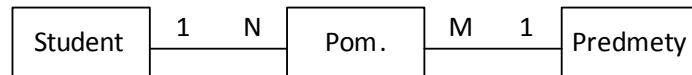
16. října 2012

10:33

- o vazba M:N nejde pomocí cizího klíče zrealizovat, musí se zavést další pomocná množina

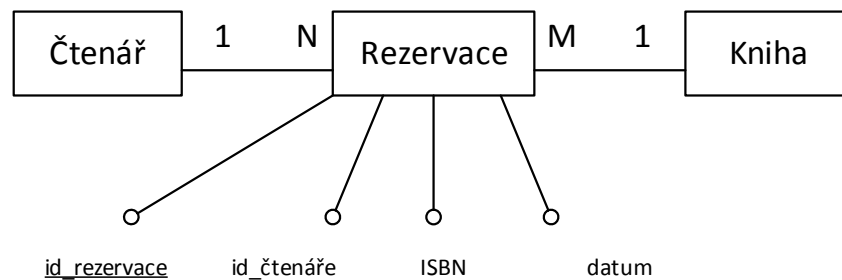
ID	Student
S1	...
S2	...

PK_Student	PK_Predmet
S1	P2
S1	P4



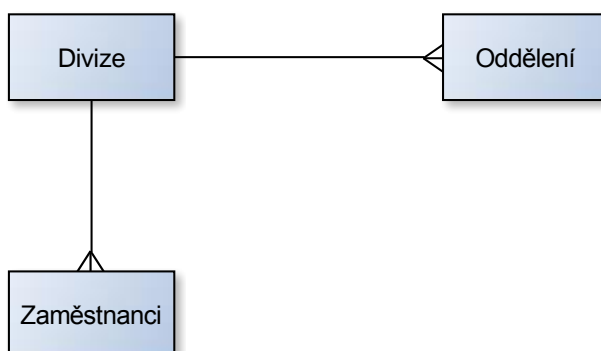
ID	Předmět
P1	...
P2	...
P3	...
P4	...
P5	...

Cizí klíče (ID_Student, ID_Predmet) jsou v pomocné entitní množině vždy a jsou primárním klíčem této pomocné entity.

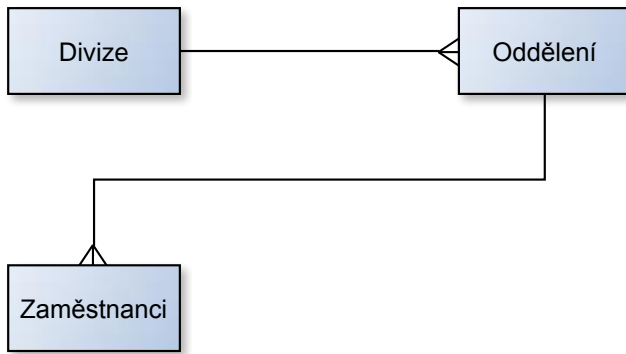


Úskalí při návrhu ERA modelů

- fan trap



Chyba - u zaměstnance není možné určit, v jakém pracuje oddělení. Oprava:



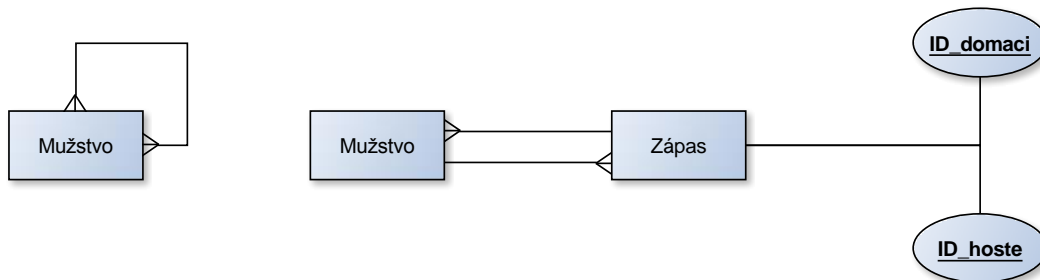
V některých případech není chybou - např. katalogy.

- **Chasm trap**
záleží na povinnosti vazby



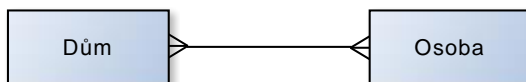
Při přesunu trafo by bez vazby mezi přívodem a vývodem zanikla informace o spojení mezi přívodem a vývodem.

- **M:N**
Unární vazba

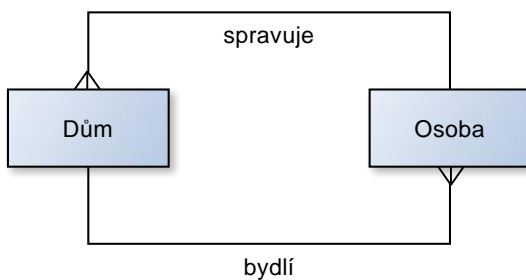


Evidence osob, které chodí do domu:

Vchodem do domu projde více osob a některé osoby chodí do více domů, tzn. kandidát na vazbu M:N:



Některé osoby však chodí do více domů ne z toho důvodu, že bydlí ve více domech, ale proto, že ty domy spravují:



Nutno se nad vazbou M:N zamyslet, v prvním případě to ve skutečnosti není přímo vazba M:N a jde rozložit.

ID_domu	...	ID_spravce
D1		O2
D2		O2

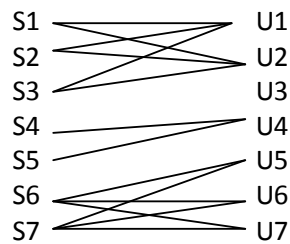
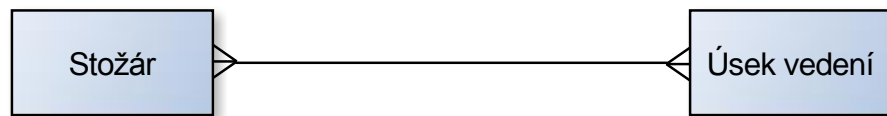
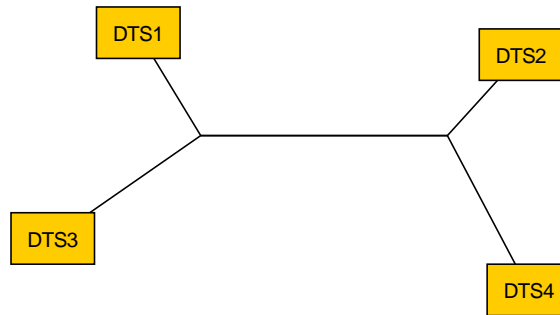
D3		O2
----	--	----

ID_osoby	...	ID_domu
O1		D1
O2		D1
O3		D1
O4		D2

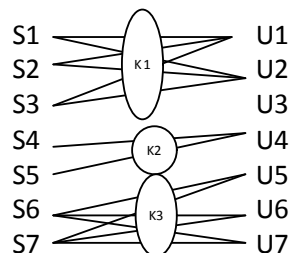
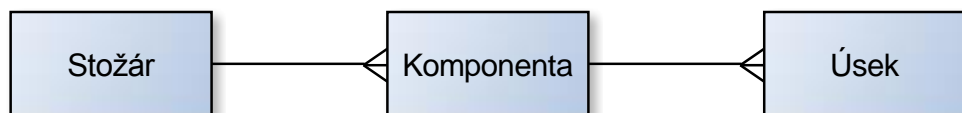
Kritérium rozhodování: $|E1| + |E2| - 1 \leq |POM|$ - s vazbou jsem se možná spletl, pokud $|E1| + |E2| - 1 > |POM|$, tak je oprávněná.

Př.

Elektrické vedení se dělí na úseky a na jednom stožáru může být až pět úseků z různých vedení.



Některé úseky spolu vedení sdílí, jiné ne. Vždy však sdílí buď celou trasu, nebo nesdílí vůbec - vznikají "úplné komponenty":



ID_stož		FK_ID_komp
S1		1

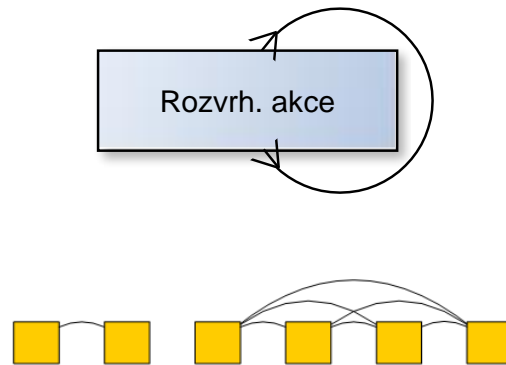
S2		1
S3		1

ID_úsek		FK_ID_komp
U1		1
U2		1

Na vazbě se podílejí úseky se stejným ID komponenty.

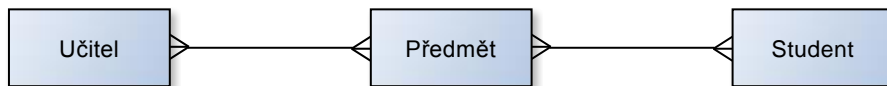
Unární vazba

STAG - rozvrhová akce - závislosti. Pokud si student zapiše jednu rozvrh. akci, tak si musí zapsat všechny ostatní, které jsou v závislosti. Tvoří spolu úplné komponenty.



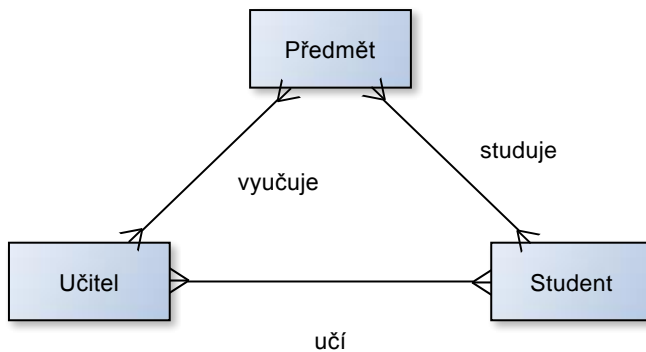
Učitel učí studenty

1)



fan trap - nelze najít souvislost mezi studentem a učitelem

2)



Vyučuje

U1	P1
U1	P2
U2	P1
U2	P2

Studuje

S1	P1
S1	P2
S2	P1

S2	P2
----	----

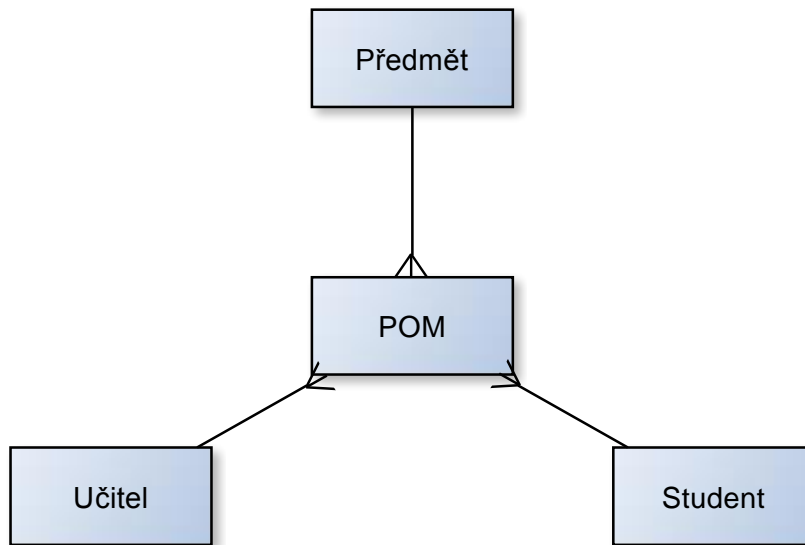
Učí

U1	S1
U1	S2
U2	S1
U2	S2

Nepoznám, jaký předmět učí učitel studenta. Model je špatně.

3)

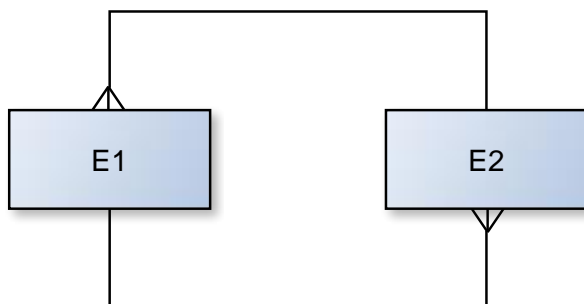
U1	P1	S1
U1	P2	S1
U2	P1	S2
U2	P2	S2



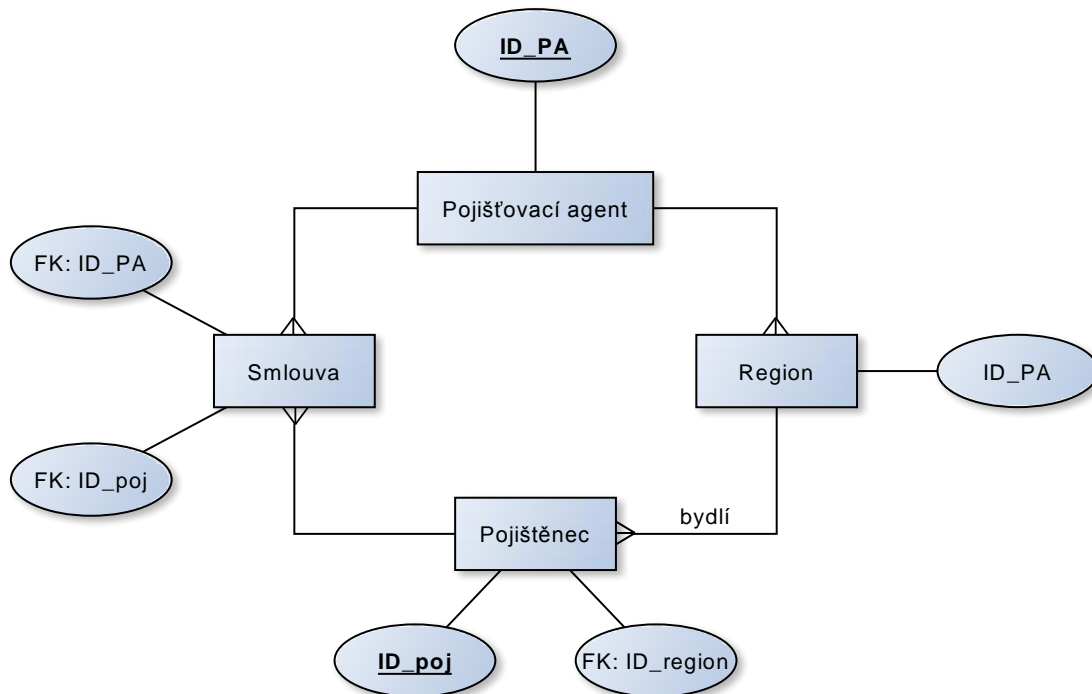
- Obecná ternární vazbu nelze nahradit třemi vazbami binárními! Nutno již na úrovni analýzy je nutno ji nahradit pomocnou entitní množinou. Pomocnou množinu je vhodné použít již při návrhu binárních vazeb.

Cykly v ERA modelech

Cyklus pro 2 entity:

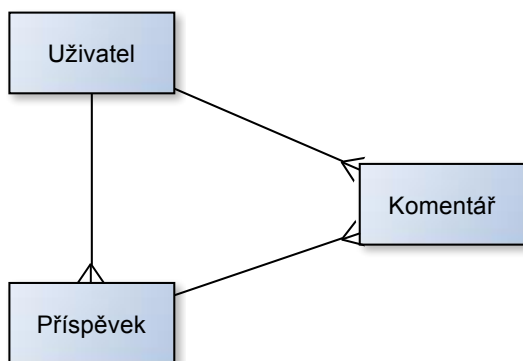


Pojišťovací agenti



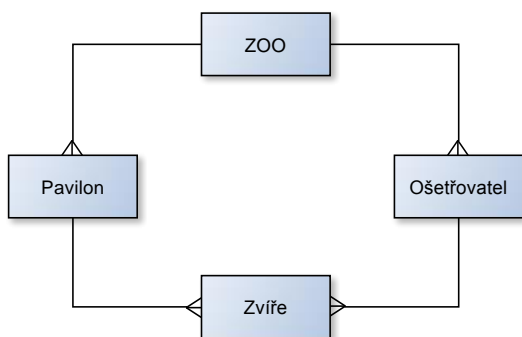
? Může pojišťovací agent uzavřít smlouvu s klientem mimo jeho region? Nevíme.

Weby



? Komentář vkládá ten samý uživatel, co vkládá příspěvek? Ne.

ZOO



? Musí mít zvíře stejné ID pavilonu, jako ošetřovatel, který jej ošetřuje? Ano.

Může se sejít ID ze stejné entity dvakrát, pokaždé však přiteče z jiné vazby.

Pro řešení přidělíme výskytu ID roli. V ZOO - ID pavilonu se u zvířete se objeví v roli že má ošetřovatele a že má pavilon. Navíc provedeme úvahu, jestli musejí být stejné, nebo ne.