

```

4780 GOTO 5000
4790 :
4800 REM
4801 REM
4802 REM
4803 REM
4810 :
4820 PRINT
4825 W=V+1
4830 FOR X
4835 FOR I
4840 PRINT
4850 NEXT:
4860 PRINT
4870 FOR I
4880 IF MD
(I+1);:GOT
4890 PRINT
4900 NEXT
4910 PRINT
4920 FOR I
4925 PRINT
4930 IF MD
Q";:GOTO 4
4935 PRINT
4940 NEXT:PRINT "
4950 PRINT"#####";
4960 FOR I=2 TO 24 STEP 2
4965 PRINT"|";
4970 IF MD$(I+W-1)=" " THEN PRINT"
M$(I)"|";:GOTO 4980
4975 PRINT M$(I);
4980 NEXT:PRINT"

```

Databázové patterny

RNDr. Ondřej Zýka



```

4780 GOTO 5000
4790 :
4800 REM -----
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM -----
4810 :
4820 PRINT" ";
4825 W=V+1:IF W<8 THEN W=W+14
4830 FOR X=1 TO 2:PRINT" ";
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT" ";
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT M$(
(I+1));GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT" ";
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)
Q";:GOTO 4940
4935 PRINT" ";
4940 NEXT:PRINT" ";
4950 PRINT" ";
4960 FOR I=2 TO 24
4965 PRINT"|";
4970 IF MD$(I+W-1)
M$(I)"|";:GOTO 4980
4975 PRINT M$(I);
4980 NEXT:PRINT" ";

```



Co to je databázový pettern



Databázové patterny

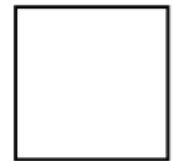
- Odzkoušené a doporučené způsoby, jak řešit často se vyskytující požadavky
- Jednoduché
 - N-ární relace
- Dědičnost
- Katalog
- Přiřazení rolí
- Klasifikace

Úrovně patternů

- Stejný typ požadavků může být řešen v databázi mnoha způsoby
- Jednoduše – výhodné pro uživatele
 - Lehce srozumitelné uživatelům, analytikům, vývojářům.
 - I při drobné změně požadavku je nutný zásah do databáze,
- Složitě – výhodné pro vývojáře
 - Hodně změn se dá vyřešit pouze změnou dat.
 - Komplikované datové struktury, uživatelsky nesrozumitelné.
 - Vždy je nutné mít jednoduché uživatelské rozhraní.
 - Koncový uživatel nesmí být zatěžován implementační složitostí.

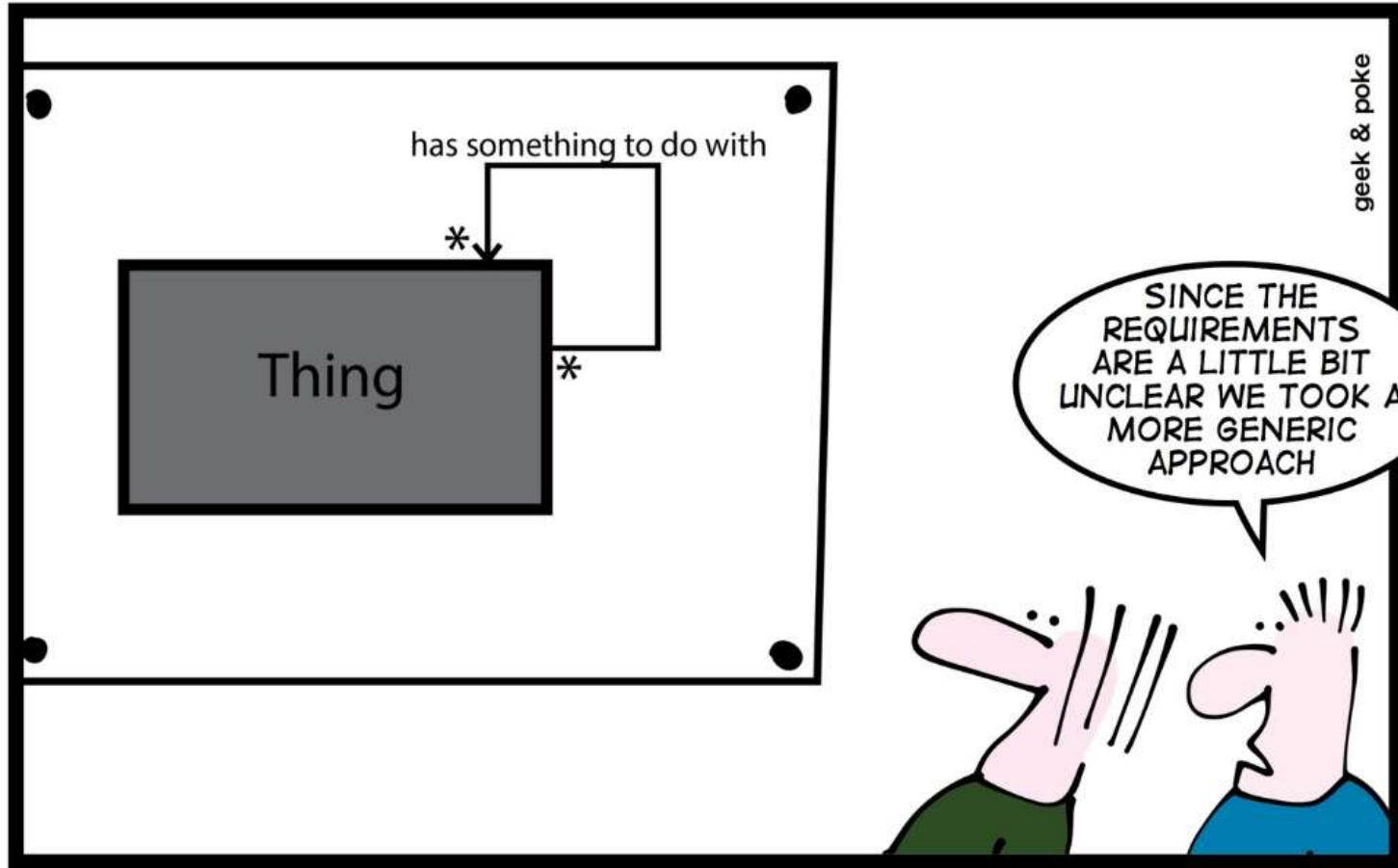


Fixní struktura



Dynamická struktura

Úrovně patternů



HOW TO CREATE A STABLE DATA MODEL

```

4780 GOTO 5000
4790 :
4800 REM -----
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM -----
4810 :
4820 PRINT" ";
4825 W=V+1:IF W<8 THEN W=W+14
4830 FOR X=1 TO 2:PRINT"XXXXXXXXXXXX";
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT"XXXXXXXXXXXX";
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT M$(
(I+1));:GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT"XXXXXXXXXXXX";
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)=
Q";:GOTO 4940
4935 PRINT" ";
4940 NEXT:PRINT" ";
4950 PRINT"XXXXXXXXXXXX";
4960 FOR I=2 TO 24
4965 PRINT"|";
4970 IF MD$(I+W-1)=
M$(I)"|";:GOTO 4980
4975 PRINT M$(I);
4980 NEXT:PRINT" ";

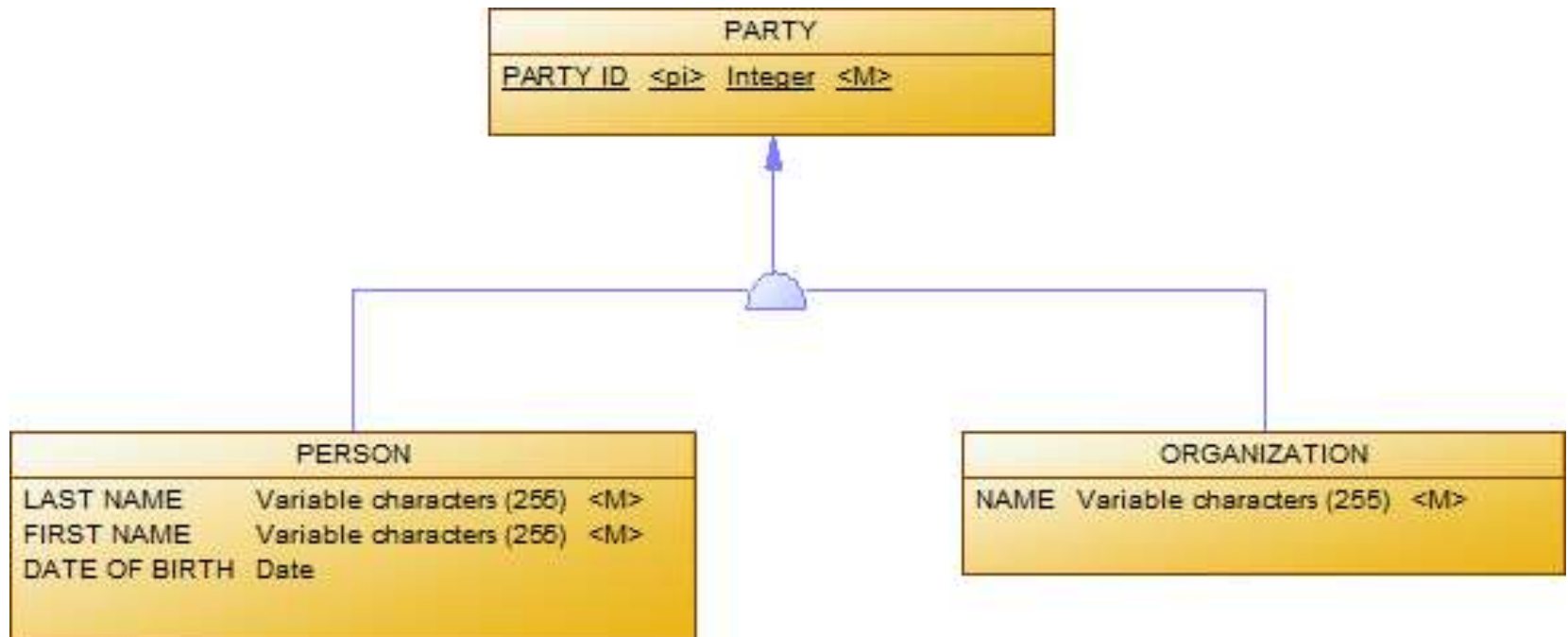
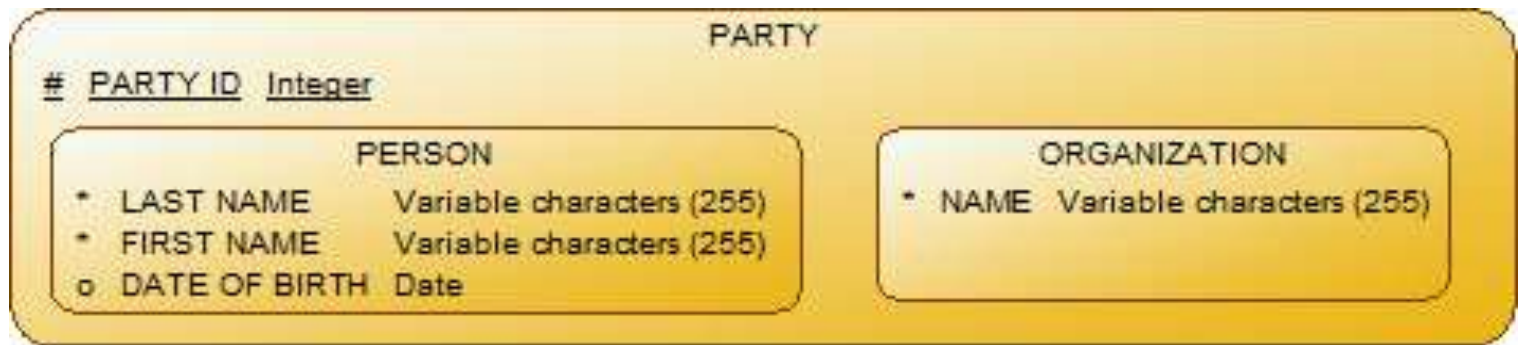
```



Dědičnost



Logický model PARTY



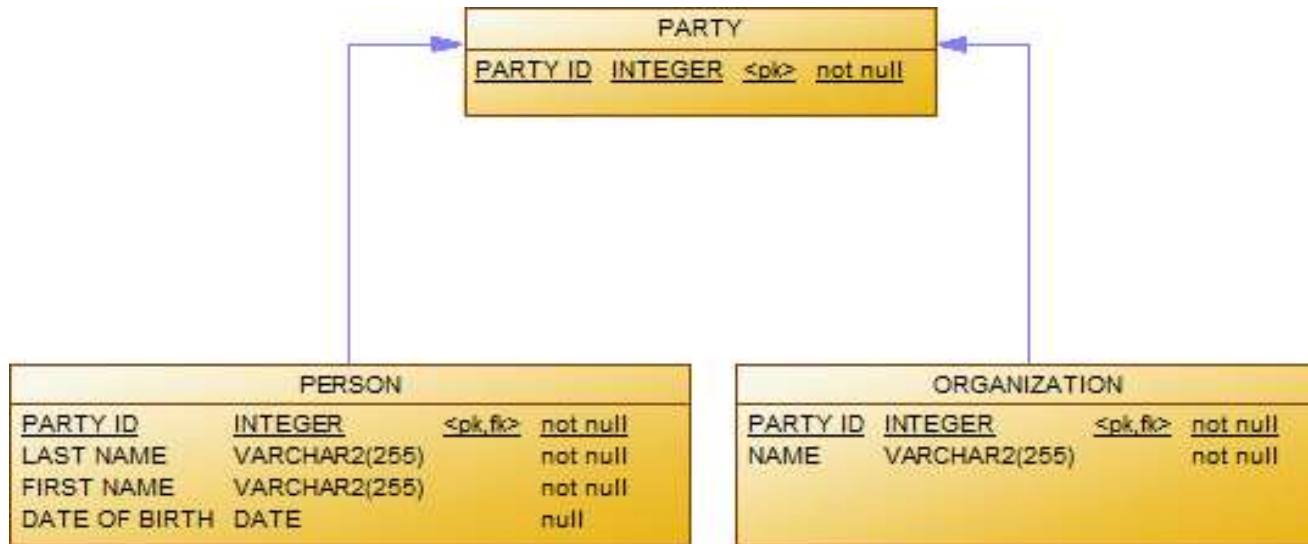
Fyzický model – Child only

PERSON			
<u>PARTY ID</u>	<u>INTEGER</u>	<pk>	not null
LAST NAME	VARCHAR2(255)		not null
FIRST NAME	VARCHAR2(255)		not null
DATE OF BIRTH	DATE		null

ORGANIZATION			
<u>PARTY ID</u>	<u>INTEGER</u>	<pk>	not null
NAME	VARCHAR2(255)		not null

- Jak vyřešit unikátnost ID
- Jak provést select přes všechny PARTY? (view?)

Fyzický model – Parent-Child



- Vhodné pro více společných atributů (v příkladu nejsou)

Fyzický model – Parent only (L-schéma)

PARTY			
<u>PARTY ID</u>	<u>INTEGER</u>	<u><pk></u>	<u>not null</u>
NAME	VARCHAR2(255)		null
LAST NAME	VARCHAR2(255)		null
FIRST NAME	VARCHAR2(255)		null
DATE OF BIRTH	DATE		null

- Jak zajistit požadavek na not null atributy v jednotlivých Child entitách?
- Jak zajistit vzájemnou výlučnost dědičnosti?
- Jak definovat typ Entity, pokud jednotlivé Child entity nemají různé not null atributy.

Fyzický model – Parent only se specifikátorem

PARTY			
<u>PARTY ID</u>	<u>INTEGER</u>	<u><pk></u>	<u>not null</u>
TYPE	VARCHAR2(255)		null
NAME	VARCHAR2(255)		null
LAST NAME	VARCHAR2(255)		null
FIRST NAME	VARCHAR2(255)		null
DATE OF BIRTH	DATE		null

- Specifikátor (TYPE) – možnost přidání číselníků
- Zajišťuje výlučnou dědičnost i při vícenásobné dědičnosti

```

4780 GOTO 5000
4790 :
4800 REM -----
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM -----
4810 :
4820 PRINT" ";
4825 W=V+1:IF W<8 THEN W=W+14
4830 FOR X=1 TO 2:PRINT"XXXXXXXXXXXX";
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT"XXXXXXXXXXXX";
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT M$(
(I+1));GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT"XXXXXXXXXXXX";
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)=
Q";:GOTO 4940
4935 PRINT" ";
4940 NEXT:PRINT"Q";
4950 PRINT"XXXXXXXXXXXX";
4960 FOR I=2 TO 24
4965 PRINT"|";
4970 IF MD$(I+W-1)=
M$(I)"Q";:GOTO 4980
4975 PRINT M$(I);
4980 NEXT:PRINT"Q";

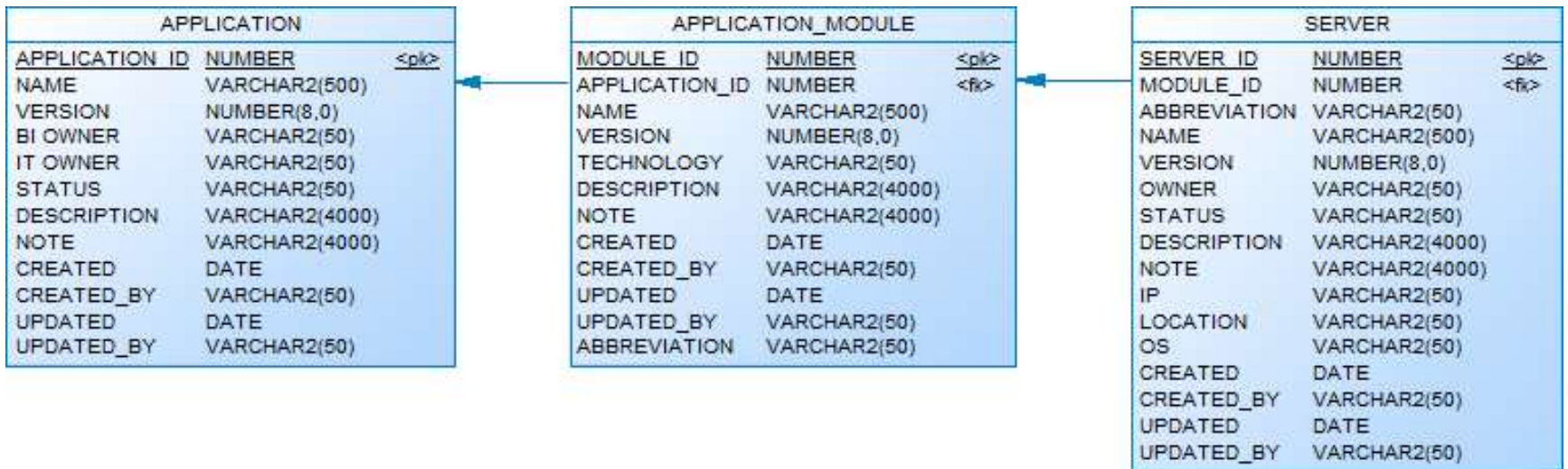
```



Metamodely

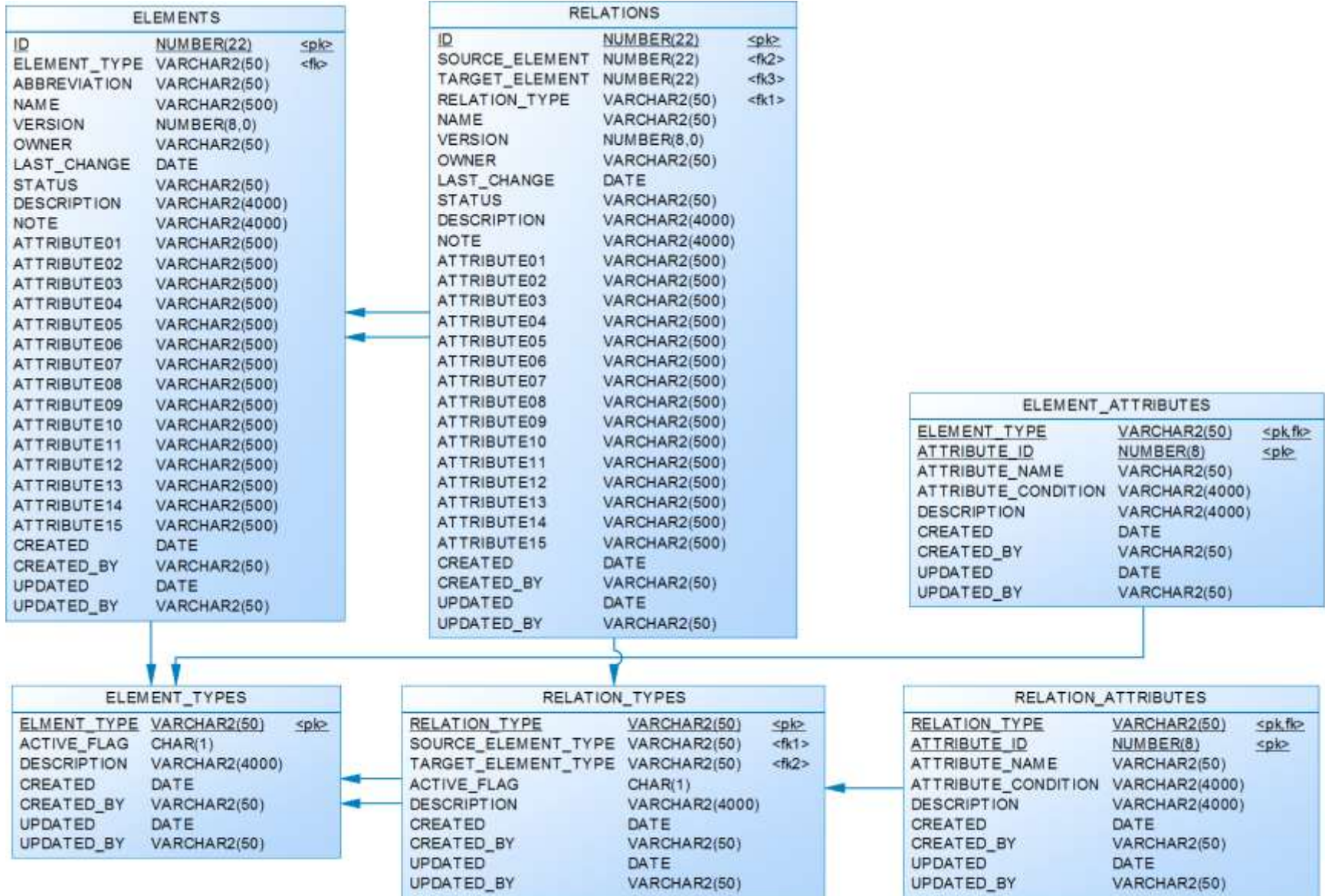


Evidence systémů – Patern I



- Srozumitelné
- Model kontroluje vazby, typy atributů, not null
- Model neumožňuje více typů vazeb mezi systémy (například vazby přenosů dat)
- Přidání atributu nebo entity vyžaduje zásah do modelu

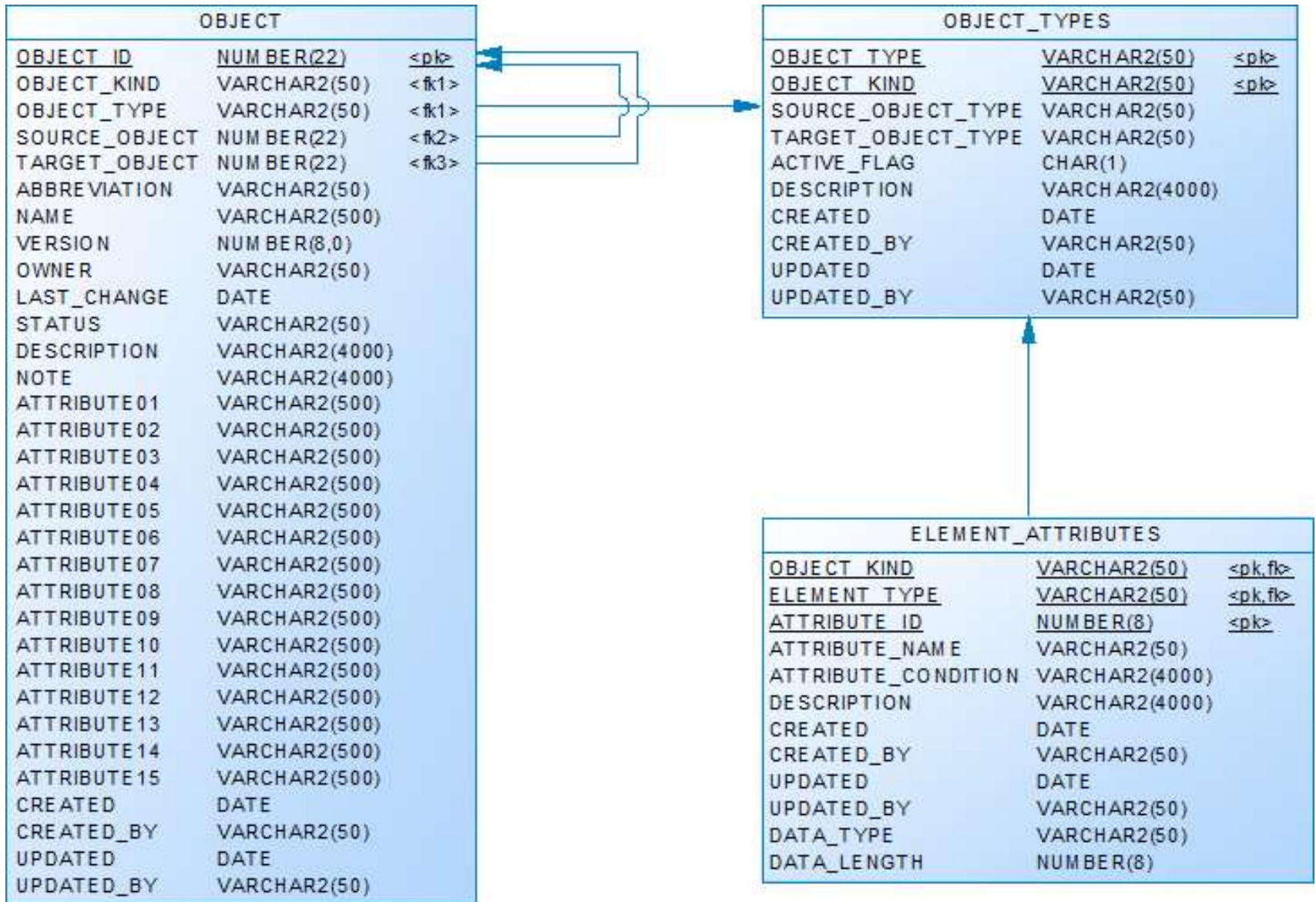
Evidence systémů – patern II



Evidence systémů – patern II

- Definice typů vazeb, vazby mají atributy
- Definice povolených vazeb pouze v datech, model neověřuje vazby mezi typy elementů, umožňuje ověření
- Definice atributů pouze v datech, model neověřuje typy, umožňuje ověření

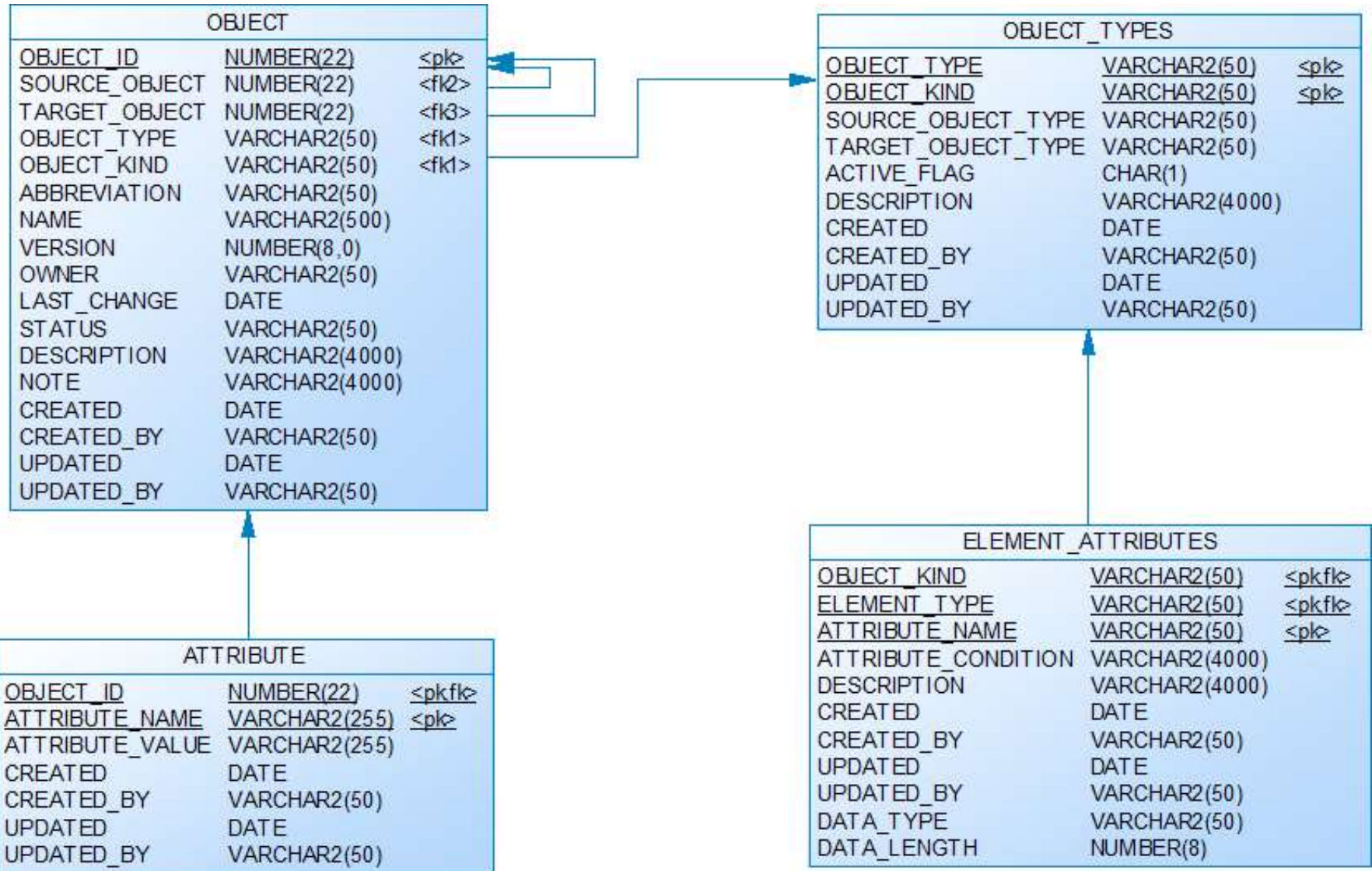
Evidence systémů – patern III



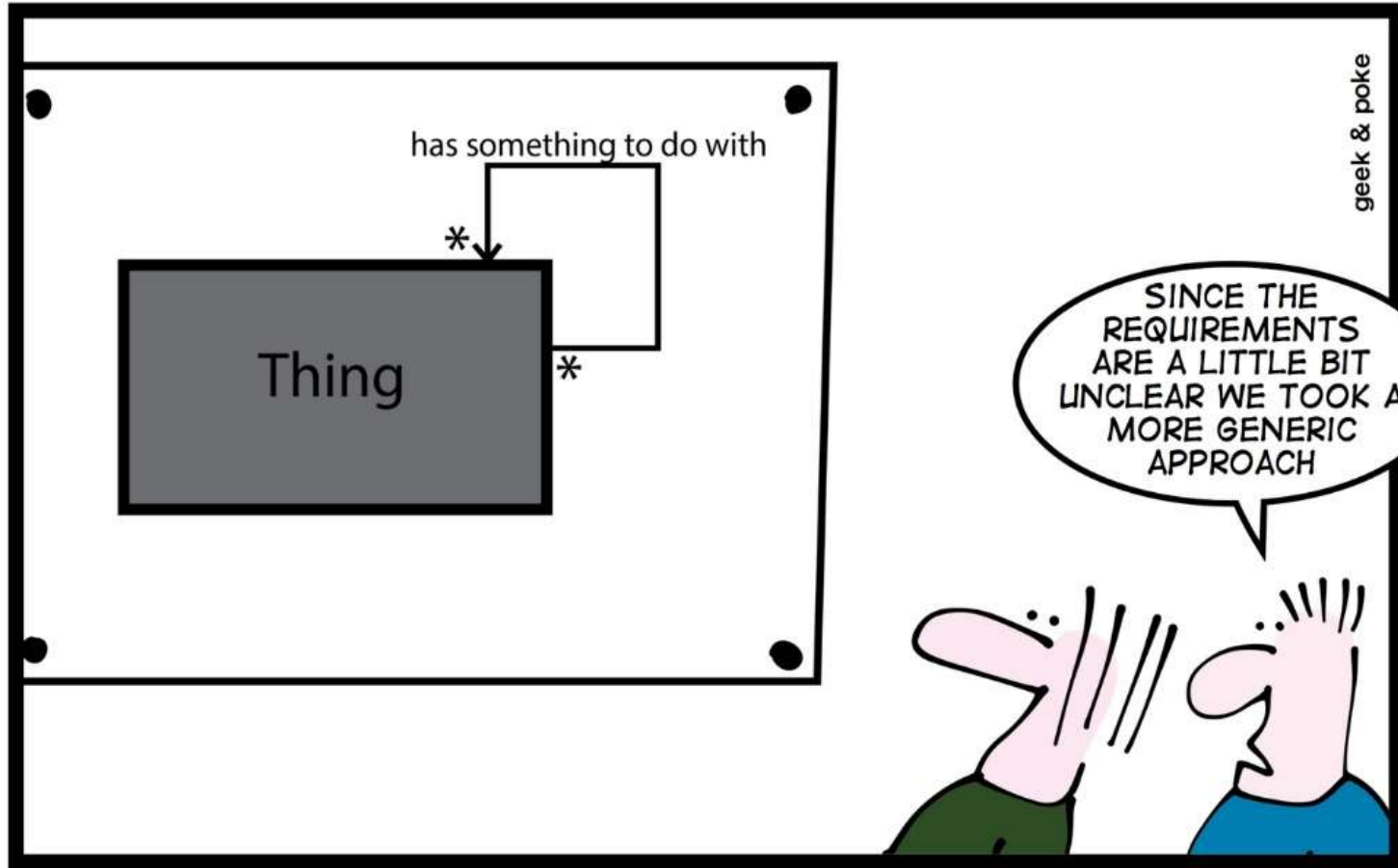
Evidence systémů – patern III

- Společná entita pro Elementy i Relace
- Snadné vyhledávání přes všechna data
- Definice povolených vazeb pouze v datech, model neověřuje vazby mezi typy elementů, umožňuje ověření
- Definice atributů pouze v datech, model neověřuje typy, umožňuje ověření
- Model nerozlišuje Elementy a Relace, nutnost kontrol podle OBJECT_KIND

Evidence systémů – patern IV



Evidence systémů – patern V

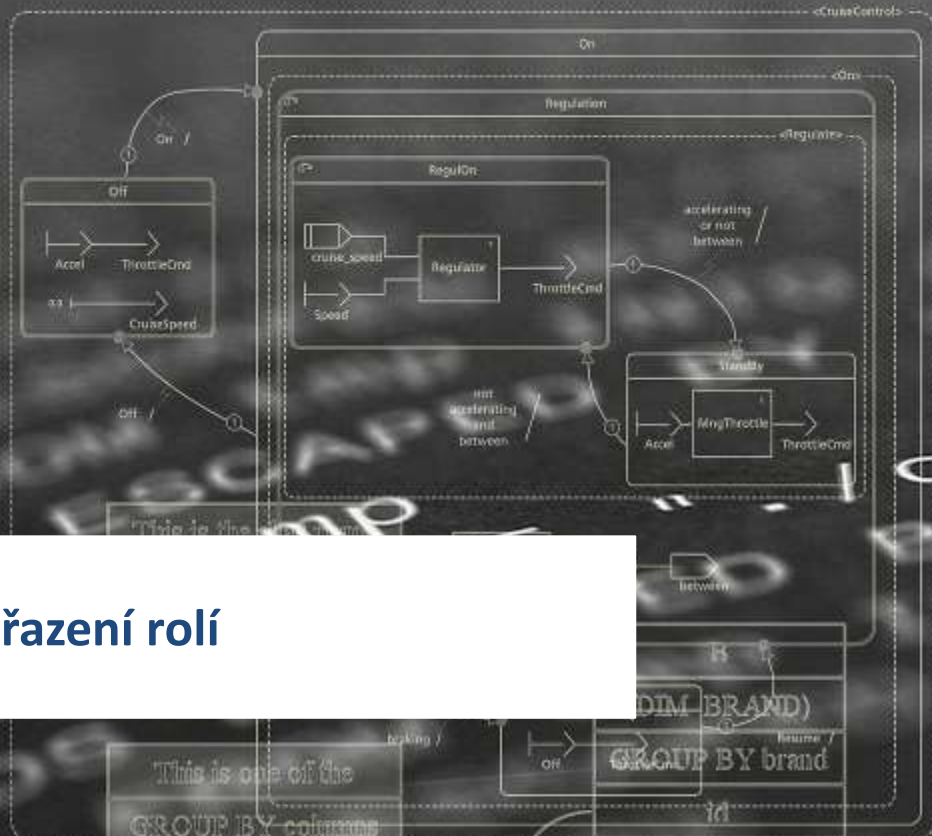


HOW TO CREATE A STABLE DATA MODEL

```

4780 GOTO 5000
4790 :
4800 REM -----
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM -----
4810 :
4820 PRINT"000";
4825 W=V+1:IF W<8 THEN W=W+14
4830 FOR X=1 TO 2:PRINT"#####";
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT"#####";
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT M$(
(I+1));GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT"#####";
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)
Q";:GOTO 4940
4935 PRINT" ";
4940 NEXT:PRINT"Q";
4950 PRINT"#####";
4960 FOR I=2 TO 24
4965 PRINT"|";
4970 IF MD$(I+W-1)="X" THEN PRINT"X"
M$(I)"Q";:GOTO 4980
4975 PRINT M$(I);
4980 NEXT:PRINT"Q";

```



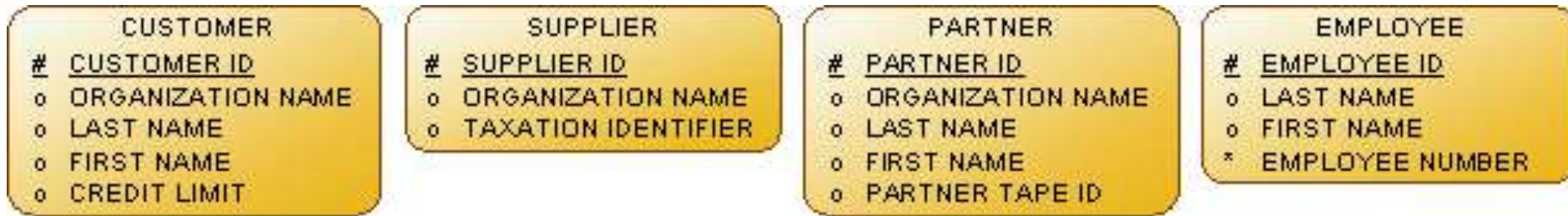
Pattern Přiřazení rolí



Pattern Přiřazení rolí

- Definice
 - Model pro správu zákazníků a dalších subjektů kooperující s podnikem
 - Příklady
 - Podnik
 - Zákazník
 - Dodavatel
 - Partner
 - Zaměstnanec
 - ...
 - Škola
 - Student
 - Zaměstnanec
 - Spolupracovník
 - Přednášející
 - ...

Pattern Přiřazení rolí I



CUSTOMER ID	ORGANIZATION/FIRST/LAST NAME	CREDIT LIMIT
100	Moje data s.r.o.	1000000 CZK
101	Tvoje Data s.r.o.	null

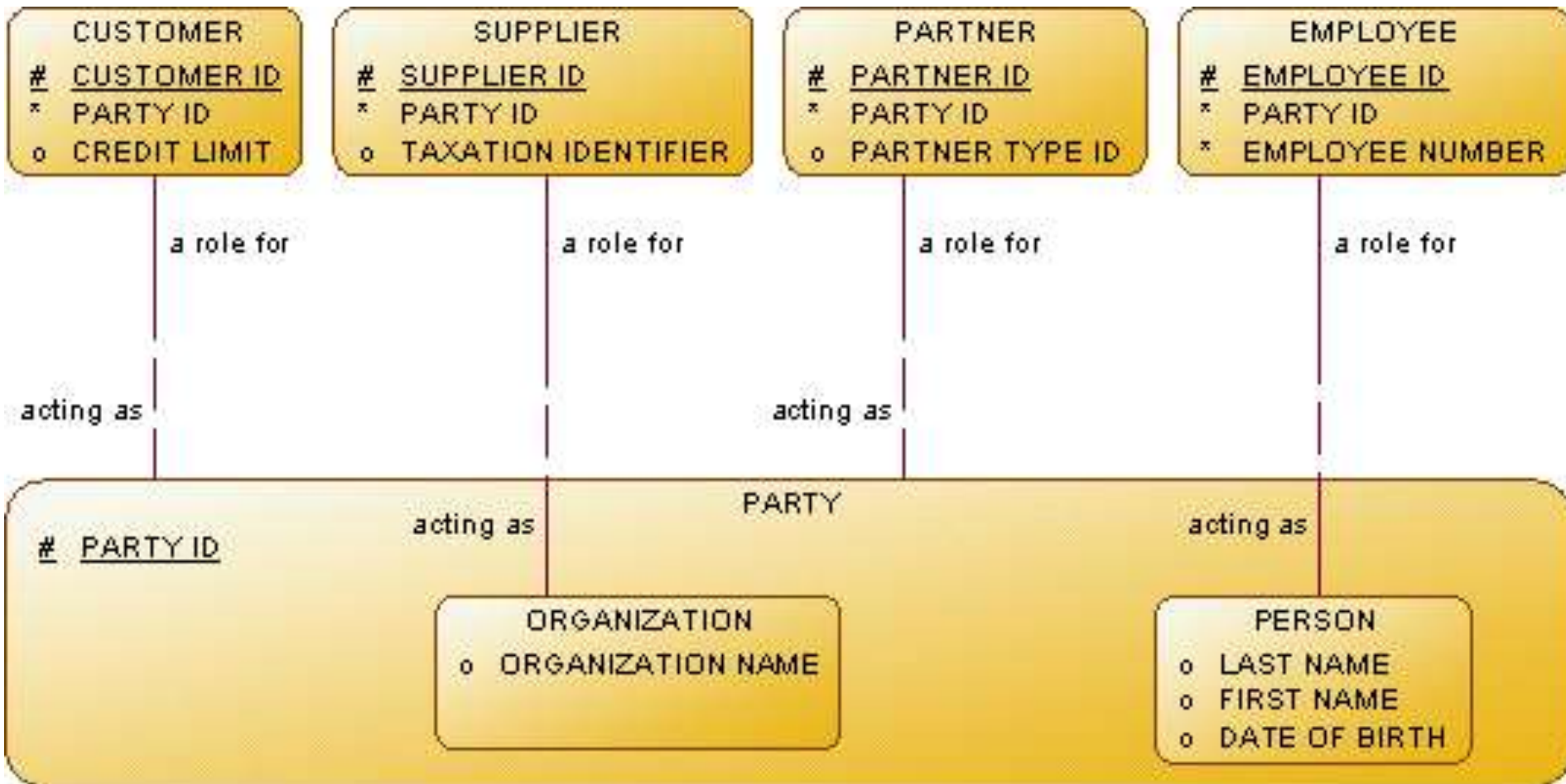
SUPPLIER ID	ORGANIZATION NAME	TAXATION IDENTIFIER
369	Moje data s.r.o.	123456789
456	Vaše Data s.r.o.	987654321

PARTNER ID	ORGANIZATION/FIRST/LAST NAME	PARTNER TYPE
1001	Moje data s.r.o.	10 (Global partner)
1002	Tvoje Data s.r.o.	20 (Software testing)

Pattern Přiřazení rolí I

- Nejjednodušší řešení - každá role jiná entitu
- Výhody
 - Přesně definované role
 - Srozumitelné řešení
 - Pravidla hlídaná modelem
 - Některé role mohou zastávat pouze organizace (SUPPLIER), některé pouze lidé (EMPLOYEE), některé jak lidé, tak organizace
- Nevýhody
 - Atributy jsou společné (Jméno) a specifické (EMPLOYEE NUMBER)
 - Stejná informace je uložena na více místech (Jak řešit změnu adresy firmy Vaše Data);
 - Jedna organizace nebo člověk může mít více rolí
 - Přidání nové role vynucuje změnu modelu
 - Těžko se skládá celkový obrázek o vazbách mezi subjekty
 - Není jasné, jak jednoznačně identifikovat subjekt

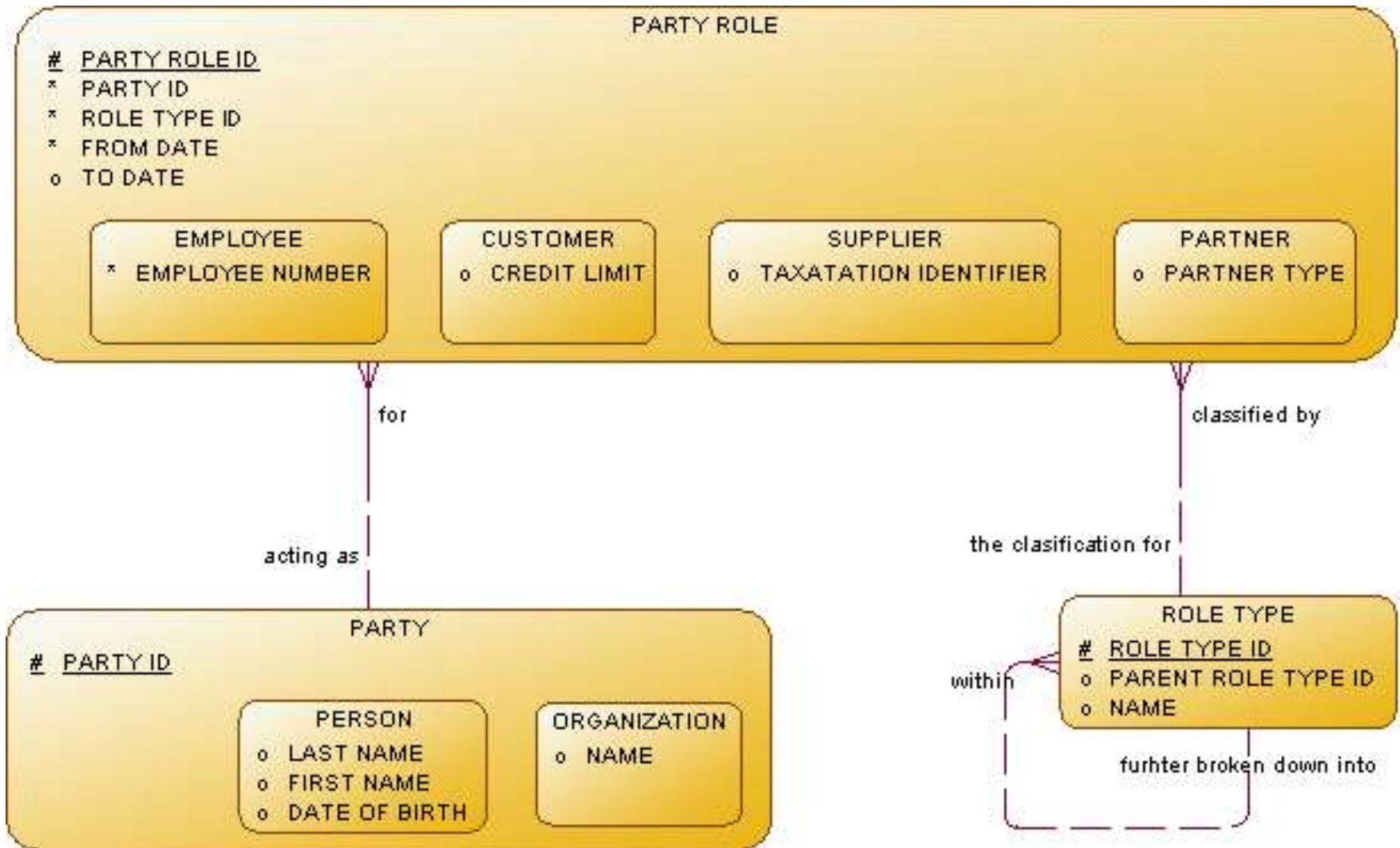
Pattern Přiřazení rolí II



Pattern Přiřazení rolí II

- Složitější řešení – jednotná identifikace, společné řešení pro základní údaje
- Výhody
 - Odstranění redundance informací o osobách a organizacích.
 - Pravidla hlídaná modelem
 - Role může být vázána na PARTY, nebo jenom na podtyp ORGANIZATION
 - Umožňuje jednoduše vázat další entity (faktura, objednávka) přímo na PARTY, není potřeba rozlišovat, zda se jedná o osobu nebo organizaci
 - Umožňuje jednoduše přidávat další role existujícím PARTY
 - Umožňuje, aby jedna PARTY vystupovala ve více rolích
- Nevýhody
 - Nutnost řešit dědičnost na úrovni PARTY
 - Jednotlivé role jsou samostatné entity: nová role = nová entita. Není vhodné pokud nové role vznikají často.
 - Uživatelé mají problém rozlišit PARTY od ROLE
 - Pattern naznačuje, že PARTY vystupuje v roli pouze jednou
 - Neumožňuje řídit informace ohledně typů rolí

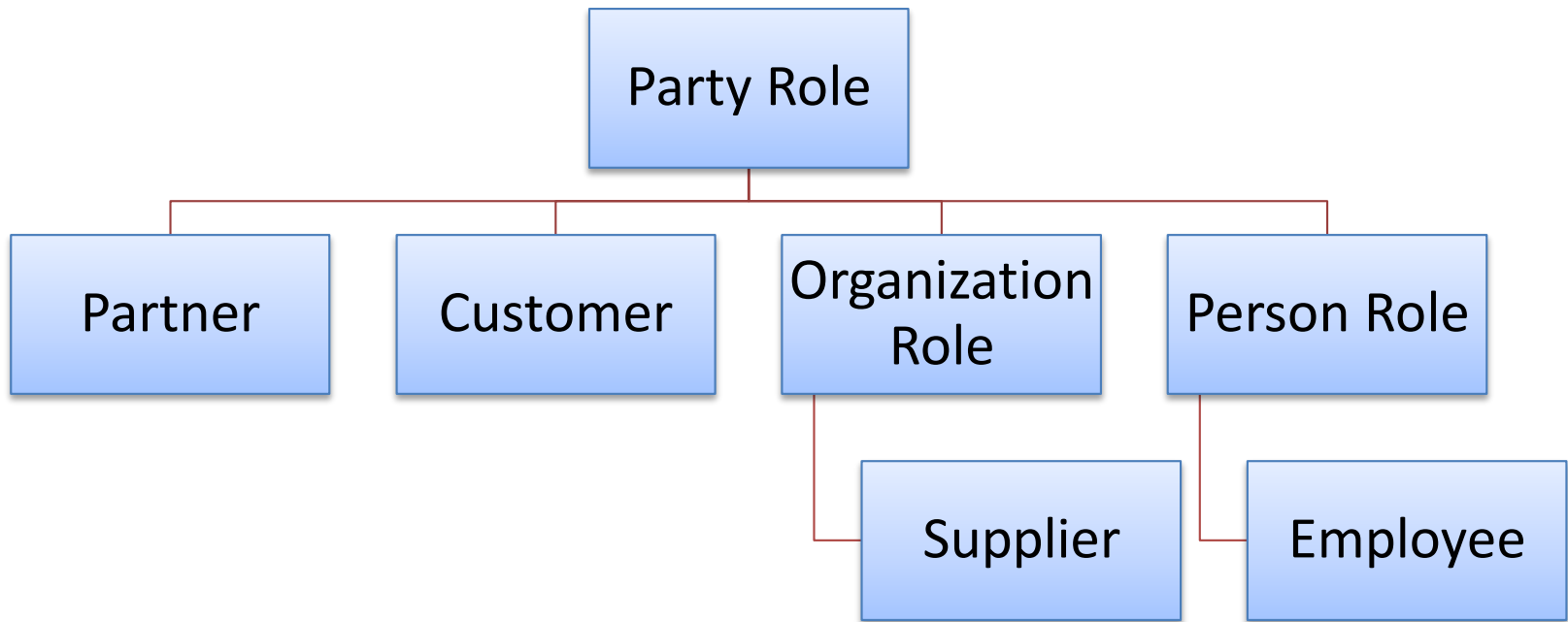
Pattern Přiřazení rolí III



Pattern Přiřazení rolí III

ROLE TYPE ID	NAME	PARENT ROLE TYPE ID	PARENT NAME
100	Party role	Null	Null
101	Customer	100	Party role
102	Partner	100	Party role
103	Organization role	100	Party role
104	Supplier	103	Organization role
105	Person role	100	Party Role
106	Employee	105	Person role
107	Manager	106	Employee
108	Debtor	100	Party role

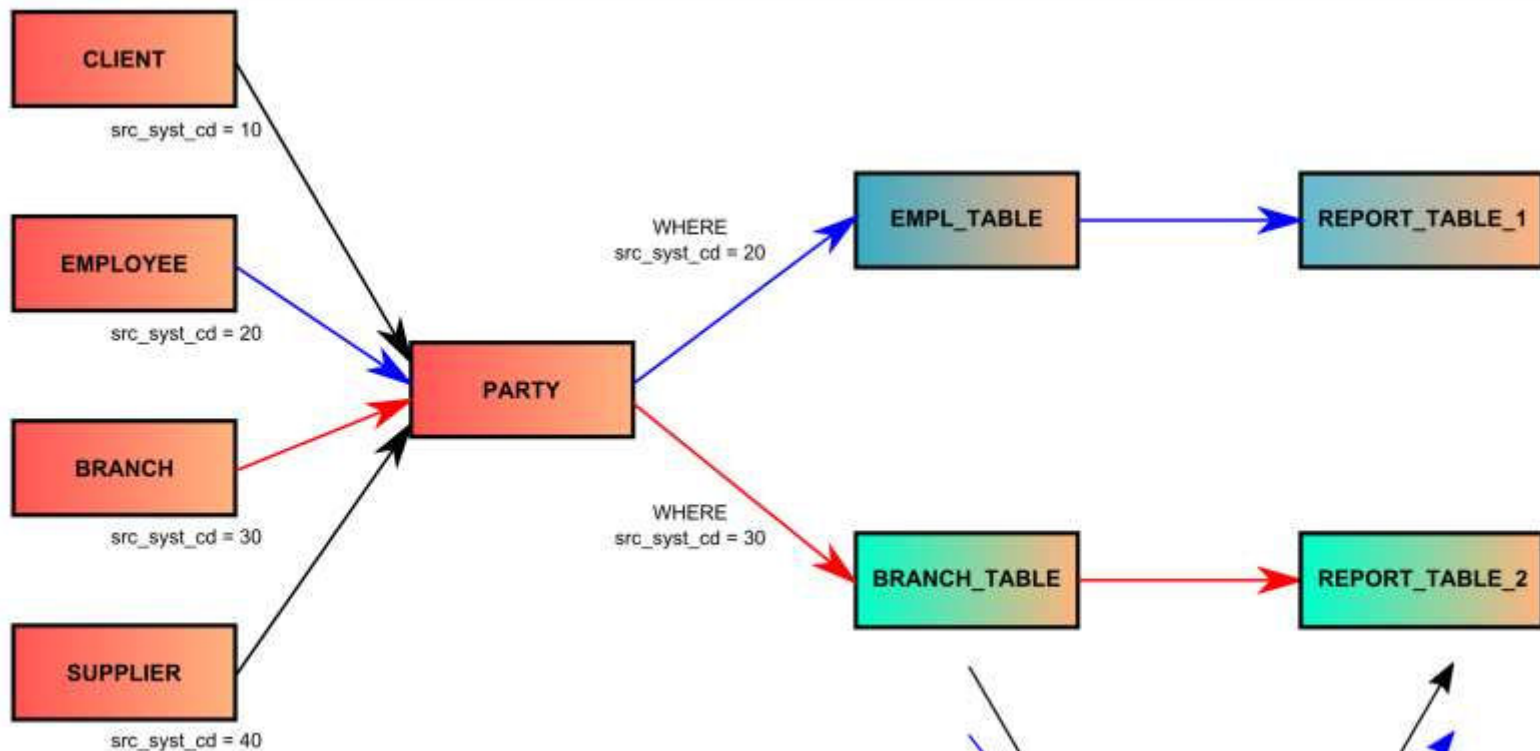
Pattern Přiřazení rolí III



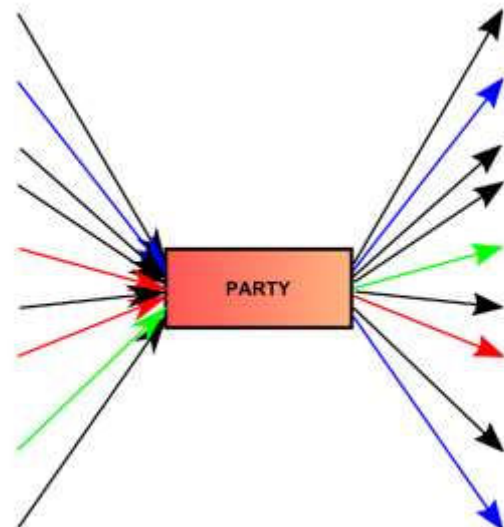
Pattern Přiřazení rolí III

- Ještě složitější přístup
 - ROLE TYPE je číselník rolí
 - PARTY ROLE je
 - Vazební entita mezi PARTY a ROLE TYPE
 - Rodičovská entita pro všechny role – Potomci obsahují specifické atributy
- Výhody
 - PARTY může přijímat mnoho rolí
 - PARTY ROLE pro jednotlivé PARTY mají časovou specifikaci
 - Existuje stromová hierarchie mezi ROLE TYPE
 - Pokud nové role nevyžadují nové atributy, nevyžaduje přidávání rolí zásah do datového modelu (možno řešit i oddělením atributů).
- Nevýhody
 - Je to složité !!!!!
 - Při uvedeném číselníku typů rolí je těžko pochopitelná vazba mezi Person role a Organization role a strukturou PARTY
 - Model hlídá méně pravidel, musí být hlídány aplikačně
 - Role může být vázána na PARTY, nebo jenom na podtyp ORGANIZATION
 - Pokud nová role vyžaduje nové atributy, je stále nutné zasáhnout do datového modelu.

Konsolidovaný model a datové toky



- Pokud se konsolidují všichni partneři do jedné tabulky, je těžko dohledatelné, odkud přišli a kde se používají data.



```

4780 GOTO 5000
4790 :
4800 REM -----
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM -----
4810 :
4820 PRINT" ";
4825 W=V+1:IF W<8 THEN W=W+14
4830 FOR X=1 TO 2:PRINT" ";
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT" ";
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT M$(
(I+1));:GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT" ";
4920 FOR I=2 TO 24 STEP 2
4925 PRINT"|";
4930 IF MD$(I+W-1)
Q";:GOTO 4940
4935 PRINT" ";
4940 NEXT:PRINT" ";
4950 PRINT" ";
4960 FOR I=2 TO 24
4965 PRINT"|";
4970 IF MD$(I+W-1)
M$(I)"|";:GOTO 4980
4975 PRINT M$(I);
4980 NEXT:PRINT" ";

```



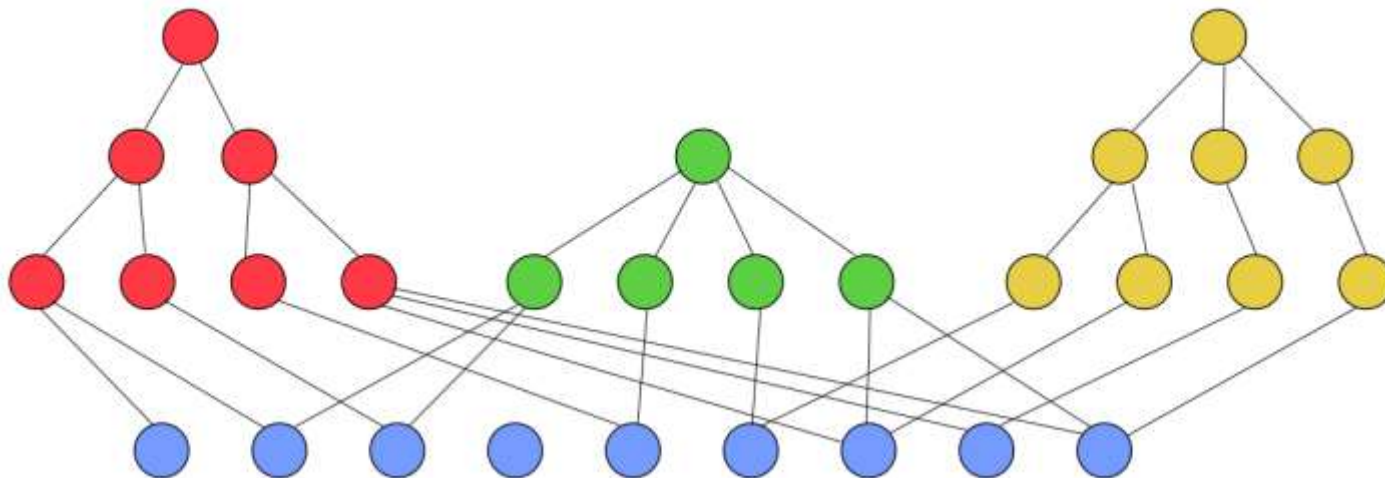
Pattern Klasifikace



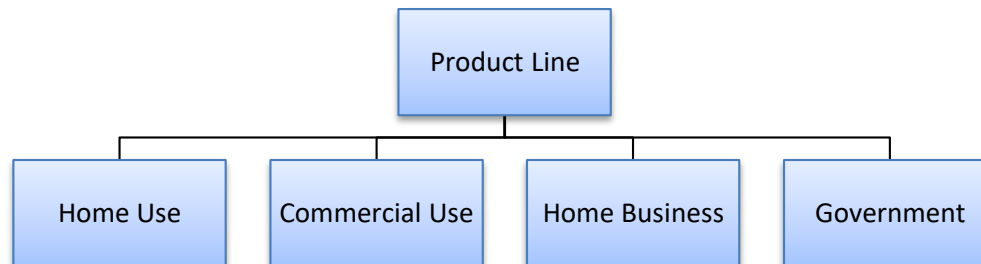
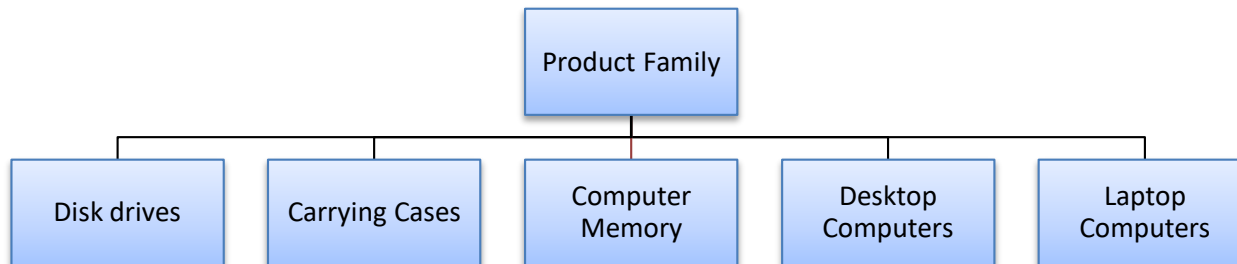
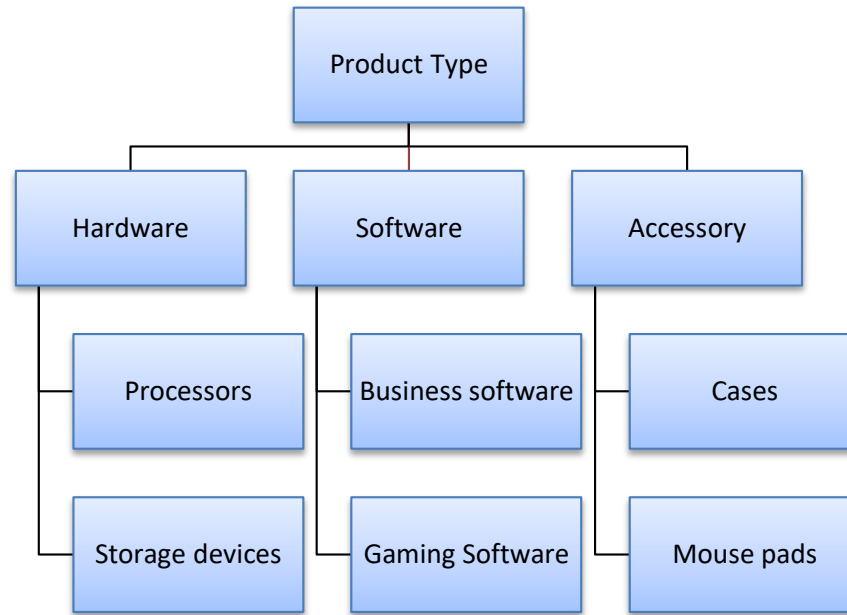
Pattern Klasifikace

▪ Definice

- Podpora členění instancí entity podle typů, do kategorií a taxonomií.
- Typy – skupiny se společnými charakteristikami
- Kategorie – kategorizace podporuje více druhů členění (Typy typů)
- Taxonomie – původně věda zabývající se klasifikací organismů; členění dle definované struktury (například Klasifikace ekonomických činností (CZ-NACE)). Každá instance právě v jedné kategorii.



Klasifikace



Pattern Klasifikace I



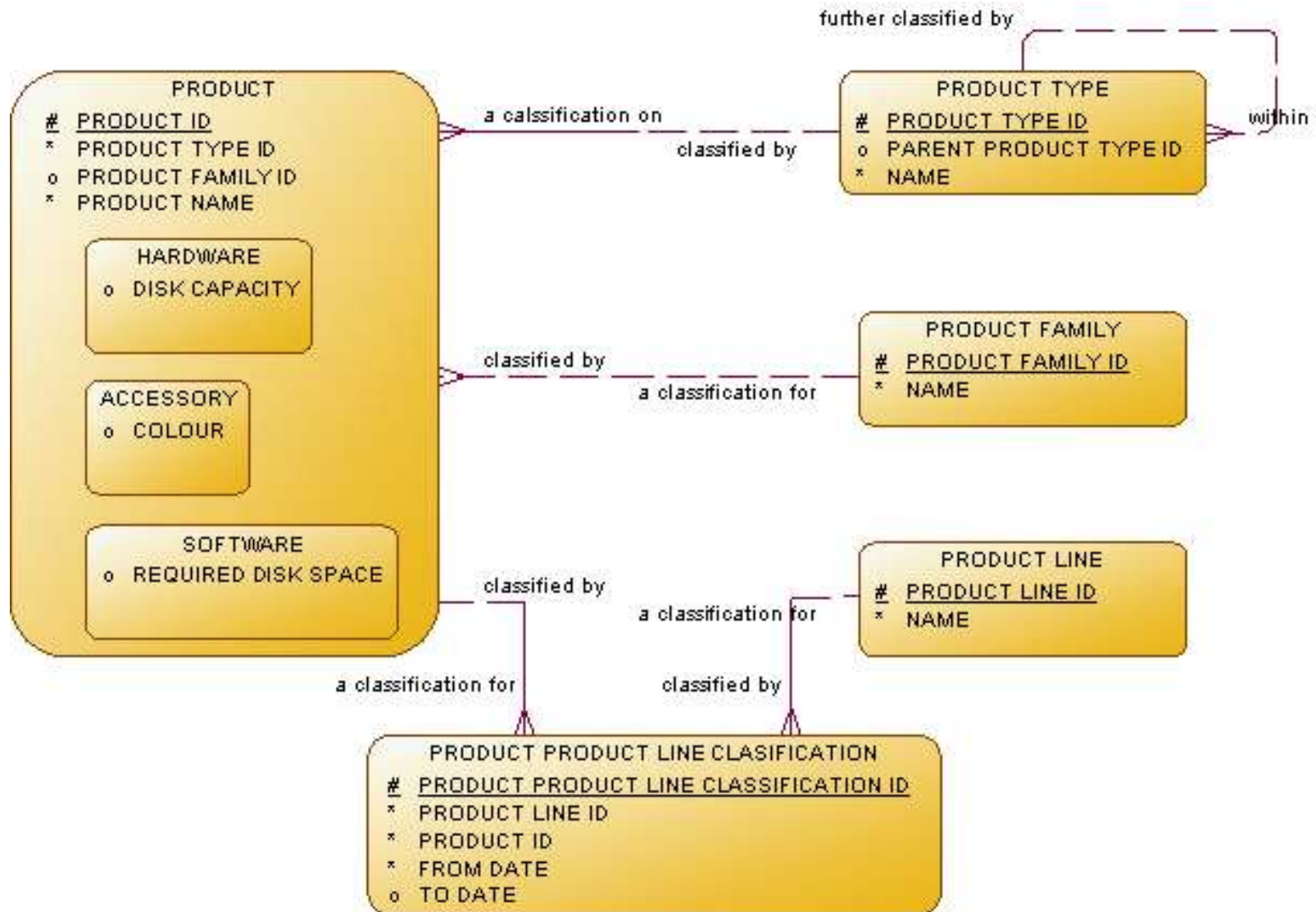
Pattern Klasifikace I

ID	NAME	TYPE	FAMILY	LINE 1	LINE 2	CAPACITY	COLOUR
100	Save Disk 2000	HW	Disk Drivers	Home use	Commercial Use	20GB	Black
101	Carry All Case	Accesory	Carrying Case	Commercial use			Green
102	HS Software package	Software		Home Business			
103	Memmory card M10	Hardware	Computer memory	Home use	Home Business	1GB	

Pattern Klasifikace I

- Každá klasifikace má své atributy
 - Taxonomie – povinný atribut
 - Násobné atributy
- Výhody
 - Velice jednoduchý model, snadno pochopitelný pro všechny uživatele
 - Vhodný jako základ (prototyp), odrazový můstek pro pochopení a podrobnější analýzu
 - Implementace může používat omezení na hodnoty ve sloupcích (nebo aplikančí omezení)
- Nevýhody
 - Složitá správa redundantních dat (HW – hardware – Hardware)
 - Velice nepružný model
 - Přidání kategorie – přidání atributu
 - Mnoho typů – mnoho atributů
 - Více typů klasifikací – více sloupců (Product line 1, Product line 2)
 - Nedají se udržovat data o klasifikacích – popis, doba platnosti a podobně
 - Model nepodporuje složitější vazby o klasifikacích – pouze povinné a nepovinné klasifikace

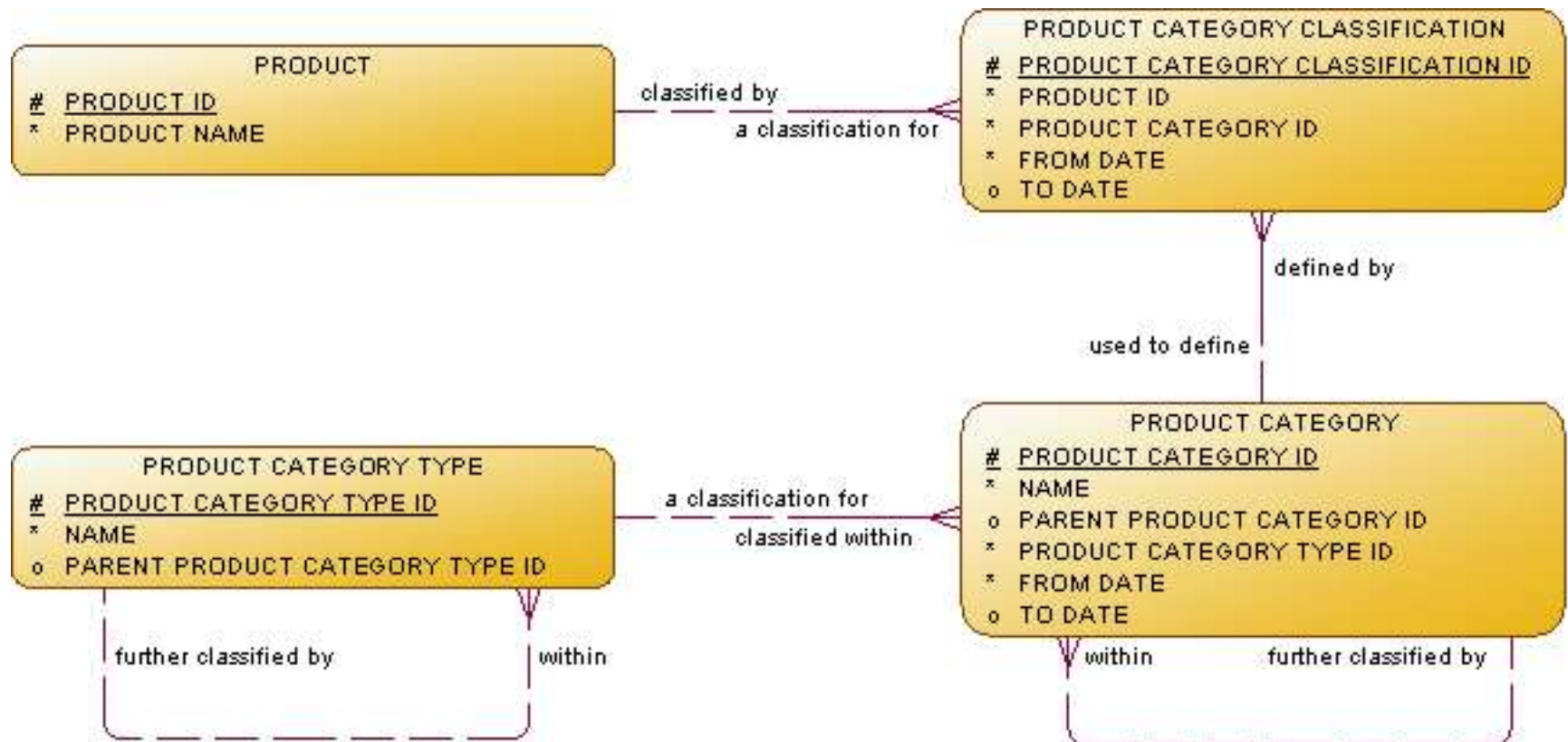
Pattern Klasifikace II



Pattern Klasifikace II

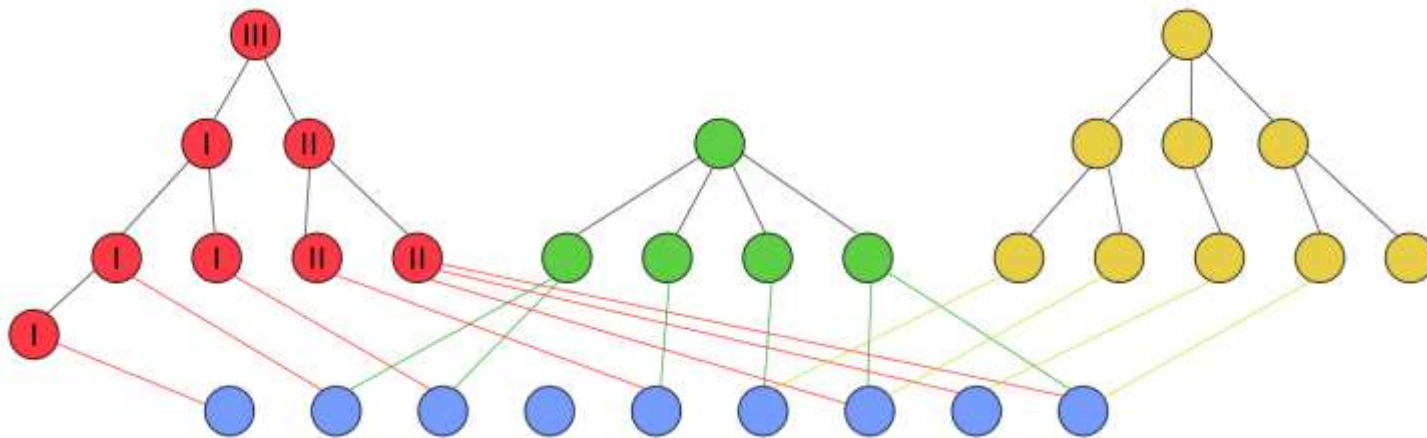
- Číselníkové tabulky
 - Taxonomie – povinný atribut
 - Vazební tabulky pro více hodnot
- Výhody
 - Snadno porozumitelný model
 - Změna klasifikace – změna číselníku
 - Pro porozumění modelu je důležité znát obsah tabulek (číselníků).
 - Umožňuje nezávislé řízení klasifikací – MDM
 - Rozdílné klasifikace mohou mít své atributy a hierarchie
- Nevýhody
 - Málo pružný model, pokud je potřeba přidávat nové klasifikace.
 - Jednotlivé klasifikace jsou udržovány v oddělených entitách.
 - Mnoho typů klasifikací – mnoho atributů, mnoho číselníků.

Pattern Klasifikace III



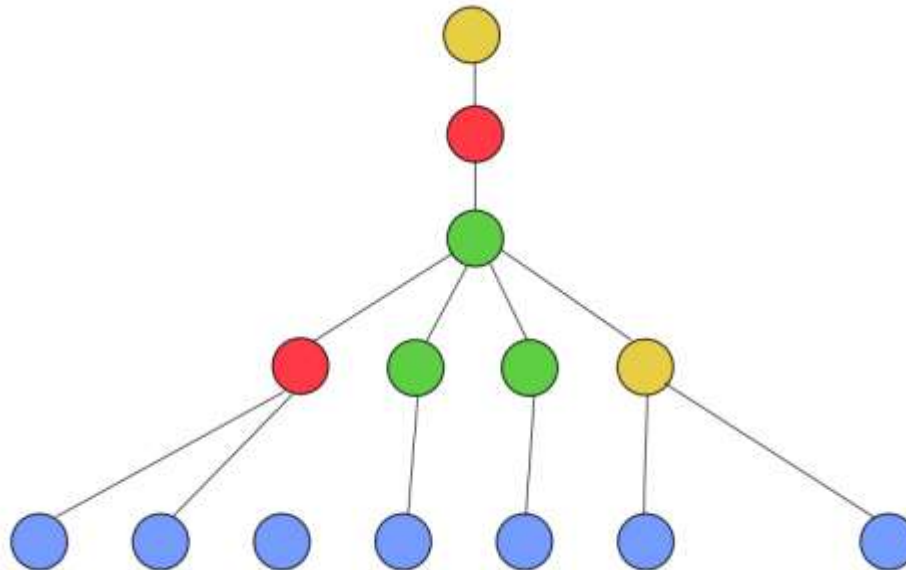
Pattern Klasifikace III

- Sjednocení všech kategorií do jedné entity
 - Zavedení klasifikace kategorií
 - Hierarchická struktura na kategoriích i typech kategorií (například pro reporting)
- Výhody
 - Jednoduché řízení klasifikací – přidávání nové kategorizace, změna hierarchie kategorií je pouze změna dat
 - Vhodné pokud je potřeba mnoho klasifikací
 - Jednoduchý po databázové stránce – jenom čtyři tabulky
 - Umožňuje jednoduše složitější analýzy podle různých klasifikací



Pattern: Klasifikace III

- Nevýhody
 - Těžký na porozumění, zejména při úpravách dat číselníků
 - Nevynucuje žádná business pravidla – nutno hlídat aplikačně
 - Není vazba mezi hierarchií Kategorii a Typů kategorií.
 - Model neumožňuje mít rozdílné atributy pro specifické typy



Shrnutí

- Řešení musí odpovídat
 - Složitosti business domény,
 - Složitosti business pravidel,
 - Schopnosti analytiků a vývojářů porozumět modelu,
 - Schopnosti uživatelů udržovat model.
- Vždy je vhodné při návrhu modelu vytvářet i data entit.

Další směry

- Řešení časové platnosti záznamu.
- Řešení více hierarchií.
- Řešení definice různých atributů pro různé typy.

Co si zapamatovat

- Co to jsou databázové patterny
- Jaké databázové patterny se používají
- Jaké patterny znáte pro řešení dědičnosti
- Jaké řešení pro pattern Rolí se používají, jaké mají slabé a silné stránky
- Jaké řešení pro pattern klasifikace se používají, jaké mají slabé a silné stránky

```

4780 GOTO 5000
4790 :
4800 REM -----
4801 REM --- DARSTELLUNG ---
4802 REM --- DES MANUALS ---
4803 REM -----
4810 :
4820 PRINT" ";
4825 W=V+1:IF W<8 THEN W=W+14
4830 FOR X=1 TO 2:PRINT" ";
4835 FOR I=0 TO 23
4840 PRINT MD$(I+W);
4850 NEXT:PRINT:NEXT
4860 PRINT" ";
4870 FOR I=0 TO 23
4880 IF MD$(I+W)=CHR$(32) THEN PRINT M$(
(I+1));:GOTO 4900
4890 PRINT MD$(I+W);
4900 NEXT
4910 PRINT:PRINT" ";
4920 FOR I=2 TO 24 STEP 2
4925 PRINT "|";
4930 IF MD$(I+W-1)=" " THEN PRINT"
";:GOTO 4940
4935 PRINT " ";
4940 NEXT:PRINT" "
4950 PRINT" ";
4960 FOR I=2 TO 24 STEP 2
4965 PRINT "|";
4970 IF MD$(I+W-1)=" " THEN PRINT"
M$(I) " ";:GOTO 4980
4975 PRINT M$(I);
4980 NEXT:PRINT" "

```



Diskuse

- Otázky
- Poznámky
- Komentáře
- Připomínky

