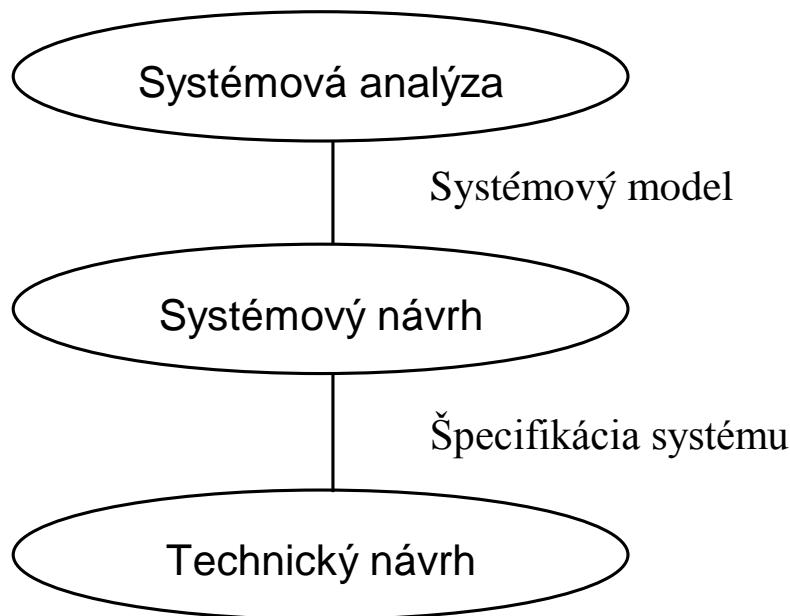


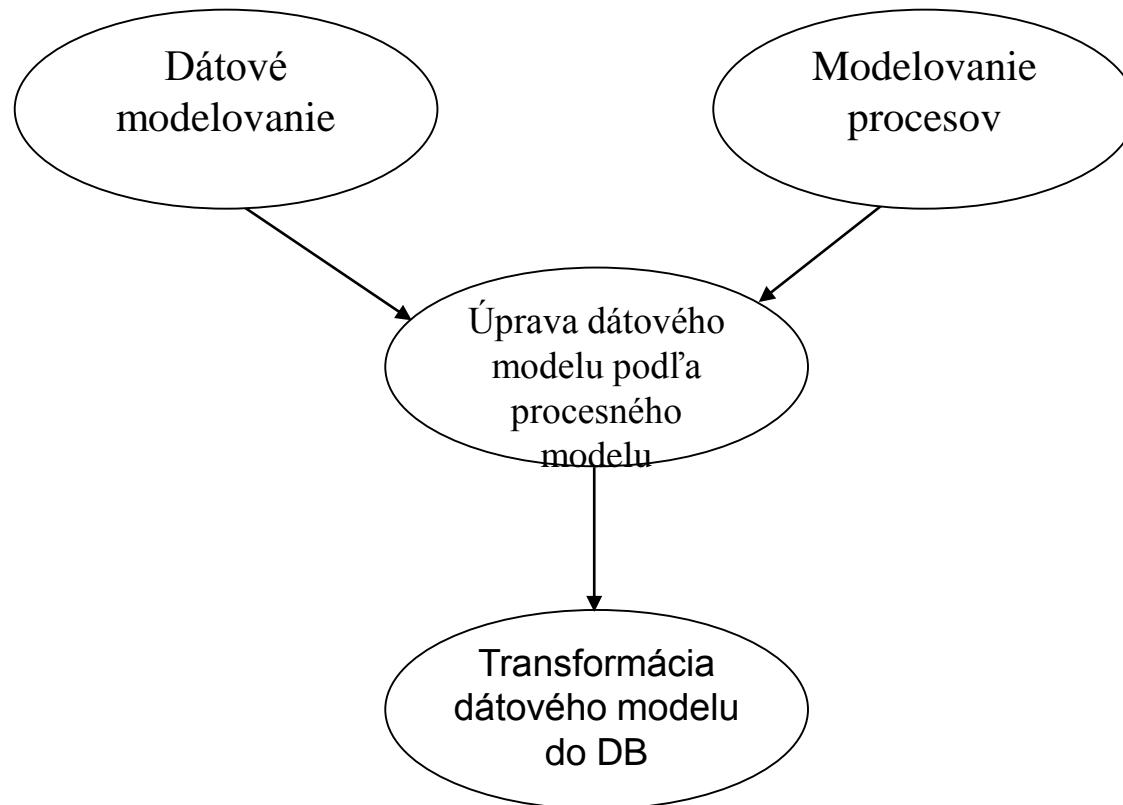
Dátové modelovanie

K. Matiaško

Postup pri vývoji IS



Systémový návrh



Dátové modely - generácie

- **Primitívne dátové modely** - Ide vlastne o typicky súborovo orientovaný prístup, pričom definície súborov sú súčasťou užívateľských programov
- **Klasické dátové modely** - sú reprezentované hlavne sietovým modelom definovaným skupinou CODASYL DBTG, hierarchickým dátovým modelom, ktorý je špeciálnym prípadom sietového a je reprezentovaný systémom IMS. K tejto skupine je možné priradiť aj **relačný dátový model**, ktorý však priniesol úplne nový pohľad na dátové modelovanie.
- **Sémantické dátové modely** - Tieto modely sa nazývajú sémantické, prípadne konceptuálne a sú založené na pojmoch entita, vzťah, vlastnosť, atribút, objekt a pod. Sémantické modely slúžia obvykle k vytvoreniu schém s následnou transformáciou na databázovú schému.

Konceptuálne modelovanie

Každý konceptuálny, či databázový model rieši tri problémy:

- ***Štruktúra dát***
Z tohto pohľadu je nutné zabezpečiť identifikáciu všetkých objektov a ich vlastností, vrátane popisu štruktúr pre vyjadrenie vzťahov medzi objektmi.
- ***Manipulácia dát***
Pre každý dátový model by bolo vhodné navrhnuť množinu prípustných operácií nad príslušným objektom dát.
- ***Integritné obmedzenia***
Pre každý objekt, jeho vlastnosti a vzťahy medzi objektmi je potrebné definovať množinu integritných obmedzení, ktoré definujú základné vlastnosti dátových objektov.

Princípy dátového modelovania

- Orientácia na objekty
- Funkcionálna podstata vztahov
- ISA-hierarchia
- Konštrukcia objektov z iných objektov

ERA model

Definícia – E-R konceptuálny model

E-R konceptuálny model (skrátene E-R model) je množina pojmov, ktoré nám pomáhajú na konceptuálnej úrovni abstrakcie popísat' užívateľ'skú aplikáciu s cieľom následnej špecifikácie štruktúry databázy.

Peter Pin-Shan Chen



....geniálne

日 (sun) + 月 (moon) = 明 (Bright/ Brightness by light)

口 (mouth) + 鳴 (bird) = 鳴 (Bird's singing)

Entita

Definícia - Entita

Entita (entity) je objekt reálneho sveta, ktorý je schopný nezávislej existencie a je jednoznačne odlišiteľný od ostatných objektov.

Vzt'ah

Definícia -Vzt'ah :

Vzt'ah (relationship) je väzba medzi dvoma (alebo viacerými) entitami.



Popisný typ

Definícia –Popisný typ

Pod popisným typom budeme rozumieť jednoduchý dátový typ, teda dvojicu (množina hodnôt, množina operácií).



Atribút

Definícia - Atribút

Atribútom (attribute) budeme rozumieť funkciu priradujúcu entitám, alebo vztahom hodnotu (popisného typu), určujúcu niektorú podstatnú vlastnosť entity, alebo vztahu.



Typ entity

Typ entity popisuje množinu vlastností entity (objektu)

Typ entity

osoba

rod_cislo (PK)

meno

priezvisko

ulica

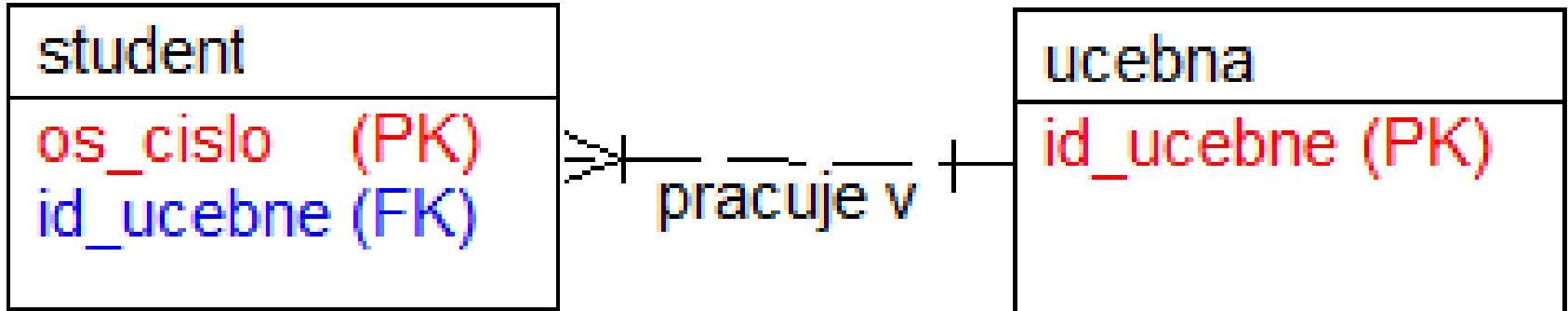
psc

obec

Typ vztahu

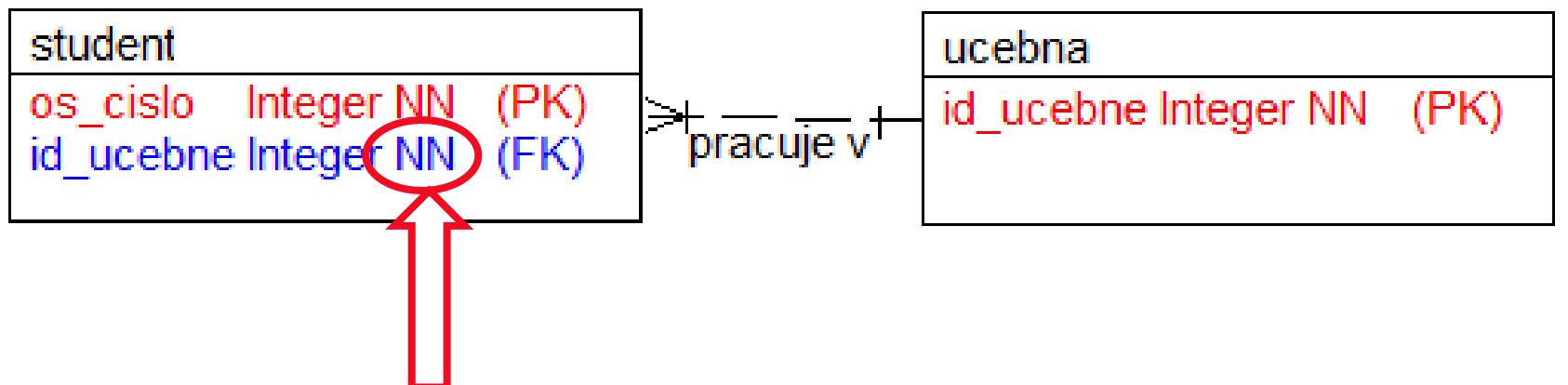
***Typ vztahu popisuje množinu vlastností
vztahu***

Vztah



Neidentifikačný vztah (`id_ucebne` = FK)

1:N, povinné členstvo



Inštancia entity

- množina hodnôt popisujúca konkrétnu entitu t.j. každý z jej atribútov má priradenú hodnotu

Inštancia vztahu

- množina hodnôt popisujúca konkrétny vztah t.j. každý z jeho atribútov má priradenú hodnotu

Atribúty

Každý entitný typ obsahuje množinu atribútov, ktoré majú určené:

- **meno atribútu**
- **typ atribútu**, v prípade atomického atribútu ide o hodnotovú množinu (**doménu**) a množinu operácií, ktoré je možné na hodnotovú množinu previesť; v rámci tejto definície býva určená aj veľkosť priestoru (v znakoch), ktorý zaberá vonkajšia reprezentácia hodnoty atribútu (napr. na obrazovke)
- príznak, či je atribút **kľúčový** (t. j. je súčasťou identifikačného kľúča)
- príznak, či atribút môže mať tzv. **prázdnú hodnotu**. Je interpretovaná ako “nedefinovaná”, “neznáma” a pod. Označuje sa obyčajne **NULL**
- príznak, že atribút musí mať unikátnu hodnotu (**UNIQUE**, **DISTINCT**)

Typy atribútov

- Atomické atribúty
- Neatomické atribúty
 - *Skupinové atribúty (štrukturované)*
 - *Viachodnotové atribúty*

Identifikačný klúč

Definícia – Identifikačný klúč

Atribút (skupina atribútov), ktorého hodnota slúži na identifikáciu konkrétnej entity sa nazýva identifikačný klúč.



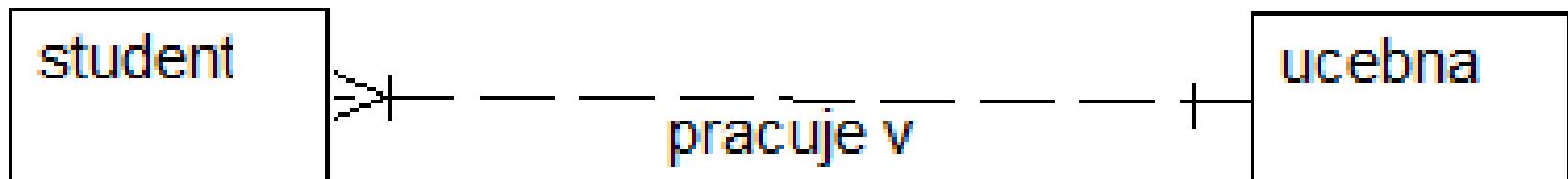
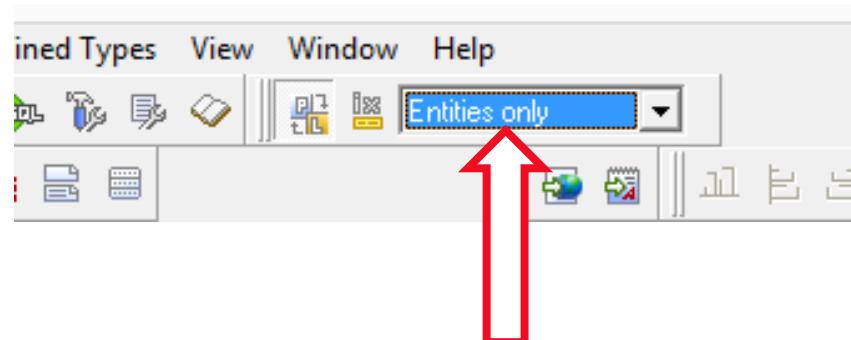
Zápis konceptuálnej schémy v modeli E-R

Poznáme dva rôzne prístupy:

- Lineárny textový zápis
- E-R diagramy

Typ entity – len meno entity

osoba



Typ entity – len mená atribútov

osoba
rod_cislo (PK)
meno
priezvisko
ulica
psc
obec

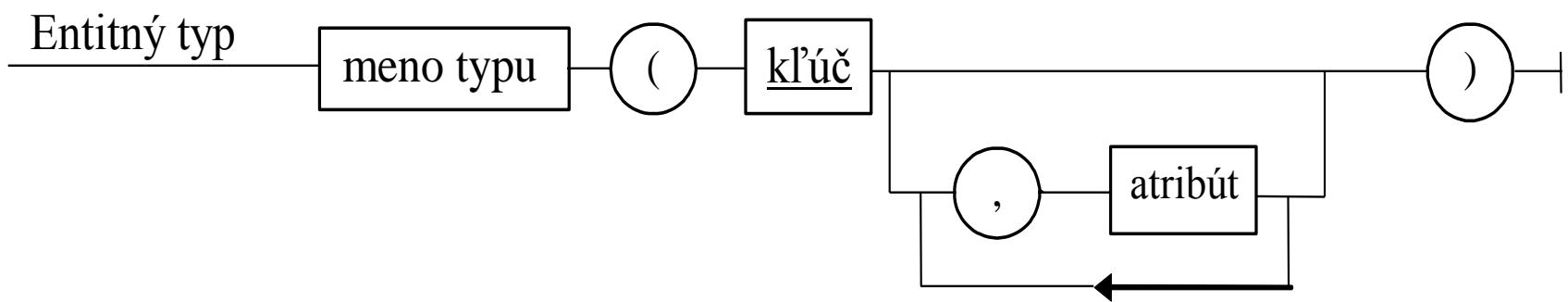


Typ entity mená atribútov, dátový typ a IO

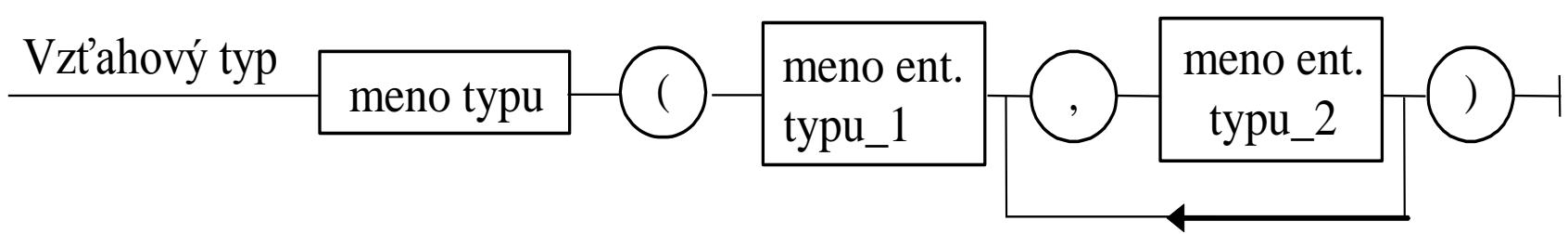


osoba				
rod_cislo	Char(11)	NN	(PK)	
meno	Varchar2(15)	NN		
priezvisko	Varchar2(15)	NN		
ulica	Varchar2(20)			
psc	Char(5)			
obec	Varchar2(50)			

Lineárny zápis



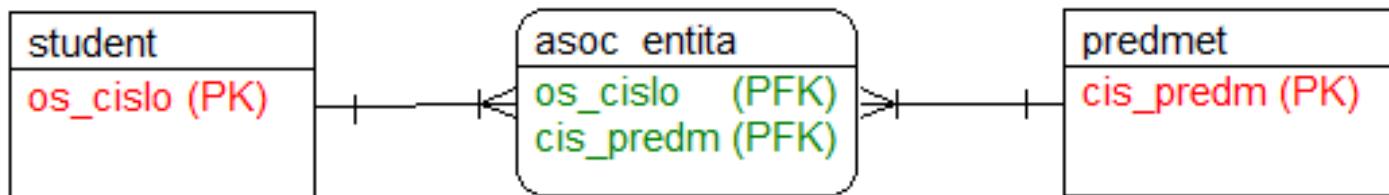
Lineárny zápis



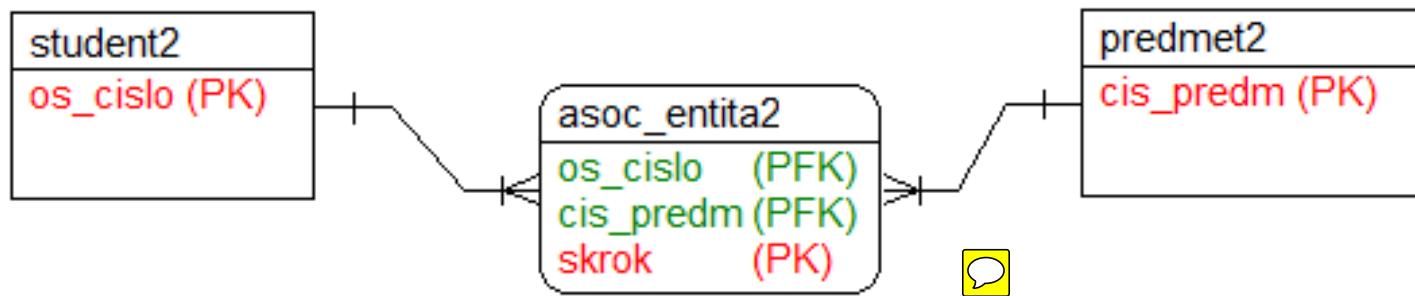
ERA diagram



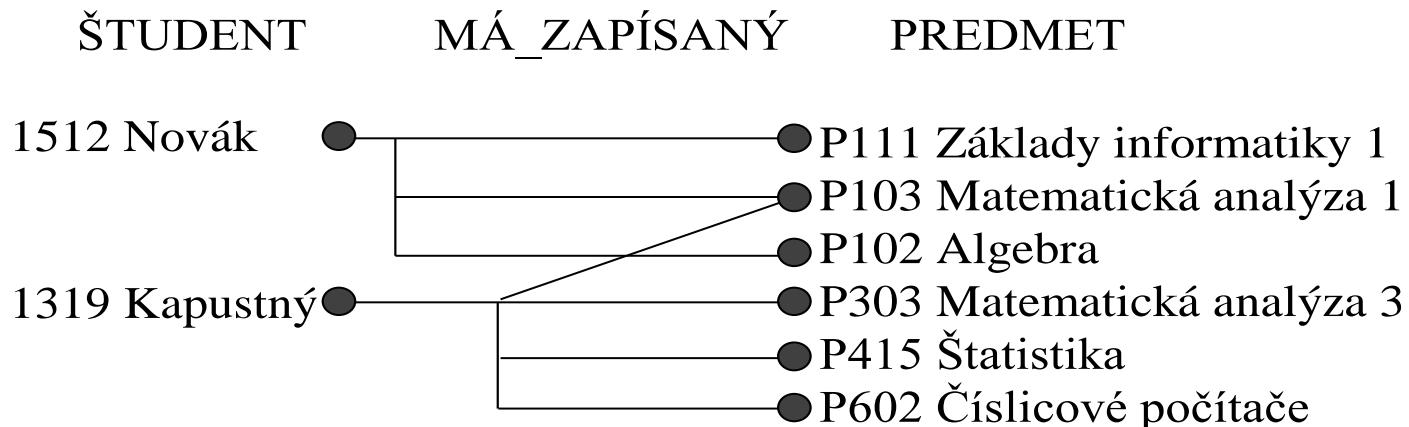
M:N => **asociatívna entita**



Študent nesmie predmet opakovat'



Výskytový diagram



Vztahy a integritné obmedzenia pre vztahy

Definícia –Kardinalita vztahu

Kardinalita vztahu je integritné obmedzenie, ktoré vyjadruje prípustný počet entít vo vztahu.



Vztahy a integrítne obmedzenia pre vztahy

Kardinalita vtahu 1:1

Kardinalita vztahu 1:N

Kardinalita vztahu M:N

Vzťahy a integritné obmedzenia pre vzťahy



Definícia – Kardinalita vzťahu 1:1

Kardinalita vzťahu 1:1 je integritné obmedzenie, ktoré vyjadruje vzťah medzi maximálne jednou entitou a maximálne jednou entitou iného, resp. rovnakého typu.



Vztahy a integritné obmedzenia pre vztahy



Definícia – Kardinalita vztahu 1:N

Kardinalita vztahu 1:N je integritné obmedzenie, ktoré vyjadruje vztah medzi maximálne jednou entitou a N entitami iného, resp. toho istého typu.



Vztahy a integritné obmedzenia pre vztahy

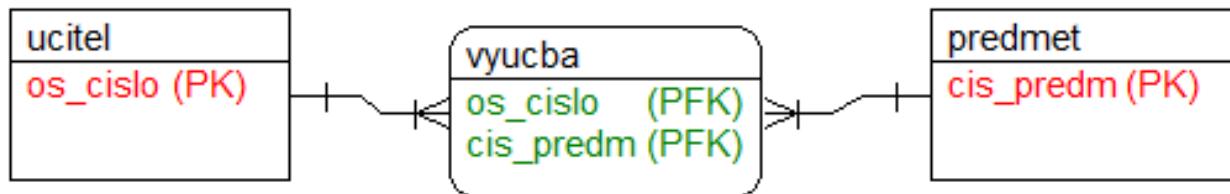
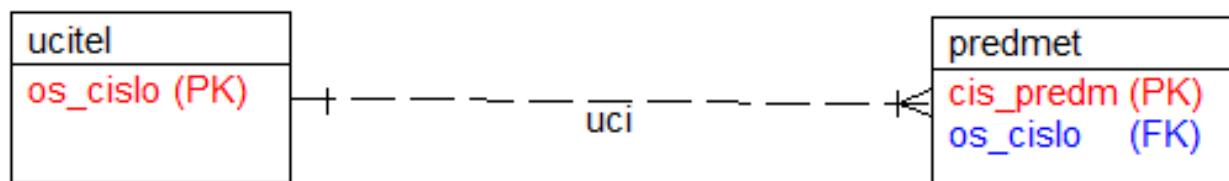
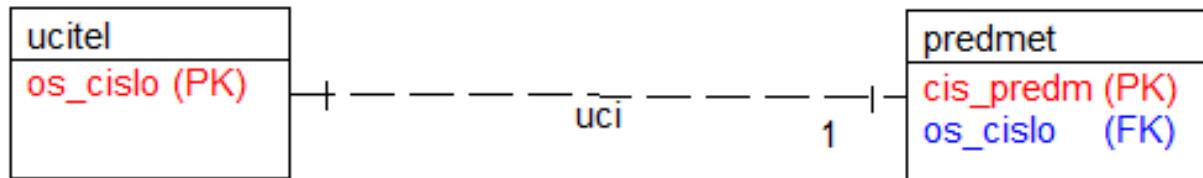


Definícia – Kardinalita vztahu M:N

Kardinalita vztahu M:N je integritné obmedzenie, ktoré vyjadruje vztah medzi M entitami jedného typu a N entitami iného, resp. toho istého typu.

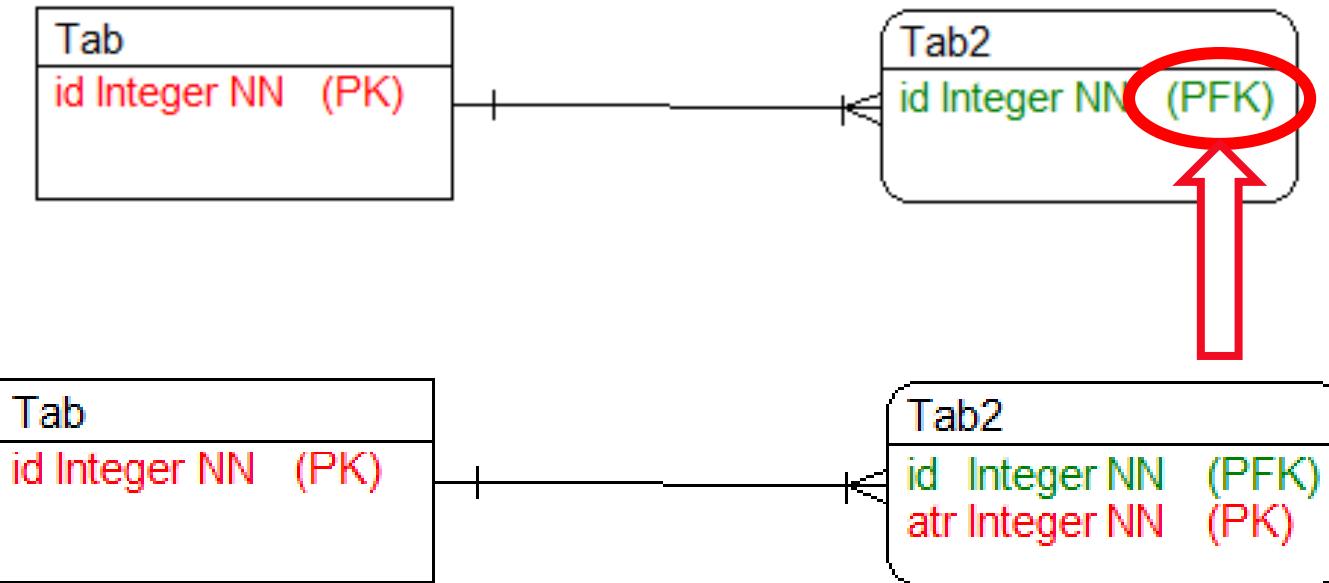


Kardinalita vzt'ahu

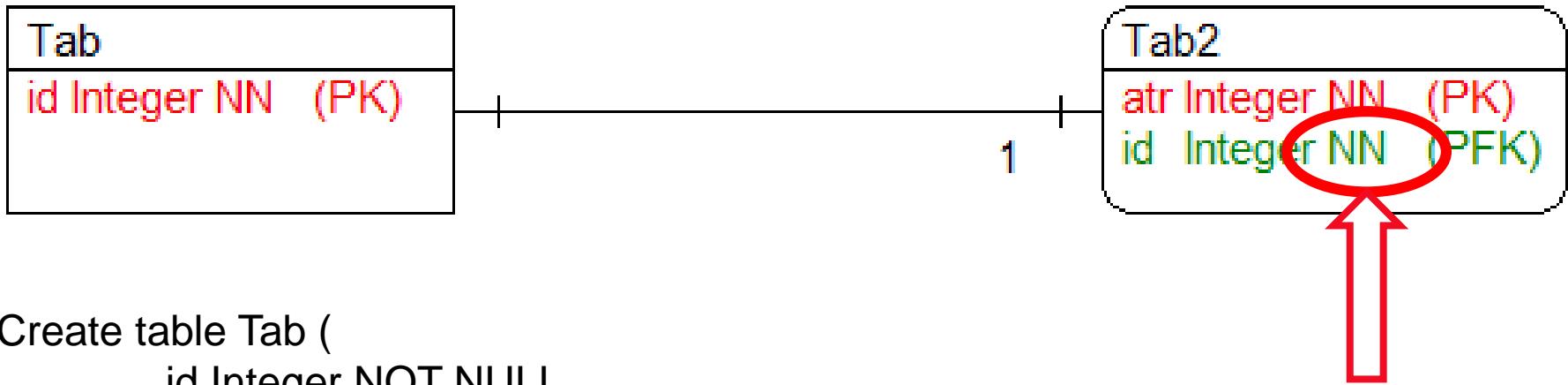


Kardinalita vzt'ahu

POZOR!!! TOTO NIE JE 1:N!!!



Kardinalita vzt'ahu

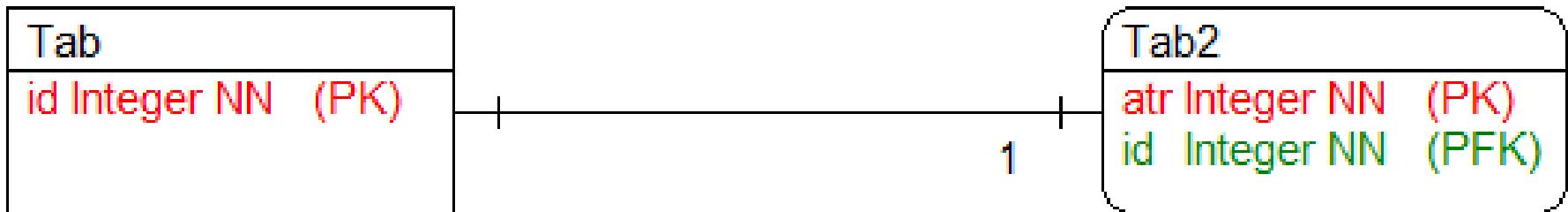


```
Create table Tab (
    id Integer NOT NULL ,
primary key (id)
);
```

```
Create table Tab2 (
    atr Integer NOT NULL ,
    id Integer NOT NULL ,
primary key (atr,id)
);
```

```
Alter table Tab2 add foreign key (id) references Tab (id) ;
```

Kardinalita vzt'ahu



Insert into TAB values(1);

Insert into TAB2 values(1,1);

Insert into TAB2 values(2,1);

Kardinalita vzt'ahu

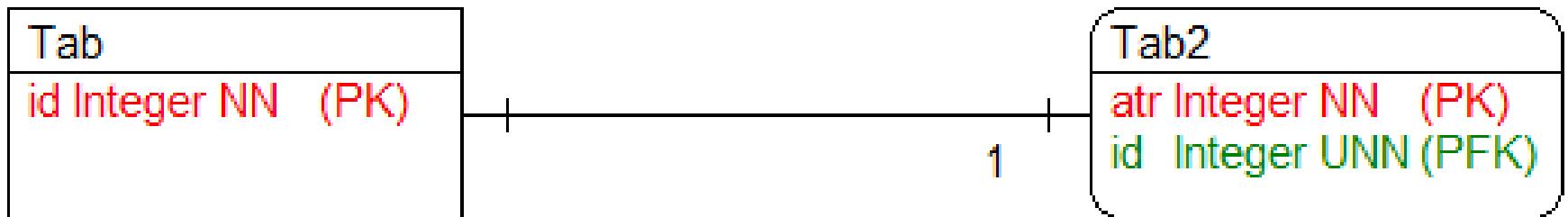


```
Create table Tab (
    id Integer NOT NULL ,
primary key (id)
);
```

```
Create table Tab2 (
    atr Integer NOT NULL ,
    id Integer NOT NULL UNIQUE ,
primary key (atr,id)
);
```

```
Alter table Tab2 add foreign key (id) references Tab (id);
```

Kardinalita vzt'ahu



Insert into TAB values(1);

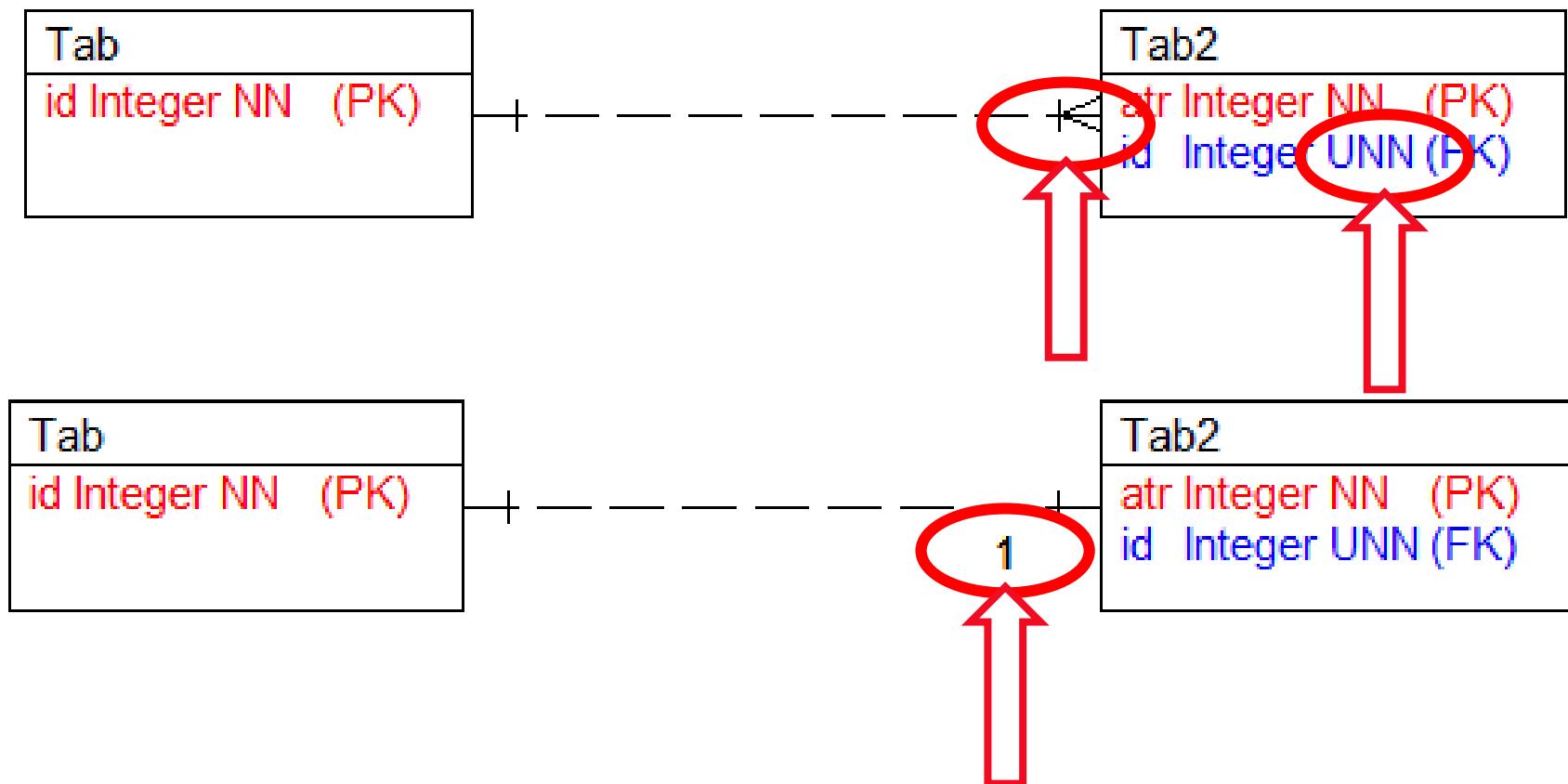
Insert into TAB2 values(1,1);

Insert into TAB2 values(2,1);

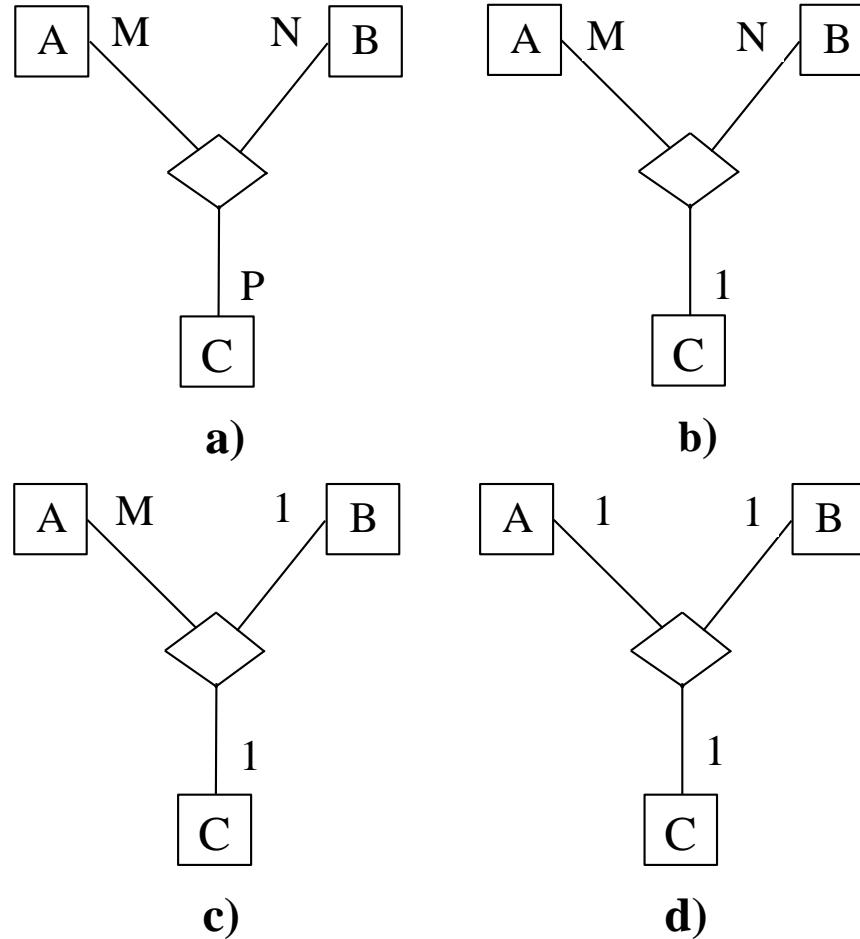
ERROR at line 1:

ORA-00001: unique constraint (KVET1.SYS_C00367481) violated

Kardinalita vzt'ahu



Ternárny vzťah



v DBS Oracle nie je možné modelovať takýto vzťah

Členstvo vo vzťahu

Definícia – Členstvo vo vzťahu

Členstvo vo vzťahu je integritné obmedzenie, ktoré vyjadruje nutnosť existencie, resp. neexistencie entity jedného typu vo vzťahu k existencii entity iného typu.



Povinné členstvo vo vztahu

Definícia –

Povinné členstvo vo vztahu je integritné obmedzenie, ktoré vyjadruje nutnosť existencie entity jedného typu vo vztahu v prípade existencie entity iného typu.

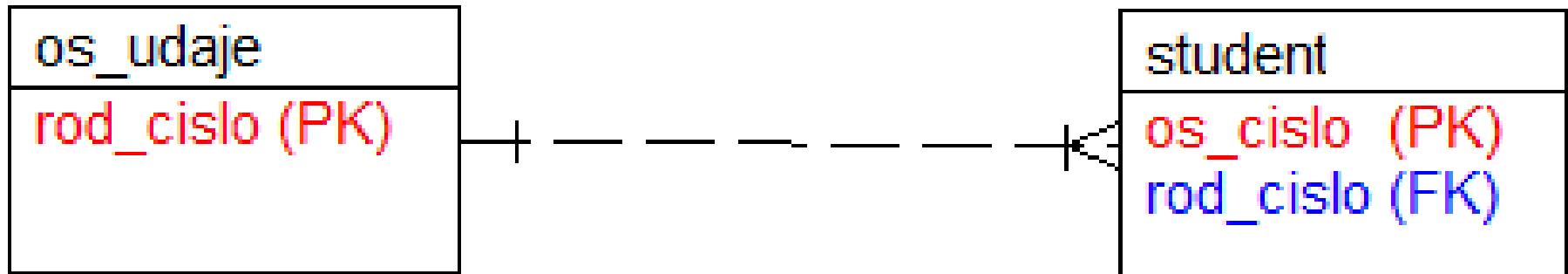


Nepovinné členstvo vo vztahu

Definícia –

Nepovinné členstvo vo vztahu je integritné obmedzenie, ktoré vyjadruje, že entita jedného typu nemusí existovať vo vztahu v prípade existencie entity iného typu.

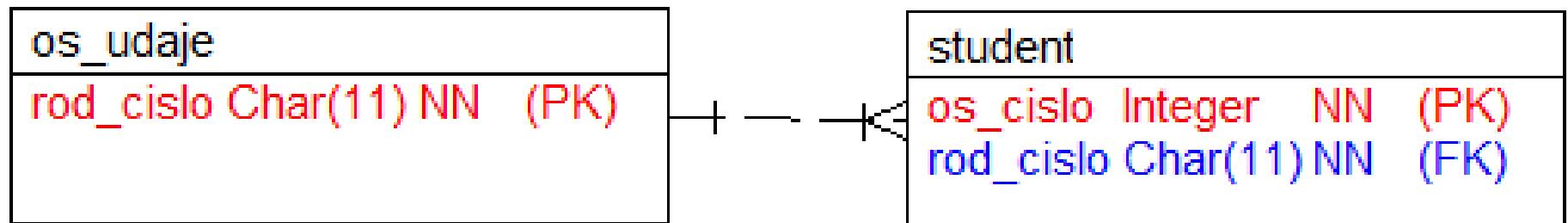
Členstvo vo vztahu



POVINNÉ ČLENSTVO

Študent musí mať vyplnené osobné údaje, osoba musí byť študentom

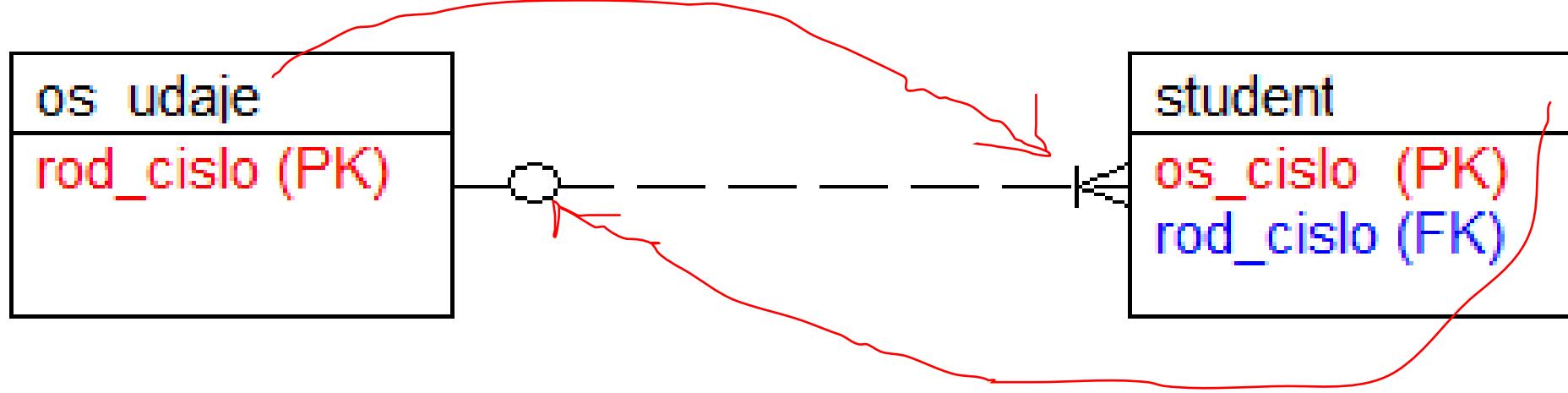
Členstvo vo vztahu



POVINNÉ ČLENSTVO

Študent musí mať vyplnené osobné údaje, osoba musí byť študentom

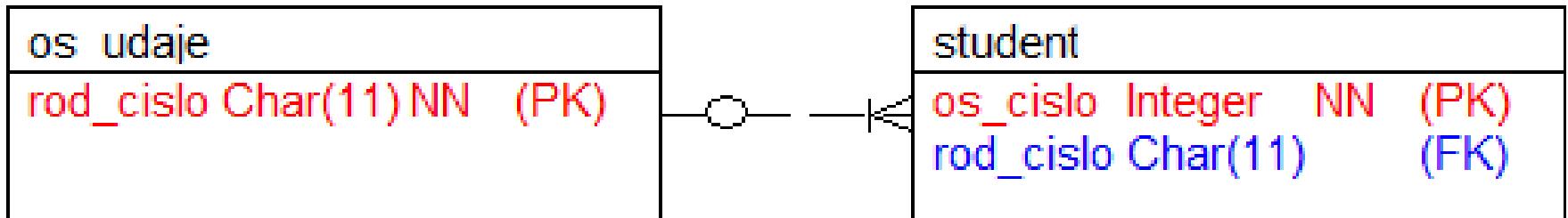
Členstvo vo vzt'ahu



NEPOVINNÉ ČLENSTVO

Študent **NEMUSÍ** mať vyplnené osobné údaje, osoba musí byť študentom

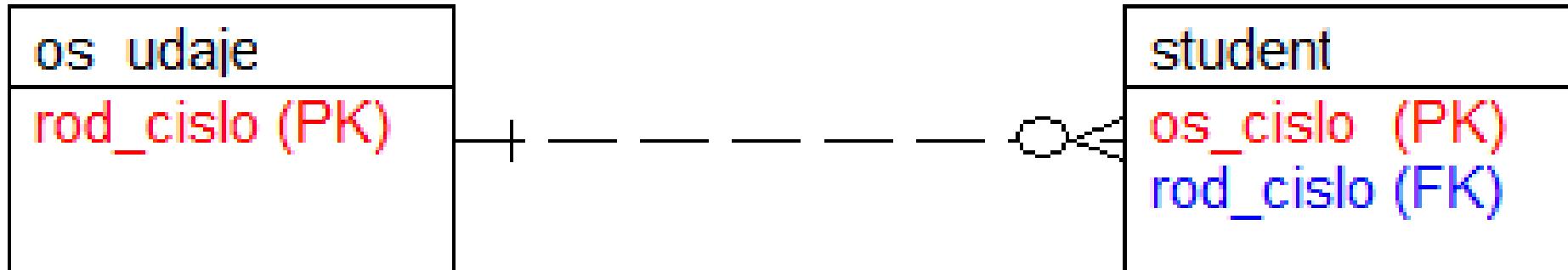
Členstvo vo vztahu



NEPOVINNÉ ČLENSTVO

Študent **NEMUSÍ** mať vyplnené osobné údaje, osoba musí byť študentom

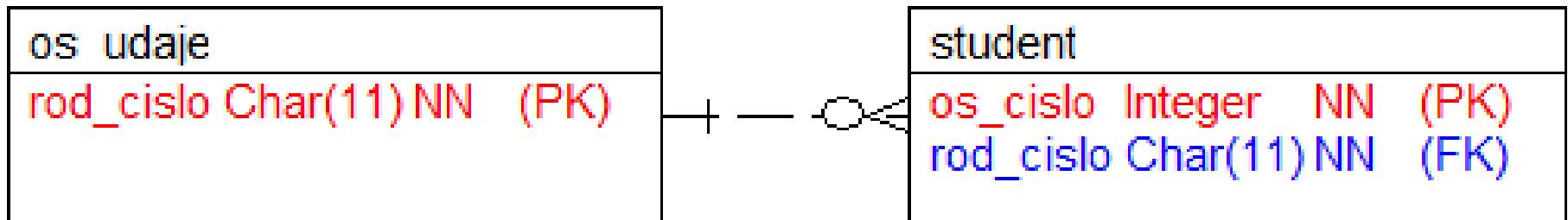
Členstvo vo vztahu



NEPOVINNÉ ČLENSTVO

Študent musí mať vyplnené osobné údaje, osoba **NEMUSÍ** byť študentom

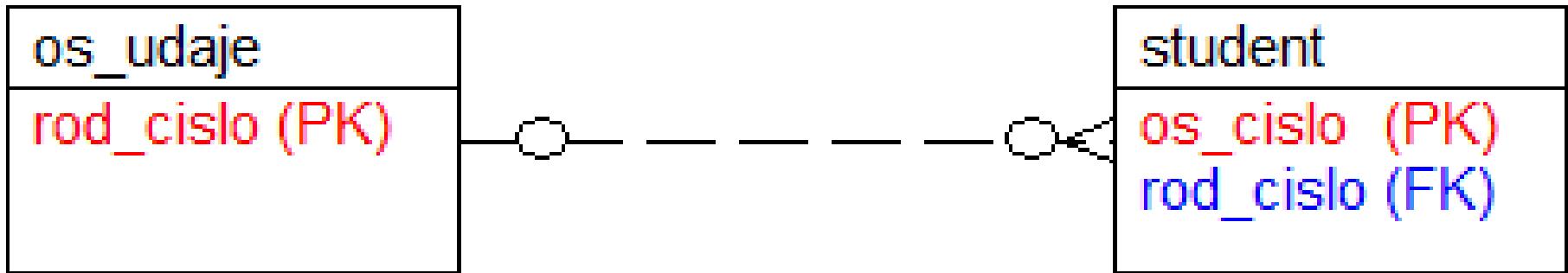
Členstvo vo vztahu



NEPOVINNÉ ČLENSTVO

Študent musí mať vyplnené osobné údaje, osoba **NEMUSÍ** byť študentom

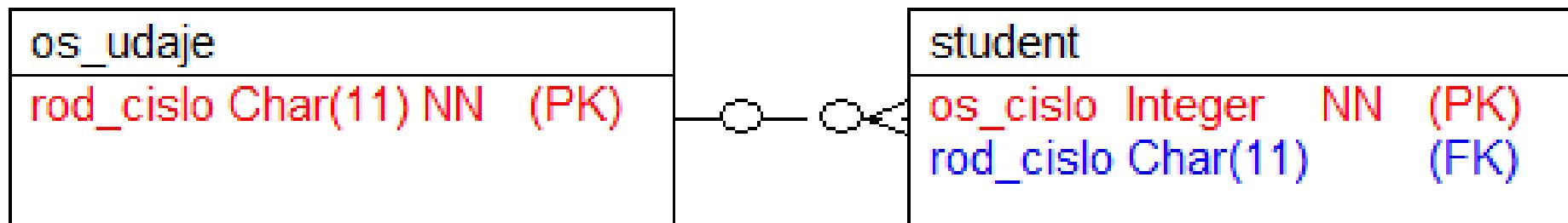
Členstvo vo vzt'ahu



NEPOVINNÉ ČLENSTVO

Študent **NEmusí** mať vyplnené osobné údaje, osoba **NEmusí** byť študentom

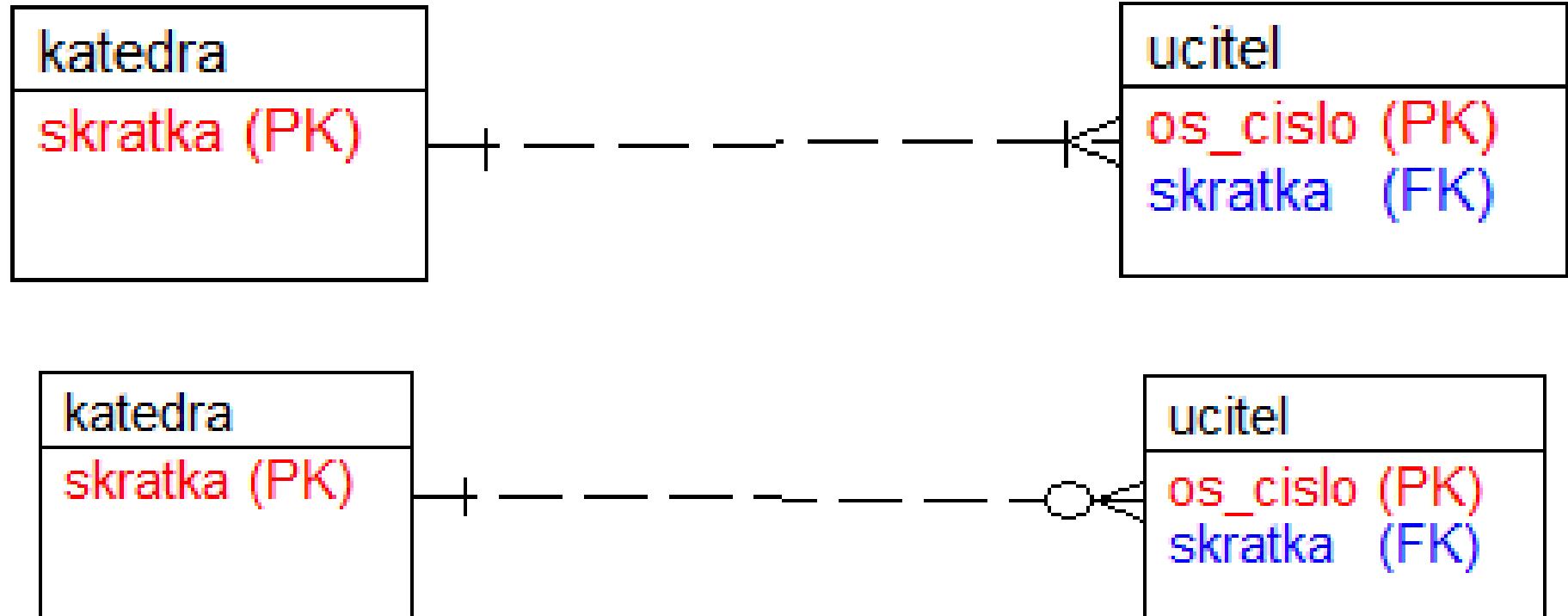
Členstvo vo vztahu



NEPOVINNÉ ČLENSTVO

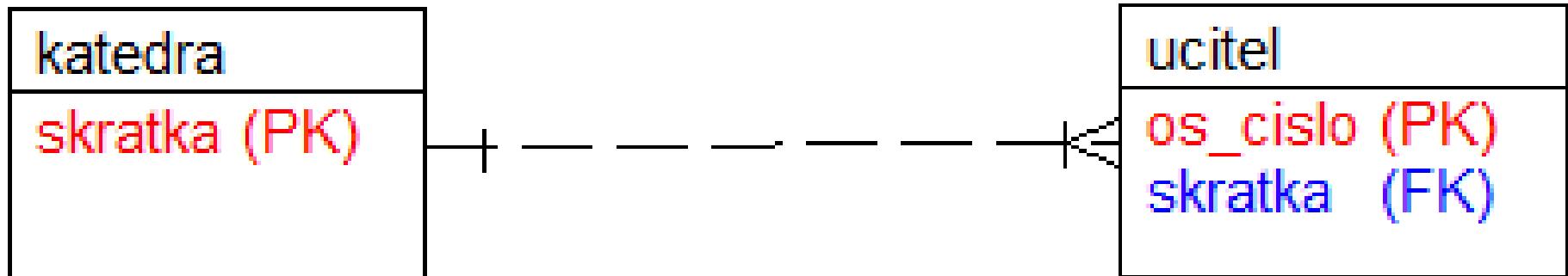
Študent **N**Emusí mať vyplnené osobné údaje, osoba **N**Emusí byť študentom

Členstvo vo vztahu



Katedra nemusí mať nejakých učiteľov,
ale učiteľ musí byť pridelený na práve jednu katedru.

Členstvo vo vztahu



```
Create table katedra (
    skratka Char (3) NOT NULL ,
primary key (skratka)
);
```

```
Create table ucitel (
    os_cislo Integer NOT NULL ,
    skratka Char (3) NOT NULL ,
primary key (os_cislo)
);
```

```
Alter table ucitel add foreign key (skratka) references
katedra (skratka);
```

SQL> insert into katedra values('KST');

1 row created.

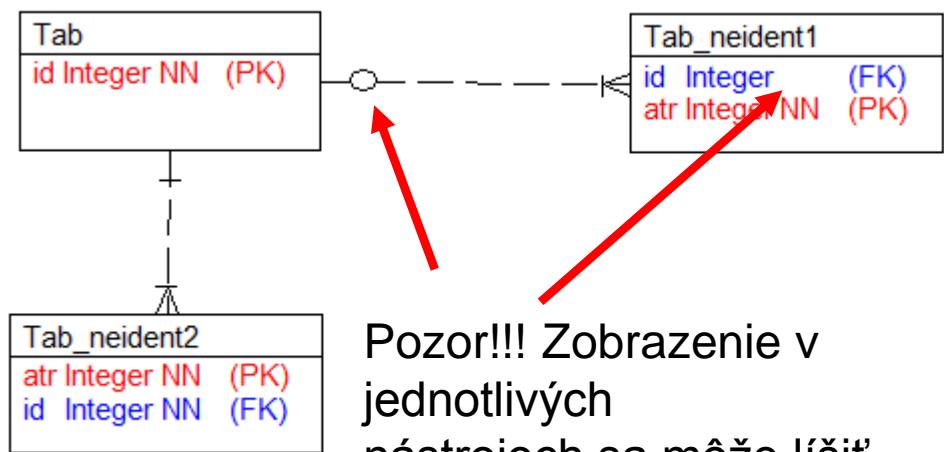
**--napriek povinnému členstvu v tab.
ucitel**

Členstvo vo vztahu

```
Create table Tab (
    id Integer NOT NULL ,
primary key (id)
)
/
```

```
Create table Tab_neident1 (
    id Integer,
    atr Integer NOT NULL ,
primary key (atr)
)
/
```

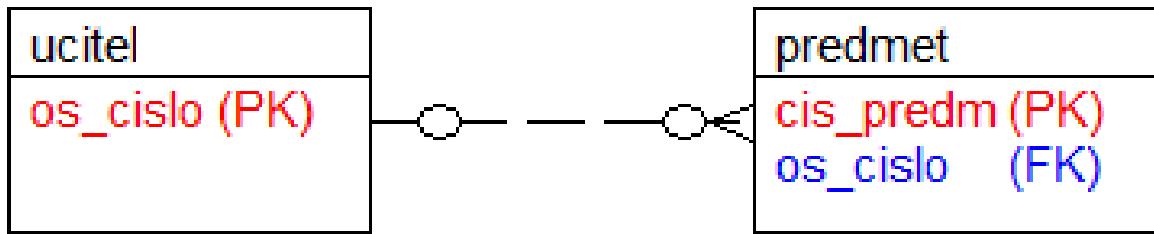
```
Create table Tab_neident2 (
    atr Integer NOT NULL ,
    id Integer NOT NULL ,
primary key (atr)
)
/
```



Pozor!!! Zobrazenie v jednotlivých nástrojoch sa môže lísiť...

Slabé entitné typy





```

Create table ucitel (
    os_cislo Integer NOT NULL ,
primary key (os_cislo)
)
/

```

```

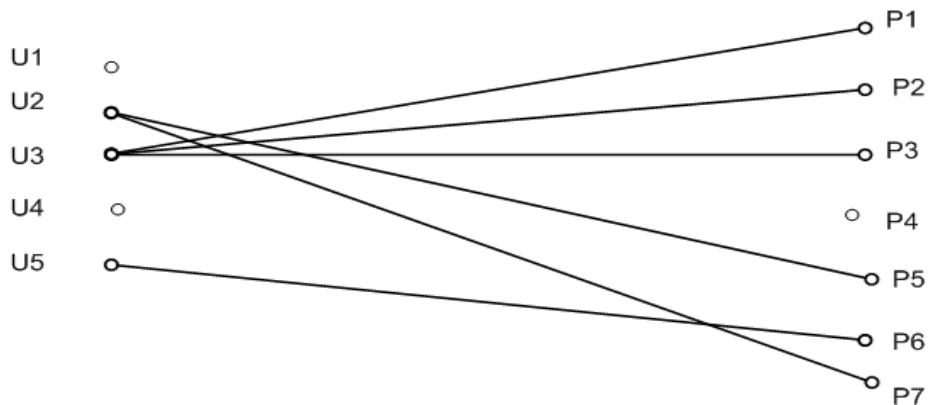
Create table predmet (
    cis_predm Char (4) NOT NULL ,
    os_cislo Integer,
primary key (cis_predm)
)
/

```

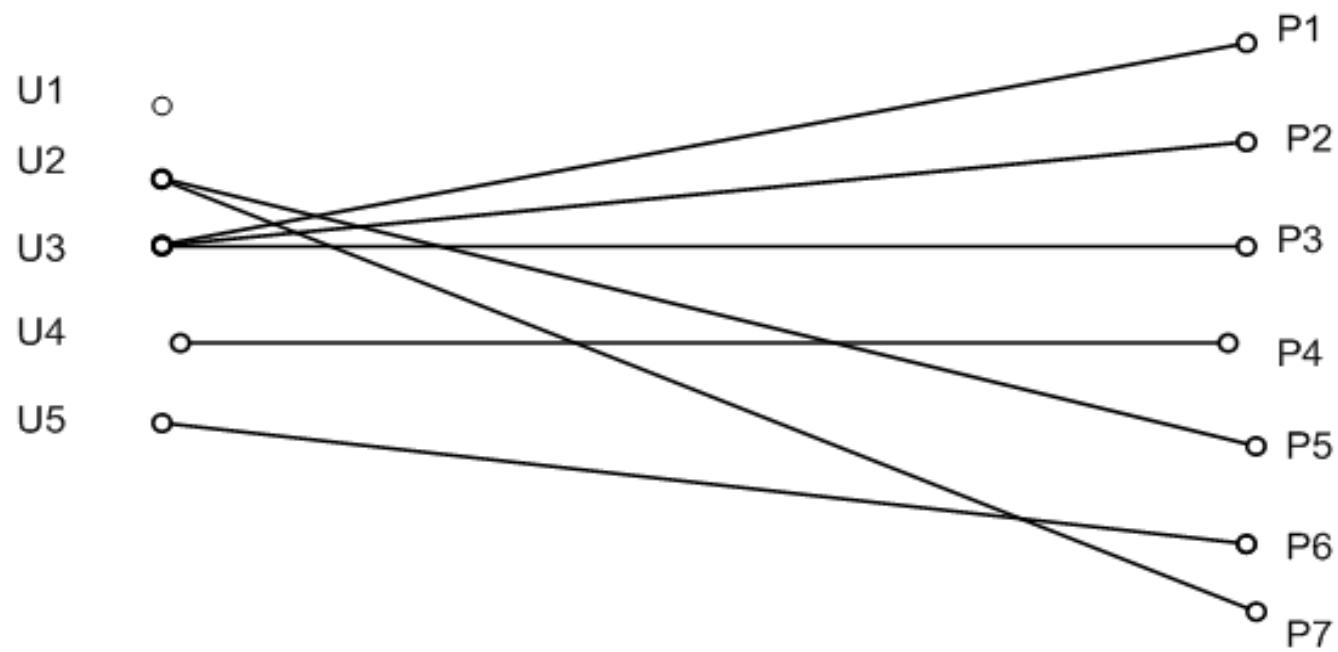
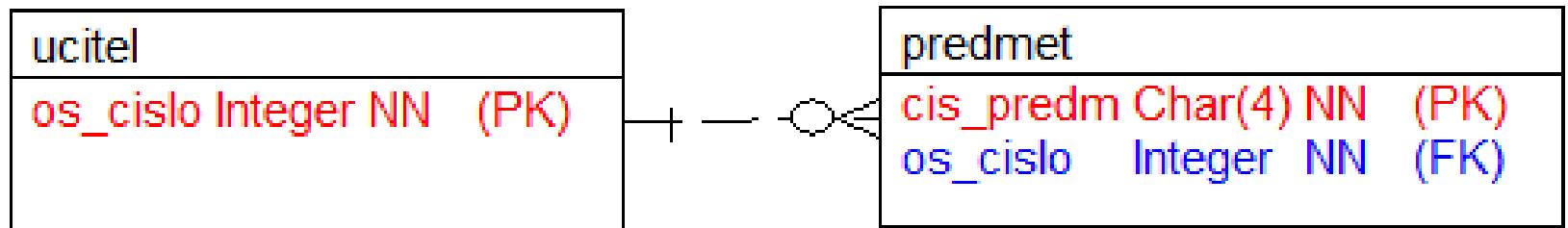
```

Alter table predmet add foreign key
(os_cislo) references ucitel (os_cislo)
/

```



Nepovinné – nepovinné členstvo



Nepovinné – povinné členstvo

Numerické vyjadrenie IO

Definícia –

Majme typ vzťahu

$R(E1:(\min, \max), E2:(\min, \max))$, kde

$E1: (\min, \max)$ a

$E2: (\min, \max)$

označuje minimálny a maximálny počet výskytov entity typu $E1$ a $E2$ vo vzťahovej množine R .

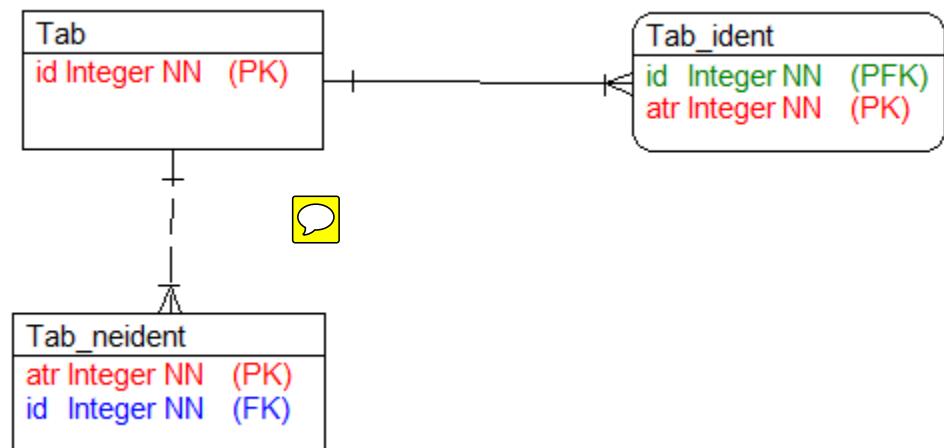


Identifikačný vs. neidentifikačný vztah

```
Create table Tab (
    id Integer NOT NULL ,
primary key (id)
)
/
```

```
Create table Tab_ident (
    id Integer NOT NULL ,
    atr Integer NOT NULL ,
primary key (id,atr)
)
/
```

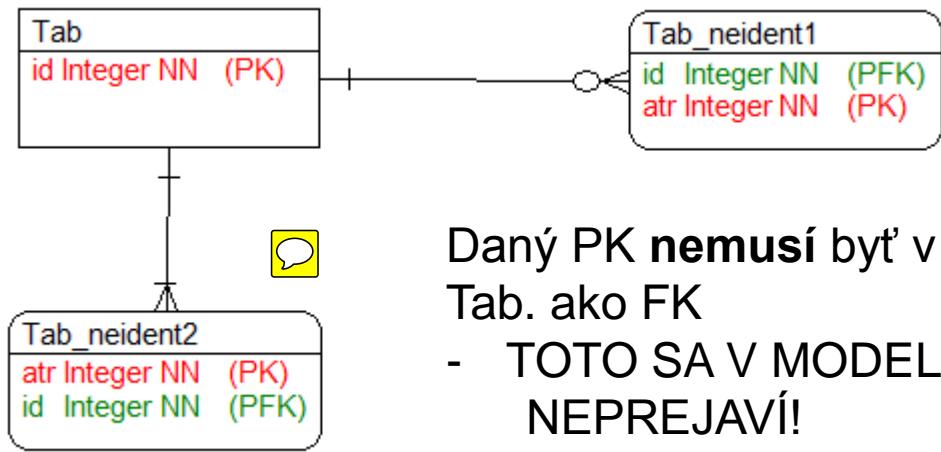
```
Create table Tab_neident (
    atr Integer NOT NULL ,
    id Integer NOT NULL ,
primary key (atr)
)
/
```



```
Create table Tab (
    id Integer NOT NULL ,
primary key (id)
)
/
```

```
Create table Tab_neident1 (
    id Integer NOT NULL ,
    atr Integer NOT NULL ,
primary key (id,atr)
)
/
```

```
Create table Tab_neident2 (
    atr Integer NOT NULL ,
    id Integer NOT NULL ,
primary key (atr,id)          --na poradí atribútov tvoriacich PK (ne)záleží!!!
)
/
```



Daný PK **nemusí** byť v druhej Tab. ako FK
- TOTO SA V MODELI NEPREJAVÍ!

Príklad

Pre binárne typy vzťahov je možné napr.
písat'

$R(E1: (1,1), E2: (1,1))$ zodpovedá 1: 1

$R(E1: (0,n), E2: (1,1))$ 1: N

$R(E1: (0,n), E2: (0,1))$ (0 alebo 1) : N

$R(E1: (0,n), E2: (0,m))$ M : N

Príklad

PREDMETY_Sem(ŠTUDENT:(0,20), PREDMET:(0,15))

Konkrétny študent nemusí mať v tomto semestri zapísaný žiadny predmet, ale môže mať súčasne zapísaných maximálne len 15 predmetov.

Na konkrétny predmet semestra je obmedzená kapacita na 20 ľudí.

Príklad

ZAP_PR(ŠTUDENT:(0,m),PREDMET:(0,60))

Konkrétny študent môže mať počas celého štúdia zapísaných maximálne 60 predmetov, pričom počet študentov na daný predmet počas existencie fakulty je neobmedzený.

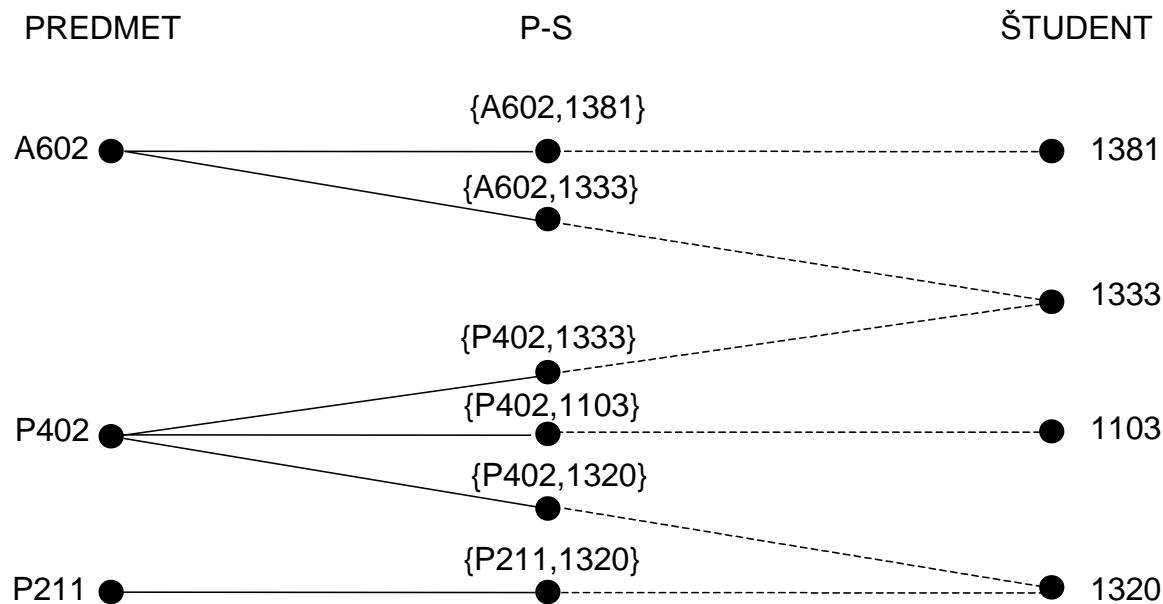
Príklad

ZAP_PRED(ŠTUDENT:(1,m), PREDMET:(1,n))

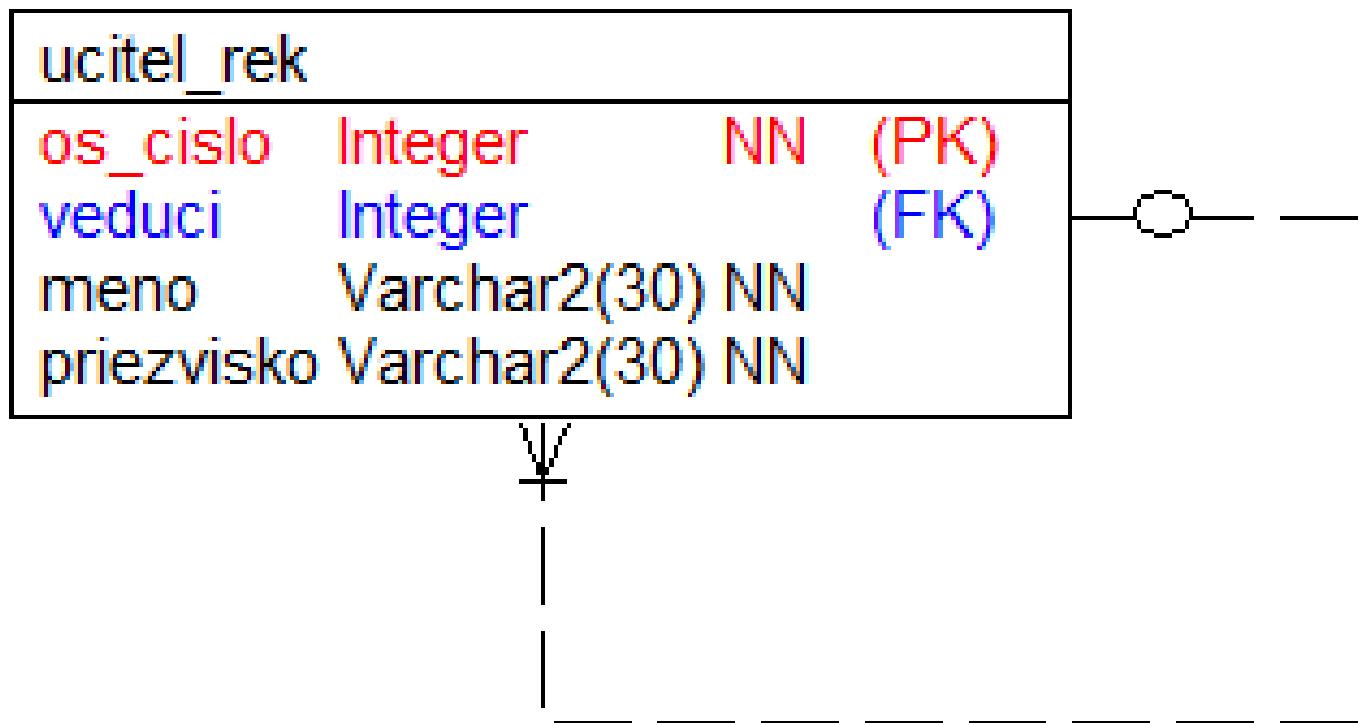
Tento prípad popisuje situáciu, kedy študent môže mať zapísaných niekoľko predmetov, avšak aspoň jeden. Pričom budeme evidovať len tie predmety, ktoré má niekto zapísané.

Dekompozícia M:N vztahu

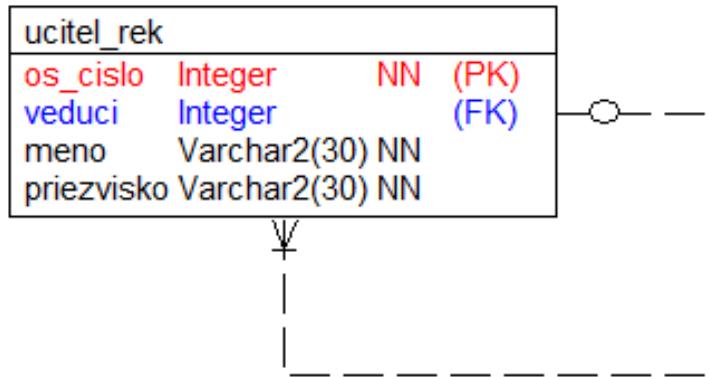
(a)



Rekurzívny vztah

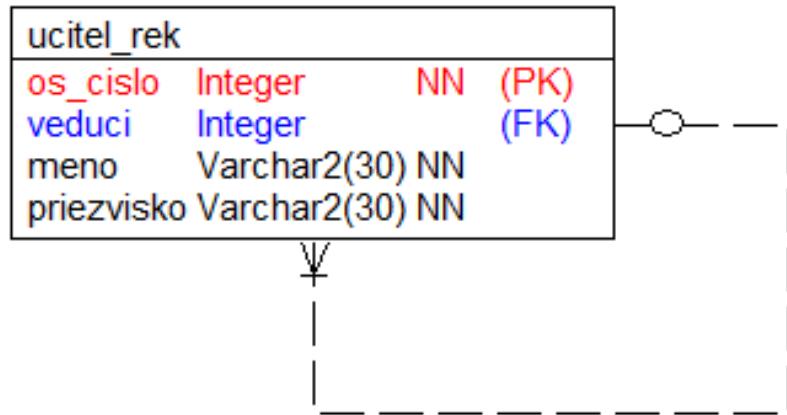


Rekurzívny vztah



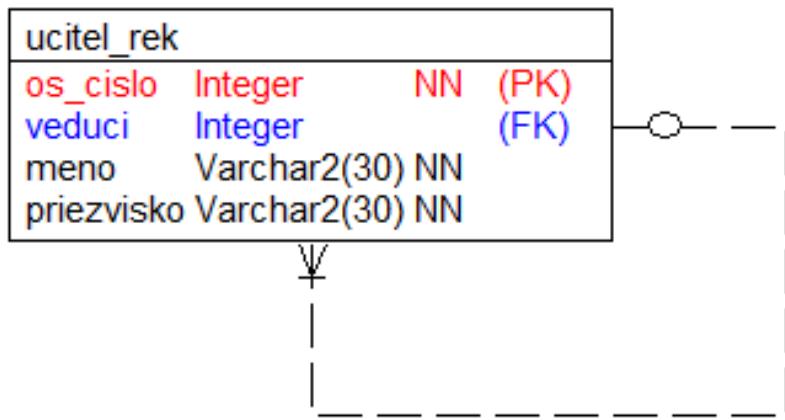
```
Create table ucitel_rek (
    os_cislo Integer NOT NULL ,
    veduci Integer,
    meno Varchar2 (30) NOT NULL ,
    priezvisko Varchar2 (30) NOT NULL ,
    primary key (os_cislo) ,
    foreign key (veduci) references ucitel_rek(os_cislo));
```

Rekurzívny vztah



os_cislo	veduci	meno	priezvisko
1	NULL	Tatiana	Čorejová
2	1	Karol	Matiaško
3	1	Emil	Kršík
4	2	Monika	Vajsová
5	2	Michal	Kvet

Rekurzívny vztah

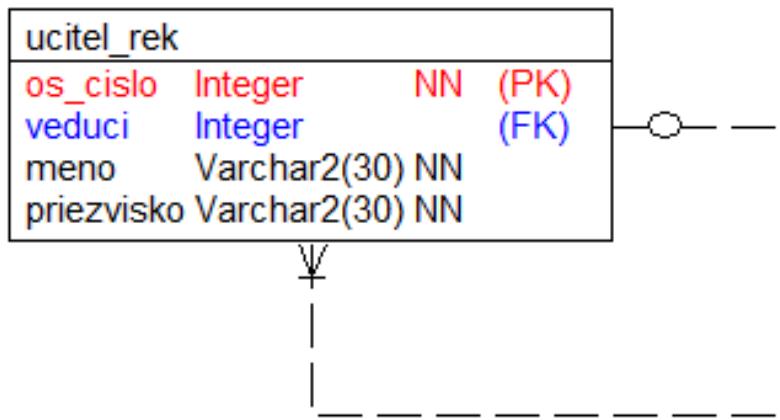


os_cislo	veduci	meno	priezvisko
1	NULL	Tatiana	Čorejová
2	1	Karol	Matiaško
3	1	Emil	Kršík
4	2	Monika	Vajsová
5	2	Michal	Kvet

Komu je vedúci Matiaško?

```
select u2.meno, u2.priezvisko  
from ucitel_rek u1 JOIN ucitel_rek u2 on (u1.os_cislo=u2.veduci)  
where u1.priezvisko like 'Matiasko%'
```

Rekurzívny vztah

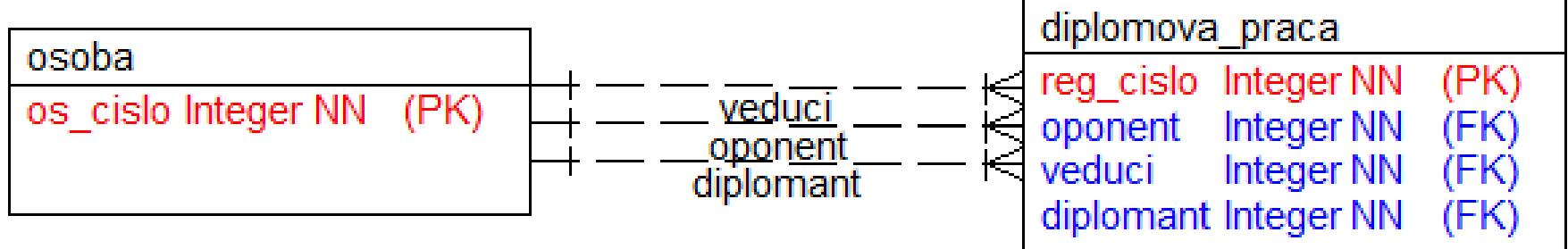


os_cislo	veduci	meno	priezvisko
1	NULL	Tatiana	Čorejová
2	1	Karol	Matiaško
3	1	Emil	Kršík
4	2	Monika	Vajsová
5	2	Michal	Kvet

Kto je vedúci Matiašovi?

```
select u2.meno, u2.priezvisko  
from ucitel_rek u1 JOIN ucitel_rek u2 on (u2.os_cislo=u1.veduci)  
where u1.priezvisko like 'Matiasko%'
```

Viacnásobný vztah



Alter table diplomova_praca add foreign key (**veduci**) references osoba (os_cislo)

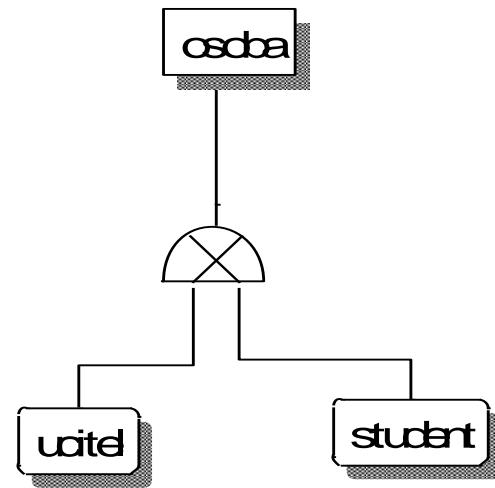
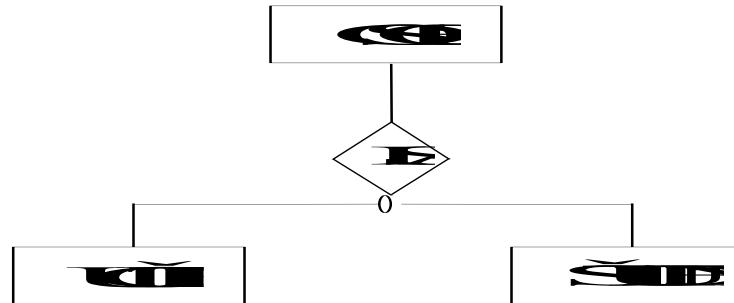
1

Alter table diplomova_praca add foreign key (**diplomant**) references osoba (os_cislo)

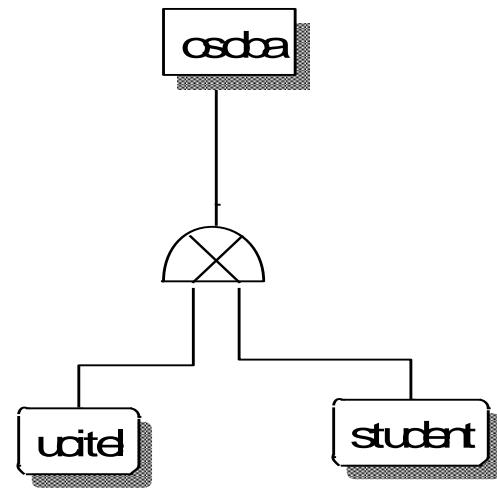
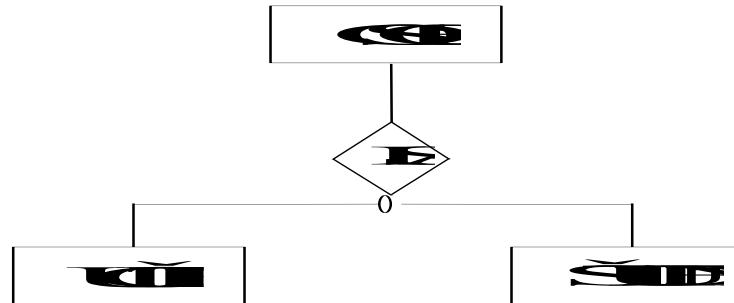
1

```
Alter table diplomova_praca add foreign key (ponent) references osoba  
(os_cislo)  
/
```

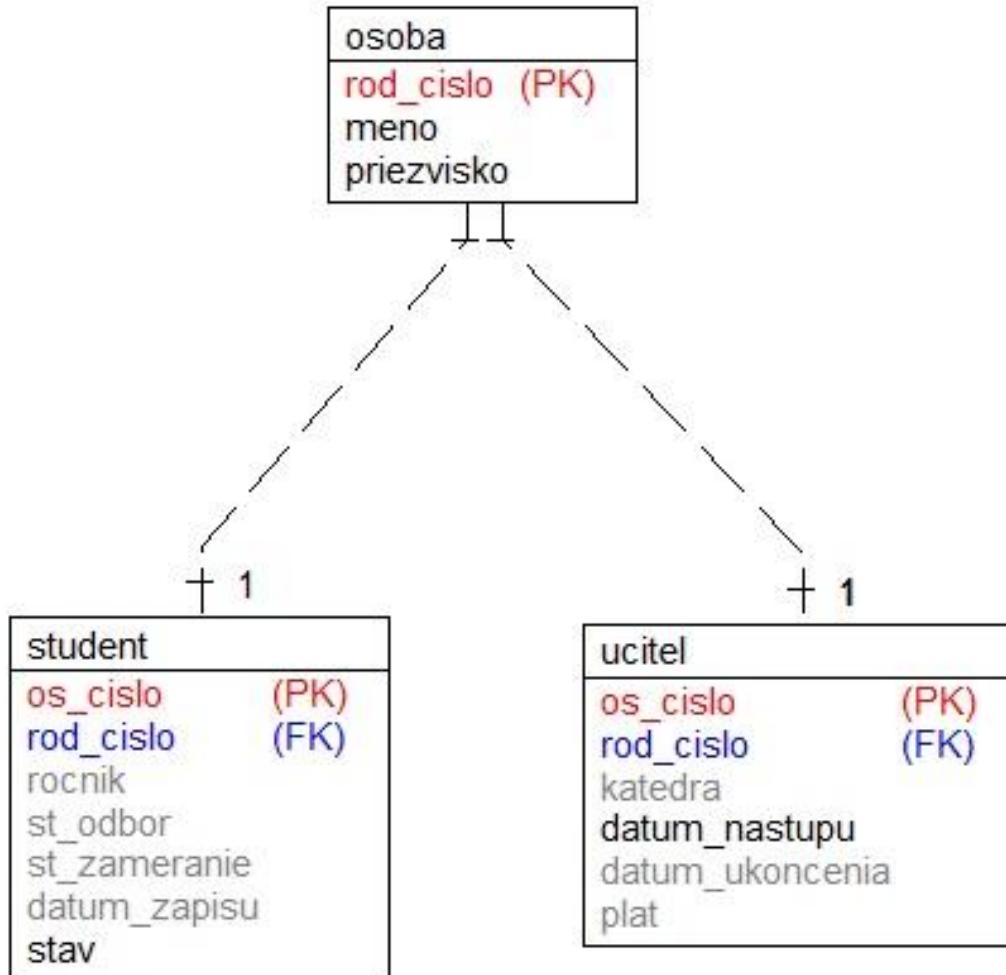
ISA hierarchia



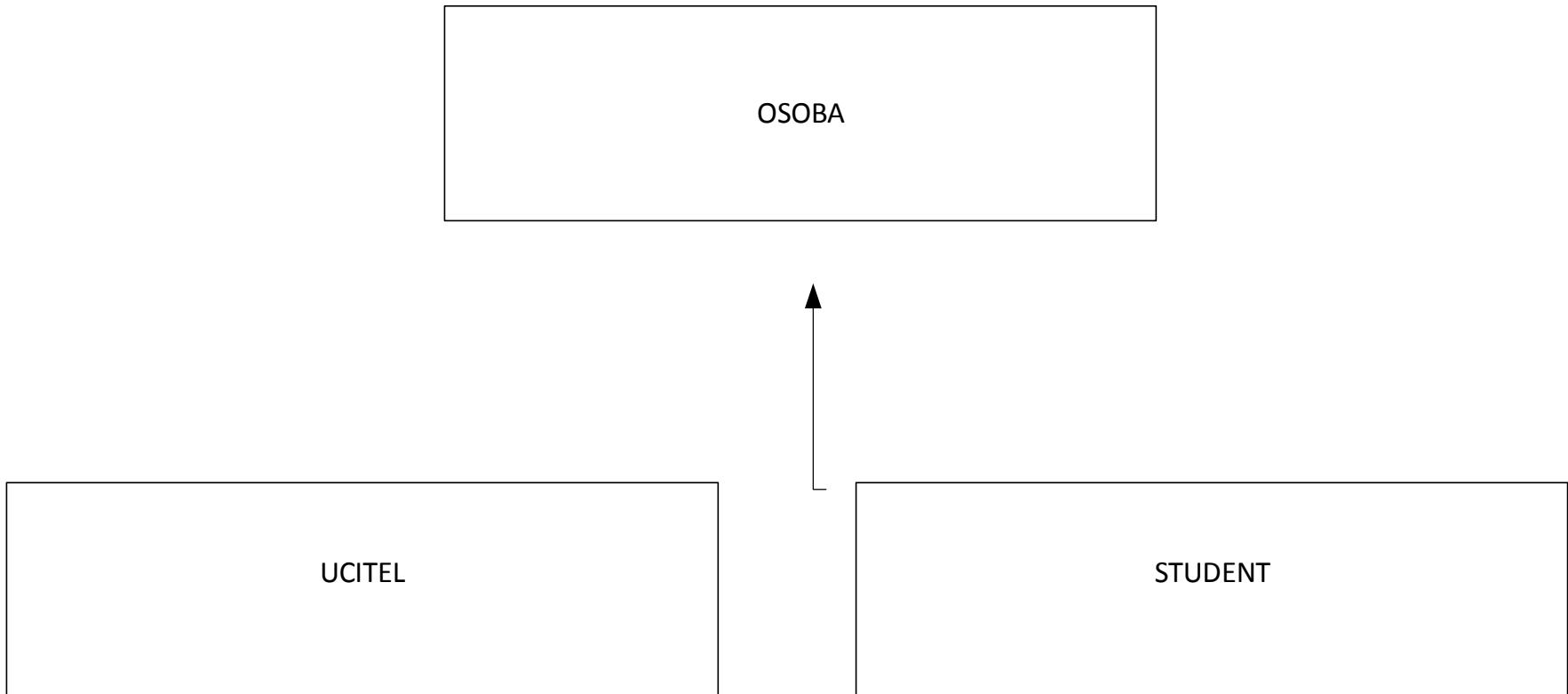
ISA hierarchia



ISA hierarchia



Generalizácia v ISA hierarchii



Špecializácia v ISA hierarchii

