

Úvod do databázových systémů

8. cvičení

Ing. Petr Lukáš
petr.lukas@nativa.cz

Ostrava, 2013

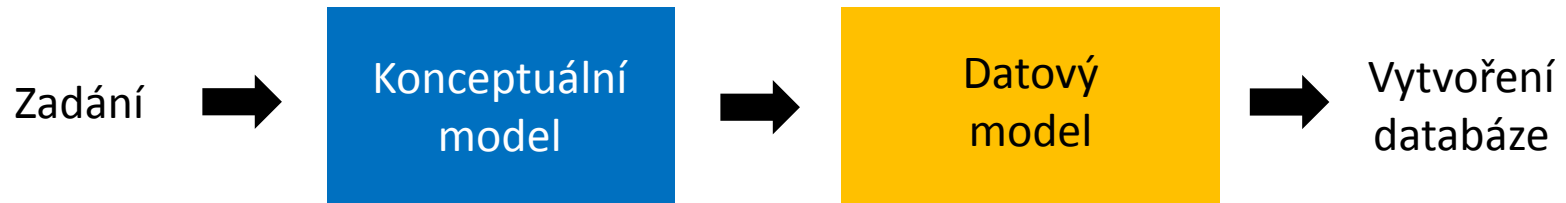
Opakování

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

Opakování

- **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

Modelování databází

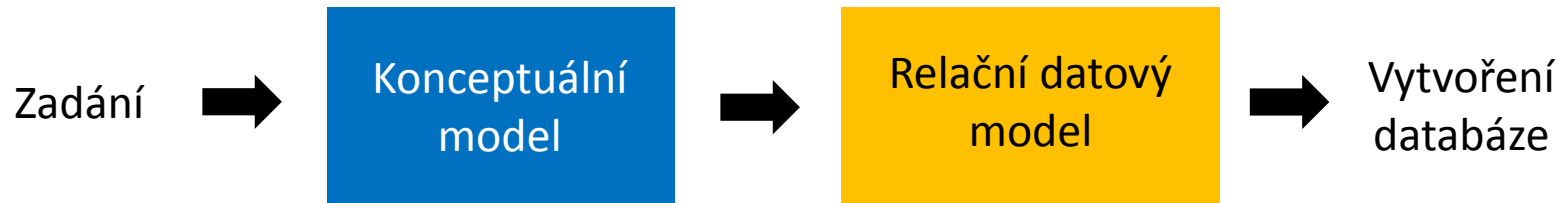


Konceptuální model – v této fázi zatím nemusíme uvažovat o implementaci nějaké databáze. Zkrátka snažíme se zachytit statický pohled na reálnou situaci.

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

Modelování databází



- 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**.

Převod konceptuálního modelu na relační

Převod konceptuálního modelu na relační

- Převod entitních typů na relační schémata
- Převod atributů
- Určení primárních klíčů
- Převod vztahů a určení cizích klíčů
- Řešení povinnosti ve vztahu
- Převod slabých entitních typů a identifikujících vztahů

Převod entitních typů a atributů, určení prim. klíče

Konceptuální model

Vyrobek

id_vyrobek

nazev

aktualni_cena

popis

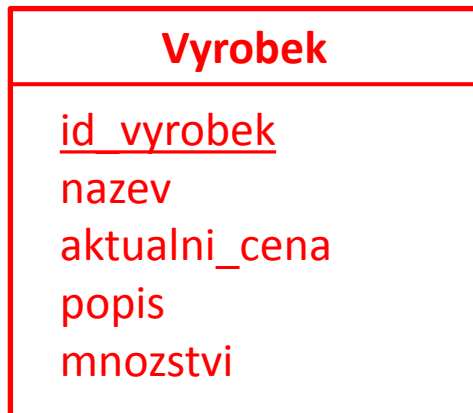
mnozstvi

Relační datový model

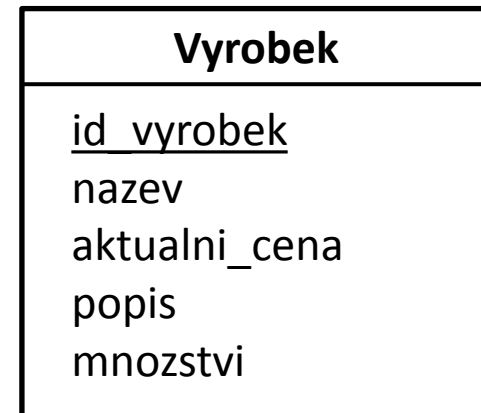
Převod entitních typů a atributů, určení prim. klíče

- Z každého entitního typu vznikne jedno **relační schéma**
- Vzniklá relační schémata přeberou všechny **atributy** entitních typů
- Atributy, které tvoří klíč entitního typu, přejdou v **primární klíč** relace

Konceptuální model

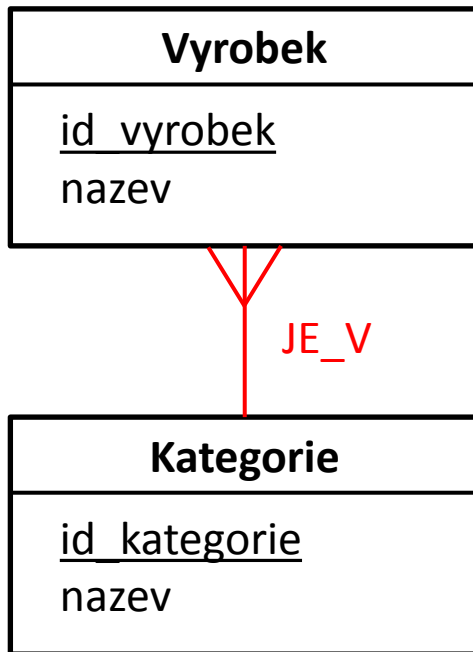


Relační datový model



Převod vztahů – vztah 1:N

Konceptuální model



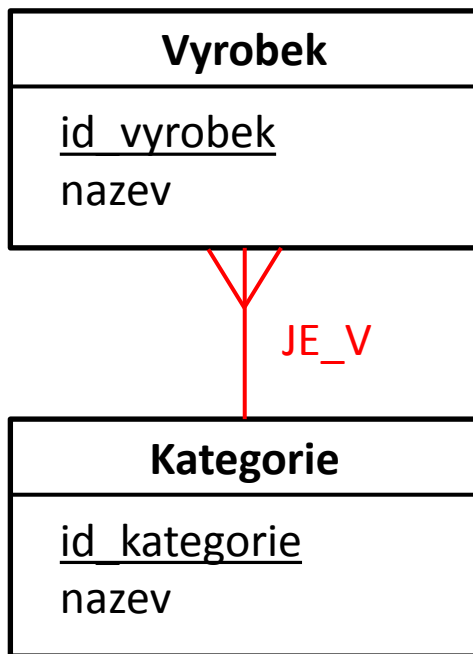
Relační datový model



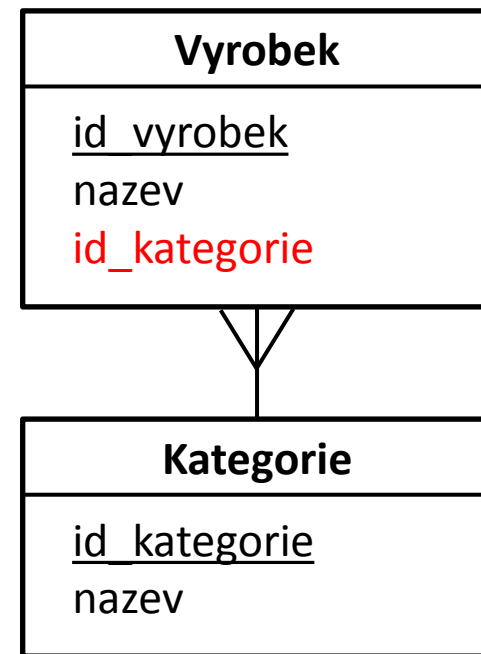
Převod vztahů – vztah 1:N

- Vztah 1:N lze převést velmi jednoduše **přidáním atributu** (popř. atributů) na stranu N
- Přidaný atribut (nebo atributy) vytvoří v jeho relaci **cizí klíč**.

Konceptuální model

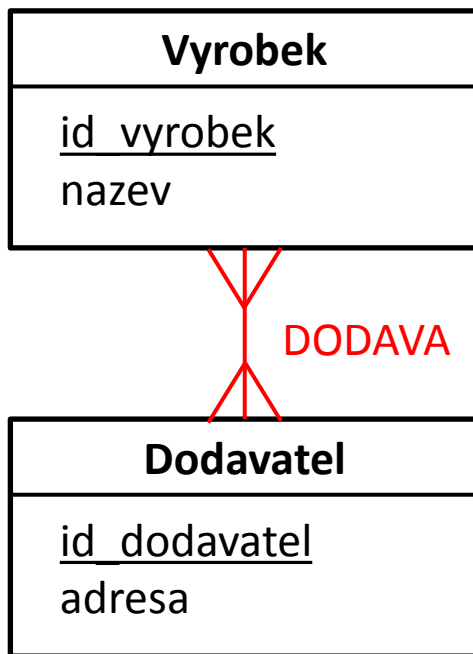


Relační datový model

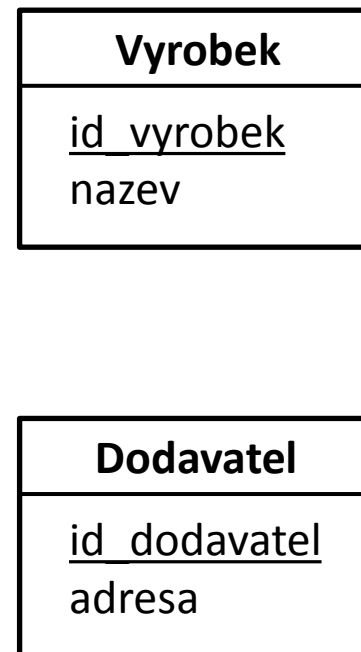


Převod vztahů – vztah M:N

Konceptuální model



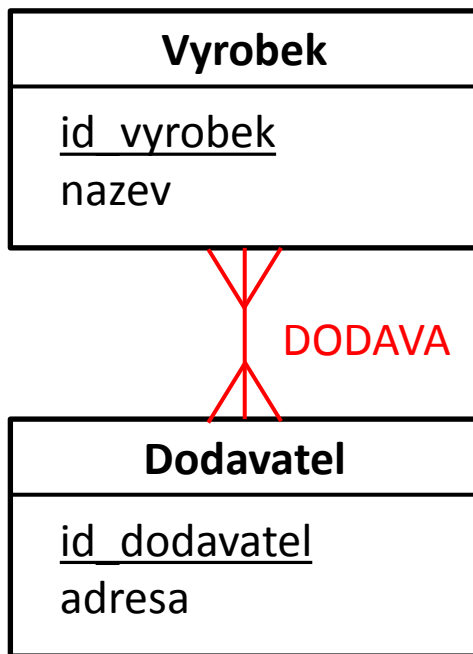
Relační datový model



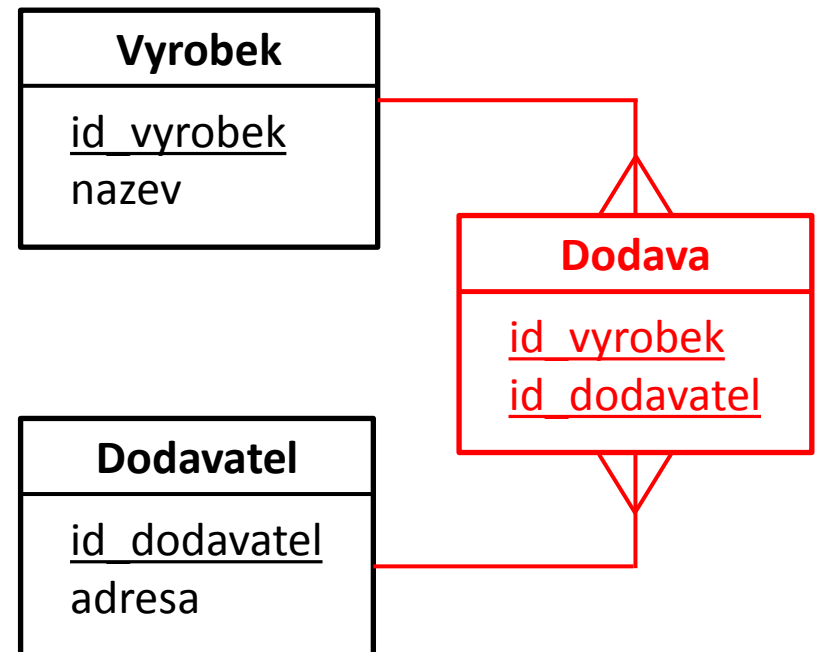
Převod vztahů – vztah M:N

- Vztah M:N je v relačním DM vždy řešen přidáním **rozkladové relace**
- Rozkladová relace obsahuje **všechny atributy, které jsou součástí primárního klíče** jedné nebo druhé relace ve vztahu M:N
- Tyto atributy jsou dohromady **primárním klíčem** a samostatně **cizími klíči**.

Konceptuální model

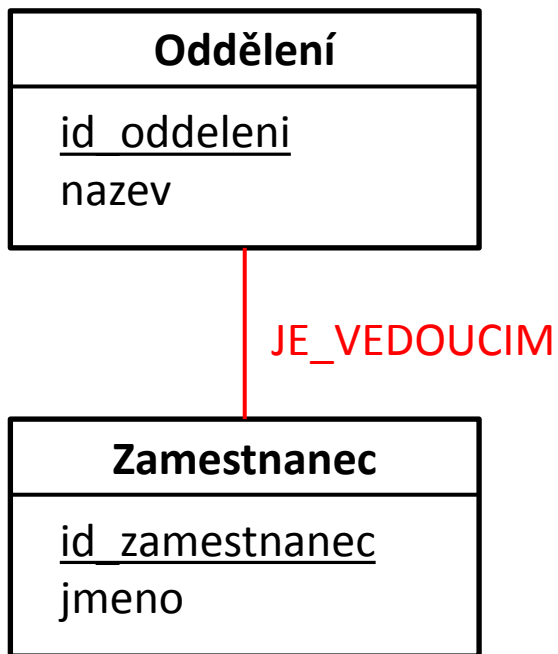


Relační datový model

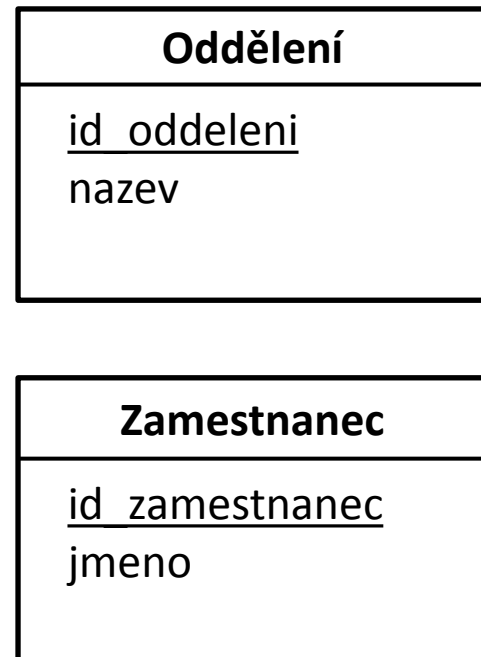


Převod vztahů – vztah 1:1

Konceptuální model



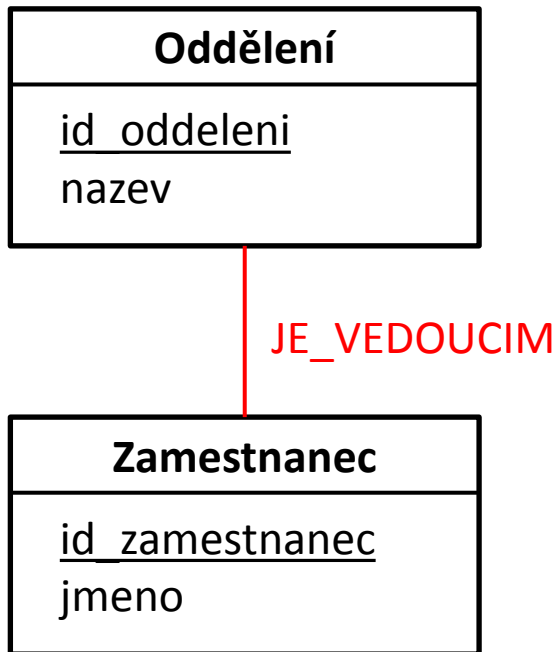
Relační datový model



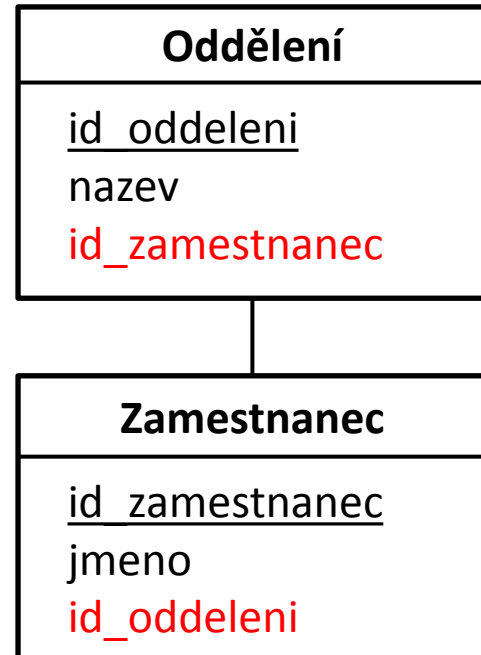
Převod vztahů – vztah 1:1

- Lze řešit **více způsoby**
- Teoreticky je „nejsprávnější“ varianta přidat **atributy do kříže**
- Prakticky se však vztah 1:1 řeší **nejčastěji jako 1:N** s tím, že se může použít nějaký dodatečný mechanismus (např. tzv. trigger), který N omezí na 1

Konceptuální model



Relační datový model



Datový slovník

Datový slovník

- Zkratka, jedná se o upřesnění vlastností atributů
- Součástí datového slovníku jsou nejčastěji: **datový typ, rozsah, klíč** (zda je atribut součástí primárního klíče), zda povolujeme hodnotu **null**, popř. jestli jsou na atribut kladeny nějaké další omezující požadavky – tzv. **integritní omezení**

Zamestnanec

Atribut	Datový typ	Rozsah	Klíč	Null	Další IO
id_zamestnanec	INT	4	ANO	NE	
jmeno	NVARCHAR	50	NE	NE	
prijmeni	NVARCHAR	50	NE	NE	
e_mail	NVARCHAR	40	NE	ANO	Musí obsahovat znak ,@'

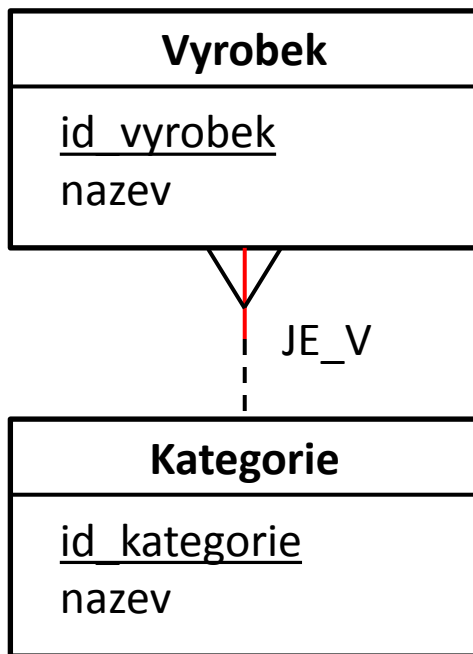
Datové typy

- **INT** Celé číslo
- **FLOAT** Desetinné číslo (plovoucí čárka)
- **DECIMAL(n,p)** Desetinné číslo (celkový počet míst a počet míst za des. oddělovačem)
- **CHAR(n)** Přesný počet znaků
- **VARCHAR(n)** Řetězec s proměnnou délkou
- **BIT** Hodnota 1 nebo 0
- **NCHAR(n)** To samé jako CHAR, ale UNICODE
- **NVARCHAR(n)** To samé jako VARCHAR, ale UNICODE
- **DATETIME** Datum a čas (razítko)

Množina dostupných datových typů je závislá na konkrétním SŘBD

Řešení povinnosti ve vztahu

Konceptuální model

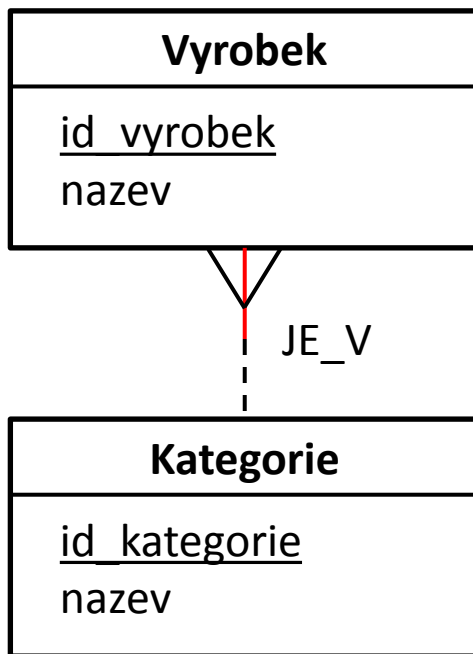


Relační datový model

Řešení povinnosti ve vztahu

- Povinnost ve vztahu 1:N ze strany N řešíme pomocí datového slovníku povolením nebo **zakázáním** hodnoty **null**.

Konceptuální model



Relační datový model

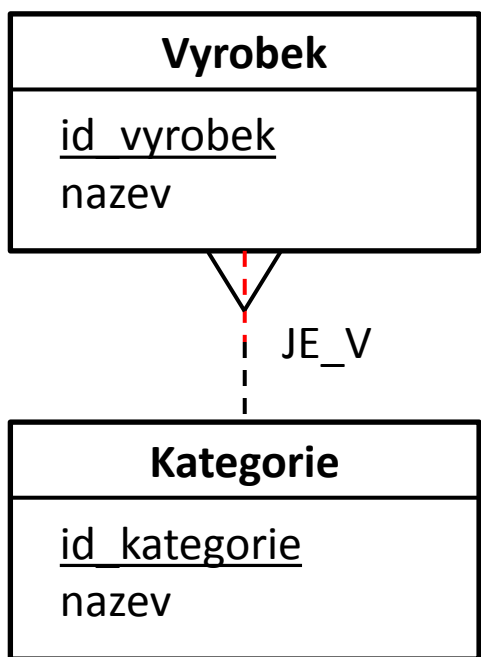
Vyrobek

Atribut			Null	
id_produkту			NE	
nazev			NE	
id_kategorie			NE	

Řešení povinnosti ve vztahu

- Povinnost ve vztahu 1:N ze strany N řešíme pomocí datového slovníku **povolením** nebo zakázáním hodnoty **null**.

Konceptuální model



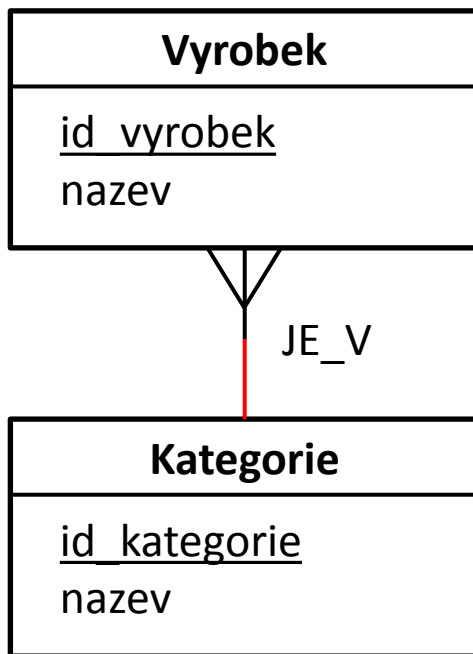
Relační datový model

Vyrobek

Atribut			Null	
id_produkту			NE	
nazev			NE	
id_kategorie			ANO	

Řešení povinnosti ve vztahu

Konceptuální model



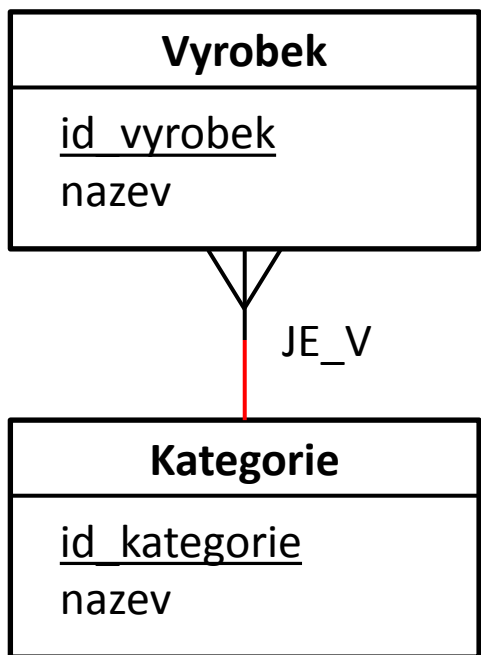
Relační datový model



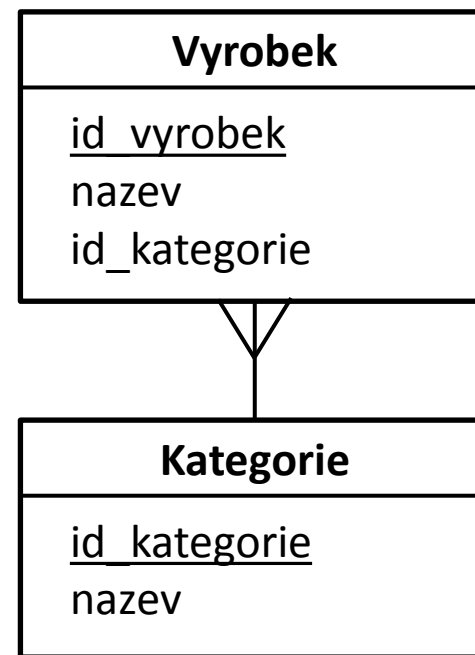
Řešení povinnosti ve vztahu

- U vztahu 1:N ze strany 1 **nelze povinnost v relačních databázích jednoduše vyřešit.**

Konceptuální model

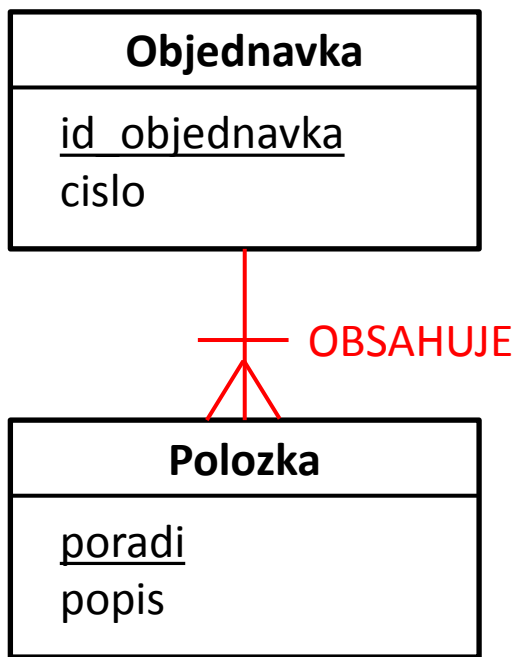


Relační datový model



Řešení slabých entitních typů a identifikujících vztahů

Konceptuální model

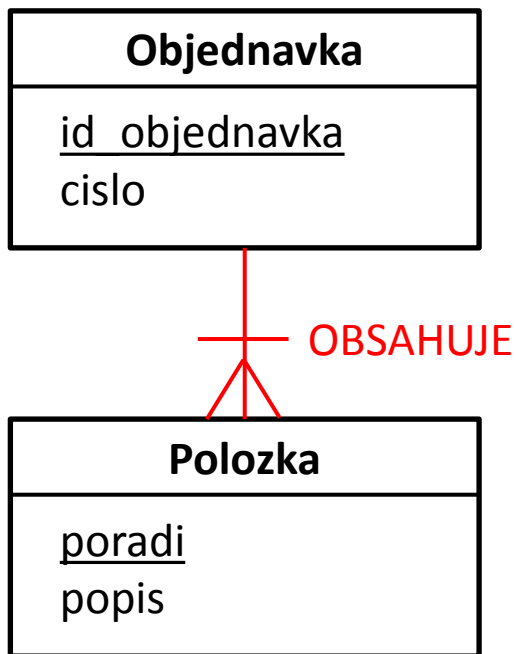


Relační datový model

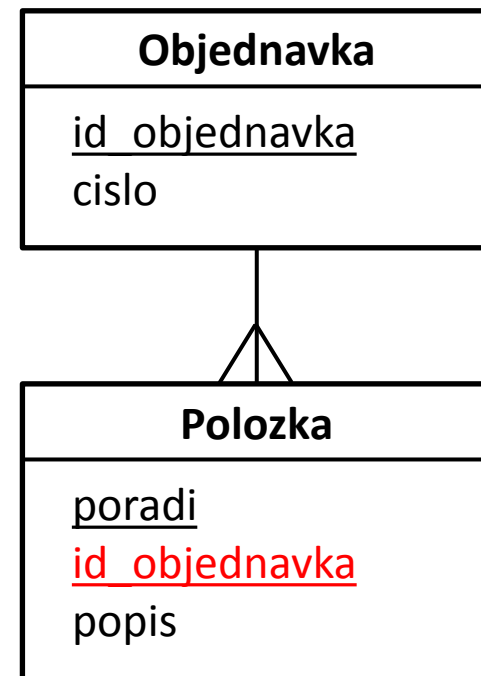
Řešení slabých entitních typů a identifikujících vztahů

- Úkolem identifikujícího vztahu je modelovat situaci, kdy je součástí klíče entitního typu atribut, který sám o sobě není součástí entitního typu.
- Identifikující vztah zajistí, že součástí relace bude **jeden (nebo více) atributů navíc**, které budou jednak **cizím klíčem** a jednak **součástí primárního klíče**.

Konceptuální model



Relační datový model



SQL jako jazyk pro definici a modifikaci dat

Rozdělení jazyků pro práci s daty

- **DDL** (Data Definition Language)
Vytváření datových struktur (prázdných tabulek)
- **DML** (Data Manipulation Language)
Vkládání, úprava a mazání dat
- **DQL** (Data Query Language)
Dotazování nad daty



SQL

DDL / DML / DQL

- Vyhledej nejmladšího zaměstnance
- Vytvoř tabulku s definovanými atributy
- Vrať všechny produkty
- Vlož záznam do tabulky
- Smaž obsah tabulky
- Smaž tabulku
- Přidej sloupec do tabulky

DDL / DML / DQL

- Vyhledej nejmladšího zaměstnance (**DQL**)
- Vytvoř tabulku s definovanými atributy (**DDL**)
- Vrať všechny produkty (**DQL**)
- Vlož záznam do tabulky (**DML**)
- Smaž obsah tabulky (**DML**)
- Smaž tabulku (**DDL**)
- Přidej sloupec do tabulky (**DDL**)

SQL jako DDL – vytvoření tabulky

SQL jako DDL – vytvoření tabulky

```
CREATE TABLE Osoba  
(  
    id_osoby INT PRIMARY KEY NOT NULL,  
    jmeno NVARCHAR(50) NOT NULL,  
    prijmeni NVARCHAR(50) NOT NULL,  
    zamestnanec BIT NOT NULL DEFAULT 0,  
    cislo_oddeleni INT NULL  
    FOREIGN KEY REFERENCES  
    Oddeleni(cislo_oddeleni)  
)
```

SQL jako DDL – vytvoření tabulky

```
CREATE TABLE Osoba
```

```
(
```

```
    id_osoby INT PRIMARY KEY NOT NULL,
```

```
    jmeno NVARCHAR(50) NOT NULL,
```

```
    prijmeni NVARCHAR(50) NOT NULL,
```

```
    zamestnanec BIT NOT NULL DEFAULT 0,
```

```
    cislo_oddeleni INT NULL
```

```
    FOREIGN KEY REFERENCES
```

```
    Oddeleni(cislo_oddeleni)
```

```
)
```


SQL jako DDL – vytvoření tabulky

Datový typ Primární klíč Povinný atribut Výchozí hodnota

```
CREATE TABLE Osoba  
(  
  id_osoby INT PRIMARY KEY NOT NULL,  
  jmeno NVARCHAR(50) NOT NULL,  
  prijmeni NVARCHAR(50) NOT NULL,  
  zamestnanec BIT NOT NULL DEFAULT 0,  
  cislo_oddeleni INT NULL  
  FOREIGN KEY REFERENCES  
  Oddeleni(cislo_oddeleni)  
)
```

Cizí klíč

Odstranění tabulky

Přidání sloupce

Úprava sloupce (např. rozsahu)

SQL jako DDL – další užitečné příkazy

Odstranění tabulky

```
DROP TABLE Osoba
```

Přidání sloupce

```
ALTER TABLE Osoba
```

```
ADD datum_narozeni DATETIME
```

Úprava sloupce (např. rozsahu)

```
ALTER TABLE Osoba
```

```
ALTER COLUMN jmeno NVARCHAR(100)
```

SQL jako DDL

CREATE ... Vytvářím něco nového (tabulku, sloupec, proceduru, funkci)

ALTER... Upravuji něco, co existuje

DROP... Odstraňuji něco, co existuje

DDL příkazy za nás ve většině případů sestavuje vhodný nástroj (např. SQL Management Studio).

SQL jako DML – vkládání dat

SQL jako DML – vkládání dat

Příkaz pro vkládání jednotlivých záznamů.

- Pozor, počet atributů uvedený za názvem tabulky musí odpovídat počtu uvedených hodnot.

```
INSERT INTO Osoba (id_osoby, jmeno, prijmeni)
VALUES (1, 'Petr', 'Lukáš')
```

SQL jako DML – úprava dat

SQL jako DML – úprava dat

Uvádíme název tabulky, jeden nebo více atributů, jejichž hodnotu chceme změnit (oddělujeme čárkami) a podmínku, pro které záznamy se má update provést.

```
UPDATE Osoba  
SET jmeno = 'Jakub', prijmeni = 'Pokusný'  
WHERE osoba_id = 1
```


SQL jako DML – úprava dat

Uvádíme název tabulky, jeden nebo více atributů, jejichž hodnotu chceme změnit (oddělujeme čárkami) a podmínku, pro které záznamy se má update provést.

```
UPDATE Osoba  
SET jmeno = 'Jakub', prijmeni = 'Pokusný'  
WHERE osoba_id = 1
```

 **Pozor, nezapomenout na WHERE**

SQL jako DML – mazání dat

SQL jako DML – mazání dat

Uvádíme název tabulky, nesmíme zapomenout na omezující podmínku.

```
DELETE FROM Osoba  
WHERE osoba_id = 1
```

 **Pozor, nezapomenout na WHERE**

Převod konceptuálního DM na relační

- Převod entitních typů, atributů, určení klíče
- Převod vztahu 1:N, M:N, 1:1
- Datový slovník
- Řešení povinnosti ve vztahu
- Identifikující vztahy

SQL jako jazyk pro definici a modifikaci dat

- Vytváření, úprava a mazání tabulky
- Vkládání, úprava a mazání dat

Shrnutí

Převod konceptuálního DM na relační

- **Převod entitních typů, atributů, určení klíče**
Triviální záležitost – atributy entitních typů přejdou v atributy relací
- **Převod vztahu 1:N, M:N, 1:1**
1:N řešíme přidáním atributu – cizího klíče na stranu N, M:N řešíme rozkladovou tabulkou, 1:1 řešíme např. atributy „do kříže“
- **Datový slovník**
Tabulka, kde uvádíme podrobně vlastnosti atributů
- **Řešení povinnosti ve vztahu**
V relačních db. řešíme pouze u vztahů 1:N ze strany N a to pomocí datového slovníku
- **Identifikující vztahy**
Řešíme podle jejich účelu, tj. přidáváme atribut – cizí klíč a tento atribut se stane součástí primárního klíče

SQL jako jazyk pro definici a modifikaci dat

- **Vytváření, úprava a mazání tabulky**
Příkazy CREATE TABLE, ALTER TABLE, DROP TABLE
- **Vkládání, úprava a mazání dat**
Příkazy INSERT, UPDATE, DELETE, pozor na podmínku WHERE!

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