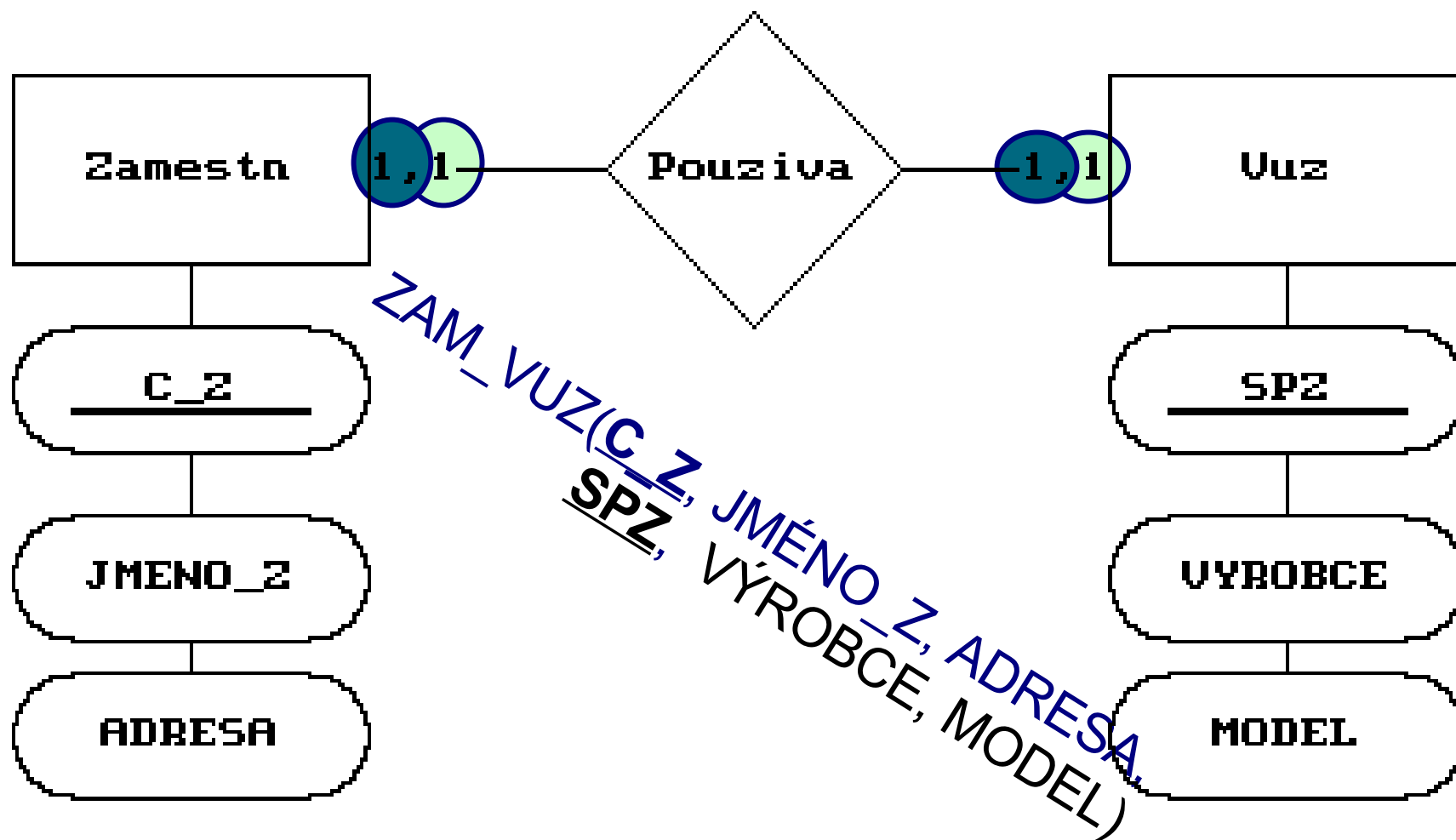


Souvislost E-R schématu s relačním

- **Reprezentace silného entitního typu**
 - entitnímu typu odpovídá schéma relace
 - atributům entitního typu odpovídají atributy relace
 - primární klíč bude tvořen atributy odpovídajícími atributům identifikačního klíče entitního typu
- **Reprezentace vztahů**
 - opět musíme použít relaci
 - vyjádření je podpořeno referenční integritou

Reprezentace vztahu 1:1 (obě účasti povinné)



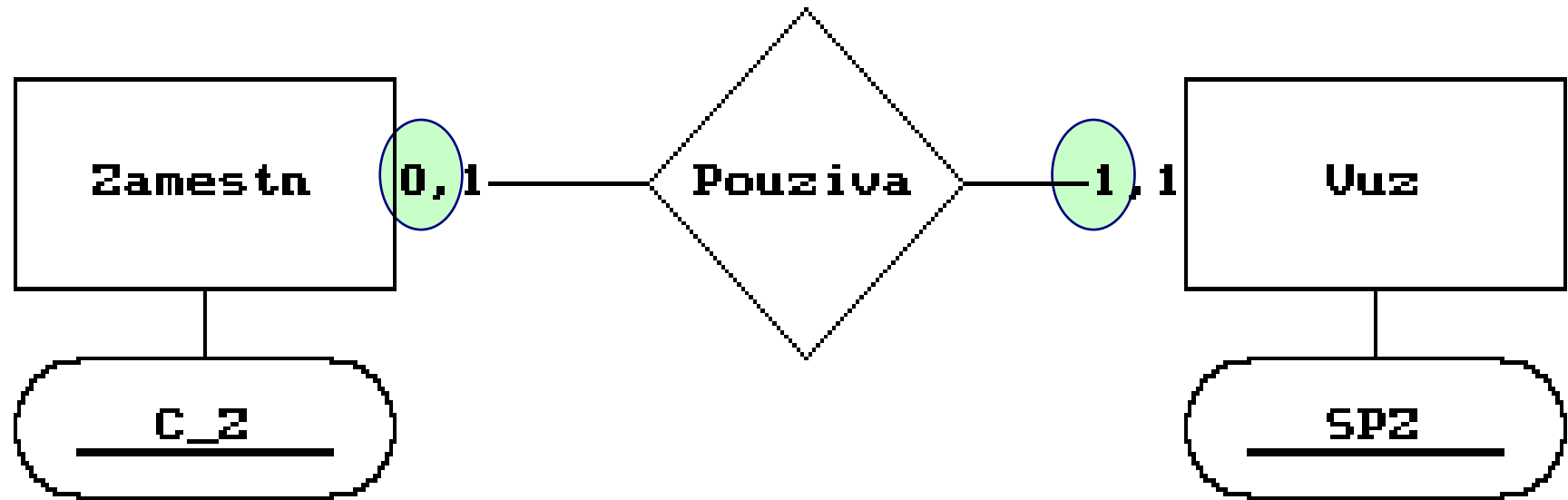
Reprezentace vztahu 1:1 (obě účasti povinné)

```
CREATE TABLE ZAM_VUZ
(C_Z          Number          PRIMARY KEY,
 JMENO_Z     Char Varying(25) NOT NULL,
 ADRESA      Char Varying(25) NOT NULL,
 SPZ         Number          NOT NULL UNIQUE,
 VYROBCE     Char Varying(10) NOT NULL,
 MODEL       Char Varying(10) NOT NULL);
```

```
CREATE VIEW VUZ
AS
SELECT SPZ, VYROBCE, MODEL
FROM ZAM_VUZ;
```

```
CREATE VIEW ZAMESTN
AS
SELECT C_Z, JMENO_Z, ADRESA
FROM ZAM_VUZ;
```

Reprezentace vztahu 1:1 (1 účast nepovinná)



ZAMESTN (C_Z, JMENO_Z, ...)

VUZ(SPZ, ..., C_Z)

VUZ[C_Z] \subseteq ZAMESTN [C_Z]

Reprezentace vztahu 1:1 (1 účast nepovinná)

```
CREATE TABLE ZAMESTN
```

```
(C_Z          Number          PRIMARY KEY ,  
 JMENO_Z     Char Varying(25) NOT NULL,  
 ADRESA      Char Varying(25) NOT NULL );
```

```
CREATE TABLE VUZ
```

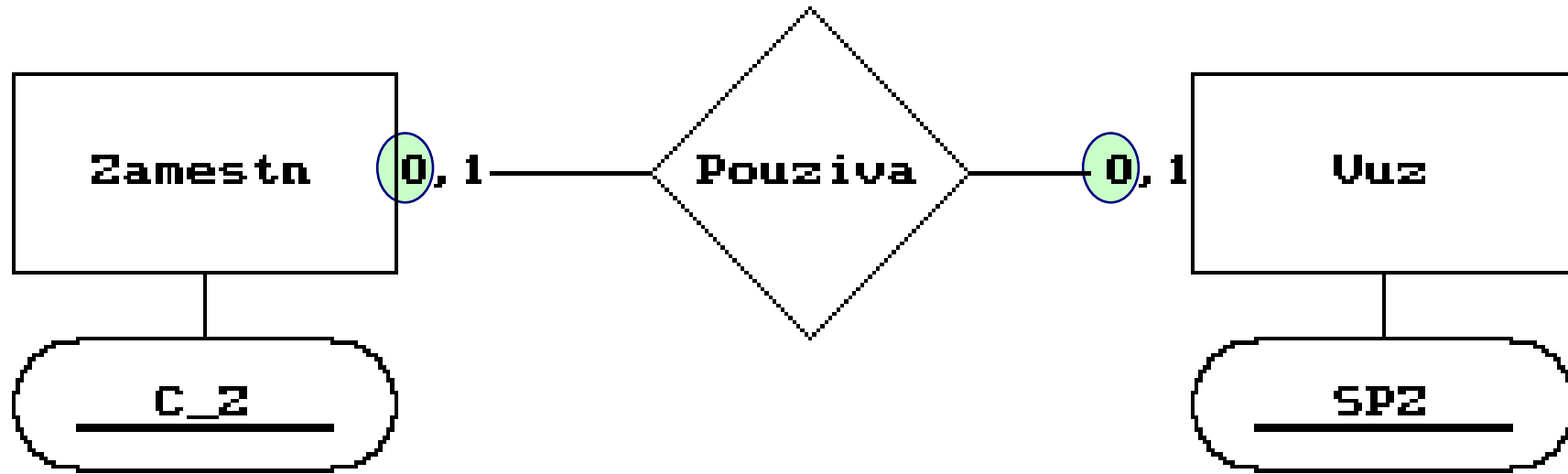
```
(SPZ          Number          PRIMARY KEY,  
 VYROBCE     Char Varying(10) NOT NULL,  
 MODEL       Char Varying(10) NOT NULL);
```

```
ALTER TABLE VUZ
```

```
ADD (V_C_Z    Number NOT NULL UNIQUE  
      REFERENCES ZAMESTN);
```

podchycení povinné účasti a kardinality 1

Reprezentace vztahu 1:1 (obě účasti nepovinné)



ZAMĚSTN (C_Z, JMÉNO_Z, ADRESA)

VŮZ(SPZ, VÝROBCE, MODEL)

POUZIVA(C_Z, SPZ)

10: $POUZIVA[C_Z] \subseteq ZAMĚSTN [C_Z]$

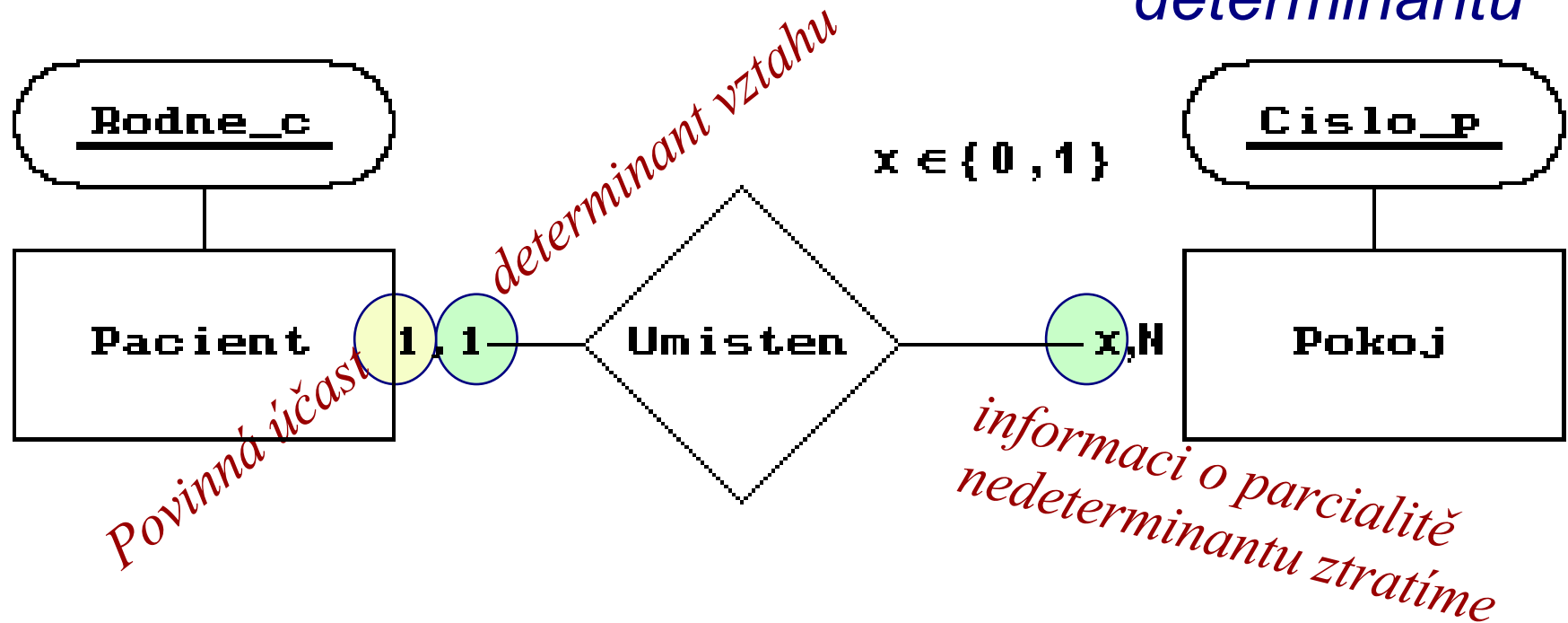
$POUZIVA[SPZ] \subseteq VŮZ[SPZ]$

*dvě entitní
relace*

vztahová relace

Reprezentace vztahu 1:N

*povinná účast
determinantu*



PACIENT(RODNE_C, PRIJMENI, ..., ČÍSLO_P)

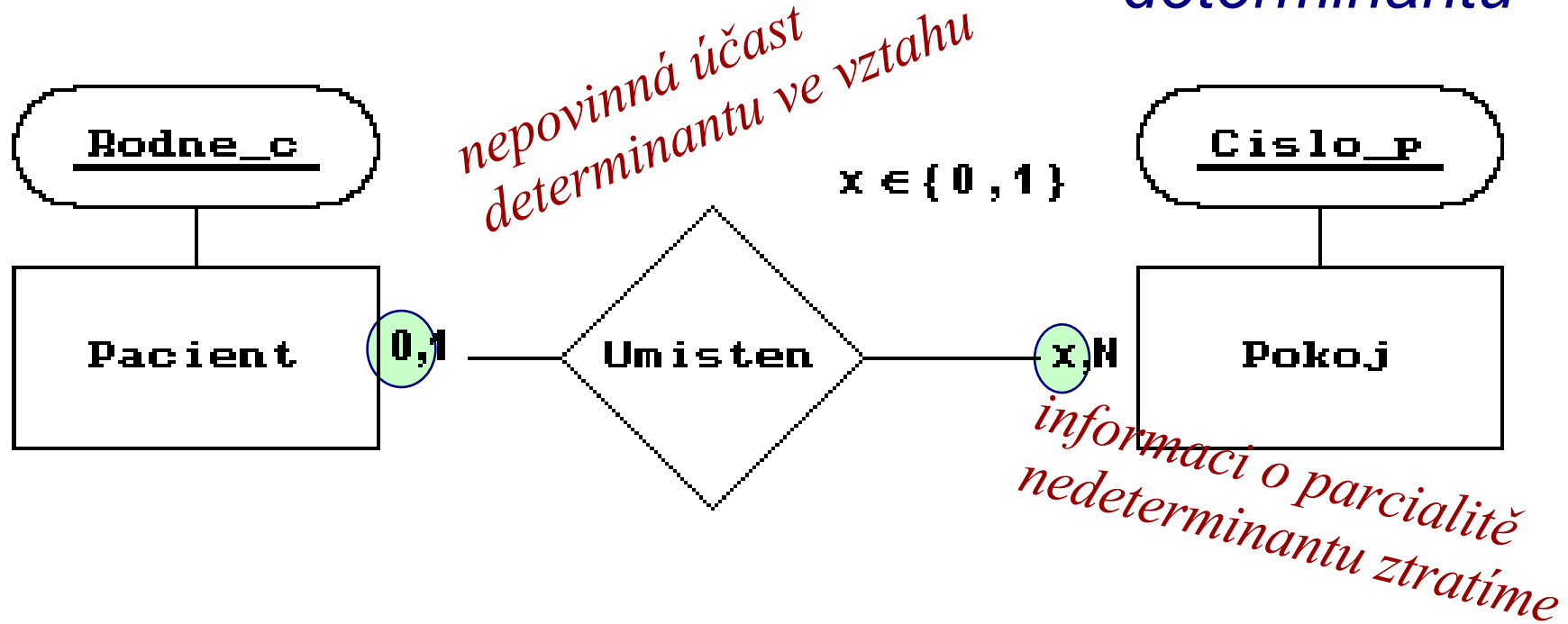
POKOJ(CISLO_P, POCET_LUZEK, ...)

IO: PACIENT[CISLO_P] \subseteq POKOJ[CISLO_P]

nesmí být NULL

Reprezentace vztahu 1:N

nepovinná účast
determinantu



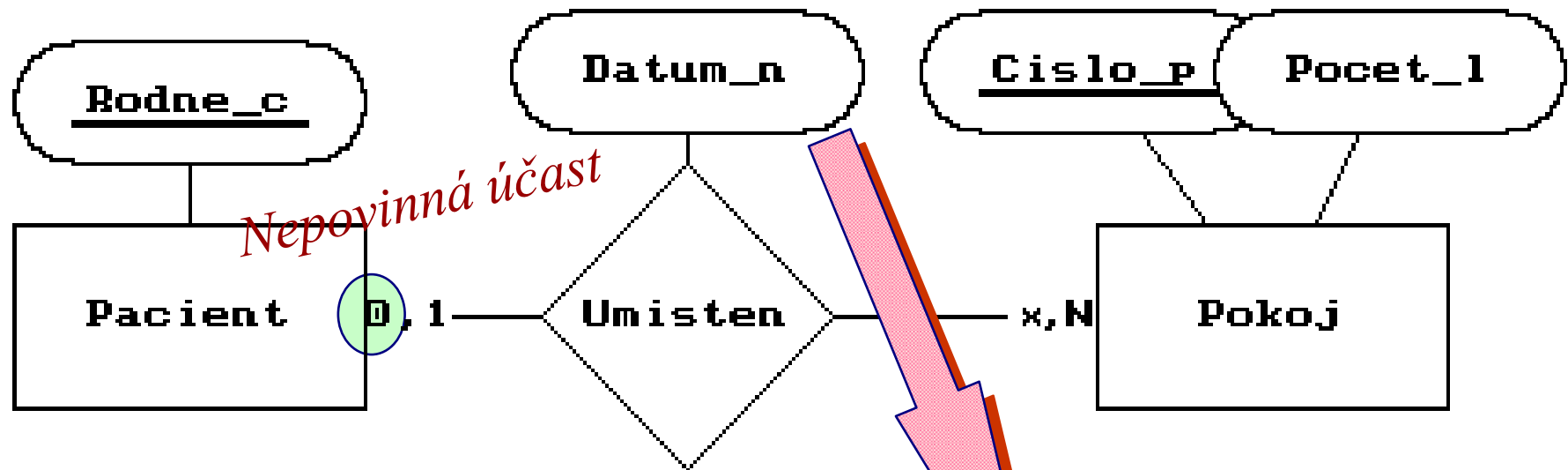
PACIENT(RODNÉ_Č, PŘÍJMENÍ, ..., ČÍSLO_P)

POKOJ(ČÍSLO_P, POČET_LŮŽEK, ...)

IO: PACIENT[ČÍSLO_P] ⊆ POKOJ[ČÍSLO_P]

miže být NULL

Reprezentace vztahu 1:N atribut vztahu varianta 1



PACIENT(RODNÉ_Č, PŘÍJMENÍ, ...)

POKOJ(ČÍSLO_P, POČET_LŮŽEK, ...)

UMÍSTĚN(RODNÉ_Č, ČÍSLO_P, **DATUM_N**)

*IO: UMÍSTĚN[RODNÉ_Č] ⊆ PACIENT[RODNÉ_Č]
_UMÍSTĚN [ČÍSLO_P] ⊆ POKOJ[ČÍSLO_P]*

pacienti bez pokoje nejsou uvedeni v UMISTEN

Reprezentace vztahu 1:N – varianta 1 (3 relace)

```
CREATE TABLE Pacient
```

```
( Rodne_c Number PRIMARY KEY,  
  Prijmeni Char(20) NULL,  
  Jmeno Char(10) NOT NULL);
```

```
CREATE TABLE Pokoj
```

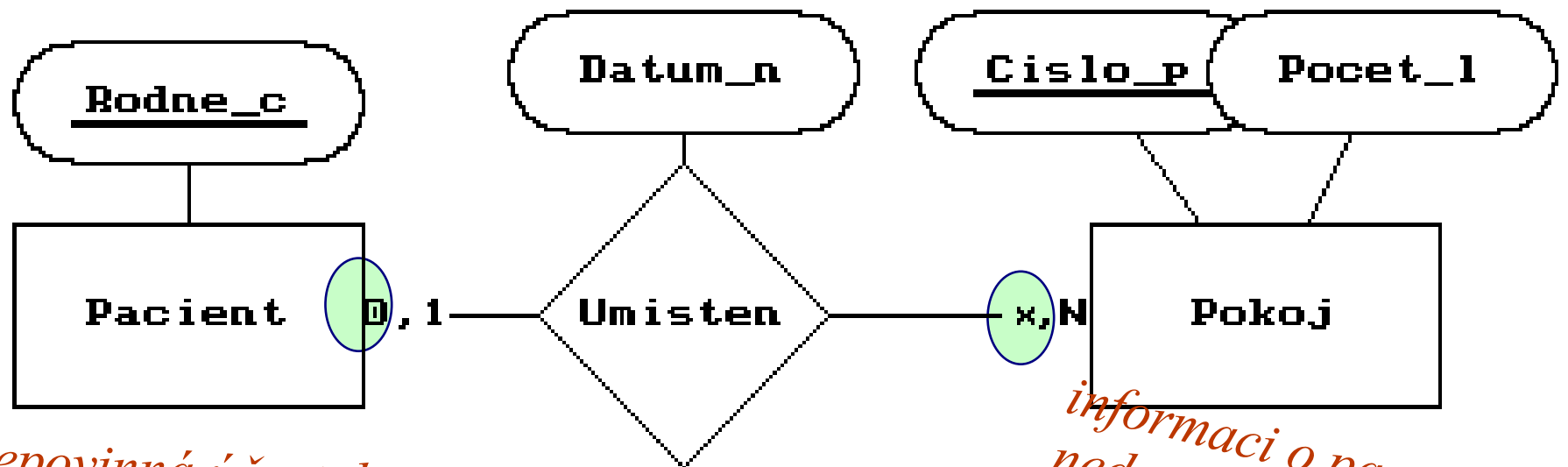
```
( Cislo_p Number PRIMARY KEY,  
  Pocet_l Number(2) NOT NULL);
```

```
CREATE TABLE Umisten
```

```
( U_Rodne_c Number REFERENCES Pacient,  
  U_Cislo_p Number NOT NULL REFERENCES Pokoj,  
  U_Datum_n Date NOT NULL,  
  PRIMARY KEY(U_Rodne_c));
```

pacienti bez pokoje nejsou uvedeni v UMISTEN

Reprezentace vztahu 1:N atribut vztahu varianta 2 (slévání)



nepovinná účast determinantu ve vztahu, který má navíc atribut

informaci o parcialitě nedeterminantu ztratíme

POKOJ(ČÍSLO_P, POČET_L,...)

PACIENT(RODNÉ_Č, PŘÍJMENÍ, ..., ČÍSLO_P, DATUM_N)

PACIENT[ČÍSLO_P] ⊆ POKOJ[ČÍSLO_P]

Obojí může být NULL pouze současně

Reprezentace vztahu 1:N - varianta 2 (2 realace)

```
CREATE TABLE Pacient
```

```
( Rodne_c Number PRIMARY KEY,  
  Prijmeni Char(20) NULL,  
  Jmeno Char(10) NOT NULL);
```

```
CREATE TABLE Pokoj
```

```
( Cislo_p Number PRIMARY KEY,  
  Pocet_l Number(2) NOT NULL);
```

```
ALTER TABLE Pacient ADD
```

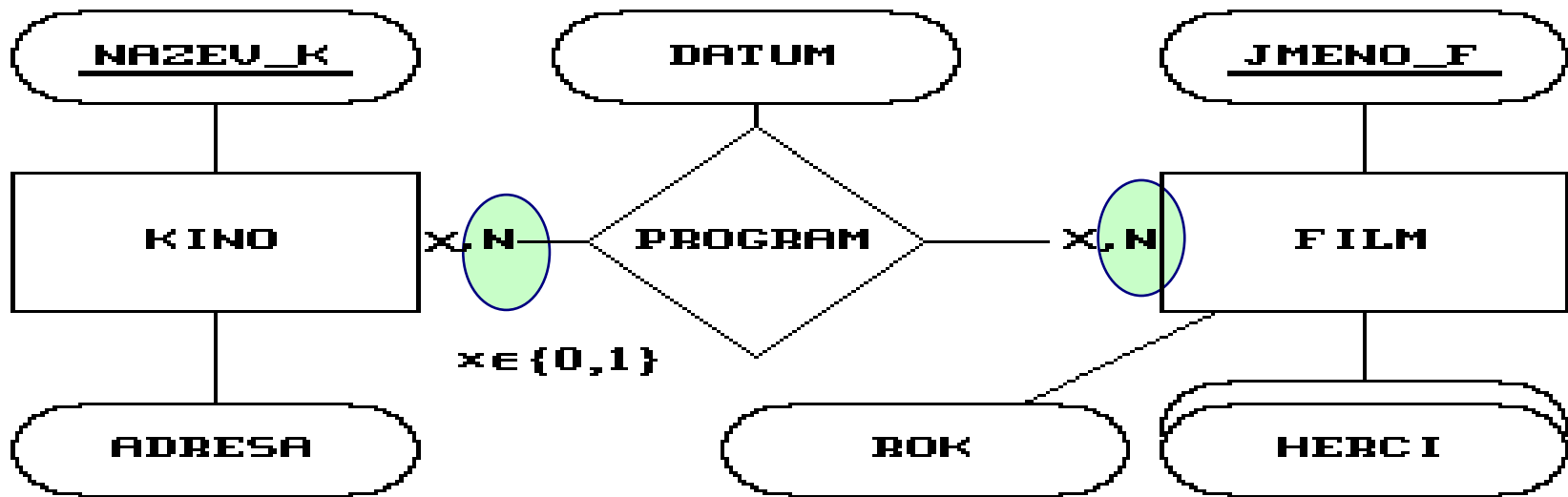
```
( Datum_n Date,  
  U_Cislo_p Number REFERENCES Pokoj,
```

```
CONSTRAINT CHECK
```

```
((U_Cislo_p IS NOT NULL AND Datum_n IS NOT NULL) OR  
(U_Cislo_p IS NULL AND Datum_n IS NULL))  
);
```

*podchycení nepovinné účasti
determinantu*

Reprezentace vztahu M:N



```
CREATE TABLE PROGRAM
```

```
(P_NAZEVI_K Char_Varying(20) NOT NULL,
```

```
P_JMENO_F Char_Varying(20) NOT NULL,
```

```
DATUM Date NOT NULL,
```

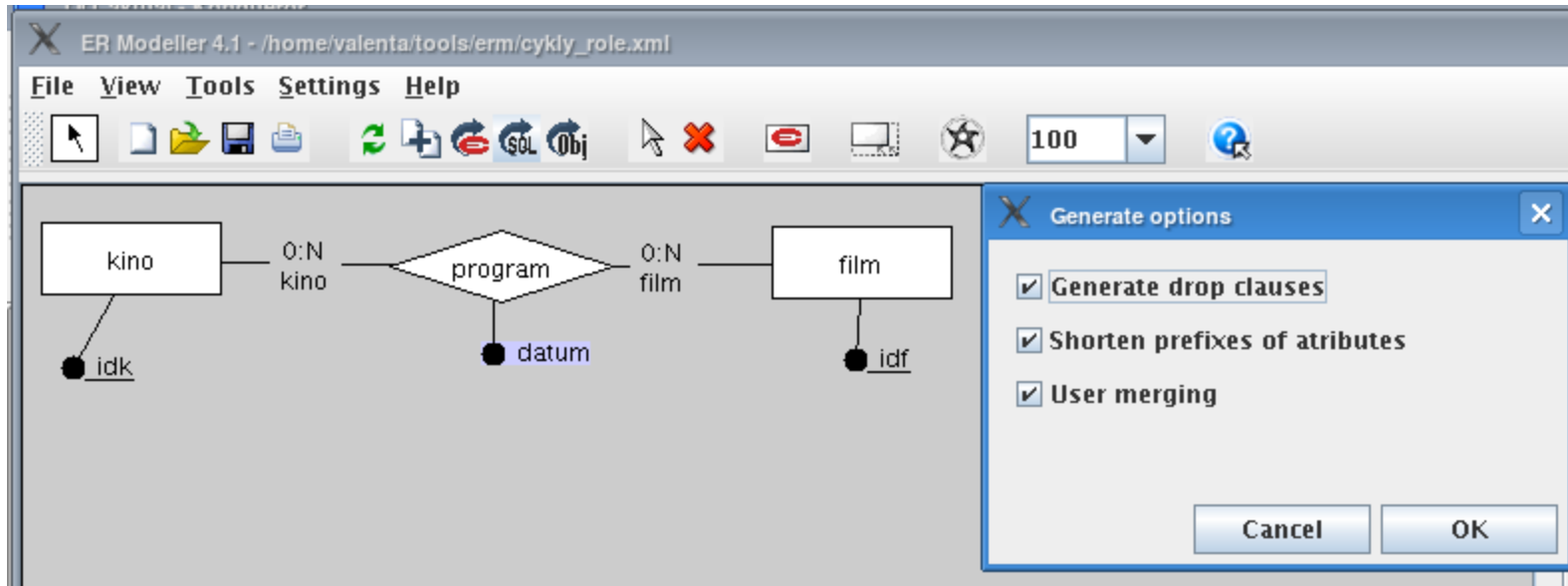
```
PRIMARY KEY (P_NAZEVI_K, P_JMENO_F), -- !!! (chceme to?)
```

```
FOREIGN KEY (P_NAZEVI_K) REFERENCES KINO,
```

```
FOREIGN KEY (P_JMENO_F) REFERENCES FILM);
```

Informaci o povinné/nepovinné účasti ztrácíme.

Vztah M:N v ER modeláři (1/2)



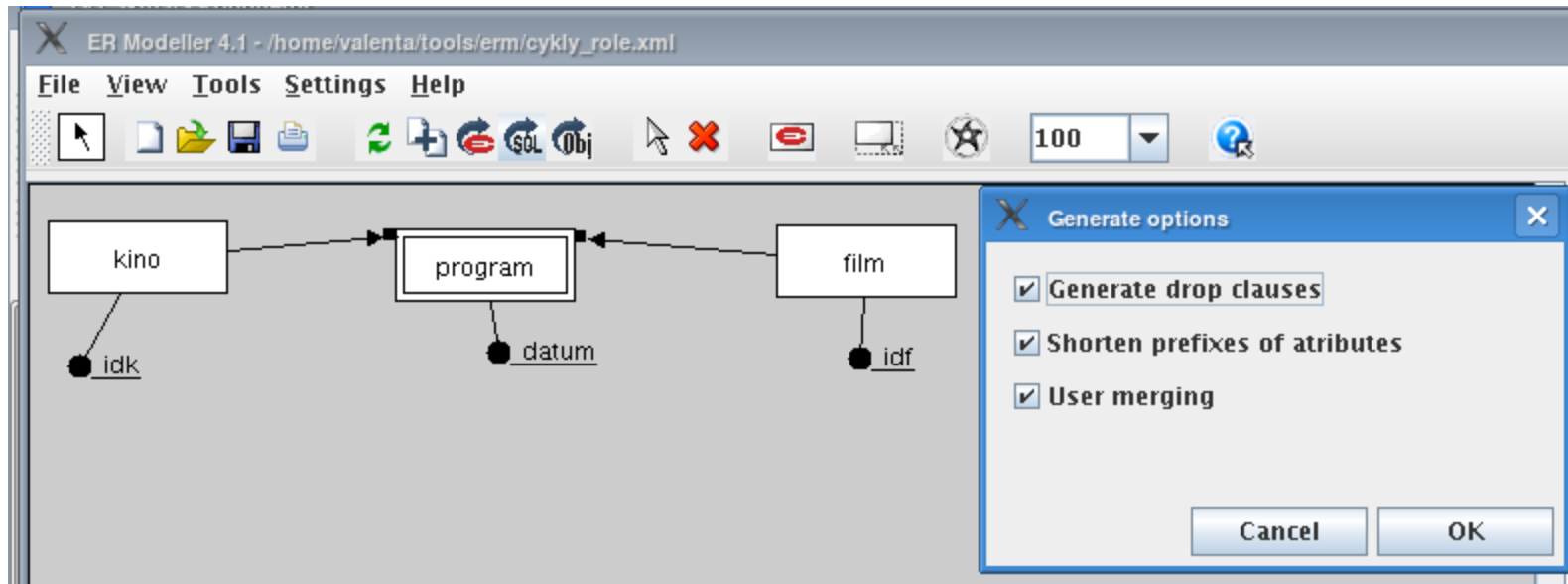
Create table kino (
idk Integer Not Null,
Constraint PK_kino Primary Key (idk));

Create table film (
idf Integer Not Null,
Constraint PK_film Primary Key (idf));

Create table program (
datum Date Not Null,
k_idk Integer Not Null,
f_idf Integer Not Null,
Constraint UNQ_program_1 Unique (k_idk, f_idf) -- opravdu to tak chceme???)

);
Alter table program add (
Constraint FK_program_1 Foreign Key (k_idk) References kino(idk),
Constraint FK_program_2 Foreign Key (f_idf) References film(idf));

Vztah M:N v ER modeláři 2/2



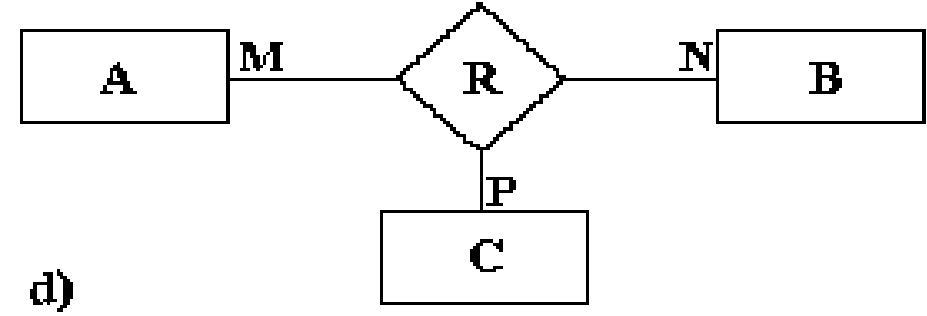
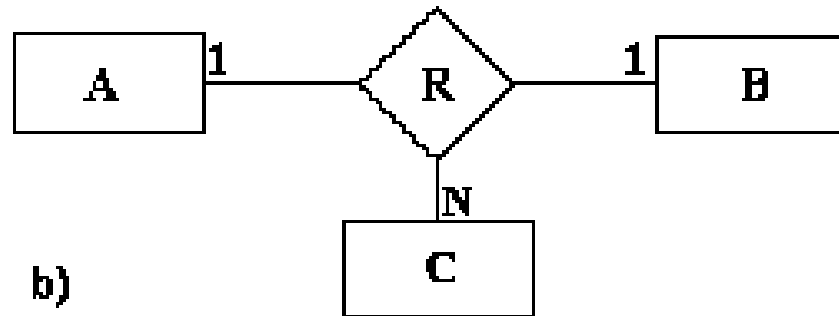
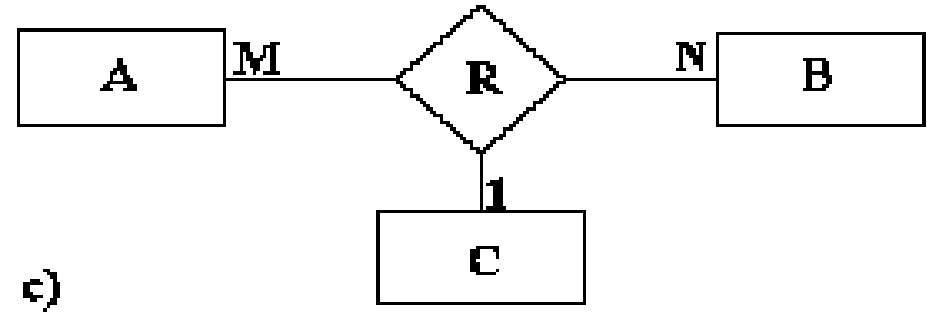
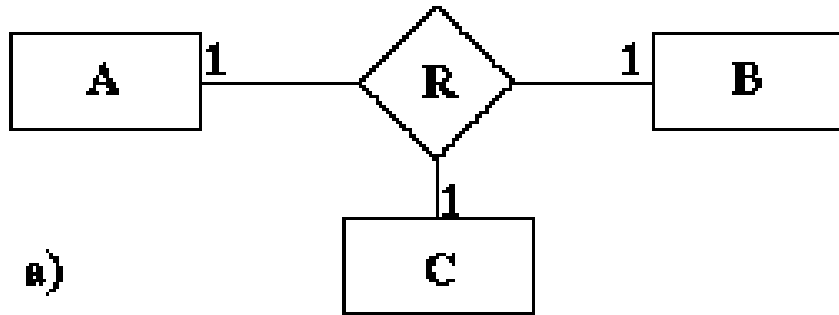
```
Create table kino (  
    idk Integer Not Null,  
    Constraint PK_kino Primary Key (idk));
```

```
Create table film (  
    idf Integer Not Null,  
    Constraint PK_film Primary Key (idf));
```

```
Create table program (  
    datum Date Not Null,  
    f_idf Integer Not Null,  
    k_idk Integer Not Null,  
    Constraint PK_program Primary Key (datum, f_idf, k_idk) -- film v jednom kině v různých dnech  
);
```

```
Alter table program add (  
    Constraint FK_program_1 Foreign Key (k_idk) References kino(idk),  
    Constraint FK_program_2 Foreign Key (f_idf) References film(idf));
```

Reprezentace vztahu n-árního



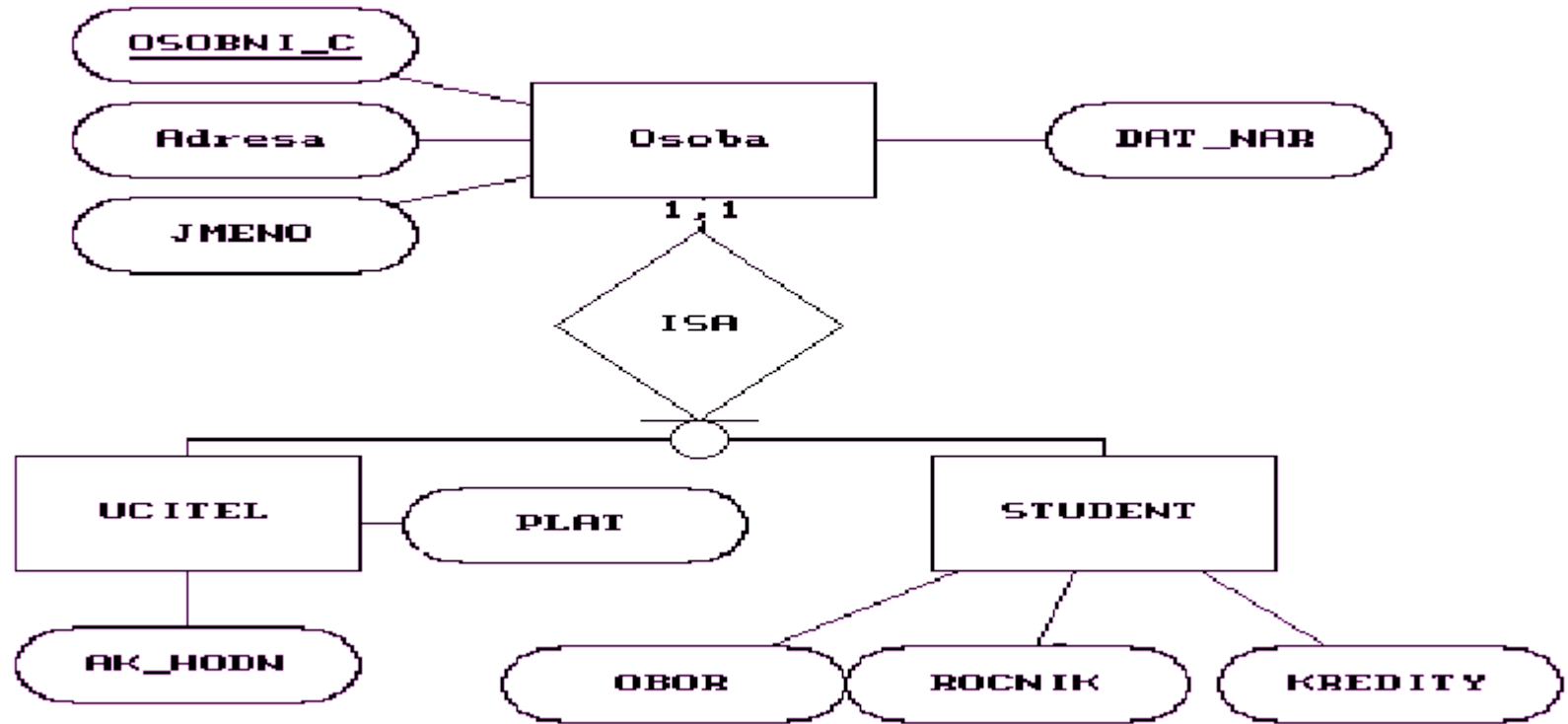
a) $R(\underline{KA}, \underline{KB}, \underline{KC}),$

c) $R(\underline{KA}, \underline{KB}, KC),$

b) $R(KA, \underline{KB}, \underline{KC}),$

d) $R(\underline{KA}, \underline{KB}, \underline{KC}),$

Reprezentace ISA vztahu



OSOBA(OS_C, ADRESA, JMENO, DAT_NAR)

UCITEL (OS_C, AK_HODNOST, PLAT)

STUDENT (OS_C, OBOR, ROCNIK, KREDIT)

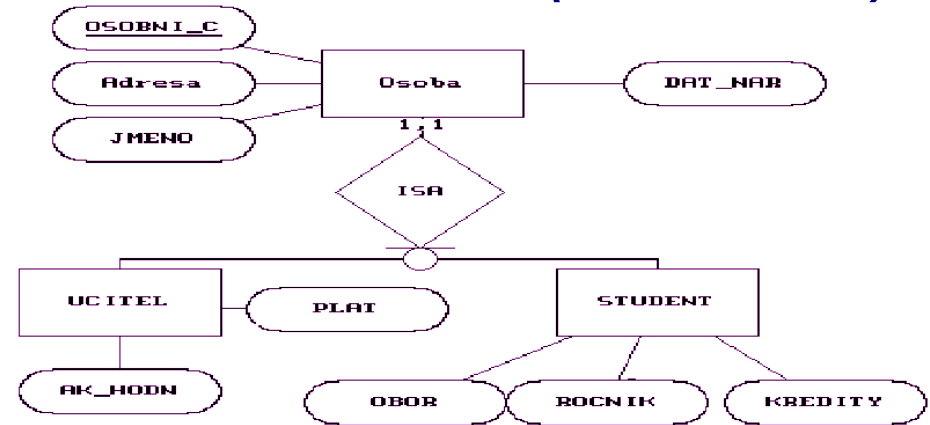
IO: $STUD[OS_C] \subseteq OSOBA[OS_C]$,
 $UCITEL[OS_C] \subseteq OSOBA[OS_C]$

Reprezentace ISA vztahu – varianta 1 (3 relace)

```
CREATE TABLE OSOBA  
(OS_C Number PRIMARY KEY,  
JMENO VARCHAR (20),  
ADRESA VARCHAR (25) ,  
PLAT NUMBER,  
DAT_NAR DATE);
```

```
CREATE TABLE UCITEL  
(OS_C Number PRIMARY KEY,  
AK_HODN Number, Plat,  
FOREIGN KEY (OS_C) REFERENCES  
OSOBA);
```

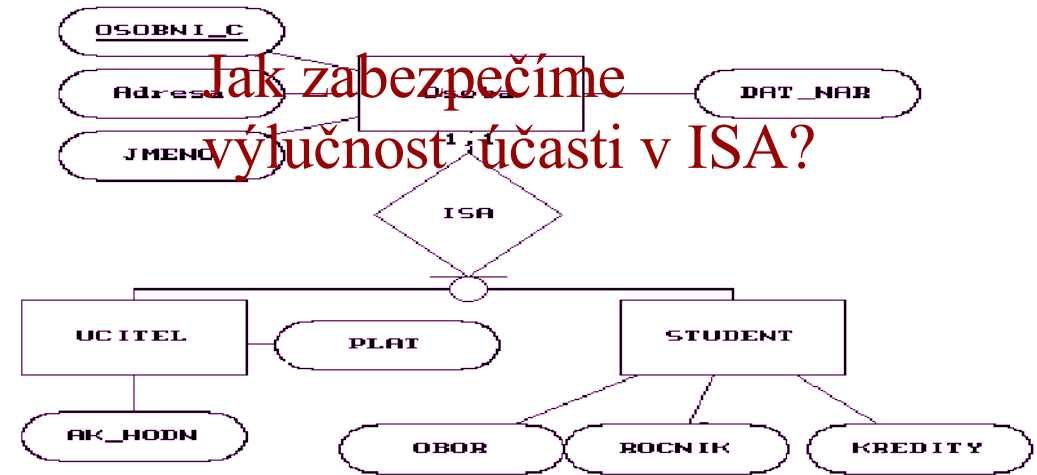
```
CREATE VIEW V_STUDENT AS  
SELECT *  
FROM OSOBA JOIN STUD  
USING (OS_C)
```



```
CREATE TABLE STUD  
(OS_C Number PRIMARY KEY,  
OBOR Number,  
ROCNIK Number,  
KREDITY Number,  
FOREIGN KEY (OS_C) REFERENCES  
OSOBA);
```

```
CREATE VIEW V_UCITEL AS  
SELECT *  
FROM OSOBA JOIN UCITEL  
USING(OS_C)
```

Reprezentace ISA vztahu – varianta 2 (1 relace)



Jak zabezpečíme
výlučnost účasti v ISA?

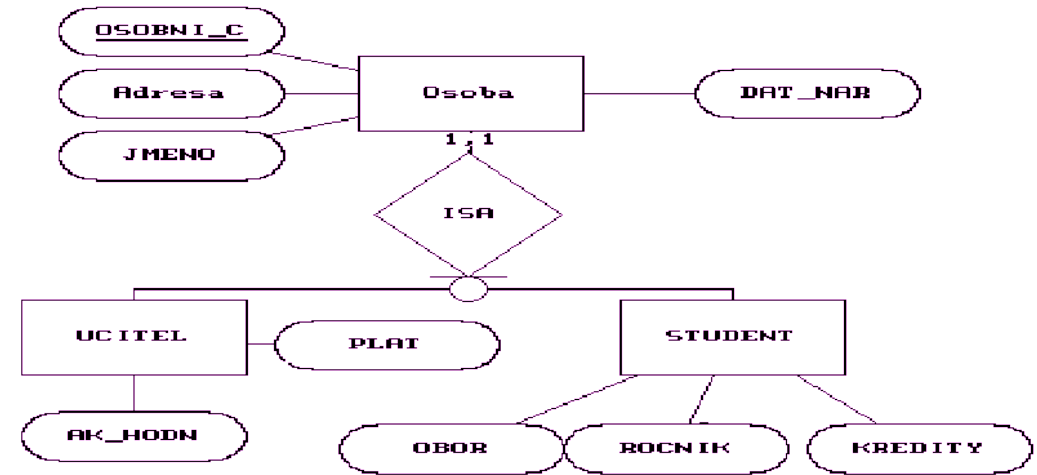
```
CREATE TABLE OSOBA  
(OS_C Number PRIMARY KEY,  
JMENO VARCHAR (20),  
ADRESA VARCHAR (25) ,  
DAT_NAR DATE,  
OBOR_st Number,  
ROCNIK_st Number,  
KREDITY_st Number,  
AK_HODN_uc Number,  
PLAT_uc Number );
```

```
CREATE VIEW STUDENT AS  
SELECT OS_C,JMENO,ADRESA,DAT_NAR,  
OBOR_st, ROCNIK_st, KREDITY_st  
FROM OSOBA  
WHERE AK_HODN_uc IS NULL;
```

```
CREATE VIEW UCITEL AS  
SELECT OS_C,JMENO,ADRESA,DAT_NAR,AK_HODN_uc,PLAT_uc  
FROM OSOBA  
WHERE AK_HODN_uc IS NOT NULL;
```

Reprezentace ISA vztahu – varianta 2 - pokračování

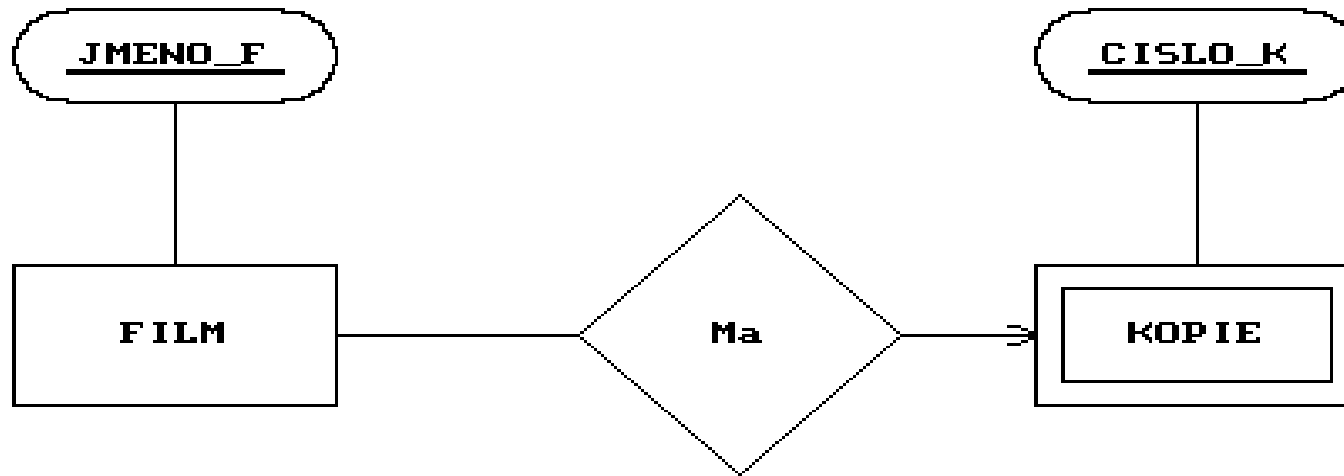
```
CREATE TABLE OSOBA  
(OS_C Number PRIMARY KEY,  
JMENO VARCHAR (20),  
ADRESA VARCHAR (25) ,  
DAT_NAR DATE,  
OBOR_st Number,  
ROCNIK_st Number,  
KREDITY_st Number,  
AK_HODN_uc Number,  
PLAT_uc Number );
```



Jak zabezpečíme
výlučnost účasti v ISA?

```
ALTER TABLE OSOBA ADD CONSTRAINT vylucnost  
CHECK(((AK_HODN_uc is NULL and PLAT_uc is NULL) and  
(OBOR_st + ROCNIK_st +KREDITY_st) is not NULL)) or  
(((AK_HODN_uc is not NULL and PLAT_uc is not NULL) and  
COALESCE(OBOR_st,ROCNIK_st,KREDITY_st) is NULL)  
);
```

Reprezentace identifikačního vztahu



FILM (JMENO_F, ROK , REZISER)

KOPIE(JMENO_F, CISLO_K, DATUM, ...)

IO: KOPIE[JMENO_F] \subseteq FILM [JMENO_F]

Reprezentace identifikačního vztahu

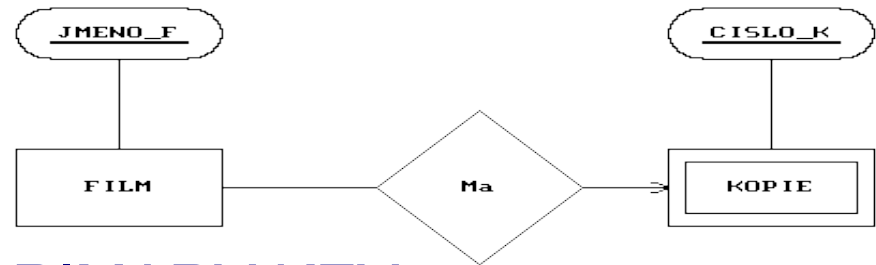
CREATE TABLE FILM

(JMENO_F CHAR(20)
ROK DATE
REZISER CHAR(20)

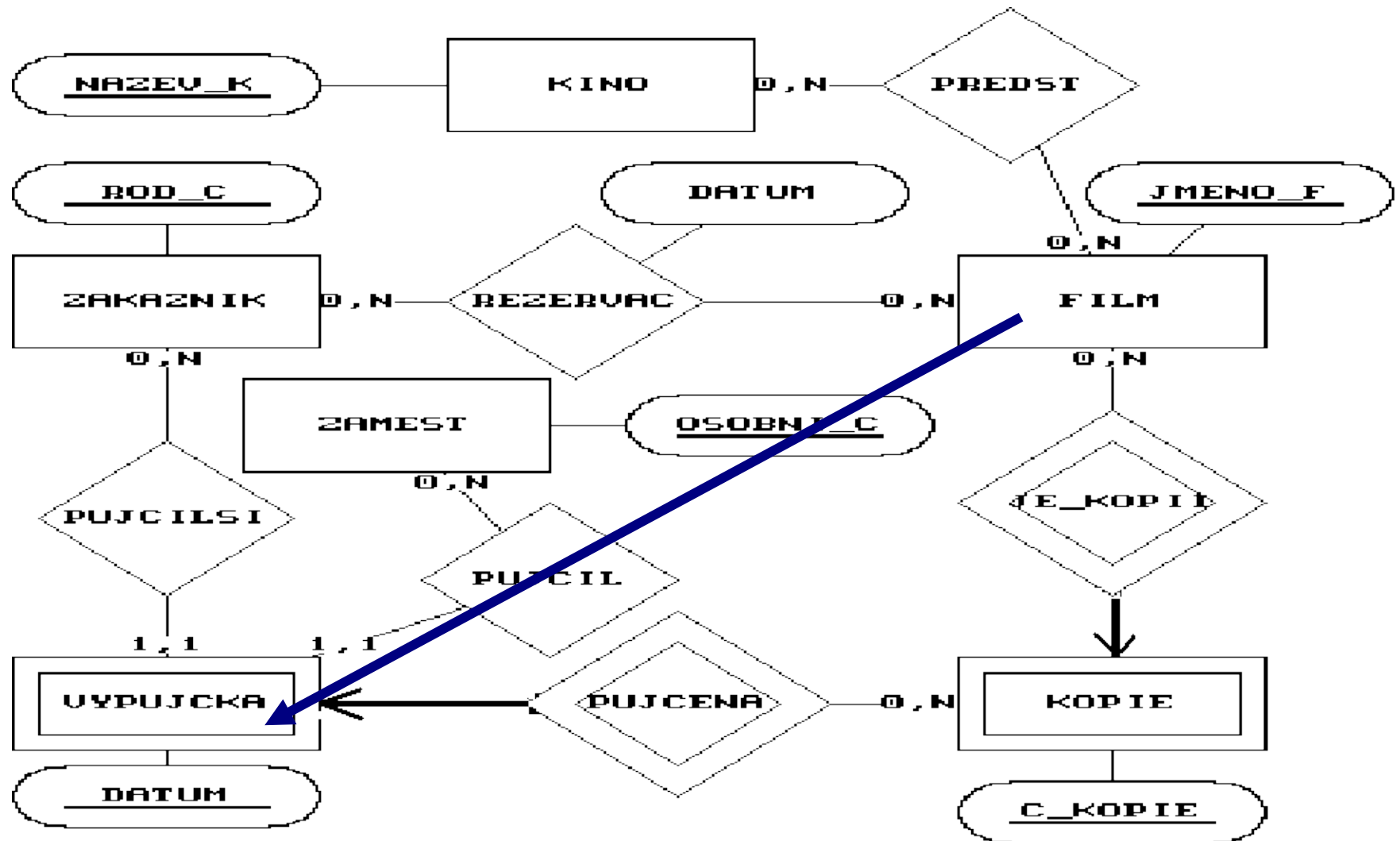
CREATE TABLE KOPIE

(JMENO_F CHAR(20) NOT NULL,
CISLO_K NUMBER NOT NULL,
DATUM DATE NOT NULL,

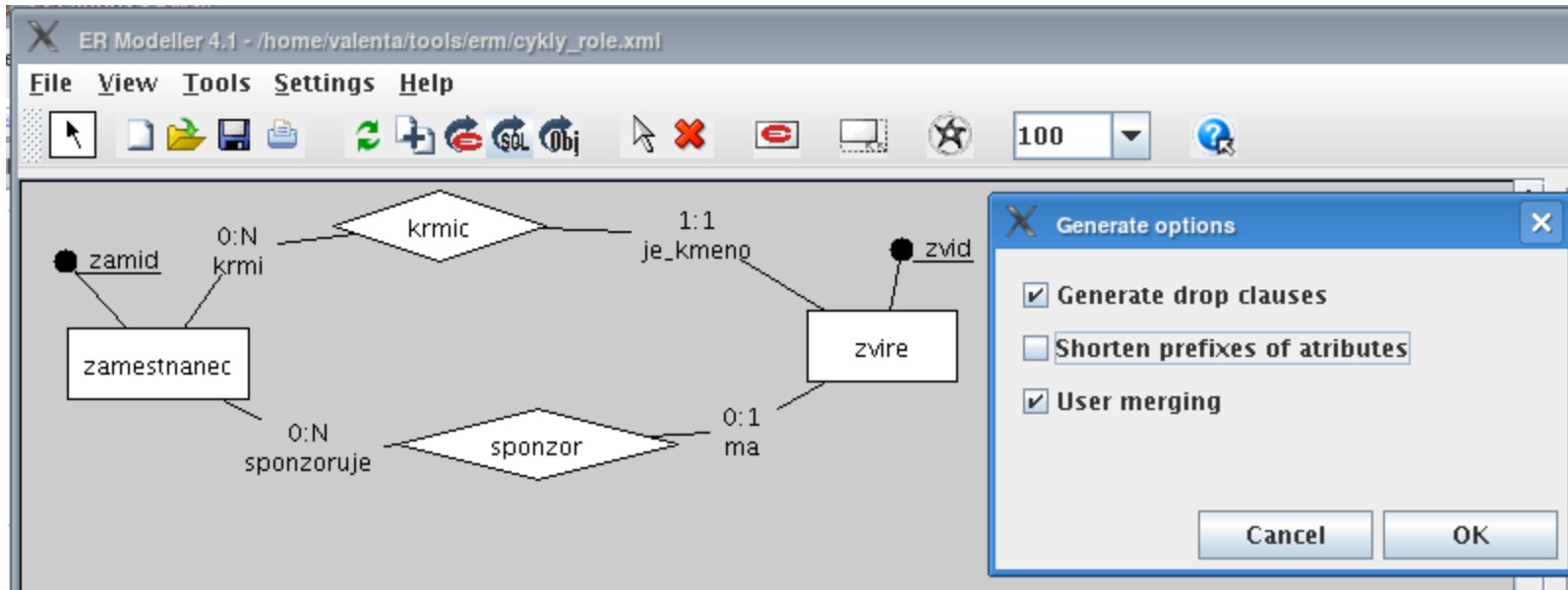
FOREIGN KEY (JMENO_F) REFERENCES FILM,
PRIMARY KEY (JMENO_F, CISLO_K));



Příklad propagace identifikačního klíče



K čemu jsou dobrá jména rolí v ER modeláři?



Create table zamestnanec (
zamid Integer Not Null,
Constraint PK_zamestnanec Primary Key (zamid));

Alter table zvire add (
Constraint FK_zvire_1 Foreign Key (sponzoruje_zamid) References zamestnanec(zamid),
Constraint FK_zvire_2 Foreign Key (krmi_zamid) References zamestnanec(zamid));

Create table zvire (
sponzoruje_zamid Integer Not Null,
krmi_zamid Integer Not Null,
zvid Integer Not Null,
Constraint PK_zvire Primary Key (zvid));

Toto řešení (povinné role vztahů) je bezpečné , ale obecně se vzdáváme zápisů:
select * from zamestnanec join zvire using (???)

=> ruční úpravy skriptu jsou na vlastní nebezpečí!