

Úvod do databázových systémů

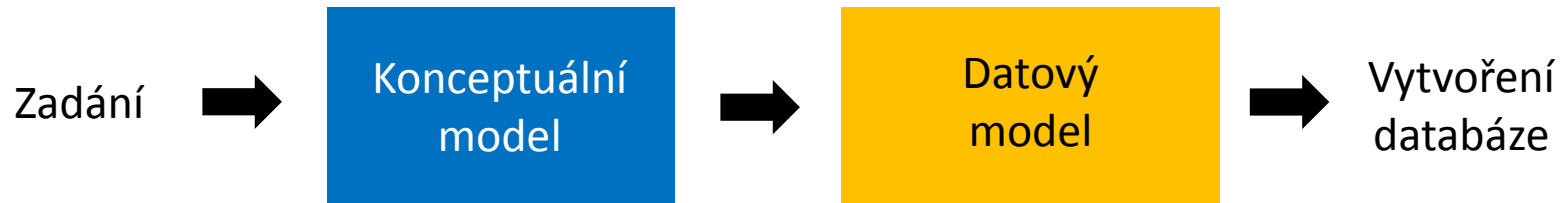
7. cvičení

Ing. Petr Lukáš
petr.lukas@nativa.cz

Ostrava, 2013

Modelování databází

Modelování databází



Konceptuální model – v této fázi zatím nemusíme uvažovat o implementaci nějaké databáze. Zkrátka snažíme se zachytit statický pohled na reálnou situaci.

Datový model – modelujeme databázi a máme jasno, zda budeme používat tabulky, XML, objekty nebo jinou organizaci dat.

- **Relační datový model** – nejběžnější, používáme tabulky
- **XML** – v určitých případech lépe modeluje reálnou situaci
- **Objektový datový model** – využívá výhody OOP jako např. dědičnost

Konceptuální datový model

Entita, entitní typ

Relační datový model

Záznam – popisuje konkrétní výskyt objektu

Schéma relace – popisuje množinu záznamů se stejnými atributy

Konceptuální model

Entita – konkrétní objekt reálného světa
např. Jan Novák narozen 1.1.1990

Entitní typ – popis množiny entit se stejnými atributy. Používáme lineární zápis.
např. Osoba (rc, jmeno, prijmeni, datum_narozeni)

Relační datový model

Primární klíč – atribut nebo více atributů, jejichž hodnota nebo kombinace hodnot je pro každý záznam jedinečná

Cizí klíč – atribut odkazující se na primární klíč

Konceptuální model

Klíč – jeden nebo více atributů, které jednoznačně identifikují entitu v množině entit

*Na úrovni konceptuálního modelu **nepoužíváme cizí klíč**, vše zaznamenávají vztahy samy o sobě (viz dále)*

Cizí klíč a konceptuální model

V konceptuální modelu neuvádíme cizí klíče!

Proč? Představme si na chvíli, že bychom místo relačního datového modelu použili XML.

```
<osoba>  
  <id>1</id>  
  <jmeno>Karel</jmeno>  
  <email>karel@gmail.com</email>  
  <email>karel@seznam.cz</email>  
  <email>karel@vsb.cz</email>  
</osoba>
```

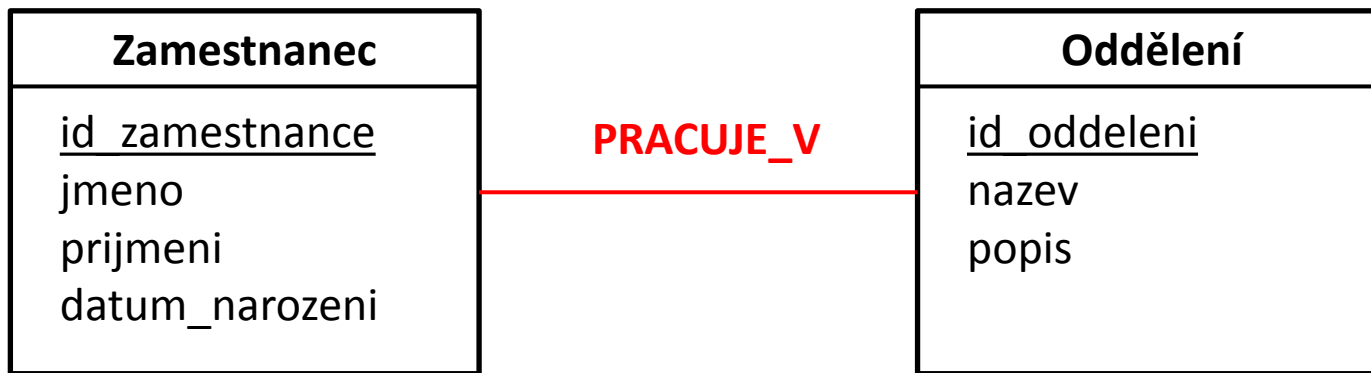
Mezi osobou a emailem je vztah 1:N. Musíme u mailu dodatečně psát, že se vztahuje k osobě s určitým id?

- Pro znázornění konceptuálního modelu používáme nejčastěji **E-R diagramy** a **lineární zápisy**.
- E-R diagramy tedy můžeme používat jak pro konceptuální, tak pro relační datový model. Proto se často setkáme s požadvkem **2 úrovní E-R diagramu**.

Vztahy

Vztah vyjadřuje fyzickou nebo konceptuální vazbu mezi entitami, resp. entitními typy.

- Každý vztah by měl být **pojmenovaný**
- Nejčastěji se setkáváme s **binárními vztahy**
- Vztahy zaznačujeme **E-R diagramem** nebo **lineárním zápisem**



PRACUJE_V (Osoba, Firma)

Vztahy

Oddělení

Zaměstnanec

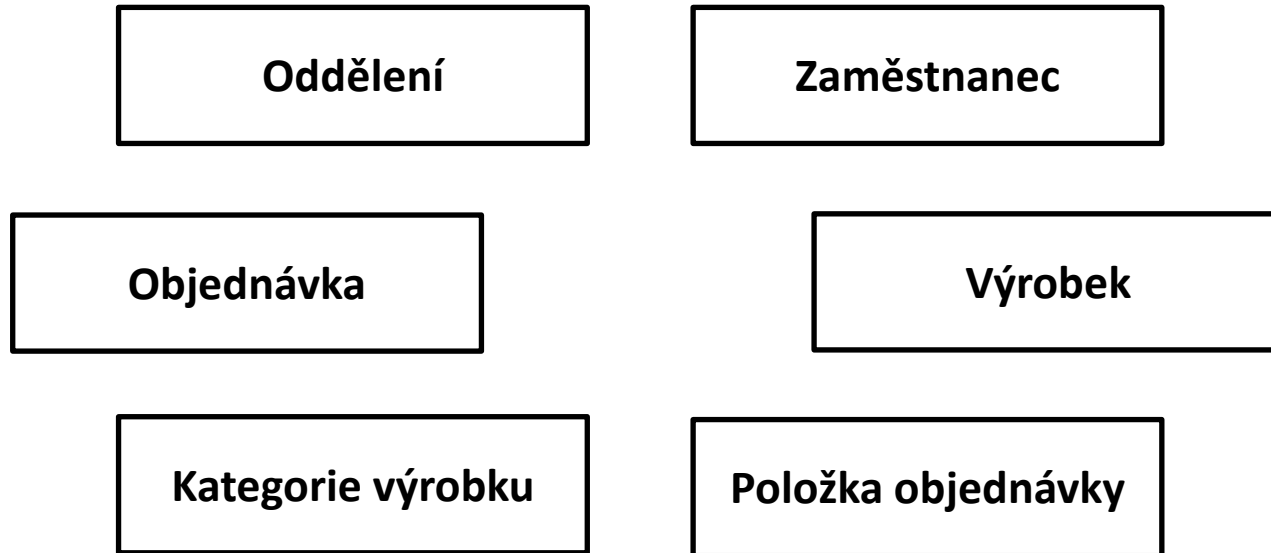
Objednávka

Výrobek

Kategorie výrobku

Položka objednávky

Vztahy



Např:

PRACUJE_V (Zaměstnanec, Oddělení)

OBSAHUJE (Objednávka, Položka objednávky)

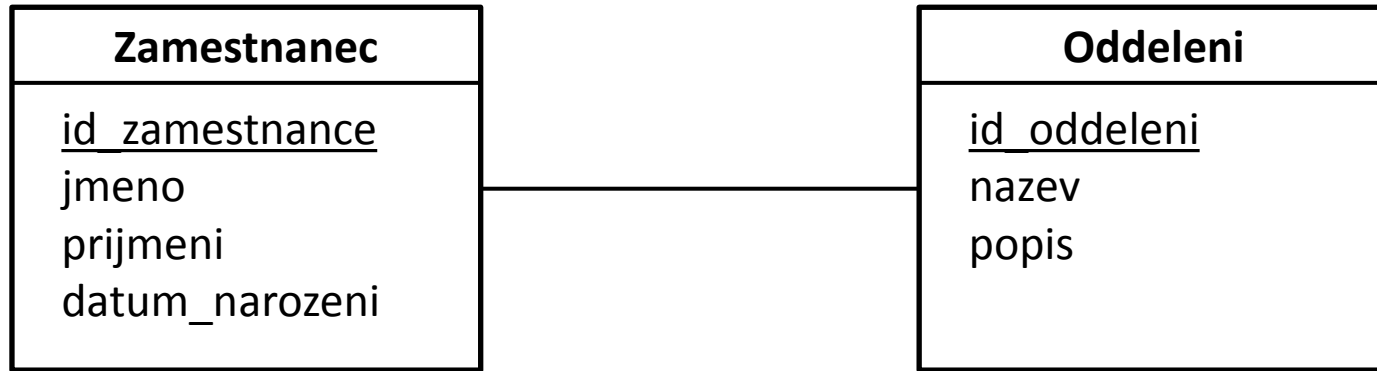
ZALOZIL (Zaměstnanec, Objednávka)

JE_V_KATEGORII (Výrobek, Kategorie výrobku)

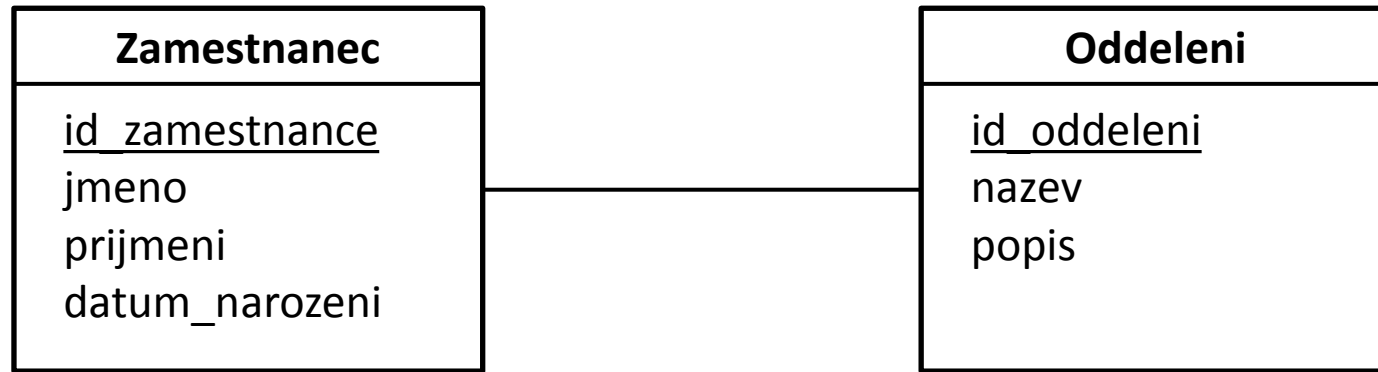
JE_NADRIZENYM (Zaměstnanec, Zaměstnanec)

JE_NA_POLOZCE (Výrobek, Položka objednávky)

Kardinalita vztahu

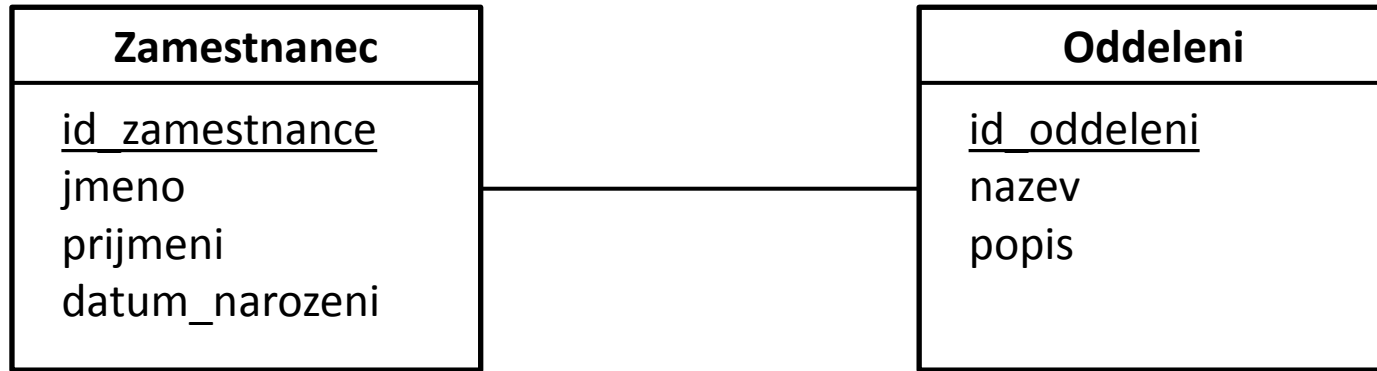


Kardinalita vztahu



- Jde jednoduše o určení, zda se jedná o vztah **1:1**, **1:N** nebo **M:N**.
- Jaká bude kardinalita vztahu na obrázku?

Kardinalita vztahu



1:1

1:N

M:N

Kardinalita vztahu

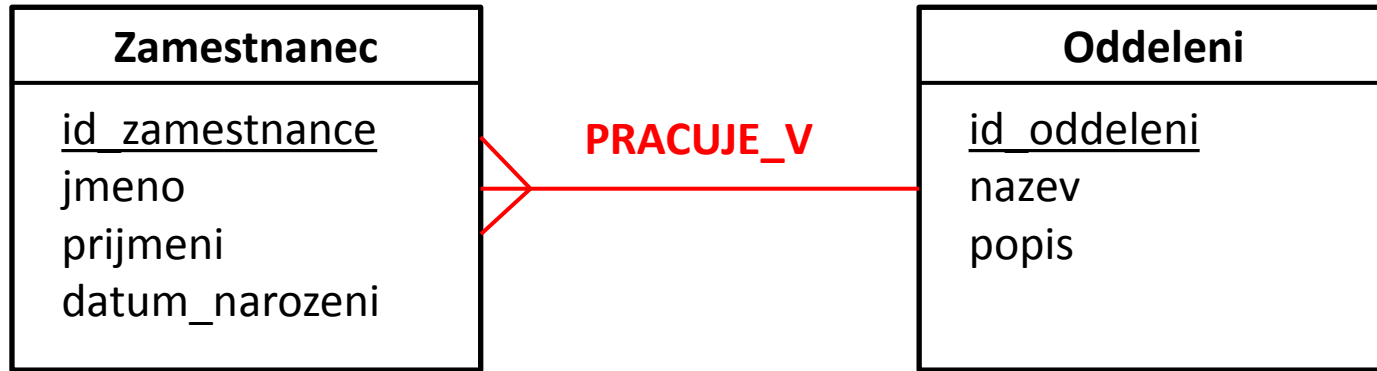


1:1 Zaměstnanec může být **vedoucím** jednoho oddělení, oddělení má jednoho vedoucího.

1:N

M:N

Kardinalita vztahu

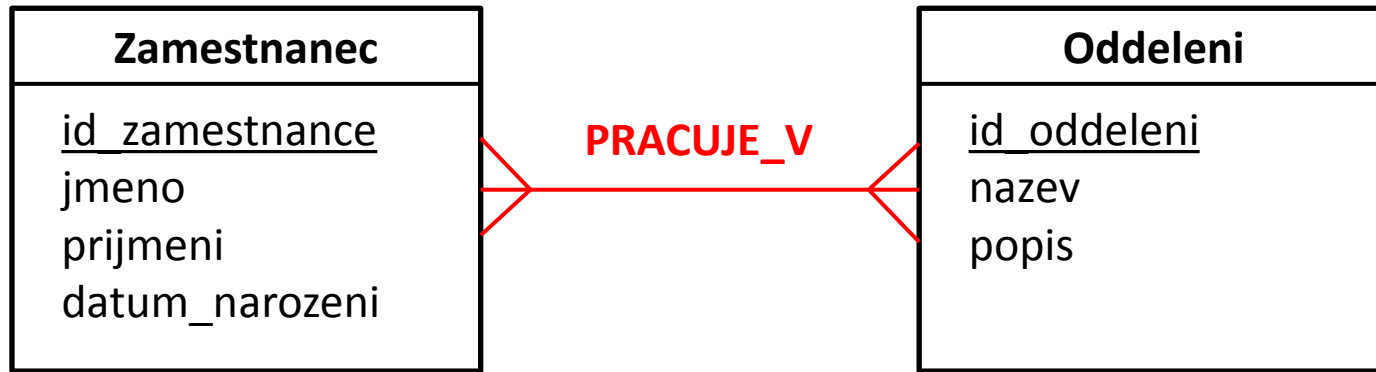


1:1 Zaměstnanec může být **vedoucím** jednoho oddělení, oddělení má jednoho vedoucího.

1:N Jedno oddělení má více zaměstnanců. Každý zaměstnanec **pracuje v** jednom oddělení.

M:N

Kardinalita vztahu

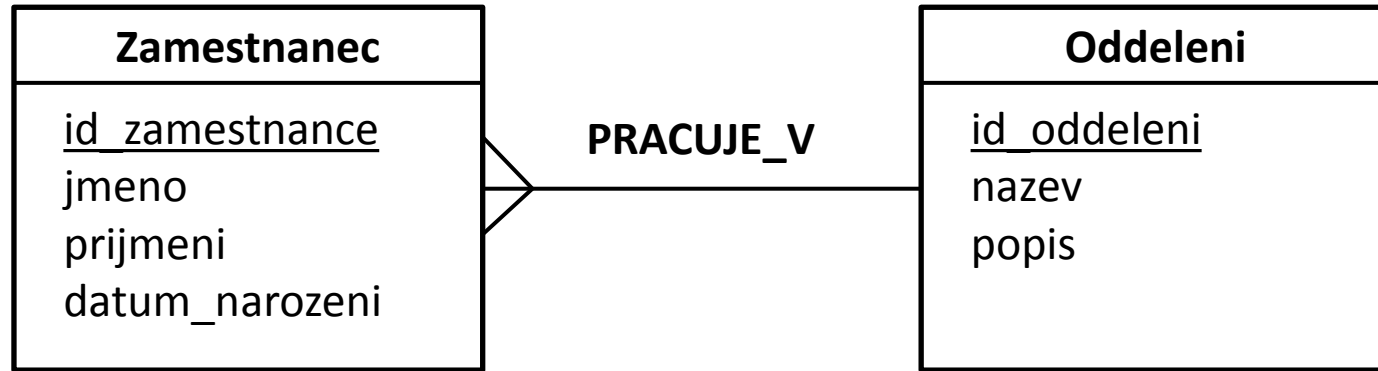


1:1 Zaměstnanec může být **vedoucím** jednoho oddělení, oddělení má jednoho vedoucího.

1:N Jedno oddělení má více zaměstnanců. Každý zaměstnanec **pracuje v** jednom oddělení.

M:N Jedno oddělení má více zaměstnanců, jeden zaměstnanec může **pracovat ve** více odděleních.

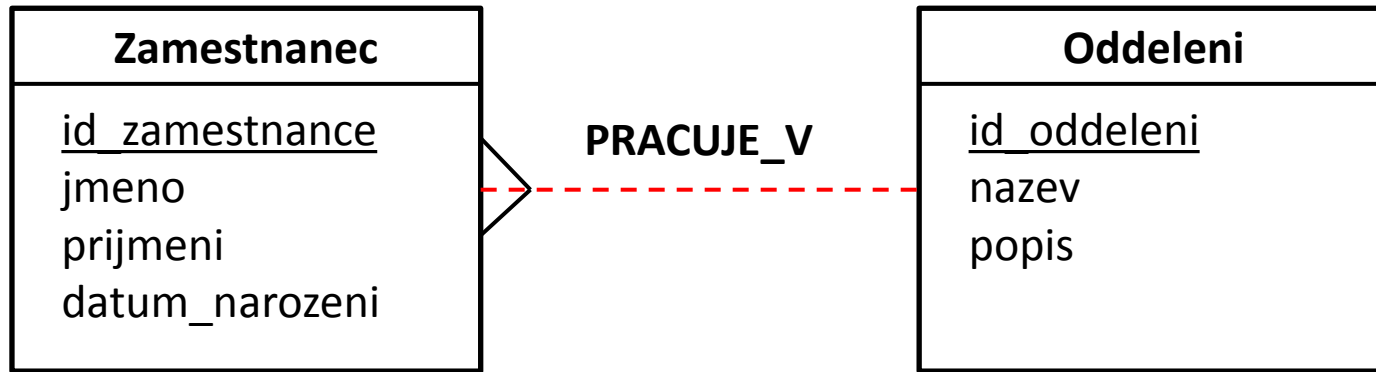
Povinnost ve vztahu



Musí každý zaměstnanec pracovat v nějakém oddělení?

Musí mít každé oddělení nějakého zaměstnance?

Povinnost ve vztahu



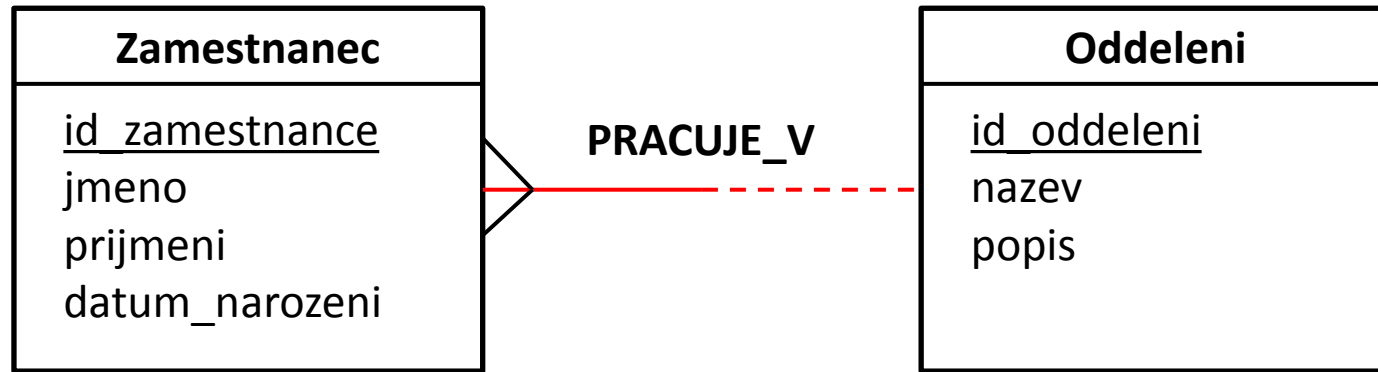
Zaměstnanec nemusí být v oddělení a oddělení nemusí mít zaměstnance.

Zaměstnanec musí pracovat v oddělení, ale oddělení nemusí mít zaměstnance.

Zaměstnanec nemusí být v oddělení, ale oddělení musí mít zaměstnance.

Zaměstnanec musí být v oddělení a oddělení musí mít zaměstnance.

Povinnost ve vztahu



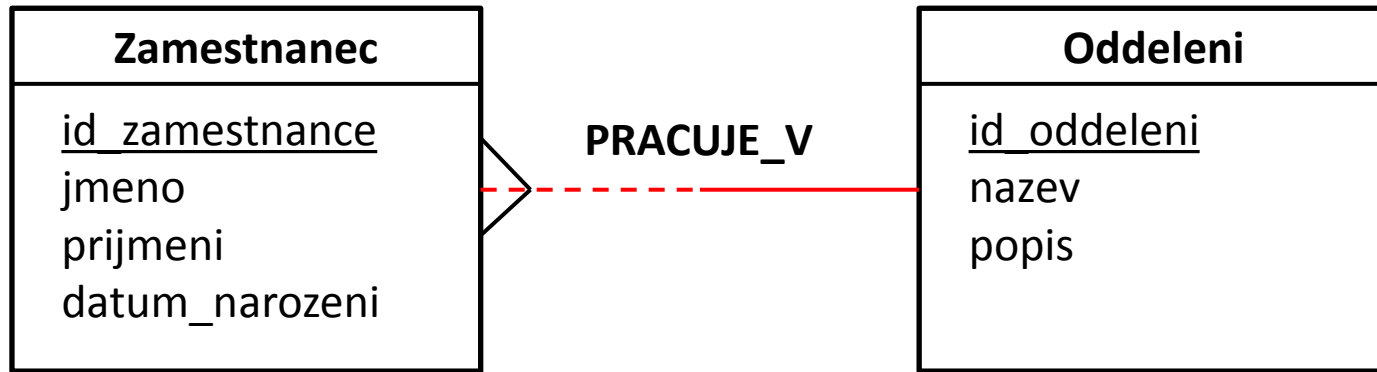
Zaměstnanec nemusí být v oddělení a oddělení nemusí mít zaměstnance.

Zaměstnanec musí pracovat v oddělení, ale oddělení nemusí mít zaměstnance.

Zaměstnanec nemusí být v oddělení, ale oddělení musí mít zaměstnance.

Zaměstnanec musí být v oddělení a oddělení musí mít zaměstnance.

Povinnost ve vztahu



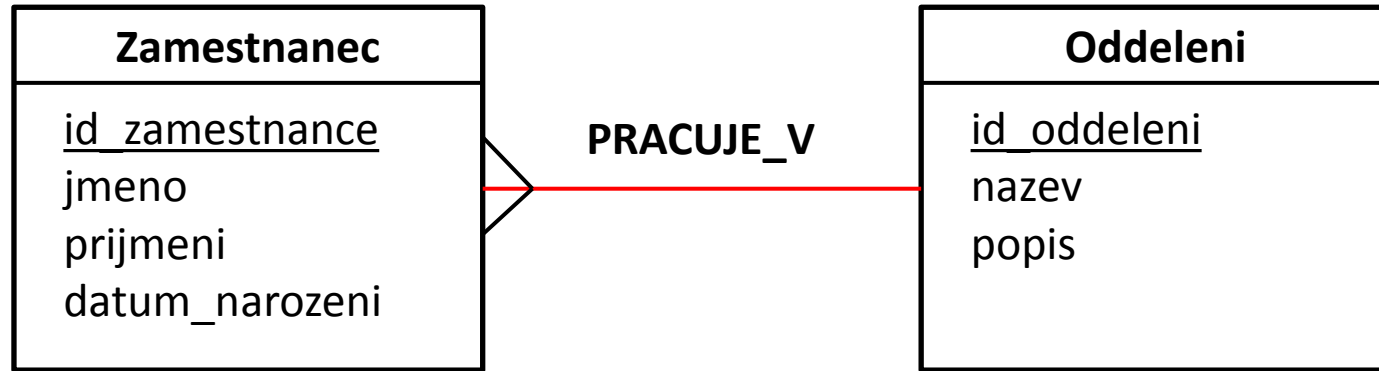
Zaměstnanec nemusí být v oddělení a oddělení nemusí mít zaměstnance.

Zaměstnanec musí pracovat v oddělení, ale oddělení nemusí mít zaměstnance.

Zaměstnanec nemusí být v oddělení, ale oddělení musí mít zaměstnance.

Zaměstnanec musí být v oddělení a oddělení musí mít zaměstnance.

Povinnost ve vztahu



Zaměstnanec nemusí být v oddělení a oddělení nemusí mít zaměstnance.

Zaměstnanec musí pracovat v oddělení, ale oddělení nemusí mít zaměstnance.

Zaměstnanec nemusí být v oddělení, ale oddělení musí mít zaměstnance.

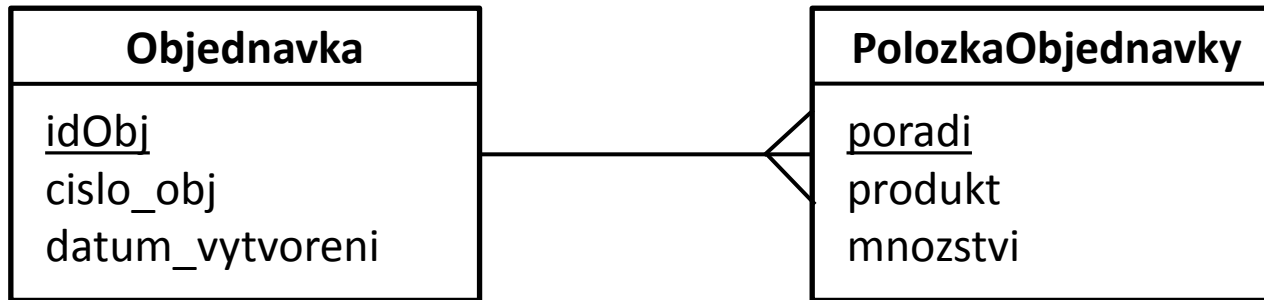
Zaměstnanec musí být v oddělení a oddělení musí mít zaměstnance.

Slabý a silný entitní typ

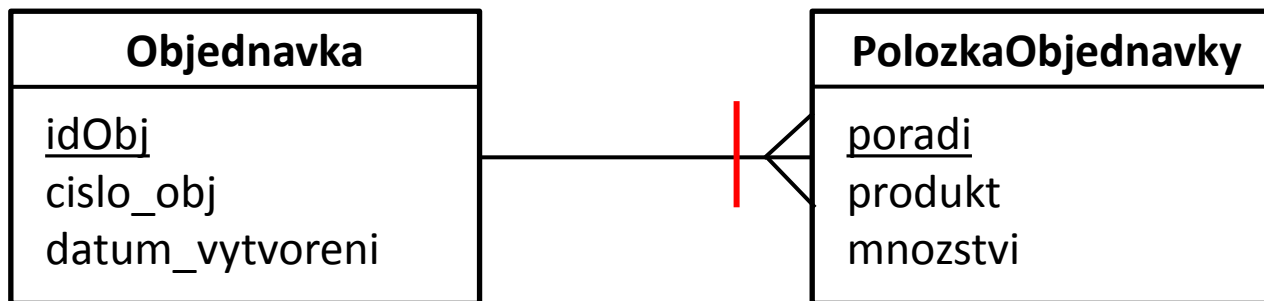
Silné entitní typy se vyznačují tím, že mají svůj vlastní klíčový atribut (popř. více atributů), který je identifikuje.

Slabé entitní typy naopak nemají vlastní atribut, který by je identifikoval. Obvykle vyjadřují objekty reálného světa, které nemohou existovat bez jiného nadřazeného objektu – jejich samostatná existence nedává smysl.

Slabý entitní typ a identifikující vztah



Slabý entitní typ a identifikující vztah



- **Položka objednávky nemá vlastní klíčový atribut.** Je identifikována klíčem z objednávky a pořadovým číslem. Jedná se tedy o slabý entitní typ.
- **Klíč položky objednávky** je složený z *odObj* a *poradi*.
- **Cizí klíče nezakreslujeme do konceptuálního modelu**, takže *poradi idObj* nemůže být součástí položky objednávky.
- Proto **používáme tzv. identifikující vztah**, který zajistí, že součástí klíče položky objednávky bude kromě pořadí také id objednávky, přestože id objednávky není součástí položky.

Shrnutí

- **Entita**
- **Entitní typ**
- **Klíč**
- **Vztah**
- **Kardinalita vztahu**
- **Povinnost ve vztahu**
- **Slabý entitní typ**
- **Identifikující vztah**

Shrnutí

- **Entita**
Objekt reálného světa
- **Entitní typ**
Označení celé třídy objektů reálného světa
- **Klíč**
Atribut nebo více atributů, které jednoznačně identifikují entitu
- **Vztah**
Fyzická nebo konceptuální vazba mezi entitami
- **Kardinalita vztahu**
1:1, 1:N, M:N
- **Povinnost ve vztahu**
Pro každý (binární) vztah máme celkem 4 možnosti
- **Slabý entitní typ**
Klíč je složen z atributů, které nejsou jeho součástí
- **Identifikující vztah**
Zajistí, aby se klíč z nadřazeného entitního typu stal součástí klíče slabého entitního typu

www.dbedu.cs.vsb.cz

- Přihlášení přes **jednotný login a heslo**
- Vpravo sloupec -> *České kurzy* -> *UDBS*

<http://homel.vsb.cz/~luk194/udbs>

- U 7. cvičení je jednoduchý „manuál“ k použití Oracle SQL DDM