

## Relační databázová technologie



# Přechod z konceptuálního na logický model

- Entitní typ → tabulka
- Atribut entitního typu → sloupec tabulky
- Vztah:
  - **vazba 1:1 a 1:N:**
    - Vztah → cizí klíč na straně N
    - atribut vztahu → sloupec tabulky na straně N
  - **vazba N:M:**
    - Vztah → „vazební“ tabulka
    - Atribut vztahu → sloupec vazební tabulky
- Doplnují se: *integritní omezení*
  - Omezení hodnot atributů, klíče, unikátnost dat, atp.

<http://www.sybase.com/products/modelingmetadata/powerdesigner>

## About PowerDesigner



### Sybase PowerDesigner (R)

Version 11.0.0.1363 Evaluation

Copyright (C) 1991-2004 Sybase, Inc. and its subsidiaries.

All rights reserved.

OK

# Relační databáze – primární klíč

**Klíč:** množina (může být i jednoprvková) atributů (sloupců), jež jednoznačně identifikuje danou entitu (záznam)

## Poznámky:

- Daný entitní typ (tabulka) *může mít více klíčů*.
  - Například (i) rodné číslo, (ii) osobní číslo zaměstnance, (iii) syntetický klíč.
- *Každá tabulka má alespoň jeden klíč*. V (opravdové) relační databázi tabulka běžně neobsahuje více řádků s týmiž hodnotami v jednotlivých sloupcích (pokud ano, ve většině případů je to chyba).

Osobní číslo	Rodné číslo	Jméno	Příjmení	Datum narození
101	8811010033	Josef	Novák	01.11.1988
	<del>8811010033</del>			
<del>101</del>				

Jméno	Příjmení	Datum narození
Josef	Novák	01.11.1988
<del>Josef</del>	<del>Novák</del>	<del>01.11.1988</del>

# Relační databáze - Cizí klíč

**Cizí klíč:** množina (může být i jednoprvková) atributů (sloupců), jejichž hodnota určuje hodnotu klíče jiné tabulky.

## Poznámky:

1. Datový typ jednotlivých atributů musí být kompatibilní s datovým typem odpovídajících atributů klíče referencované tabulky.
2. Cizí klíč není (nemusí být) v dané tabulce klíčem.
3. Vztah mezi entitami se realizuje asociativní vazbou

**cizí klíč -> primární klíč.**

4. Vzhledem k tomu, že tabulka může mít více klíčů, je dobré jeden z nich zvolit a systematicky ho používat k realizaci vztahů s ostatními tabulkami. Takto zvolený klíč nazýváme klíčem **primárním**.

Osobní číslo	Rodné číslo	Jméno	Příjmení	Datum narození
101	8811010033	Josef	Novák	01.11.1988

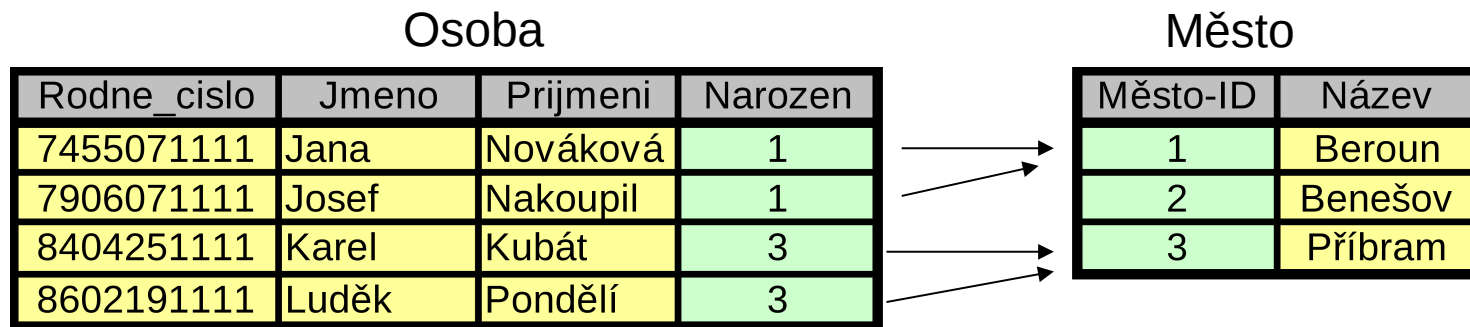
Osobní číslo	Popis	Ráže
101	Kalašnikov	7,65

Rodné číslo	SPZ	Značka
8811010033	BE-04-30	Škoda

# Referenční integrita

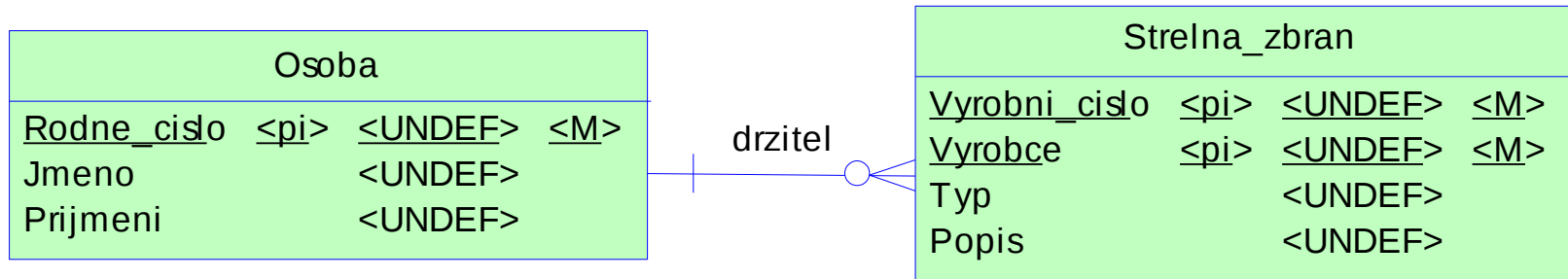
Mějme dvě tabulky A a B takové, že tabulka B obsahuje cizí klíč odkazující to tabulky A.

**Referenční integrita** je integritní omezení, které v databázi definuje (zajišťuje) vztah (souvislost) mezi cizím a primárním klíčem.



V databázi nesmí dojít k tomu, že v nějakém řádku tabulky B odkazuje hodnota cizího klíče na neexistující řádek v tabulce A. V takovém případě jsou **data nekonzistentní** a v databázi bude docházet k problémům.

# Referenční integrita



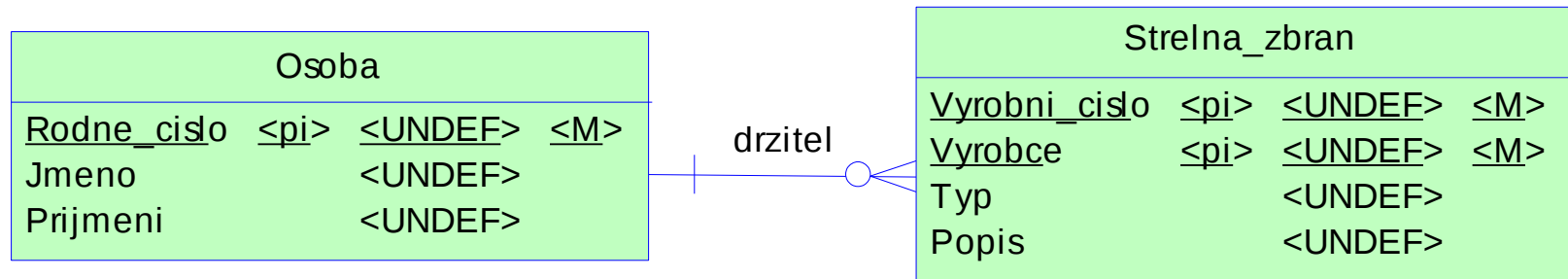
Osoba		
<i>Rodne_cislo</i>	<i>Jmeno</i>	<i>Prijmeni</i>
1	Josef	Novák
2	Jaroslav	Novotný

Zbran				
<i>Drzitel</i>	<i>Vyrobni_cislo</i>	<i>Vyrobce</i>	<i>Typ</i>	<i>Popis</i>
1	101	Zbrojovka	Vzduchovka	Slavie
2	202	Zbrojovka	Kalashnikov	Vzor 57

Jak může dojít k porušení referenční integrity?

- Zrušíme záznam v tabulce *Osoba* => cizí klíč příslušného řádku tabulky *Zbran* bude odkazovat na neexistující řádek tabulky *Osoba*.
- Změníme hodnotu primárního klíče některého řádku v tabulce *Osoba* => cizí klíč příslušného řádku tabulky *Zbran* bude odkazovat na neexistující řádek tabulky *Osoba*.

# Referenční integrita - Implementace



```
create table Strelna_zbran (  
Vyrobní_cislo      CHAR(10)    not null,  
Vyrobcce          CHAR(10)    not null,  
Rodne_cislo       CHAR(10)    not null,  
Typ               CHAR(10)    null,  
Popis             CHAR(10)    null,  
constraint PK_STRELNA_ZBRAN primary key (Vyrobní_cislo, Vyrobcce)  
constraint FK_STRELNA__DRZITEL_OSOBA foreign key (Rodne_cislo)  
  references Osoba (Rodne_cislo)  
  on delete restrict on update restrict;  
);
```



# Referenční integrita - Implementace

Reference Properties - drzitel (drzitel)

Child Table: Strelna\_zbran

Parent Table: Osoba

Constraint name: **FK\_STRELNA\_DRZITEL\_OSOBA**

Implementation:

- No action
- Restrict**
- Cascade
- Set null
- Set Default

Mandatory parent

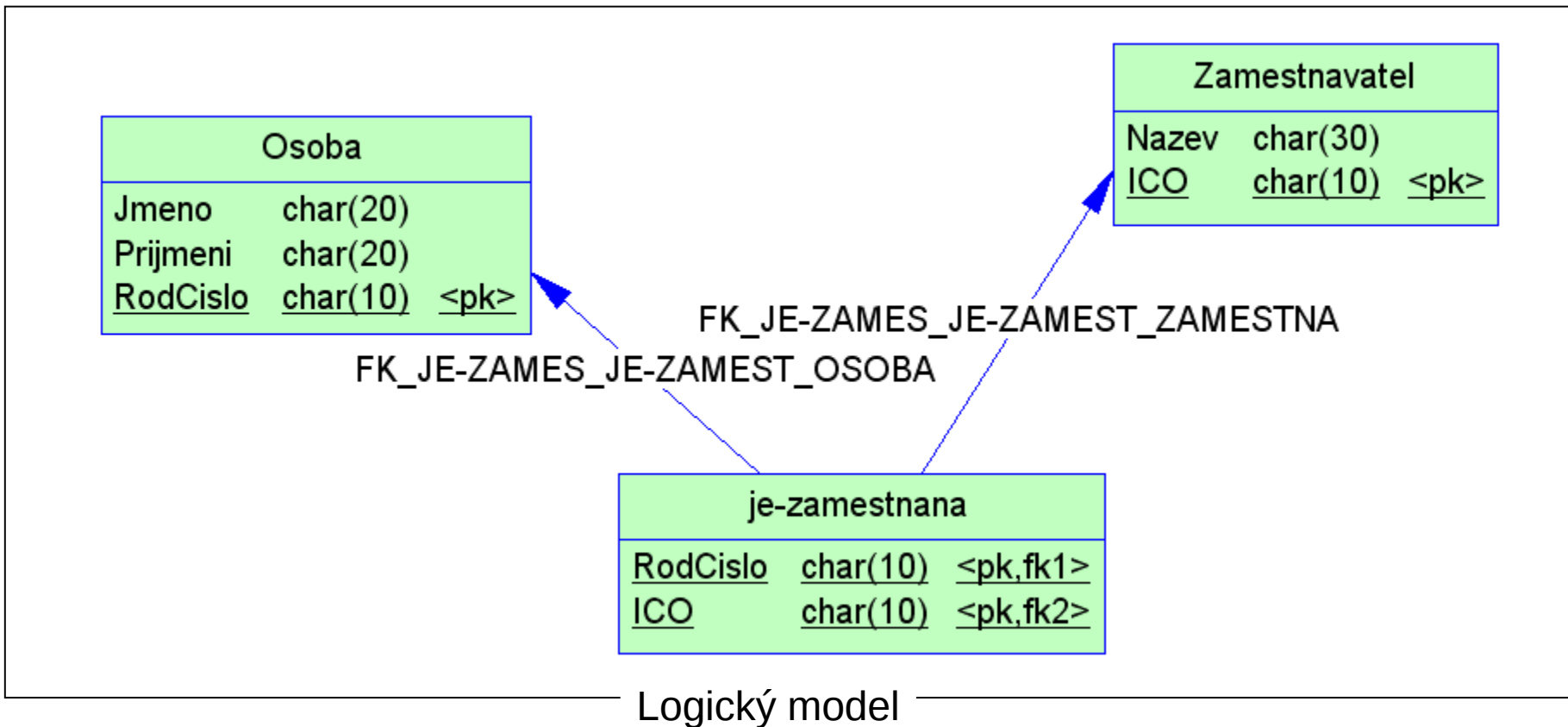
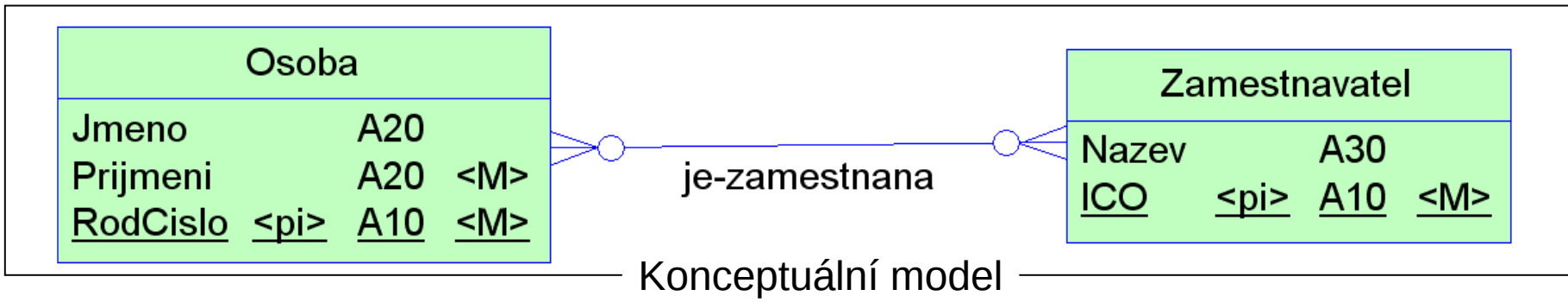
Change parent allowed

Check on commit

Buttons: OK, Storno, Použít, nápověda

**constraint FK\_STRELNA\_DRZITEL\_OSOBA foreign key (Rodne\_cislo) references Osoba (Rodne\_cislo) on delete restrict on update restrict;**

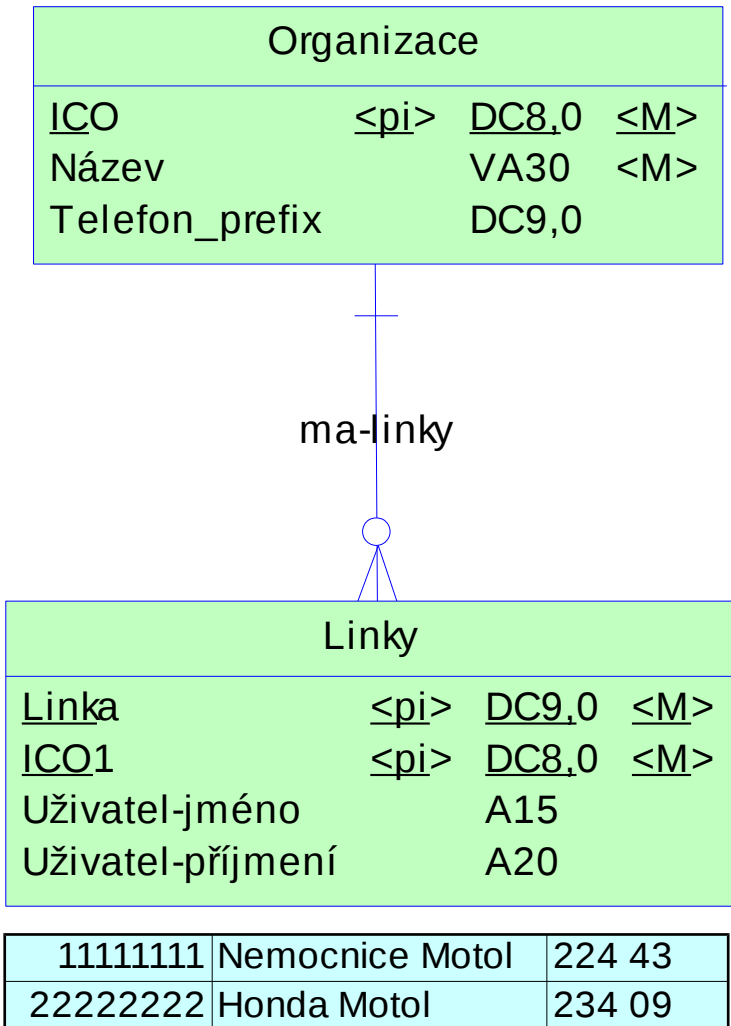
# Transformace N:M vazby na 1:N vazby



# Slabý entitní typ

**Slabý entitní typ** modeluje situaci, kdy jeden entitní typ je *existenčně závislý* na jiném entitním typu.

Příklad: Telefonní linky v organizaci

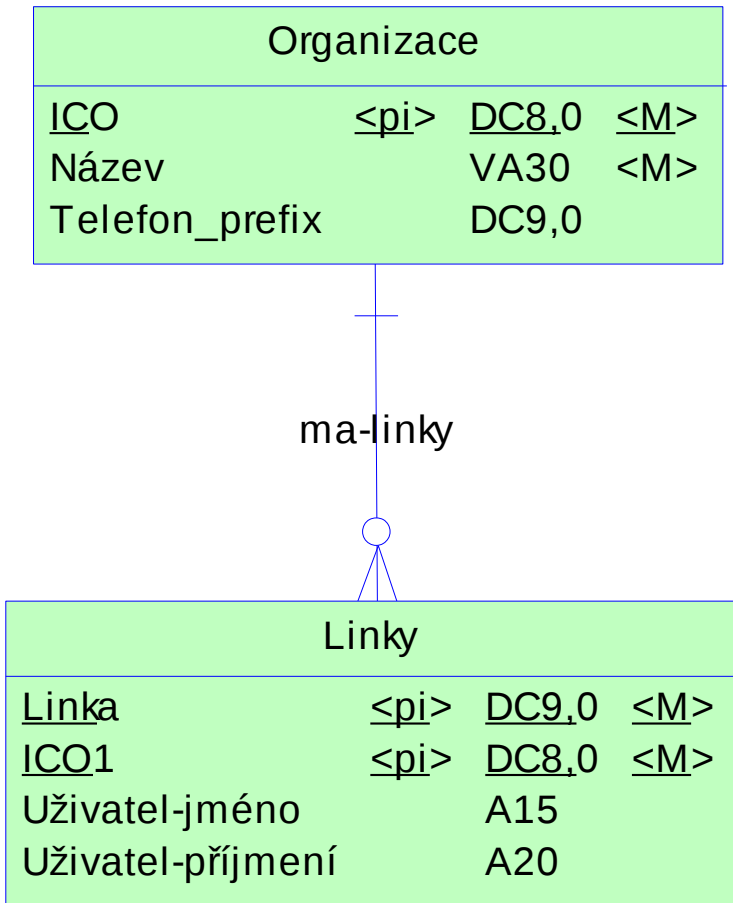


11111111	1111	Ústředna
11111111	2920	Kardiocentrum
11111111	2101	Novorozenecké oddělení
22222222	1111	Recepce
22222222	6690	Prodej nových vozů

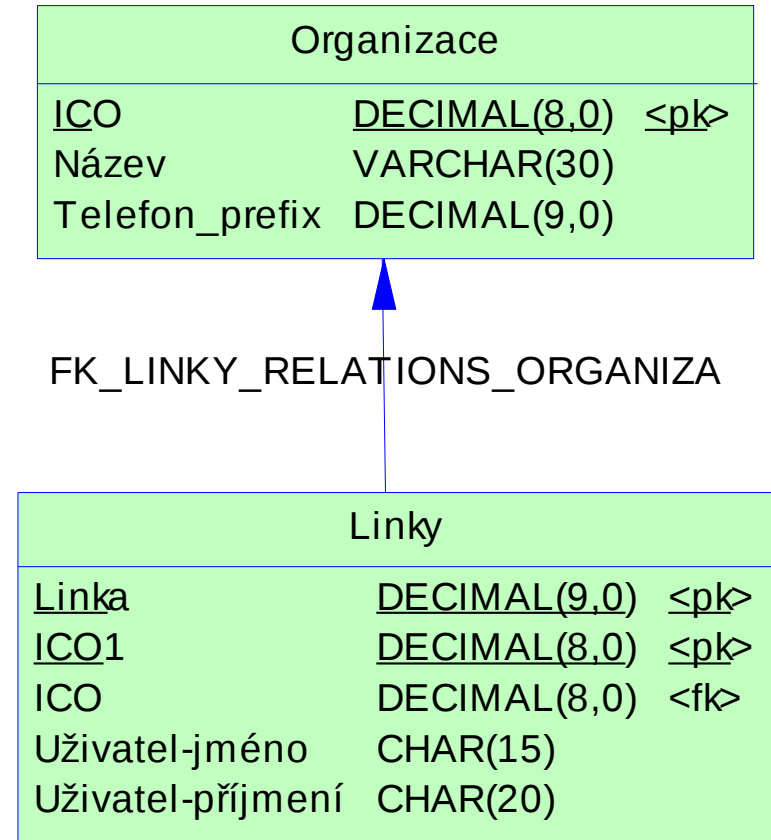
Cizí klíč

Primární klíč

# Slabý entitní typ – logický model - motivace



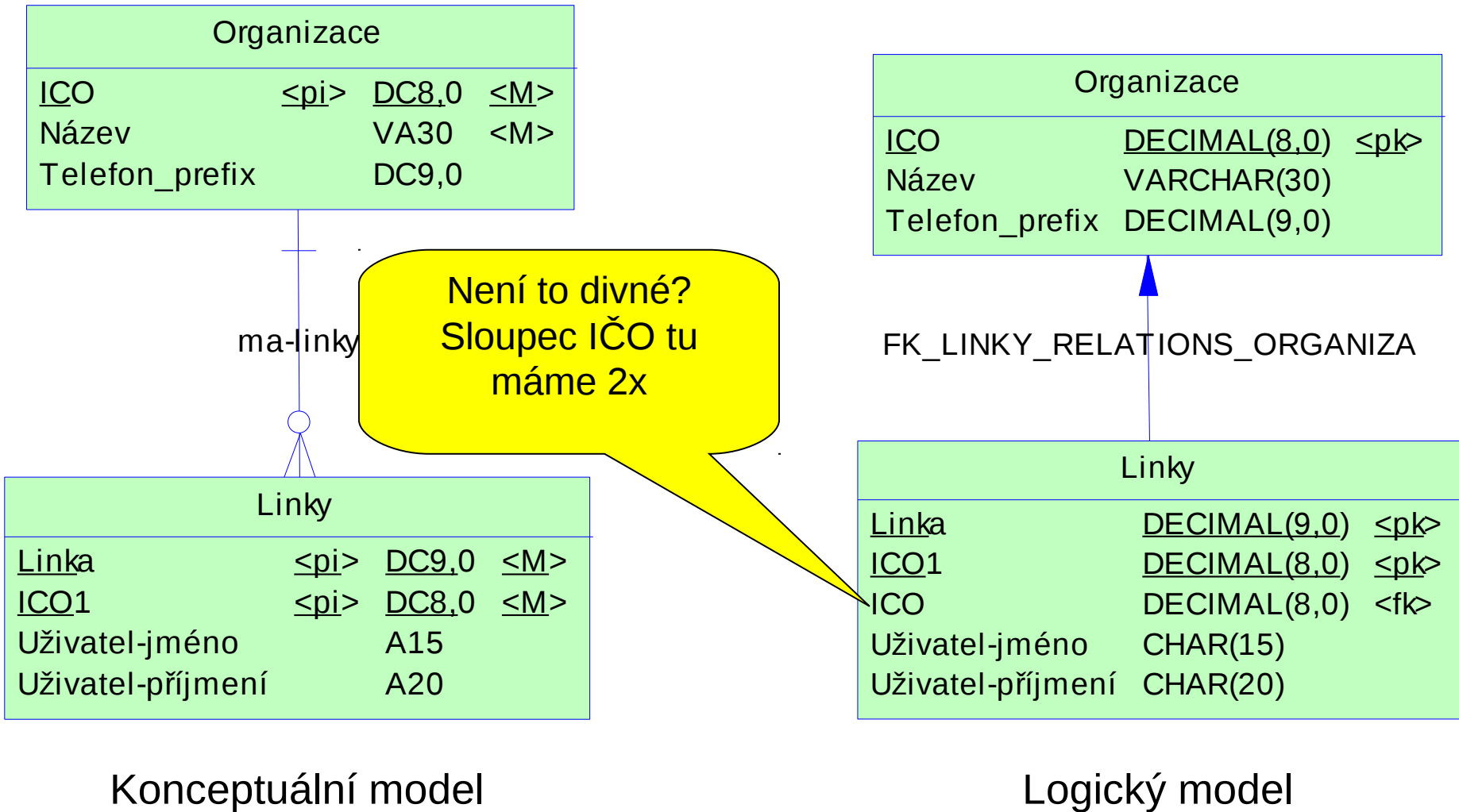
Konceptuální model



Logický model

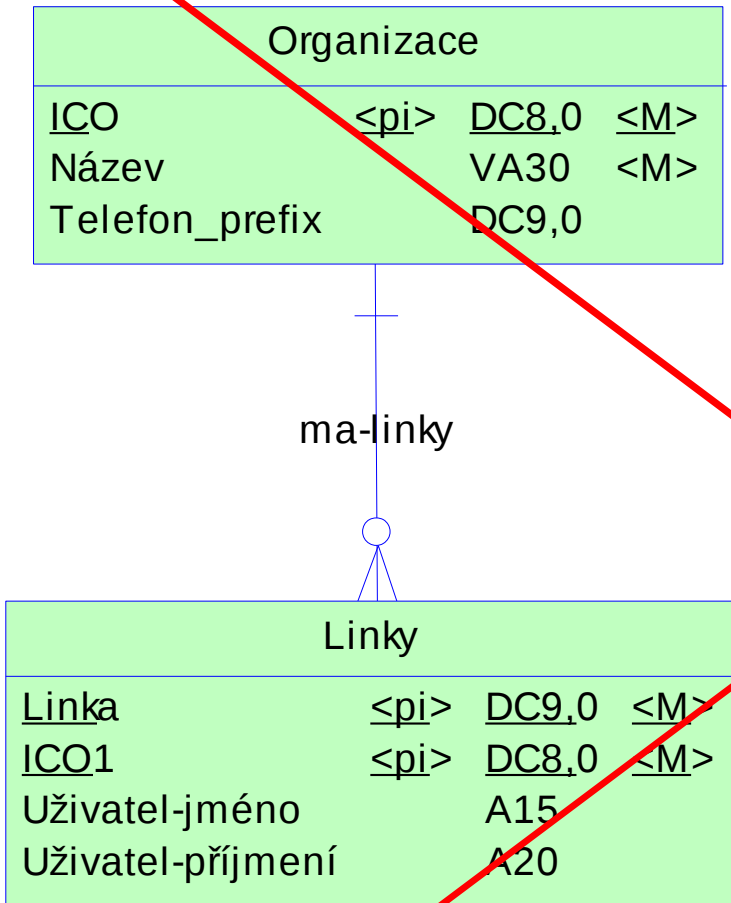
Při přidání vazby dojde k přidání cizího klíče, ale....

# Slabý entitní typ – logický model - motivace

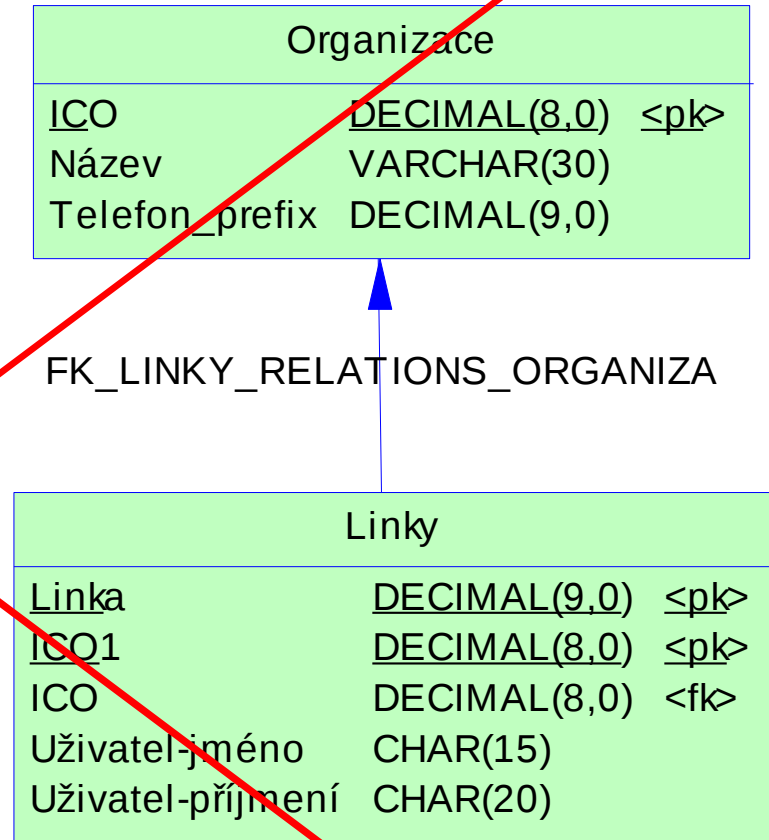


Při přidání vazby dojde k přidání cizího klíče, ale....

# Slabý entitní typ – logický model - motivace



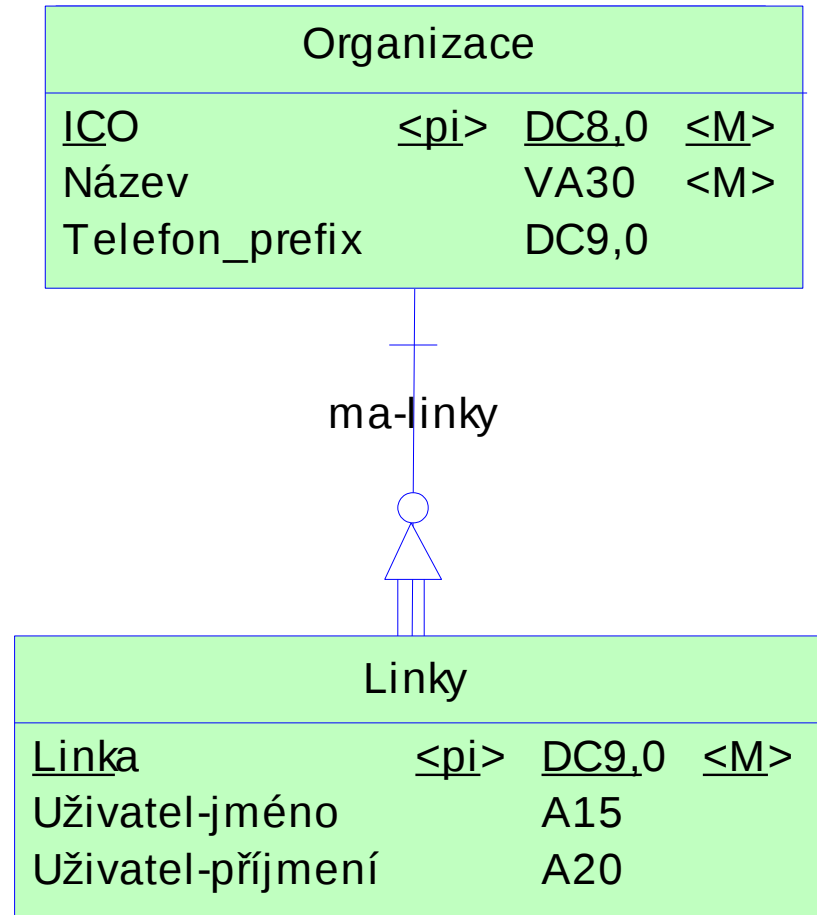
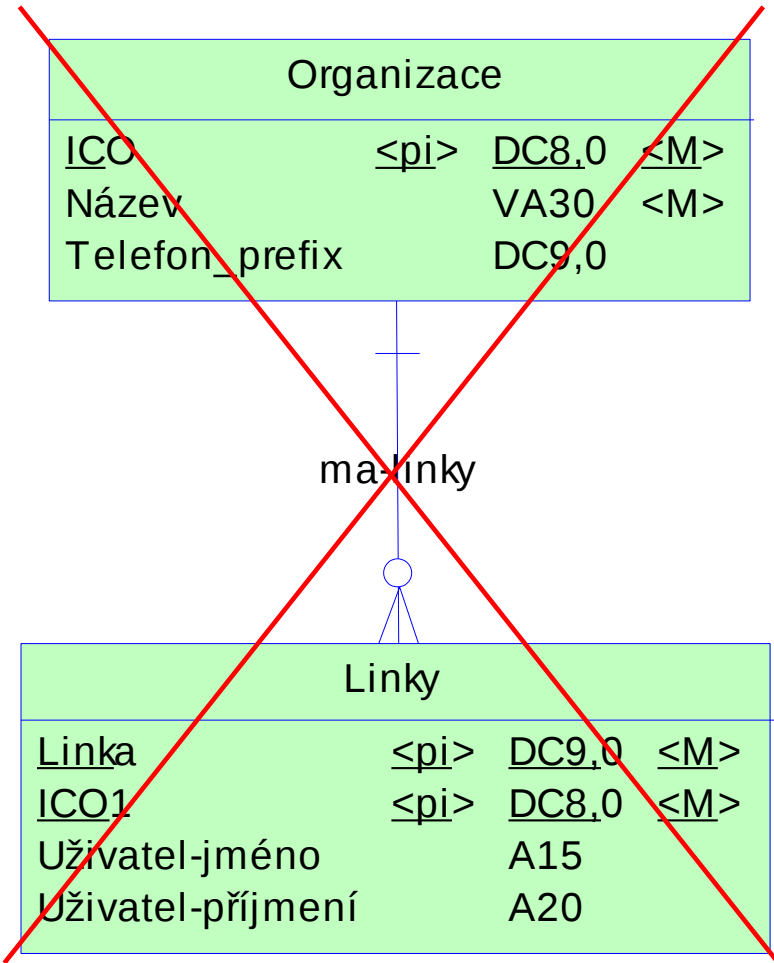
Konceptuální model



Logický model

Při přidání vazby dojde k přidání cizího klíče, ale....

# Slabý entitní typ – správná vazba



# Slabý entitní typ – správná vazba

Řídící entitní typ

Organizace			
<u>ICO</u>	<pi>	<u>DC8,0</u>	<M>
Název		VA30	<M>
Telefon_prefix		DC9,0	

Organizace			
<u>ICO</u>	<pi>	<u>DC8,0</u>	<M>
Název		VA30	<M>
Telefon_prefix		DC9,0	

ma-linky

ma-linky

Slabý entitní typ

Identifikující vztah (vazba)

Linky			
<u>Linka</u>	<pi>	<u>DC9,0</u>	<M>
<u>ICO1</u>	<pi>	<u>DC8,0</u>	<M>
Uživatel-jméno		A15	
Uživatel-příjmení		A20	

Linky			
<u>Linka</u>	<pi>	<u>DC9,0</u>	<M>
Uživatel-jméno		A15	
Uživatel-příjmení		A20	

11111111	Nemocnice Motol	224 43
22222222	Honda Motol	234 09

11111111	1111	Ústředna
11111111	2920	Kardiocentrum
11111111	2101	Novorozenecké oddělení
22222222	1111	Recepce
22222222	6690	Prodej nových vozů



# Slabý entitní typ – Logický model

Organizace			
<u>ICO</u>	<pi>	DC8,0	<M>
Název		VA30	<M>
Telefon_prefix		DC9,0	

ma-linky



Linky

<u>Linka</u>	<pi>	DC9,0	<M>
Uživatel-jméno		A15	
Uživatel-příjmení		A20	

Organizace		
<u>ICO</u>	<u>DECIMAL(8,0)</u>	<pk>
Název	VARCHAR(30)	
Telefon_prefix	DECIMAL(9,0)	

FK\_LINKY\_MA-LINKY\_ORGANIZA

Linky

<u>ICO</u>	<u>DECIMAL(8,0)</u>	<pk,fk>
<u>Linka</u>	<u>DECIMAL(9,0)</u>	<pk>
Uživatel-jméno	CHAR(15)	
Uživatel-příjmení	CHAR(20)	

# Slabý entitní typ – SQL pro vytvoření

Organizace		
<u>ICO</u>	DECIMAL(8,0)	<pk>
Název	VARCHAR(30)	
Telefon_prefix	DECIMAL(9,0)	

```
create table Linky (  
  ICO          DECIMAL(8,0)    not null,  
  Linka        DECIMAL(9,0)    not null,  
  Uzivatel-jmeno CHAR(15)      null,  
  Uzivatel-prijmeni CHAR(20)    null,  
  constraint PK_LINKY primary key (ICO, Linka)  
);
```

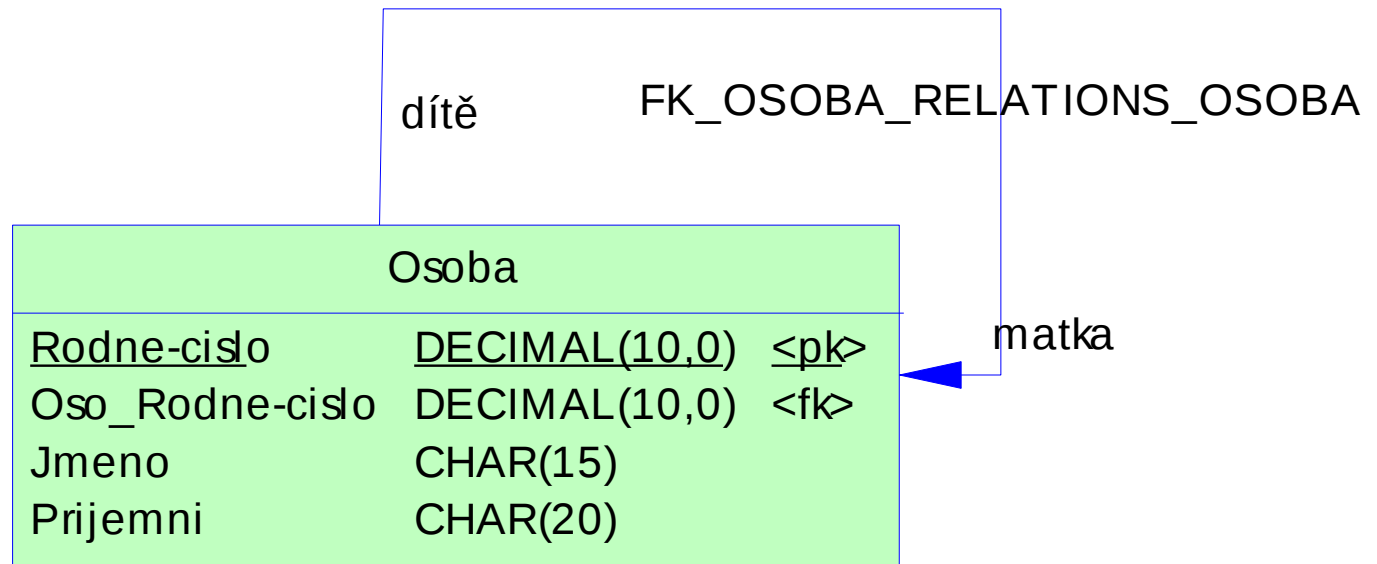
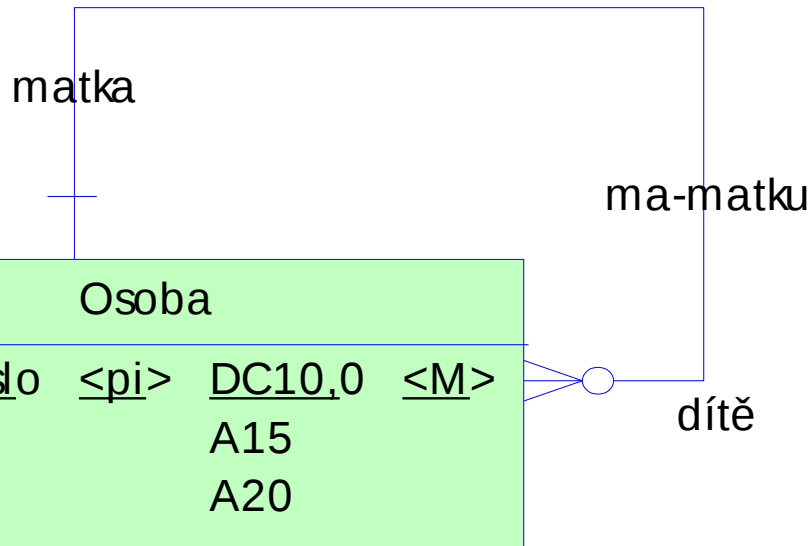
```
create table Organizace (  
  ICO          DECIMAL(8,0)    not null,  
  Nazev        VARCHAR(30)     not null,  
  Telefon_prefix DECIMAL(9,0)  null,  
  constraint PK_ORGANIZACE primary key (ICO)  
);
```

```
alter table Linky  
add constraint "FK_LINKY_MA-LINKY_ORGANIZA" foreign key (ICO)  
references Organizace (ICO)  
on delete restrict on update restrict;
```

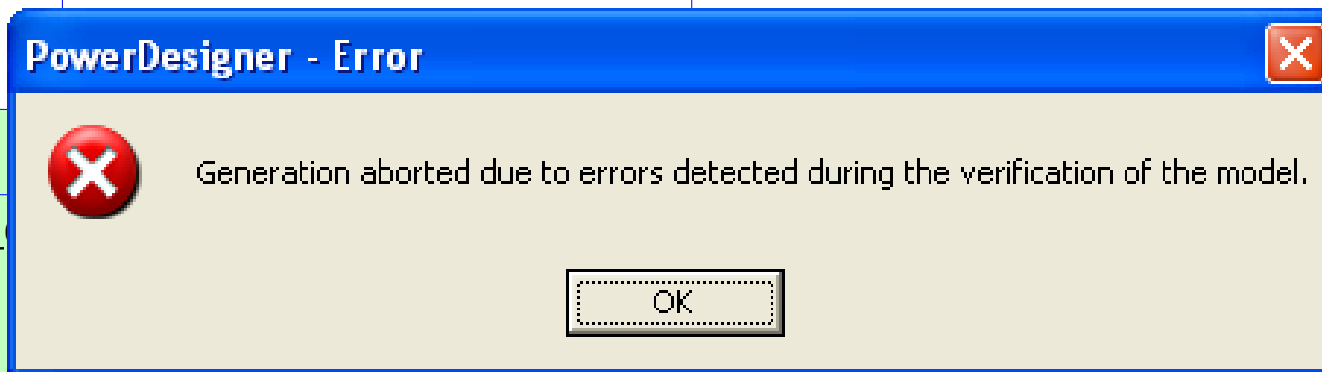
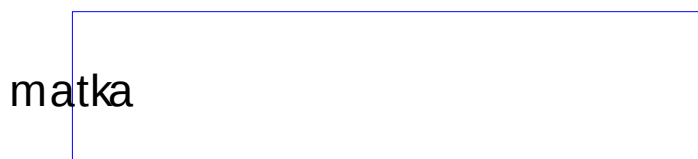
FK\_LINKY\_MA-LINKY\_ORGANIZA

Linky		
<u>ICO</u>	<u>DECIMAL(8,0)</u>	<pk,fk>
<u>Linka</u>	<u>DECIMAL(9,0)</u>	<pk>
Uživatel-jméno	CHAR(15)	
Uživatel-příjmení	CHAR(20)	

# Reflexivní vztah



# Reflexivní vztah

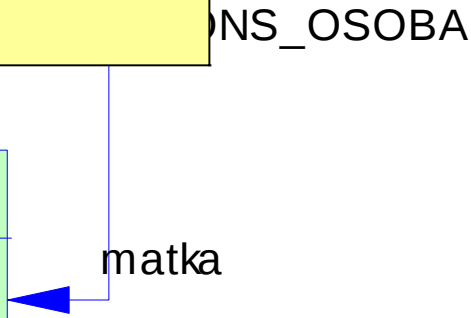


Rodne-cislo
Jmeno
Prijemni

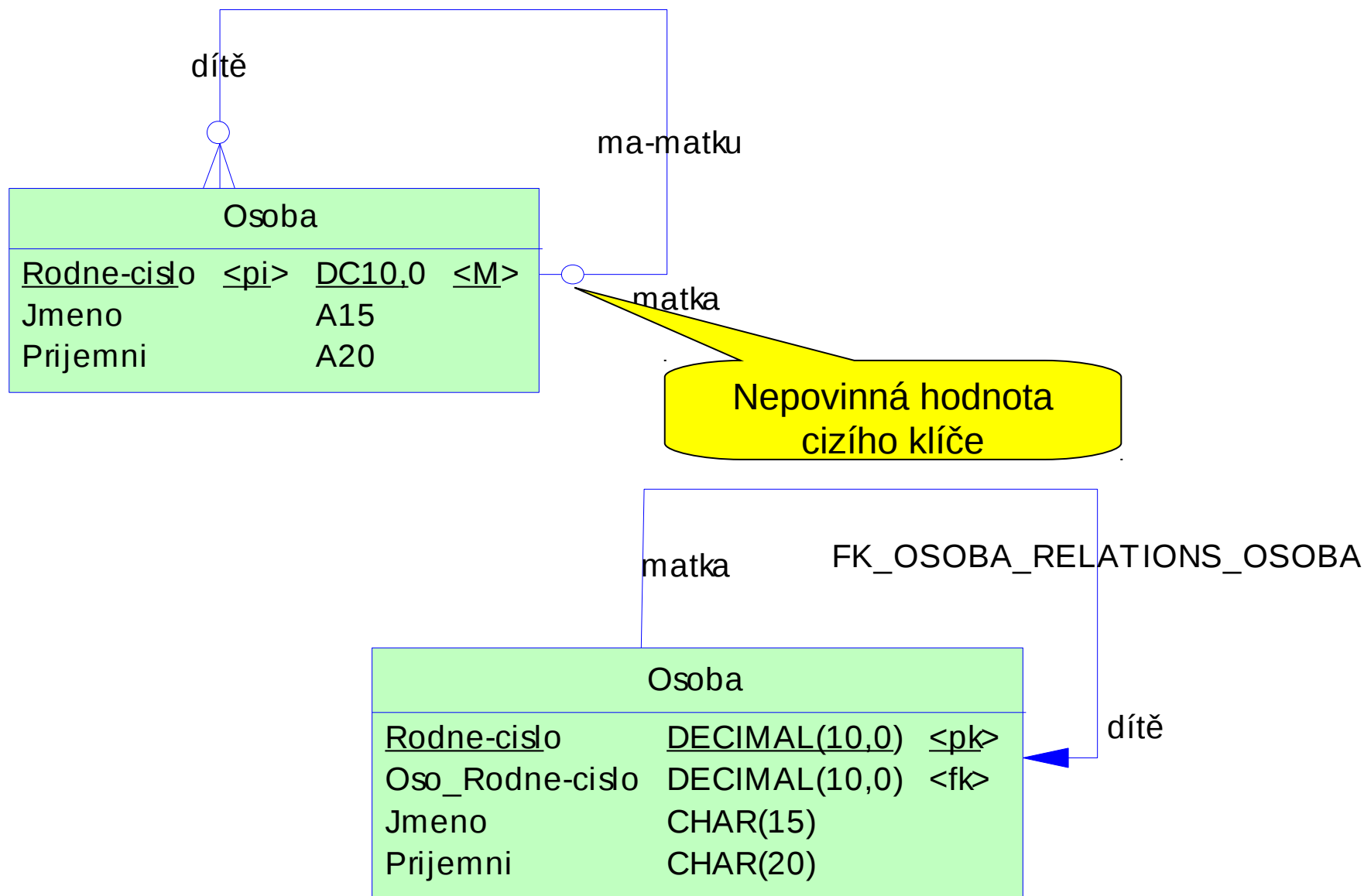
**Reflexive mandatory reference**

A reflexive reference exists should not have a mandatory parent which could lead to inconsistent joins.

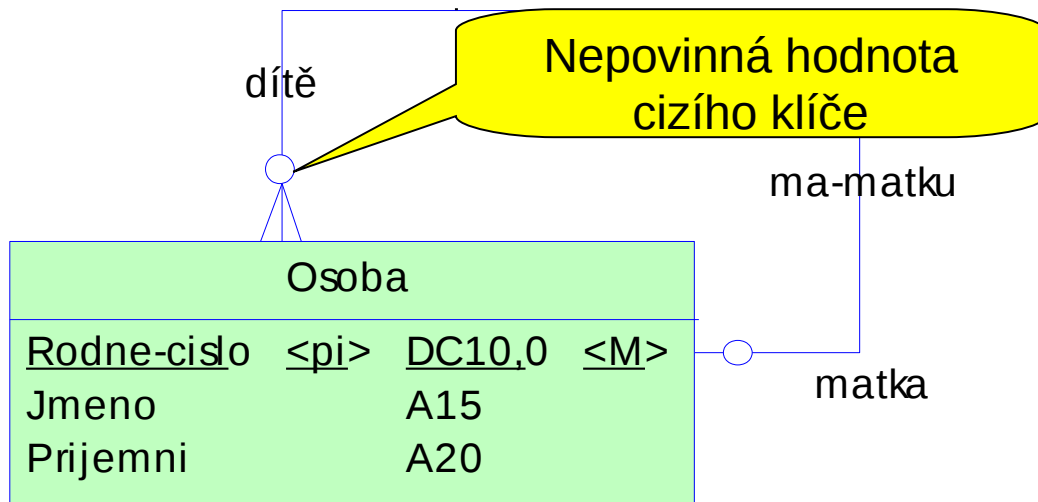
Osoba		
<u>Rodne-cislo</u>	<u>DECIMAL(10,0)</u>	<pk>
Oso_Rodne-cislo	DECIMAL(10,0)	<fk>
Jmeno	CHAR(15)	
Prijemni	CHAR(20)	



# Reflexivní vztah



# Reflexivní vztah



```
create table Osoba (  
Rodne_cislo          DECIMAL(10,0)  not null,  
Oso_Rodne_cislo     DECIMAL(10,0)  null,  
Jmeno                CHAR(15)        null,  
Prijmeni             CHAR(20)        null,  
constraint PK_OSOBA primary key (Rodne_cislo)  
);
```

```
alter table Osoba  
add constraint FK_OSOBA_RELATIONS_OSOBA foreign key (Oso_Rodne_c  
references Osoba ("Rodne-cislo")  
on delete restrict on update restrict;
```