

NÁČELNÍK

Databázové variace na téma pozdního středověku



Rev.: 08 / 2023
Aut.: Hom50

OBSAH

OBSAH.....	i
1 Prolog.....	1
2 O několik století dále ...	2
3 Úloha NÁČELNÍK - verze jednoduchá.....	2
3.1 První pokus o pochopení situace.....	2
3.2 Jaká data v jakých tabulkách.....	3
3.3 Vazby mezi tabulkami, datové typy.....	4
3.4 Dotazy	4
3.4.1 Úloha 1.....	4
3.4.2 Úloha 2.....	5
3.4.3 Úloha 3.....	5
3.4.4 Úloha 4.....	6
3.4.5 Úloha 5.....	6
4 Úloha NÁČELNÍK - verze náročná	7
4.1 Změna logiky	7
4.2 Cvičná data.....	8
4.3 Struktura tabulek	10
4.3.1 Datová pole, data	10
4.3.2 Kritické poznámky.....	12
4.4 Vazby mezi datovými zdroji	13
4.5 Dotazy	14
4.5.1 Úloha 1 - detailně krok po kroku.....	14
4.5.2 Úloha 2.....	19
4.5.3 Úloha 3.....	20
4.5.4 Úloha 4.....	22
4.5.5 Úloha 5.....	22

1 Prolog

Jsem prefektem přímořské prefektury 苏州 (Su-čou) provincie 江苏 (Ťiang-su) daleko na jih od sídelního města. V něm nedávno bílí barbaři ze severu svrhli naši zákonnou císařskou dynastii 明国 (Ming) a zavádějí podivné pořádky. Jak mně povídal můj kapitán, který včera odtud připlul, tak například - představte si ten nesmysl - prý muž může mít jen jednu ženu! Ovšem kapitán byl opilý, naše výborné rýžové víno rychle stoupá do hlavy.

To já, 刘冠 (Liu-taj), prefekt a současně náčelník - hlava rodu 周 (ZHOU) mám žen několik (asi čtyři). Všechny jsou, samozřejmě, ze stejně vážených rodů (asi třech dalších - rody u nás sídlí v různých okresech, asi třech okresech naší prefektury) jako já. Třeba moje První Dáma je z rodu 一个 (CHU), a protože jsem s ní byl spokojen, vzal jsem si jako jednu z mých dalších Dam její sestru. Řeknu vám, jsem šťastný, že si je všechny mohu dovolit. Dávám jim podle starého dobrého zvyku na jejich potřeby měsíčně (všem stejně, jistě, aby byl doma klid) jeden Zlatý Peníz. Ten, jak dobře víte, dělíme na sto Stříbrných Penízků.

S mými Dámami mám hromadu dětí (asi čtyři). Většinou jsou již dospělé. Synové založili své vlastní rodiny, dcery se výhodně provdaly do spřátelených rodů, a to i v jiných okresech naší prefektury. Já jsem už mnohonásobným (asi trojnásobným) dědečkem. U nás je rod dítěte samozřejmě rodem otce. Protože dětí a dětí jejich dětí neustále přibývá, musel jsem vzít několik čtvrtek našeho skvělého papíru, štětec a tuš, a udělat si v tom pořádek.

To byl výborný nápad, hlavy všech okolních rodů mně výsledek závidí. Když se každou neděli sejdeme nad obvyklou sklenkou saké, udivuji je rychlostí, s jakou dovedu uspokojit jejich zvědavost (někdy jsou to věci hodně osobní, ale mezi námi muži a navíc přáteli se to tak nebere). Třeba hned minulou neděli, čtvrtého dne sedmého měsíce toho roku, který naši námořníci po vzoru bílých obyvatel dalekých západních zemí číslují jako 1674:

Soudce a náčelník rodu 的 (TI) Šeng chtěl jako první vědět, kolik mých Dam je ze stejného (stejných) rodů a z jakých okresů jsou.

A hned náčelník rodu 听力 (ČOU) Ťüan, teprve nedávno se oženil, prý se kterou mojí Dámou také ještě nemám žádné děti?

Třetí dáma obchodníka a hlavy rodu 吴 (WU) Saj se ho už několikrát ptala, jak je stará moje nejstarší vnučka mého rodu a v kolika letech svého otce se mu narodila? Tady vidíte problémy, kterými žijí naše Dámy ...

Mne při té příležitosti napadlo, kdy přibližně budu muset jet za obchodníkem WU se svou Druhou Dámou, o níž vím, že už dvanáct týdnů neutrácí a střádá si na nové kimono za - považte - sedm Zlatých a padesát Stříbrných!

Moje zápisky mně závidí generál a hlava rodu 十一 (XI) Keng a chtěl mně trochu vyzkoušet, možná pozlobit. Kolik že roků má (nebo by měl) můj nejstarší tchán?

No, myslím, že jsem před přáteli opět obstál. Už se těším na další setkání - ale vím, že moje zápisky budu muset ještě doplnit, na některé otázky bych z nich zatím nedokázal odpovědět.

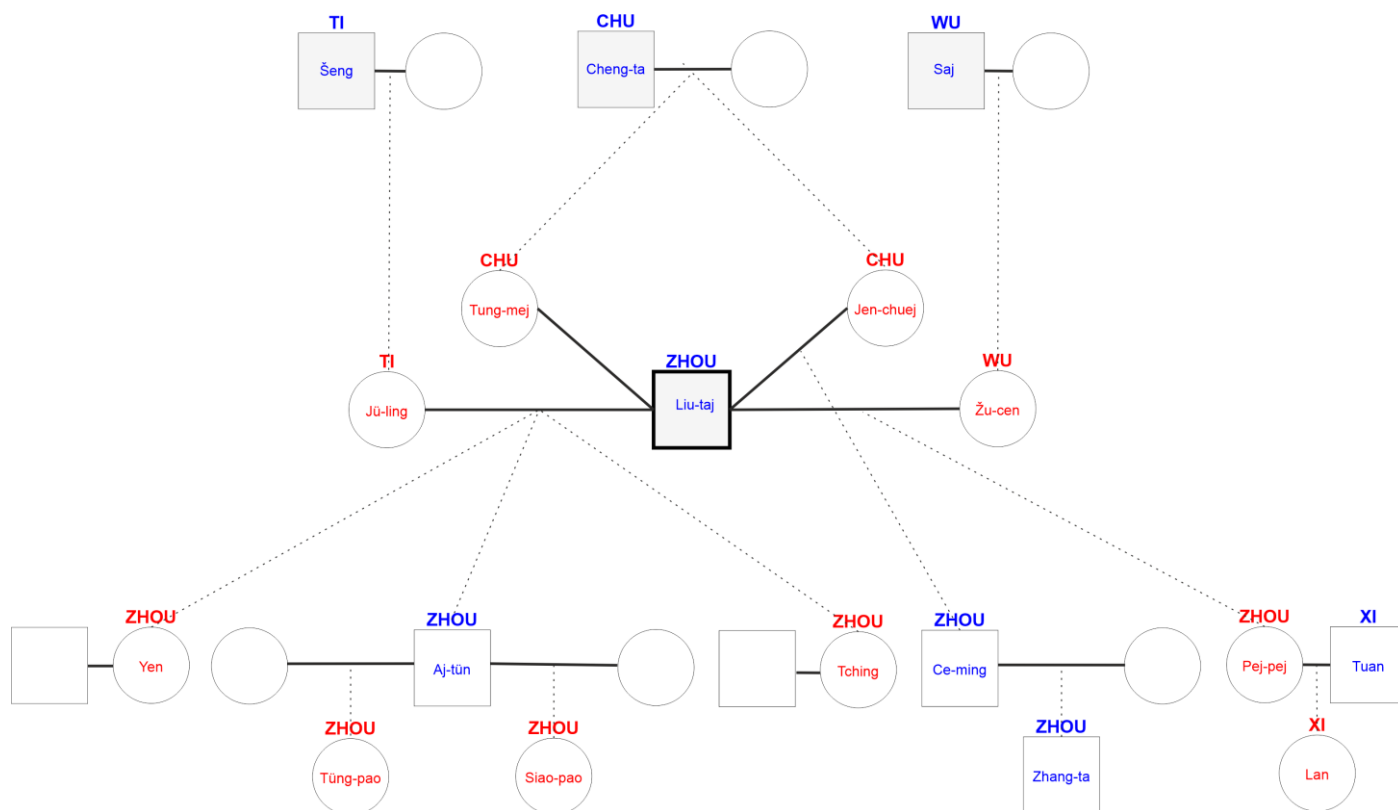
2 O několika století dále ...

Jsem pilným studentem provinční univerzity daleko na východ od sídelního města. V něm mám mnoho spolužáků z předchozího studia a pozorně naslouchám těm, kteří už jsou v praxi. Jejich zmínky o tom, kolik jim to sype, mně připadají přeci jen nadsazené, ale znáte to - není šprochu, aby ... I já bych chtěl, aby mně to sypalo, nejlépe také jako ajťákovi právě v onom sídelním městě. Proto se učím a trénuji na nejrůznějších příkladech. Onehdy jsem se setkal při četbě populárních historických románů s milým spokojeným sebehodnocením dávného obyvatele rozlehlé říše (viz prolog shora). Nedalo mně, zda nezkusit změřit své síly s oním náčelníkem, on s papírem a štětcem, já s počítačem a databázemi.

3 Úloha NÁČELNÍK - verze jednoduchá

3.1 První pokus o pochopení situace

Zatím nejsem příliš zkušeným datařem - a upřímně: komu se hned po prvním přečtení prologu začne rýsovat řešení, kterak formulovat dotazy odpovídající na všetečné otázky náčelníků? Já jsem četl (a pokusil se chápat) třikrát. I pak jsem usoudil, že nejrozumnější bude si tu situaci vzorově nakreslit:



Použil jsem přitom symboliku dle doporučení genealogů: kruhem je znázorněna žena, čtvercem muž, vodorovná spojnice váže manžela s manželkou, spojnice shora dolů směřují od rodičů k dětem. Uprostřed obrazce je jméno osoby, nad ním jeho rod. Tučným čtvercem je zvýrazněn vypravěč prologu, podbarvení čtverců znázorňuje náčelníky. Abych zbytečně nezabřednul do problematiky historických rodinných vztahů starých říší, pojmám označení rodu jako to, co dnes chápeme jako rodné jméno. Na druhé straně jsem si

zjednodušil úvahy tím, že jsem se omezil na jedinečné jméno každé z osob. Stejně na mne historie dýchla, když jsem pátral po tvaru křestních jmen a příjmení, o způsobu jejich zápisu (pro většinu z nás nečitelnými) fonty ani nemluvě :-)

Z textu prologu jsem pochopil (a definitoricky přijal do řešení):

- Protože dámy náčelníka Liu-taje jsou ze stejné vážených rodů jako on, hlava rodu, jsou to dcery rovněž hlav svých rodů.
- Z toho tedy dále plyne, že jeho tchánové jsou hlavy rodů (protože tchán je otec manželky).
- Má-li náčelník čtyři manželky, má maximálně čtyři tchány. V jeho případě však jen tři, protože dvě z jeho dam jsou sestry (a mají tedy stejného otce, nemusí však mít stejné matky).
- Všechny jeho děti jsou stejného rodu jako on. Jakmile se dcera provdá, ponechává si svůj rod, nepřejímá rod manžela.
- Ovšem všichni jeho vnuci a vnučky jsou stejného rodu jako on pouze tehdy, jde-li o děti jeho synů. Děti jeho dcer jsou rodu manžela jeho dcery.
- Rod (potažmo jeho hlava) sídlí v nějakém okrese. Ze znění prologu vyplývá, že prefektura je částí provincie a okres částí prefektury. Okresy v ději vystupující jsou ve stejné prefektuře, a protože se jinde v textu o provincii a prefektuře nemluví, jsou pro mé výsledné řešení irelevantní.
- V nejjednodušším případě lze abstrahovat od jiných rodů a zabývat se jen osobami patřících do nejbližších předků a následníků vypravěče Liu-taje.

3.2 Jaká data v jakých tabulkách

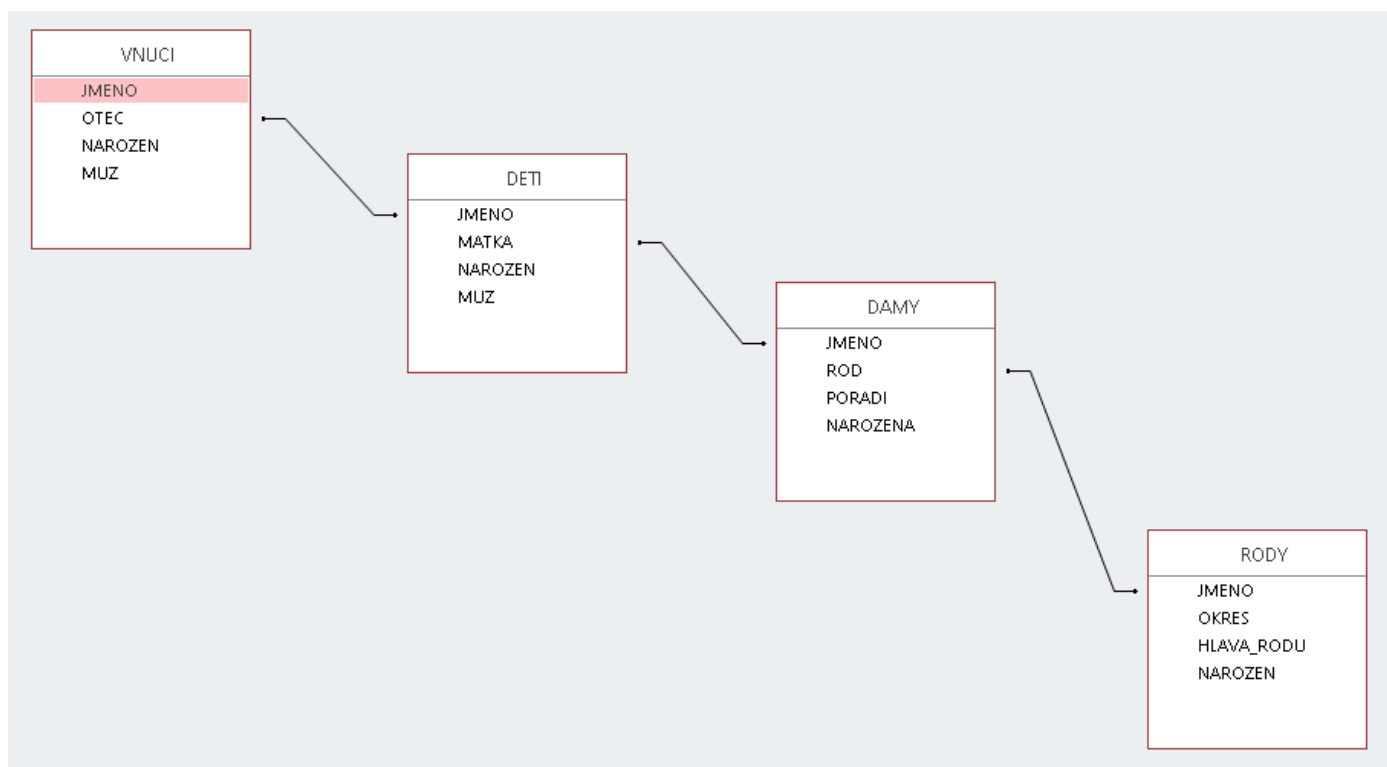
Z prologu a otázek v něm popsaných mně vyplynuly čtyři tabulky:

1. Především tabulka s výčtem jednotlivých rodů. Každý rod má nějaké (jedinečné) jméno a sídlí v nějakém okrese. Není vyloučeno, aby v jednom okrese sídlilo více rodů. Každý rod má svého náčelníka - hlavu rodu. Z nich se rekrutují tchánové našeho náčelníka, a protože padla otázka na věk nejstaršího s nich, musí se sledovat i datum jejich narození. Protože v jiném kontextu se o hlavách rodu nemluví, patří informace o nich do této tabulky.
2. Dále tabulka s výčtem dam náčelníka Liu-taje (jen jeho manželkami se toto řešení zabývá). Ze znění nevyplývá, že by se o manželkách mělo sledovat něco jiného než jméno a rod. V mém řešení jsem si přidal (nepoužité) datum narození a pořadí dámy.
3. Náčelník Liu-taj má hromadu dětí, evidentně nesmí chybět tabulka dětí. Protože se zabýváme jen potomky tohoto náčelníka (tj. stejného rodu jako on) s jedinečným jménem, zdánlivě postačí jen jméno, jméno matky a informace o tom, zda to je mužský nebo ženský potomek. Protože však mužský potomek může mít dceru (tj. vnučku Liu-taje) a bude se zjišťovat, v kolika letech otce (tj. tohoto mužského potomka) se narodila, musí být v tabulce dětí uvedeno i datum jejich narození.
4. Konečně tabulka vnuků resp. vnuček. Jistě bude v tabulce jméno, datum narození a informace o pohlaví (bude se pátrat po nejstarší vnučce). Dále musí být přítomno jméno otce (bude se zjišťovat, v kolika letech otce se vnučka narodila). Otec vnučky je dítětem Liu-taje a je proto přítomen v tabulce dětí včetně data svého narození.

Označení jednotlivých datových polí viz následující odstavec.

3.3 Vazby mezi tabulkami, datové typy

Pro toto jednoduché řešení nejsou relace (relation) jako takové snad ani zapotřebí. Vůbec mně však neuškodilo nechat si graficky znázornit vztahy (relationship) mezi tabulkami, protože usnadní návrh dotazů používajících konstrukci datových zdrojů pomocí klauzule join:



Datové typy:

- NAROZEN resp. NAROZENA: Logicky typ Datum a čas. O cílovém databázovém systému se předpokládá, že interní uložení hodnoty tohoto typu je racionální číslo nahlížené v jednotkách [den]. Přesněji, že rozdíl A-B dvou hodnot A a B tohoto typu je racionální číslo v jednotkách [den].
- MUŽ: Hodnota typu Boolean, Logical, Yes/No (dle cílového databázového systému). V grafickém návrhovém prostředí může být použito národních ekvivalentů typů, např. Ano/Ne.
- HLAVA_RODU: Stejně jako MUŽ, Ano nebo Ne.
- POŘADÍ dámy: Celé číslo, stačí na 1 bytu :-)
- Ostatní datová pole jsou typu Text. Délka je vcelku libovolná, pokud však ve dvou různých tabulkách jsou sloupce ve stejném logickém významu (např. jméno matky a jméno dámy) sloužící k propojení těchto tabulek, musí být délka textu v obou takových sloupcích obecně stejná.

3.4 Dotazy

3.4.1 Úloha 1

Kolik mých Dam je ze stejného (stejných) rodů a z jakých okresů jsou?

Už formulace úlohy přímo naznačuje řešení: vytvořit skupiny dam stejného rodu. Výsledkem má být počet dam v každé takové skupině - ovšem jen tam, kde je ten počet větší než 1. Protože součástí odpovědi má být i okres jejich rodu, využije se v dotazu propojení tabulky dam na tabulku rodů.

```
/* D01_KolikManzelekSester */

select
  Count (D.JMENO) as POCET, D.ROD, R.OKRES
from DAMY D left join RODY R on D.ROD = R.JMENO
group by D.ROD, R.OKRES
having Count (D.JMENO)>1
```

3.4.2 Úloha 2

Se kterou mojí Dámou také ještě nemám žádné děti?

Tato úloha je tak jednoduchá, že jsem vymyslel hned dva principiálně různé dotazy.

A. Dáma bez dětí má prázdný (Null) cíl v propojení tabulky dam na tabulku dětí.

```
/* D02_ManzelkyBezDeti A */

select D.*
from DAMY D left join DETI C on D.JMENO = C.MATKA
where IsNull (C.MATKA)
```

B. Dáma bez dětí není uvedena (náповěda: neexistuje :-) v tabulce dětí jako matka.

```
/* D02_ManzelkyBezDeti B */

select D.*
from DAMY D
where not exists
  (select * from DETI C where D.JMENO = C.MATKA)
```

3.4.3 Úloha 3

Jak je stará moje nejstarší vnučka mého rodu a v kolika letech svého otce se mu narodila?

Tak tady jsem se trochu zapotil. Vypravěč to vypráví v jistém okamžiku, prý dne 4 / 7 / 1674. Toto datum musím vzít jako konstantu, ke které se v toku času vztahují děje minulé i budoucí. Ovšem v mnoha verzích SQL něco jako deklarace konstanty neexistuje. Proto jsem tuto datumovou hodnotu nechal vložit do výsledného dotazu; ten se na ni ve svých dalších částech odvolává jménem jejího „sloupce“ - zvolil jsem DNES.

Další skutečnost (jsem hrd, že už předjímám do budoucnosti možné dotazy oponentů): Protože mám mnoho žen a mnoho dětí, nejstarších vnuček (narodivších se ve stejný den různým mým dětem - a to pomijím dvoj- a více -čata) může být více.

```

/* D03_NejstarsiVnucka */

select
  V.JMENO,
  V.NAROZEN as NAROZEN,
  V.MUZ,
  #7/4/1674# as DNES,
  Round((DNES-V.NAROZEN)/365.25,2) as ROKU,
  D.JMENO as OTEC,
  D.NAROZEN,
  Round((V.NAROZEN-D.NAROZEN)/365.25,2) as ROKU_OTCE
from VNUCI V left join DETI D on V.OTEC = D.JMENO
where
  not V.MUZ and
  V.NAROZEN=(select Min(W.NAROZEN) from VNUCI W where not W.MUZ)

```

3.4.4 Úloha 4

Kdy přibližně budu muset jet za obchodníkem WU se svou Druhou Dámou, o níž vím, že už dvanáct týdnů neutrácí a strádá si na nové kimono za sedm Zlatých a padesát Stříbrných?

Tento dotaz vlastně vůbec nepotřebuje žádný datový zdroj, protože jen počítá ze zadaných „konstant“. Příkaz Select však povinně musí datový zdroj obsahovat, může však být jakýkoliv s alespoň jedním řádkem. V uváděném řešení to však musí být DÁMY a navíc musí obsahovat jediný řádek s pořadím dámy č. 2 - viz podmínka **where**. Pokud by to měl být fakt jakýkoliv datový zdroj s alespoň jedním řádkem, vynechá se klauzule **where** a použije se sice nepříliš elegantní, ale fungující **select top 1**.

```

/* D04_KdyProKimono */

select
  #7/4/1674# as Dnes,
  12*7 as UzStradaDni,
  Round(7.5*(365.25/12),2) as MusiStradatDni,
  Dnes-UzStradaDni as StradaOdDne,
  StradaOdDne+MusiStradatDni as OdjezdKolem
from DAMY D
where D.PORADI=2

```

3.4.5 Úloha 5

Kolik roků má (nebo by měl) můj nejstarší tchán?

Potenciální tchánové jsou uvedeni v tabulce RODY jakožto hlavy rodů i se svým datem narození. Stačí tedy zjistit toho s nejmenším datem narození - ovšem nikoliv ze všech; musí to být z množiny otců mých dam. S první verzí A jsem byl hotov celkem brzy:

```

/* D05_NejstarsiTchan_A */

select
  Y.JMENO as ROD,
  Y.HLAVA_RODU,
  Y.NAROZEN,
  Round((#7/4/1674#-Y.NAROZEN)/365.25,2) as MA_ROKU
from RODY Y
where Y.NAROZEN =
  (
    select Min(R.NAROZEN)
    from RODY R
    where R.JMENO in (select ROD from DAMY)
  )

```

Pak jsem si ale uvědomil toto: Kdyby v tabulce rodů byl uveden také jeden, jehož náčelník sice není mým tchánem, ale má stejné datum narození jako můj nejstarší tchán, pak jako výsledek uvidím oba. Proto je správnější řešení verze B (to samozřejmě nevylučuje, že třeba dva moji stejně staří tchánové jsou nejstarší a ty oba dva uvidím):

```

/* D05_NejstarsiTchan_B */

select
  Y.JMENO as ROD,
  Y.HLAVA_RODU,
  Y.NAROZEN,
  Round((#7/4/1674#-Y.NAROZEN)/365.25,2) as MA_ROKU
from
  (
    select R.*
    from RODY R
    where R.JMENO in (select ROD from DAMY)
  ) Y
where Y.NAROZEN =
  (
    select min(R.NAROZEN)
    from RODY R
    where R.JMENO in (select ROD from DAMY)
  )

```

4 Úloha NÁČELNÍK - verze náročná

4.1 Změna logiky

Můj první jednoduchý pokus dopadl celkem obstojně. Jsem si sice vědom mnoha míst ke zdokonalení - např. kontrola dat v tabulkách, přidávání dalších dat, zamyšlení nad krajními případy, o strukturálních změnách ani nemluvě (což takhle děti vnuků?). Začal jsem tedy přemýšlet nad pojetím celé úlohy jakožto informačního systému, který by v prologu uvedené dotazy obsahoval jako svou součást.

Hned bylo zřejmé, že nelze rozšiřovat počet tabulek např. zaváděním vždy nové tabulky pro každou další generaci. Byl by to stejný nesmysl, jako mít tabulku manželů a v ní tolik sloupců pro manželky, kolik je předpokládaný maximální počet manželek.

Tedy opačně, zcela od začátku. Co je subjektem zamýšleného informačního systému? Přeci každá jednotlivá osoba náležející do jistého rodu (ten do jistého okresu, ten do jisté prefektury ...). Řečeno dnešní terminologií, tyto osoby tvoří množinu M obyvatel rodu, okresu, ..., planety. Každá taková osoba disponuje vlastnostmi, jejichž hodnoty jsou pro informační systém zdrojem dat (jméno, datum narození, adresa bydliště atd.).

Nabízí se tedy vlastnosti jedné nebo více osob zapsat do běžné tabulky. Co osoba, to jeden řádek tabulky. Co vlastnost, to jeden sloupec tabulky. V této fázi úvah není rozhodující, zda všechny vlastnosti všech osob budou v budoucnu vyplněné. Označil jsem si takovou tabulku - zatím pracovně - OSOBY.

Informační systém by měl sledovat nejen osoby jako takové, ale i jejich děti, manžele, tchány ap. Ovšem moje dítě, moje manželka, můj tchán - ti všichni jsou rovněž členy rodu, okresu, planety - jsou tedy také prvky množiny výše označené jako M, ve které jsem i já. A tedy zcela zřejmě mají stejnou strukturu vlastností jako já. Řečeno databázově, patří do stejné tabulky jako já. Jinak: se mnou jsou v té tabulce i moji sourozenci, moji potomci, moji předkové atd.

Poznámka: *Taková tabulka by evidentně byla jednak obrovská, jednak by směrem do minulosti byla čím dál prázdnější - tak, jak by mizely schopnosti získat o dávných předcích vůbec nějaké údaje. O tom viz dále.*

Po praktické stránce: Kdyby taková tabulka byla k dispozici, jsem schopen si v ní najít údaje o nějaké osobě X. Ale potřebuji také údaje o vnučce V osoby X (viz otázky v prologu). Tak vezmu prst a jedu po řádcích tabulky tak dlouho, až vnučku V najdu. Jenže ji musím hledat podle nějakého kritéria, které je vázáno na výchozí osobu X (tedy třeba dědu). V řádku dědy X tedy musí být něco, co mně dovolí vnučku V najít (i třeba napotřetí).

Když jsem tento aspekt domyslel do konce: u každé osoby musí být údaj, který identifikuje otce, matku, a manžela (muže) pro případ mnohoženství nebo manželku (ženu) pro případ mnohomužství.

Stejně jako ve shora diskutovaném řešení pomocí čtyř tabulek jsem pro zjednodušení přijal jako jedinečný identifikátor osoby její jméno; její rod nadále chápu jako ekvivalent našeho příjmení. V nějaké provozuschopné realizaci je náhrada jednoznačné identifikace osoby zřejmá - pro počet osob menšího rozsahu něčím jako obvyklé osobní číslo, pro velké počty např. automaticky generovaným celým číslem.

A stejně tak jako v případě čtyř tabulek, nakonec mně kromě tabulky OSOBY zbyla ještě tabulka popisující RODY.

4.2 Cvičná data

V tomto okamžiku si připadám čím dál zkušenější. To ještě netuším, že dlouho před dokončením této úlohy se opět začnu cítit jako trapný začátečník. Nicméně (správně) usuzuji, že nejen pro pochopení situace, ale hlavně pro konstrukci a testování posléze vytvářených dotazů jsou rozhodující vhodně připravená testovací data nejlépe založená (pokud to lze) na vhodném grafickém schématu. Tohle jsem už použil shora při jednoduchém řešení a celkem mně to pomohlo.

Začal jsem tedy připravovat data. Nakreslil jsem si (zatím prázdnou) tabulku a začal ji vyplňovat. Nejzábavnější a konec konců i nejpoučnější pro humanitní vzdělání sama sebe bylo hledání alespoň trochu reálných jmen jak osob, tak rodů. Sice asi moc neodpovídají tvarům před čtyřmi staletími, to jsem ale trochu korigoval četbou historických detektivek výtečného pana Gulika a jeho soudce Ti.

Za pomoci Wikipedie a jiných internetových zdrojů jsem takhle vytvořil první řádek, samozřejmě náčelníka Liu-taje. Jeho rodiče jsem nevyplňoval, v žádném dotazu nejsou zapotřebí. Stejně tak jsem nechal prázdná pole o manželovi (náčelník je muž a tedy není manželem jiného muže). K vyplnění údaje o manželce toto pole nelze použít, protože on má čtyři ženy a proto by v tabulce OSOBY musel být uveden čtyřikrát.

Avšak už když jsem vymýšlel datum jeho narození, začal jsem tušit obtíže. Pokud v prologu uvádí, že s přáteli popíjel roku 1674, kolik mu asi v tu dobu bylo let? Nemohlo mu být 15, i 95 let vypadá nepravděpodobně. Tak jsem si tipnul něco kolem padesátky. Vymyslel jsem datum narození prosinec 1627.

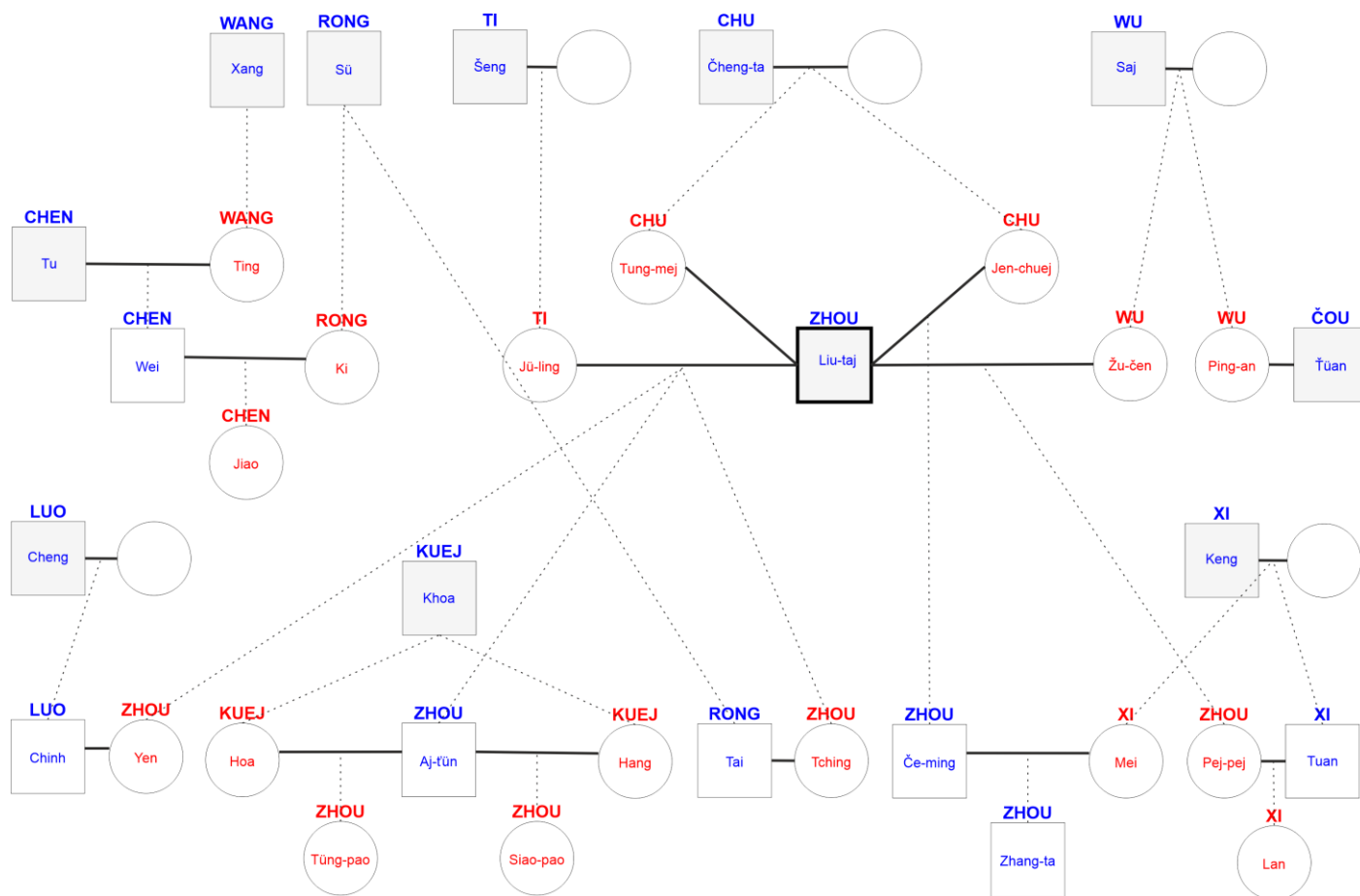
No a pak to začalo. Musel jsem vyplnit 4 řádky dam, kde byl Liu-taj uveden jako manžel. Jak s věkem jeho dam? Tenkrát asi nebylo zvykem pojímat za manželky ženy starší :-)) ale ani to nebyli pedofilové zaměřeni na pětileté dívky :-)) Tedy řekněme tak o 5 až 15 roků mladší. V kolika letech se však Liu-taj asi poprvé ženil? Dále: měl s nimi děti. V kolika letech jeho manželky se jim narodily?

Manželky mají otce (tchány Liu-taje), ti musí být v tabulce OSOBY uvedeni, protože mezi nimi se bude hledat nejstarší tchán. Jak s datem jejich narození? Nutně musí brát ohled na datum narození jeho dcery (dcer). To ale není všechno. Pro test správnosti dotazů musí být v tabulce uvedeni další náčelníci, kteří jsou tchánové (otcové) jiných osob. Někteří z těchto tchánů by měli být starší než nejstarší tchán Liu-Taje, aby dotaz na nejstaršího tchána Liu-taje vybral fakt jen jeho nejstaršího tchána, ne jakéhokoliv nejstaršího tchána.

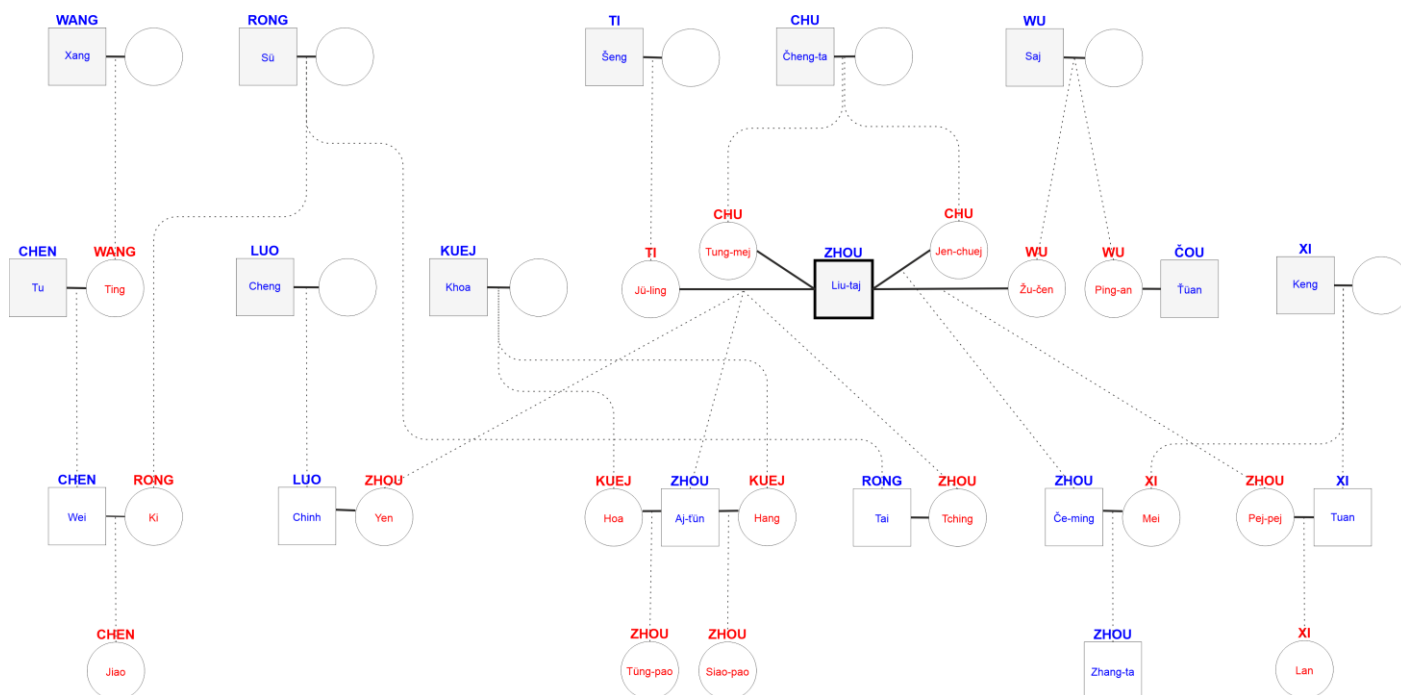
Liu-taj má děti. Pro odzkoušení správnosti to musí být jak synové, tak dcery. Ale ne se všemi manželkami, alespoň jedna musí kvůli požadovaným dotazům zůstat bezdětná. A opět data narození těchto dětí. U dcer musí být uveden manžel, a ten musí být rovněž v tabulce OSOBY. Budou vnuci a vnučky. Musí to být děti dětí, jak mužského tak ženského pohlaví, děti jak synů, tak dcer. Atd atd atd.

Už jakmile cvičně plněná tabulka začala přesahovat 10 - 12 řádků, začala být orientace mezi jednotlivými osobami značně obtížná. A s ohledem na odzkoušení krajních situací to začalo vypadat tak k padesátce fiktivních osob, mezi kterými musí být alespoň trochu reálné vztahy jak věkové, tak příbuzenské.

Jediné co mně v tento okamžik napadlo, bylo nakreslení analogického diagramu, jako pro výše uvedený jednoduchý případ.



nebo pro ty čtenáře, kteří disponují lepším rozlišením obrazovky



Zvláště z druhého schématu je vidět věkové rozvrstvení účinkujících osob: první pás shora = generace předchůdců (matek, otců, tchánů), druhý = generace současníků, třetí = generace následníků (synů a dcer), čtvrtý = generace vnuček a vnuků. Vycházejí z tohoto schématu jsem pak daleko jednodušeji odvodil data narození jednotlivých osob. Hlavně jejich změna pro účely testů kritických míst dotazů pak byla celkem hračkou.

4.3 Struktura tabulek

4.3.1 Datová pole, data

Na výše uvedených schématech jsem pak založil strukturu obou potřebných tabulek (OSOBY a RODY) následovně:

Tabulka OSOBY		
Pole	Typ	Význam
JMENO	Text (31)	(Jednoznačné) křestní jméno
ROD	Text (31)	Rod osoby
MUZ	Yes/No	Yes je-li osobou muž, No není-li
NAROZEN	Date/Time	Datum narození osoby
OTECJ	Text (31)	Jméno otce
OTECR	Text (31)	Rod otce
MATKAJ	Text (31)	Jméno matky

MATKAR	Text (31)	Rod matky
HLAVAR	Yes/No	Yes je-li osoba hlavou rodu (=náčelníkem), No není-li
MANZELJ	Text (31)	Jméno manžela
MANZELR	Text (31)	Rod manžela
PORADIM	Celé číslo	Pořadí manželky

Tabulka RODY		
Pole	Typ	Význam
PROVINCIE	Text (31)	Jméno provincie*
PREFEKTURA	Text (31)	Jméno prefektury*
OKRES	Text (31)	Jméno okresu
OKRES_CH	Text (31)	Okres čínský*
ROD	Text (31)	(Jednoznačné) jméno rodu
ROD_CH	Text (31)	Rod čínský*
NACELNIK	Text (31)	Jméno náčelníka (= hlavy rodu)
NACELNIK_CH	Text (31)	Jméno náčelníka čínský*

Poznámka: Pole označená * nejsou v řešení použita, jsou uvedena spíše pro zajímavost. Jejich hodnoty však podle mého pátrání asi odpovídají současné realitě.

Tabulky po vyplnění:

JMENO	ROD	MUZ	NAROZEN	OTECJ	OTECR	MATKAJ	MATKAR	HLAVAR	MANZELJ	MANZELR	PORADIM
Aj-tün	ZHOU	ano	19.10.1648	Liu-taj	ZHOU	Jü-ling	TI	ne			
Če-ming	ZHOU	ano	23.10.1650	Liu-taj	ZHOU	Jen-chuej	CHU	ne			
Čheng-ta	CHU	ano	24.12.1608					ano			
Hang	KUEJ	ne	07.08.1651	Khoa	KUEJ			ne	Aj-tün	ZHOU	1
Hoa	KUEJ	ne	09.02.1650	Khoa	KUEJ			ne	Aj-tün	ZHOU	2
Cheng	LUO	ano	09.09.1625					ano			
Chinh	LUO	ano	04.08.1647	Cheng	LUO			ne			
Jen-chuej	CHU	ne	23.12.1629	Čheng-ta	CHU			ne	Liu-taj	ZHOU	1
Jiao	CHEN	ne	01.02.1666	Wei	CHEN	Kí	RONG	ne			
Jü-ling	TI	ne	04.05.1627	Šeng	TI			ne	Liu-taj	ZHOU	4
Keng	XI	ano	24.04.1629					ano			
Khoa	KUEJ	ano	02.11.1626					ano			
Kí	RONG	ne	10.08.1647	Sü	RONG			ne	Wei	CHEN	1
Lan	XI	ne	17.01.1668	Tuan	XI	Pej-pej	ZHOU	ne			
Liu-taj	ZHOU	ano	23.12.1627					ano			
Mei	XI	ne	15.05.1651	Keng	XI			ne	Če-ming	ZHOU	1
Pej-pej	ZHOU	ne	05.09.1649	Liu-taj	ZHOU	Žu-čen	WU	ne	Tuan	XI	1
Píng-an	WU	ne	18.04.1631	Saj	WU			ne	Ťüan	ČOU	1
Saj	WU	ano	07.10.1609					ano			
Siao-pao	ZHOU	ne	03.08.1672	Aj-tün	ZHOU	Hang	KUEJ	ne			
Sü	RONG	ano	14.07.1606					ano			
Šeng	TI	ano	04.05.1607					ano			
Tai	RONG	ano	07.12.1649	Ťüan	ČOU			ne			
Tching	ZHOU	ne	30.07.1650	Liu-taj	ZHOU	Jü-ling	TI	ne	Tai	RONG	1
Tíng	WANG	ne	17.05.1626	Xang	WANG			ne	Tu	CHEN	1
Tu	CHEN	ano	17.05.1624					ano			
Tuan	XI	ano	20.02.1651	Keng	XI			ne			
Ťüan	ČOU	ano	01.11.1628					ano			
Tung-mej	CHU	ne	14.02.1628	Čheng-ta	CHU			ne	Liu-taj	ZHOU	3
Tüng-pao	ZHOU	ne	22.12.1669	Aj-tün	ZHOU	Hoa	KUEJ	ne			
Wei	CHEN	ano	09.07.1646	Tu	CHEN	Tíng	WANG	ne			
Xang	WANG	ano	19.01.1605					ano			
Yen	ZHOU	ne	30.10.1649	Liu-taj	ZHOU	Jü-ling	TI	ne	Chinh	LUO	1
Zhang-ta	ZHOU	ano	05.09.1670	Če-ming	ZHOU	Mei	XI	ne			
Žu-čen	WU	ne	07.10.1630	Saj	WU			ne	Liu-taj	ZHOU	2

Tab. 1: Cvičná data v tabulce OSOBY

PROVINCIE	PREFEKTURA	OKRES	OKRES_CH	ROD	ROD_CH	NACELNIK	NACELNIK_CH
Ťiang-Su	Su-Čou	Ling-pi	灵璧县	KUEJ	马匹	Khoa	霍阿
Ťiang-Su	Su-Čou	Ling-pi	灵璧县	WU	吴	Saj	萨伊
Ťiang-Su	Su-Čou	Ling-pi	灵璧县	ČOU	听力	Ťüan	团
Ťiang-Su	Su-Čou	Siao	萧县	CHU	一個	Čheng-ta	契城达
Ťiang-Su	Su-Čou	Siao	萧县	ZHOU	周	Liu-taj	刘冠
Ťiang-Su	Su-Čou	Siao	萧县	RONG	二	Sü	洗
Ťiang-Su	Su-Čou	Siao	萧县	TI	的	Šeng	看！
Ťiang-Su	Su-Čou	Tang-šan	砀山县	LUO	罗	Cheng	住
Ťiang-Su	Su-Čou	Tang-šan	砀山县	XI	十一	Keng	带来
Ťiang-Su	Su-Čou	Tang-šan	砀山县	CHEN	陈	Tu	经过
Ťiang-Su	Su-Čou	Tang-šan	砀山县	WANG	王	Xang-li	祥力

Tab. 2: Cvičná data v tabulce RODY

4.3.2 Kritické poznámky

Teď tedy mám připravená data a můžu začít s dotazy. Když však prohlížím obě tabulky a připomínám si náročnost jejich přípravy, vybaví se mně dva aspekty. Jeden ryze praktický, jeden databázově profesionální. Jako autor komplexního informačního systému bych je v reálném provozu musel mít vyřešené.

A. Při provozování bude někdo udržovat data v aktuálním stavu. Bude vkládat nová data, bude opravovat resp. upravovat některé stávající hodnoty, bude event. vypouštět již nepotřebná data. Určitě to nebude databázový specialista, ale odborník např. v historických vědách. Proto při všech naznačených činnostech bude spoléhat na to, že používaný informační systém mu bude pomáhat. A to jak vhodným prostředím (např. přehlednými formuláři), tak zejména aby tam „nevkládal blbosti“ - tak nazývají počítačová laici kontrolu dat. V mém případě namátkou třeba to, aby každý zadaný rod v tabulce OSOBY byl uveden v tabulce RODY. A protože se zadávání bude provádět pomocí formulářů, pak jejich přehlednost a kontrolní funkce jsou svázány.

Pro mě jakožto autora je aspekt A relativně samostatnou oblastí k řešení. Už se na ni těším - na formuláře, čarokrásně vybarvené, předkládající uživateli texty v desítkách úžasných fontů (tady vidíte, že - jak mně upozornil kolega - jsem zdaleka ještě datařsky nevyspěl). Mým hlavním cílem teď je však zvládnout strukturu dat a na ni aplikovat dotazy.

B. Jako autora návrhu právě té struktury mně však profesionálně mrzí (ale což, jsem teprve začátečník), že jsem přehlédl minimálně jednu věc. V tabulce RODY je uveden jeho NÁČELNÍK (= hlava rodu, principiálně by tam mohl mít uvedena i další svá data). Ovšem tento náčelník je současně i osobou svého rodu, svého okresu, své prefektury. Musí tedy současně být uveden i v tabulce OSOBY. V této tabulce mám i datové pole HLAVA RODU (=náčelník) [Ano, Ne]. A to je právě ta situace, která mně s odstupem času velmi připomíná situaci s obávanými redundantními údaji: změna jednoho (nebo více) podmiňuje změnu jiného (nebo více). Když dojde ke smutné události a náčelník zemře, musí se změnit současně jak tabulka RODY, tak dokonce dvě OSOBY (původní náčelník už není náčelník, a jeho nástupce tuto hodnotu nabyt). Z hlediska návrhu struktury to ale vůbec není všechno: co když někdo sice změní v tabulce RODY náčelníka na nového, tomu v tabulce OSOBY změní pole HLAVA RODU na „Ano“, ale zapomene původnímu náčelníkovi změnit toto pole na „Ne“?

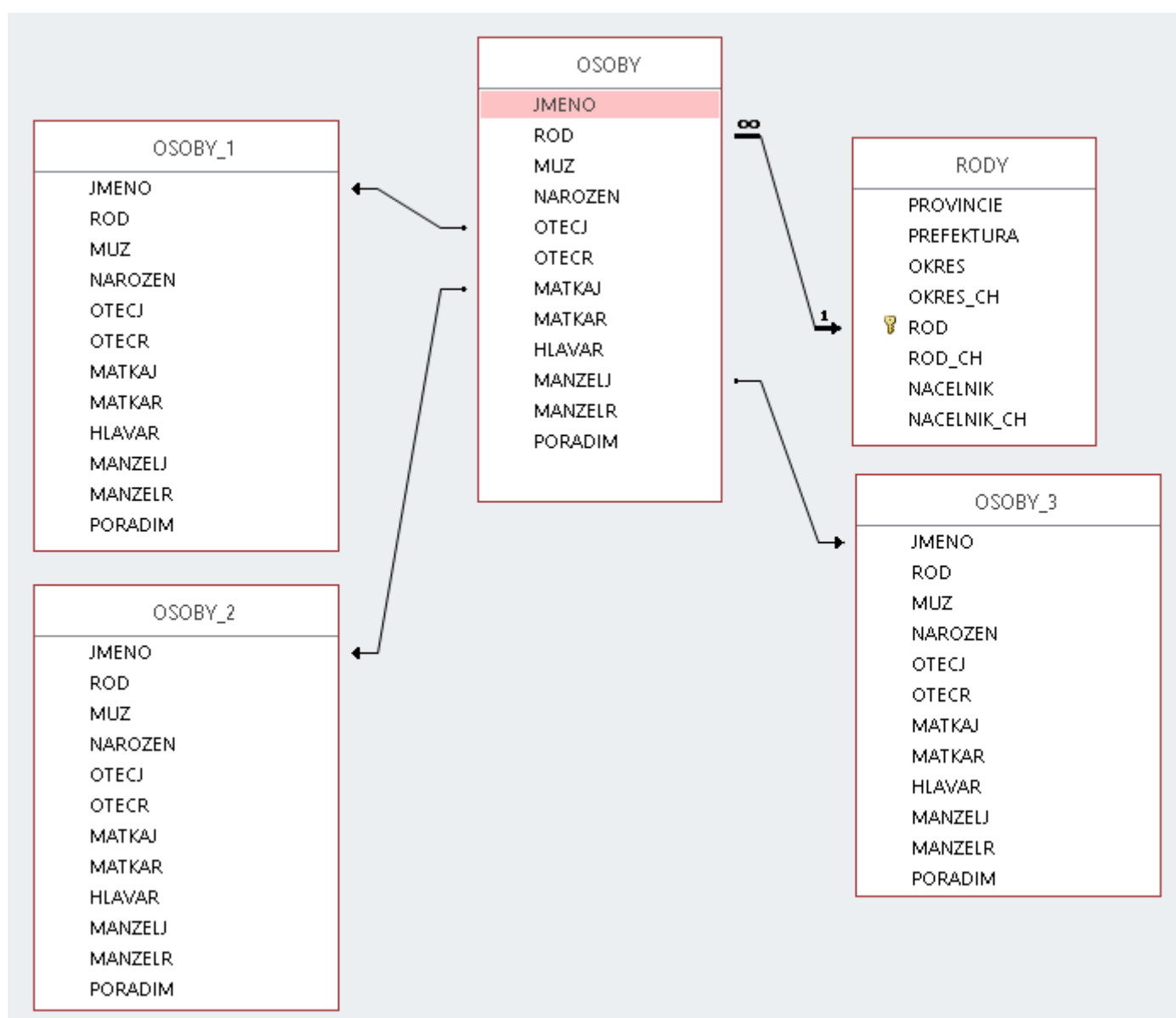
Nějaké nápady už mám v hlavě. Proč vůbec mám v tabulce RODY pole NÁČELNÍK? Vždyť pro jeho nalezení stačí vyhledat v tabulce OSOBY tu osobu s daným rodem, která má v poli HLAVA RODU hodnotu „Ano“. To by pak ale znamenalo vyřešit výše zmíněnou situaci se dvěma náčelníky jednoho rodu. Atd atd ...

No, zatím nechám data tak jak jsou. Nezbyvá ale než parafrázovat poslední větu náčelníka z prologu: Víím, že moje řešení budu muset ještě upravit, na některé situace by v reálném provozu zatím nedokázalo správně reagovat.

4.4 Vazby mezi datovými zdroji

Řešení nastíněné v úvodu této kapitoly jsem si mohl dovolit, protože vím, že v jednom dotazu je možno odkazovat na více datových zdrojů současně. Podstatné však je, že i na jeden (stejný) datový zdroj několikrát v různém kontextu - ten chápu jen já jako autor dotazu, databázovému systému je to fuk. Musím mu ale jeho nástroji sdělit, v jakém vzájemném vztahu ty dva nebo více přístupů ke stejnému zdroji jsou. Především je tedy musím nejen já sám, ale hlavně databázový systém rozlišit. V textu dotazu se k tomu využívá přidělení tzv. alias.

Tuto možnost lze pak výhodně použít pro definici propojení z jednoho exempláře tabulky OSOBY do jiného exempláře téže tabulky. Názorně to dokumentuje např. jedno z možných schémát:



Při procházení tabulky OSOBY zpřístupní databázový systém řádek otce v téže fyzické tabulce, ovšem logicky zpřístupněné pod alias OSOBY_1 (mnou chápané v kontextu tabulky otců). Analogicky pro matky a manžele.

Důležitá poznámka: Pokud se v hluboké minulosti nepodaří dopátrat údajů o otci nějaké osoby a pole OTECJ v tabulce OSOBY zůstane prázdné, pak mají všechna datová pole řádku ve zdroji OSOBY_1 hodnotu **null**. Analogicky pro matky (OSOBY_2) a manžele (OSOBY_3). Stejně pak platí pro vztah mezi polem ROD a tabulkou RODY.

4.5 Dotazy

Dotazy směřující do datového zdroje takto organizovaného budou o poznání složitější než v případě shora popsaného jednoduchého řešení. Tam totiž všechny tabulky měly jednoznačnou strukturu s jednoznačným významem jednotlivých datových polí. Bylo tedy možno vytvořením pouhého propojení mezi dvěma z nich vytvořit „virtuální“ strukturu obsahující vše potřebné (tam definované, už datově připravené) pro jeden řádek výstupu dotazu. V případě jediné, vícenásobně přístupované tabulky (zde OSOBY) směřujeme k témuž. Při procházení (event. filtrování, event. řazení, event. seskupování) zdroje OSOBY ale pro konstrukci každého řádku výstupu potřebuji řádek souvisejícího datového zdroje - ale to je jiný (případně tentýž) řádek téže tabulky OSOBY (např. pro řádek Dámy potřebuji i řádek Manžela). Jeden řádek výstupu tedy bude tvořen datovými poli nebo výrazy z nich vytvořenými, z nichž část pochází z jednoho „vydání“ tabulky OSOBY a část z druhého „vydání“ téže tabulky (přístupné ovšem pod jiným alias). Čerpání ze dvou zdrojů lze obecně dosáhnout především buď použitím jejich kartézského součinu nebo vytvořením propojení (join). Stejně je to samozřejmě v případě, kdy jde o „dvojnásobný“ přístup do téhož datového zdroje rozlišeného pomocí alias.

Dotazovací jazyk SQL v dnešní době disponuje i mechanismem vnořených dotazů, tzv. poddotazů. To je ale ten okamžik, kdy moje sebevědomí kleslo k nule. Proto, než dotaz jediný, ale díky poddotazům složitý a pro mou úroveň zatím těžce pochopitelný, jsem raději nejprve vytvořil jeden (nebo i více) „pomocných“ dotazů D0, z jehož (nebo z jejichž) výsledků čerpá můj finální dotaz D1, odpovídající na žádanou otázku.

Níže uvedená řešení jsou jen některá z několika možných. K jednotlivým úlohám jsem přidal stručné komentáře. K té první úloze jsem se snažil naznačit detailněji celý myšlenkový proces, který mně vedl právě k takovému řešení. A protože nedosahuji kdovíjakých databázových výšin, určitě existují řešení lepší.

4.5.1 Úloha 1 - detailně krok po kroku

Kolik mých Dam je ze stejného (stejných) rodů a z jakých okresů jsou?

Otázky z prologu jsou formulovány cíleně pro jedinou osobu, náčelníka Liu-taje. Abych se trochu pocvičil, zkusil jsem nejprve zadání trochu zobecnit: kolik manželek jednotlivých manželů je ze stejného rodu? Odpověď na konkrétní otázku z prologu pak bude jen odfiltrováním z této obecnější.

Dána jsou tedy data dle Tab. 1. Ta obsahuje mnoho osob, ale jen některé jsou manželkami. Osoby (= řádky tabulky), které jsou manželkami, se poznají neprázdným polem ve sloupci MANZELJ; tam uvedeno jméno manžela. Pro naše účely je však lepší formulace tato: Nemají pole ve sloupci MANZELJ prázdné (není Null) - to proto, že v dotazech mohu použít funkci IsNull, která tu „prázdnotu“ testuje a tato formulace mně na ni přímo navádí. Tedy dotaz (proč barevná odlišnost viz níže)

```
/* D00_ManzelkySeznam */  
  
select *  
from OSOBY  
where not IsNull(MANZELJ)
```

má jako výsledek seznam všech manželek

JMENO	ROD	MUZ	NAROZEN	OTECJ	OTECR	MATKAJ	MATKAR	HLAVAR	MANZELJ	MANZELR	PORADIM
Jen-chuej	CHU	ne	23.12.1629	Čheng-ta	CHU			ne	Liu-taj	ZHOU	1
Tung-mej	CHU	ne	14.02.1628	Čheng-ta	CHU			ne	Liu-taj	ZHOU	3
Hang	KUEJ	ne	07.08.1651	Khoa	KUEJ			ne	Aj-tün	ZHOU	1
Hoa	KUEJ	ne	09.02.1650	Khoa	KUEJ			ne	Aj-tün	ZHOU	2
Ki	RONG	ne	10.08.1647	Sü	RONG			ne	Wei	CHEN	1
Jü-ling	TI	ne	04.05.1627	Šeng	TI			ne	Liu-taj	ZHOU	4
Ting	WANG	ne	17.05.1626	Xang-li	WANG			ne	Tu	CHEN	1
Ping-an	WU	ne	18.04.1631	Saj	WU			ne	Ťüan	ČOU	1
Žu-čen	WU	ne	07.10.1630	Saj	WU			ne	Liu-taj	ZHOU	2
Mei	XI	ne	15.05.1651	Keng	XI			ne	Če-ming	ZHOU	1
Pej-pej	ZHOU	ne	05.09.1649	Liu-taj	ZHOU	Žu-čen	WU	ne	Tuan	XI	1
Tching	ZHOU	ne	30.07.1650	Liu-taj	ZHOU	Jü-ling	TI	ne	Tai	RONG	1
Yen	ZHOU	ne	30.10.1649	Liu-taj	ZHOU	Jü-ling	TI	ne	Chinh	LUO	1

To vybere skutečně všechny dámy, které jsou manželkami, ale v dalším kroku se potřebuji dozvědět, kolik jich je u jednotlivých manželů z jednotlivých rodů. Je-li jich více než 1, jsou to sestry. Je tedy třeba výsledek předchozího dotazu nechat seskupit podle jména manžela a rodu dámy, a nechat spočítat, kolik řádků v každé takové skupině je. Mám-li předchozí dotaz vytvořený a uložený (rozumí se text dotazu, nikoliv jeho výsledek!), mohu ho použít:

```
/* D00_PoctyManzelekRodu */

select ROD, MANZELJ, Count(ROD) as ZeStejnýchRodu
from D00_ManzelkySeznam
group by ROD, MANZELJ
```

dává (pozor! ROD je rodem Dámy, nikoliv Manžela!)

ROD	MANZELJ	ZeStejnýchRodu
CHU	Liu-taj	2
KUEJ	Aj-tün	2
RONG	Wei	1
TI	Liu-taj	1
WANG	Tu	1
WU	Liu-taj	1
WU	Ťüan	1
XI	Če-ming	1
ZHOU	Chinh	1
ZHOU	Tai	1
ZHOU	Tuan	1

Popsaným postupem jsem si vyzkoušel, že vytvořený a uložený dotaz lze použít jako datový zdroj jiného dotazu. Napadlo mě však, zda by náhodou nešlo obejít nutnost vytvoření a uložení prvního dotazu a teprve pak jeho následné použití. Vždyť přece v počtech také místo

```
x = 2 + 3
y = 4 * x
```

lze psát rovnou

```
y = 4 * (2 + 3)
```

Tak jsem to zkusil (barevně jsem si označil ten dříve ukládaný dotaz ManzelkySeznam - viz výše); prostě jsem ho jen zkopíroval Ctrl/C + Ctrl/V:

```

/* D00_PoctyManzelekRodu */

select ROD, MANZELJ, Count(ROD) as ZeStejnýchRodu
from
(
    select *
    from OSOBY
    where not IsNull(MANZELJ)
)
group by ROD, MANZELJ

```

a ono to skutečně funguje. Ušetřil jsem jeden nejprve ukládaný dotaz, výsledek je zcela stejný jako dříve. Dokonce lze takto vytvořenému datovému zdroji přidělit alias (= posloupnost písmen a čísel, začínající písmenem, tedy nejméně jedno písmeno - zde např. P), aby tento datový zdroj byl přehledněji použitelný ve složitějších případech:

```

/* D00_PoctyManzelekRodu */

select P.ROD, P.MANZELJ, Count(P.ROD) as ZeStejnýchRodu
from
(
    select *
    from OSOBY
    where not IsNull(MANZELJ)
) as P
group by P.ROD, P.MANZELJ

```

A ještě jeden nápad při zamyšlení se nad tímto dílčím krokem: Zde nejprve vytvořím vnořený dotaz, který odfiltruje (na vstupu) prázdná datová pole MANZELJ, a pak jeho výsledek nechám seskupit. Ten vnořený dotaz tady vůbec nic jiného nedělá, jen filtruje. Nešlo by čerpat přímo z tabulky OSOBY, přidat filtr na vstupu a ponechat seskupování? Vždyť databázový systém filtr na vstupu použije před vytvářením skupin. Zkusme

```

/* D00_PoctyManzelekRodu */

select ROD, MANZELJ, Count(ROD) as ZeStejnýchRodu
from OSOBY
where not IsNull(MANZELJ)
group by ROD, MANZELJ

```

a vidíme, že výsledek je opět úplně stejný. Pro účely budovaného dotazu jsou však zapotřebí jen ty řádky, kde počty ZeStejnýchRodu jsou větší než 1. Stačí předchozí dotaz doplnit o filtr na výstupu:

```

/* D00_PoctyManzelekRodu */

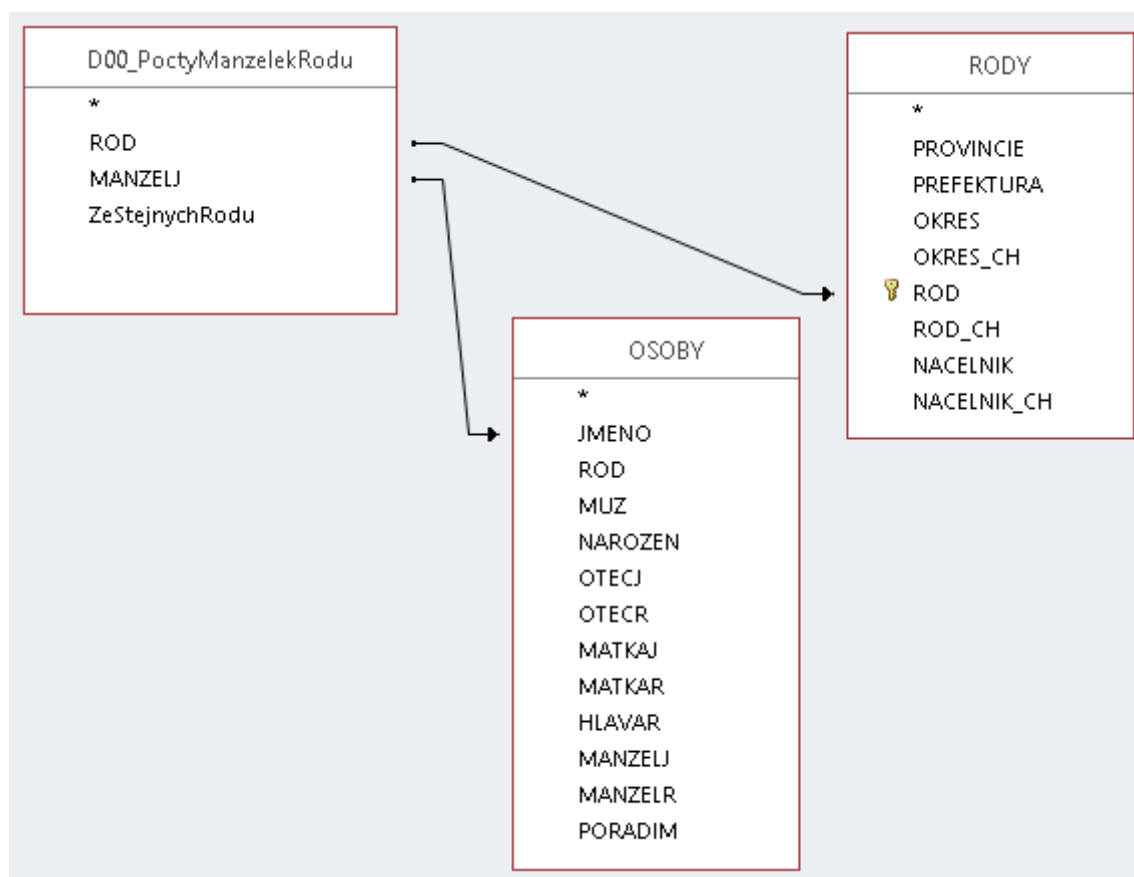
select ROD, MANZELJ, Count(ROD) as ZeStejnýchRodu
from OSOBY
where not IsNull(MANZELJ)
group by ROD, MANZELJ
having Count(ROD) > 1

```

ROD	MANZELJ	ZeStejnýchRodu
CHU	Liu-taj	2
KUEJ	Aj-tün	2

Poznámka: Až do tohoto místa jsem používal pro text příkazů SQL tu písmo tučné, tu barevně odlišené. Bylo to jen pro moji lepší orientaci, co je vlastně co. Protože interpreti SQL znají pouze jediný typ písma, budu - až na komentované výjimky - používat pro příkazy jen jednotné „normální“ písmo.

Otázka z prologu však požaduje také okres rodu dam. A když už v tom jsme, přidáme na výstup dotazu i rod manžela (případně další údaje o manželovi). K tomu je třeba při procházení všech řádků výsledku dotazu D00_PoctyManzelekRodu připojit k nim vždy ten jediný ze souvisejících tabulek (left join) OSOBY a RODY:



```
/* D01_RodyManzelekSester */
```

```

select
  D00_PoctyManzelekRodu.MANZELJ as Manzel,
  MANZELE.ROD as RodManzela,
  D00_PoctyManzelekRodu.ROD as RodDam,
  D00_PoctyManzelekRodu.ZeStejnýchRodu as DamRodu,
  RODY.OKRES as OkresDam,
  RODY.OKRES_CH as OkresDamCh
from
  (
    D00_PoctyManzelekRodu left join RODY on D00_PoctyManzelekRodu.ROD = RODY.ROD
  )
  left join OSOBY as MANZELE on D00_PoctyManzelekRodu.MANZELJ = MANZELE.JMENO
  
```

což dává

Manzel	RodManzela	RodDam	DamRodu	OkresDam	OkresDamCh
Liu-taj	ZHOU	CHU		2 Siao	萧县
Aj-tün	ZHOU	KUEJ		2 Ling-pi	灵璧县

No a finální krok: odfiltrovat jen vypravěče prologu Liu-taje je hračkou:

```
/* D01_RodyMychManzelekSester */

select
  D00_PoctyManzelekRodu.MANZELJ as Manzel,
  MANZELE.ROD as RodManzela,
  D00_PoctyManzelekRodu.ROD as RodDam,
  D00_PoctyManzelekRodu.ZeStejnýchRodu as DamRodu,
  RODY.OKRES as OkresDam,
  RODY.OKRES_CH as OkresDamCh
from
  (
    D00_PoctyManzelekRodu left join RODY on D00_PoctyManzelekRodu.ROD = RODY.ROD
  )
  left join OSOBY as MANZELE on D00_PoctyManzelekRodu.MANZELJ = MANZELE.JMENO
where
  D00_PoctyManzelekRodu.MANZELJ = "Liu-taj"
```

Manzel	RodManzela	RodDam	DamRodu	OkresDam	OkresDamCh
Liu-taj	ZHOU	CHU		2 Siao	萧县

Úžasné! Tak teď mně trošinku stouplо sebevědomí. Se zálibou si prohlížím výsledek - ale jedna věc se mně přeci jen moc nelíbí. Ten uložený dotaz D00 jsem si nazval sice dobře vypovídajícím identifikátorem D00_PoctyManzelekRodu, ale je to hodně dlouhé. To dost znepřehledňuje čtení a hlavně pochopení funkce toho finálního dotazu D01 (viz kousek nahoře). Pojmenovat ten první dotaz třeba D00_ABC se mně nechce, za měsíc bych nevěděl, co jsem tím myslel. Tak mně napadlo uvnitř finálního dotazu D01 použít alias pro datový zdroj D00_PoctyManzelekRodu (např. PMR); bude to trochu přehlednější a přímo v tom dotazu mně zůstane zachován identifikátor dotazu D00:

```
/* D01_RodyMychManzelekSester */

select
  PMR.MANZELJ as Manzel,
  MANZELE.ROD as RodManzela,
  PMR.ROD as RodDam,
  PMR.ZeStejnýchRodu as DamRodu,
  RODY.OKRES as OkresDam,
  RODY.OKRES_CH as OkresDamCh
from
  (
    D00_PoctyManzelekRodu as PMR left join RODY on PMR.ROD = RODY.ROD
  )
  left join OSOBY as MANZELE on PMR.MANZELJ = MANZELE.JMENO
where
  PMR.MANZELJ = "Liu-taj"
```

Že tohle pořád funguje mně v tomto případě ani moc nepřekvapuje. Co ale hned padne do oka je to, že identifikátor D00_PoctyManzelekRodu (schválně jsem si ho zvýraznil tučně a jinou barvou) se vyskytuje jen jednou, a to v klauzuli **from**. Už jednou jsem shora použil místo volání uloženého dotazu přímo jeho zápis do textu následného dotazu, viz to **y = 4 * (2 + 3)**. Nešlo by to i tady? ŠLO:

```
/* D01_RodyMychManzelekSester */

select
  PMR.MANZELJ as Manzel,
  MANZELE.ROD as RodManzela,
  PMR.ROD as RodDam,
  PMR.ZeStejnýchRodu as DamRodu,
  RODY.OKRES as OkresDam,
  RODY.OKRES_CH as OkresDamCh
from
  (
    (
      select ROD, MANZELJ, Count(ROD) as ZeStejnýchRodu
      from OSOBY
      where not IsNull(MANZELJ)
      group by ROD, MANZELJ
      having Count(ROD) > 1
    ) as PMR
    left join RODY on PMR.ROD = RODY.ROD
  )
  left join OSOBY as MANZELE on PMR.MANZELJ = MANZELE.JMENO
where
  PMR.MANZELJ = "Liu-taj"
```

opět se stejným výsledkem:

Manzel	RodManzela	RodDam	DamRodu	OkresDam	OkresDamCh
Liu-taj	ZHOU	CHU	2 Siao	萧县	

Tak ale tohle mně překvapuje dvojnásobně. Jednak že takovou složitou věc databázový systém zvládne. Co mně ale překvapuje víc je to, že jsem tohle vymyslel já a to vlastní hlavou :-)

4.5.2 Úloha 2

Se kterou mojí Dámou také ještě nemám žádné děti?

Tady je myšlenka jednodušší, než je následný zápis textu dotazu. Je to s tou Dámou nebo s těmi Dámami, které jsou 1) manželkami, a to mými, a 2) nejsou uvedeny jako matka u žádné osoby.

Nechám tedy procházet OSOBY v kontextu Manželů (Levá tabulka, alias A), které zleva připojují OSOBY v kontextu Dam (pravá tabulka, alias B). Připomenu si, co left join znamená: zahrne všechny záznamy z levé tabulky (mnou chápané v kontextu manželů), i když neexistuje související záznam v pravé tabulce (mnou chápané v kontextu manzelek) - v tom případě odevzdá odkaz na hodnotu kteréhokoliv pole pravé tabulky Dam hodnotu null. Jinak: bude-li k řádku levé tabulky (k manželovi) vůbec připojen nějaký řádek pravé tabulky (dámy), bude to ta dáma, které je osoba levé tabulky manželem.

Na výstup nechám poslat jen ty osoby, kde:

- Jméno osoby jakožto manžela je Liu-Taj.
- Připojený řádek pravé tabulky není null.
- Jméno připojené dámy není vyplněno jako matka u žádné osoby tabulky všech osob.

```

/* D02_MojeManzelkyBezDeti */

select
  B.JMENO as JmDamy, B.ROD as RodDamy,
  A.JMENO as JmMuze, A.ROD as RodMuze
from
  OSOBY A left join OSOBY B
    on A.JMENO=B.MANZELJ
where
  A.JMENO="Liu-taj"
  and (not IsNull(B.JMENO) )
  and B.JMENO not in
    (
      select C.MATKAJ
      from OSOBY C
      where not IsNull(C.MATKAJ)
    )
order by
  A.JMENO

```

Výsledek:

JmDamy	RodDamy	JmMuze	RodMuze
Tung-mej	CHU	Liu-taj	ZHOU

4.5.3 Úloha 3

Jak je stará moje nejstarší vnučka mého rodu a v kolika letech svého otce se mu narodila?

Zkusil jsem podobný postup, kterým jsem začal první úlohu. Tam jsem nejprve vybral všechny manželky, tady zkusím nejprve vybrat všechny vnučky. Odpovědí na dotaz bude odfiltrovaná část vybraných vnuček.

Manželky jsem si definoval jako osoby s neprázdným manželem. To byly evidentně ženy - a pokud jsem správně nachystal data, nemusel jsem tu ženskost testovat. Analogicky si vnučky můžu definovat jako osoby s neprázdným dědečkem. Tady ovšem testovat na ženu musím, protože děda může mít i vnuky a dotaz zní na vnučky.

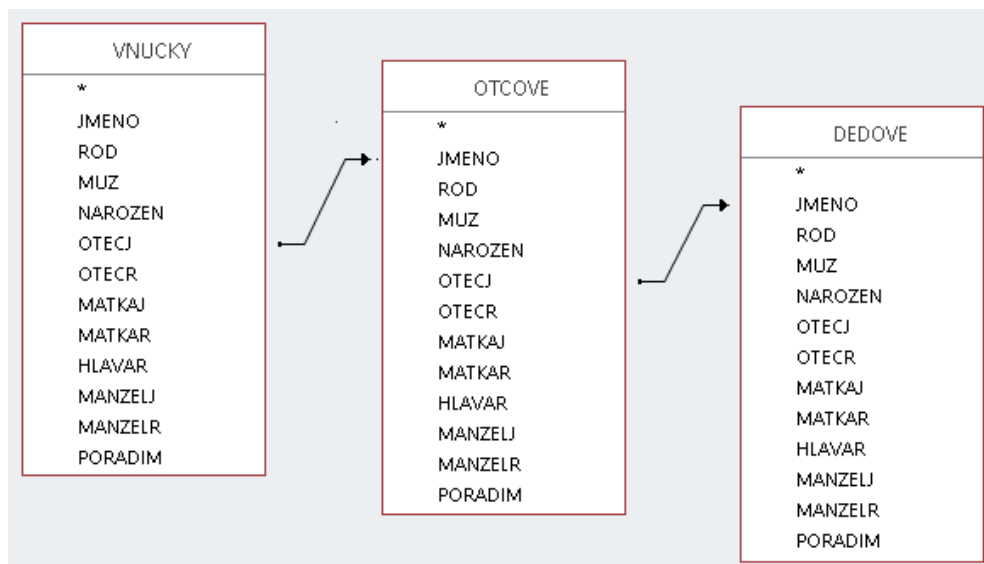
To ale není jediný zádrhel. Každá vnučka má dva dědy: z otcovy strany a z matčiny strany. I vypravěč Liu-taj může mít mnoho vnuček, tedy dcer jeho dětí. Dcery jeho synů jsou vnučky jeho rodu, ale dcery jeho dcer jsou sice jeho vnučky, ale nejsou jeho rodu (jsou rodu manžela jeho dcery).

Bude to asi komplikovanější. Možná ani ne tak sestavení samotných dotazů, ale ověření jejich správností. Teď jsem se musel vrátit k úpravě cvičných dat (výsledek už je zahrnut v tabulce shora). Pokud hodlám sestavit dotaz na všechny vnučky, musí výsledek počítat i s těmito situacemi (viděno očima vypravěče):

- Nestarší vnučka je sice vnučka, ale nikoliv jeho.
- Nejstarší jeho vnučka je sice jeho, ale ne jeho rodu.

Proto jsem přidal do dat na úroveň vnuček dvě: Jiao rodu CHEN (úplně nejstarší, s rodem vypravěče nemá nic společného), a Lan rodu XI (nejstarší z vnuček vypravěče, ale nikoliv jeho rodu). Obě se musí v seznamu všech vnuček objevit, ale žádná z nich nesmí mít v poli svého rodu rod vypravěče a nesmí se objevit v odpovědi na výsledný dotaz (jinak mám blbě testovací data nebo sestavený dotaz).

Pro sestavení dotazů mně hodně pomohlo schéma závislostí jednotlivých datových zdrojů:



Pokud dotaz budu realizovat pomocí klauzulí „left join“ naznačeným způsobem (z Vnuček přes jejího otce do Otců, z Otců přes jeho otce do Dědů), vyřeší se tím i výběr dědy jen rodu vnučky. Stačí pro každou vnučku testovat dědu na „not null“ - to zahrne i případ, že už ani otec nemůže být „null“ (protože jinak by nutně i děda byl „null“):

```

/* D03a_VsechnyVnucky - zeny s neprazdnym dedou */

select
  VNUCKY.JMENO as JmenoVnucky, VNUCKY.ROD as RodVnucky,
  VNUCKY.MATKAJ as JmenoMatky, VNUCKY.MATKAR as RodMatky,
  OTCOVE.JMENO as JmenoOtce, OTCOVE.ROD as RodOtce,
  DEDOVE.JMENO as JmenoDedy, DEDOVE.ROD as RodDedy,
  Round((#7/4/1674#-VNUCKY.NAROZEN)/365.25,2) as VekVnucky,
  Round((VNUCKY.NAROZEN-OTCOVE.NAROZEN)/365.25,2) as VekOtce
from
  (
    OSOBY as VNUCKY left join OSOBY as OTCOVE
      on VNUCKY.OTECJ = OTCOVE.JMENO
  )
  left join OSOBY as DEDOVE
    ON OTCOVE.OTECJ = DEDOVE.JMENO
where
  not IsNull(DEDOVE.JMENO) and
  not VNUCKY.MUZ

```

Výsledek pro všechny vnučky zatím dobrý, jsou tam i Jiao i Lan:

JmenoVnucky	RodVnucky	JmenoMatky	RodMatky	JmenoOtce	RodOtce	JmenoDedy	RodDedy	VekVnucky	VekOtce
Siao-pao	ZHOU	Hang	KUEJ	Aj-tün	ZHOU	Liu-taj	ZHOU	1,92	23,79
Tüng-pao	ZHOU	Hoa	KUEJ	Aj-tün	ZHOU	Liu-taj	ZHOU	4,53	21,17
Jiao	CHEN	Kí	RONG	Wei	CHEN	Tu	CHEN	8,42	19,57
Lan	XI	Pej-pej	ZHOU	Tuan	XI	Keng	XI	6,46	16,91

Následný dotaz vybere z předchozích VsechVnuček jen vnučky vypravěče Liu-taje a z nich tu (nebo ty) jeho nejstarší. Výsledek:

```

/* D03b_MojeNejstarsiVnucka */

select *
from D03a_VsechnyVnucky
where
  JmenoDedy = "Liu-taj" and
  VekVnucky = (select Max(VekVnucky)
               from D03a_VsechnyVnucky
               where JmenoDedy="Liu-taj")

```

Výsledek:

JmenoVnucky	RodVnucky	JmenoMatky	RodMatky	JmenoOtce	RodOtce	JmenoDedy	RodDedy	VekVnucky	VekOtce
Tung-pao	ZHOU	Hoa	KUEJ	Aj-tün	ZHOU	Liu-taj	ZHOU	4,53	21,17

4.5.4 Úloha 4

Kdy přibližně budu muset jet za obchodníkem WU se svou Druhou Dámou, o níž vím, že už dvanáct týdnů neutrácí a strádá si na nové kimono za sedm Zlatých a padesát Stříbrných?

Stejná poznámka jako u stejné shora uvedené úlohy s jednoduchým řešením se čtyřmi tabulkami: Tento dotaz vlastně vůbec nepotřebuje žádný datový zdroj, protože jen počítá ze zadaných „konstant“. Příkaz Select však povinně musí datový zdroj obsahovat, může však být jakýkoliv s alespoň jedním řádkem. Na druhé straně by měl po svém provedení odevzdat právě jeden řádek výsledku (ať nevidím několik stejných řádků). Já zvolil tabulku OSOBY. Jediného řádku výsledku dosáhnou např. vhodně zadaným filtrem na vstupu (třebas JMENO="Liu-taj" - ten tam je a právě jednou), nebo použitím nepříliš elegantního, ale fungujícího Select Top 1.

```

/* D04a_KdyProKimono */

select top 1
  #7/4/1674# as Dnes,
  12*7 as UzStradaDni,
  Dnes-UzStradaDni as StradaOdDne,
  7.5 * (365.25 / 12) as MusiStradatDni,
  StradaOdDne+Round(MusiStradatDni) as OdjezdKolem
from OSOBY

```

Výsledek:

Dnes	UzStradaDni	StradaOdDne	MusiStradatDni	OdjezdKolem
04.07.1674	28	06.06.1674	228,125	20.01.1675

4.5.5 Úloha 5

Kolik roků má (nebo by měl) můj nejstarší tchán?

Už dvakrát (u manželek a vnuček) se mně celkem vyplatilo nejprve shromáždit všechny, a teprve pak konkretizovat ty požadované - minimálně proto, abych se jako začátečník pocvičil. Možná to jde rovnou jinak, ale já zatím preferuji jistotu. Protože jsem se hned na začátku do toho pořádně zamotal, půjdu teď raději pomalu krok po kroku. Zkusím nejprve vybrat všechny tchány. Tchán je osoba, která je otcem manželky. Manželka je osoba s vyplněným datovým polem MANŽELJ.

Budu čerpat z tabulky OSOBY; abych měl kontrolu, nechám na výstup poslat manžela, manželku a tchána. To ale jen v případě, že datové pole MANŽELJ není „null“. Na výstupu tedy budu mít pouze řádky manželek (!) = žen, kterým je [MANŽELJ] manželem:

```
/* T05a_Manzel_Manzelka_Otec */

select
MANZELJ, MANZELR,
JMENO as MANZELKAJ, ROD as MANZELKAR,
OTECJ, OTECR
from OSOBY
where
not IsNull(MANZELJ) and
not IsNull(OTECJ)
order by MANZELJ, JMENO
```

MANZELJ	MANZELR	MANZELKAJ	MANZELKAR	OTECJ	OTECR
Aj-tün	ZHOU	Hang	KUEJ	Khoa	KUEJ
Aj-tün	ZHOU	Hoa	KUEJ	Khoa	KUEJ
Če-ming	ZHOU	Mei	XI	Keng	XI
Chinh	LUO	Yen	ZHOU	Liu-taj	ZHOU
Liu-taj	ZHOU	Jen-chuej	CHU	Čheng-ta	CHU
Liu-taj	ZHOU	Jü-ling	TI	šeng	TI
Liu-taj	ZHOU	Tung-mej	CHU	Čheng-ta	CHU
Liu-taj	ZHOU	Žu-čen	WU	Saj	WU
Tai	RONG	Tching	ZHOU	Liu-taj	ZHOU
Tu	CHEN	Ting	WANG	Xang	WANG
Tuan	XI	Pej-pej	ZHOU	Liu-taj	ZHOU
Ťüan	ČOU	Ping-an	WU	Saj	WU
Wei	CHEN	KI	RONG	Sü	RONG

To přesně jsem (pro kontrolu) zamýšlel. Nyní další úvaha: V prvním sloupci je jméno manžela, v předposledním jméno otce manželky (!) = jméno tchána manžela. Abych se dobral nejstaršího tchána, k tomu vlastně vůbec nepotřebuji manželky. To je jednoduché: prostě vypustím v předchozím dotazu řádek [JMENO as MANZELKAJ, ROD as MANZELKAR,] Manžel ale může mít za manželky sestry (ty mají stejného otce), proto pro tohoto manžela by byl jeho stejný řádek vícekrát. Pro mne jakožto sice trochu pokročilejšího, ale přeci jen pořád začátečníka, bude nejjednodušší použít odstranění duplicit na výstupu dotazu uvedením klíčového slova „distinct“ (česky „odlišný“) - nesmím to zaměnit s podobně vypadajícím klíčovým slovem „distinctrow“, které odstraňuje duplicity na vstupu dotazu, a tam žádné nejsou. Také mohu sloupce obsahující otce manželek pojmenovat z kontextu vyplývajícím názvem Tchán:

```
/* T05b_Manzel_Tchan */

select distinct
MANZELJ, MANZELR,
OTECJ as TCHANJ, OTECR as TCHANR
from OSOBY
```

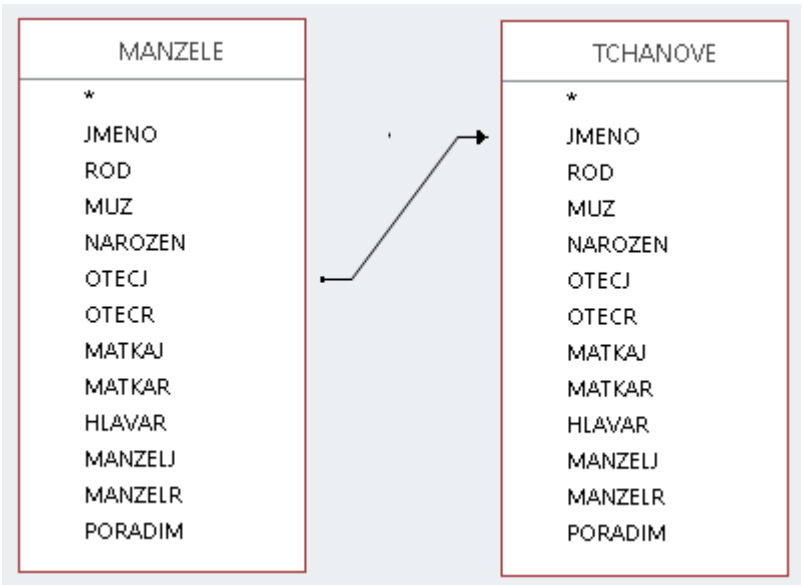
```

where
  not IsNull(MANZELJ) and
  not IsNull(OTECJ)
order by MANZELJ

```

MANZELJ	MANZELR	TCHANJ	TCHANF
Aj-tün	ZHOU	Khoa	KUEJ
Če-ming	ZHOU	Keng	XI
Chinh	LUO	Liu-taj	ZHOU
Liu-taj	ZHOU	Čheng-ta	CHU
Liu-taj	ZHOU	Saj	WU
Liu-taj	ZHOU	Šeng	TI
Tai	RONG	Liu-taj	ZHOU
Tu	CHEN	Xang	WANG
Tuan	XI	Liu-taj	ZHOU
Ťüan	ČOU	Saj	WU
Wei	CHEN	Sü	RONG

Dále: výsledek předchozího dotazu je zapotřebí doplnit o sloupec VĚK TCHÁNA. Tedy z předchozího dotazu při procházení tabulky OSOBY (kterou si nazvu názorněji MANŽELÉ) z každého jejího řádku získat přístup do téže tabulky (které jakožto druhý zdroj přidělím alias TCHÁNOVÉ) do řádku odpovídajícího tchána. Schematicky (pozor, OTECJ je otcem MANŽELKY, viz hned první dotaz této úlohy):



```

/* T05c_Manzel_Tchan_VekTchana */
select distinct
  MANZELE.MANZELJ, MANZELE.MANZELR,
  MANZELE.OTECJ as TCHANJ, MANZELE.OTECR as TCHANR,
  TCHANOVE.NAROZEN,
  Round((#7/4/1674#-TCHANOVE.NAROZEN)/365.25,2) as VekTchana
from
  OSOBY as MANZELE left join OSOBY as TCHANOVE
    on MANZELE.OTECJ = TCHANOVE.JMENO
where
  not IsNull(MANZELE.MANZELJ) and
  not IsNull(MANZELE.OTECJ)
order by MANZELE.MANZELJ

```

Pro moji kontrolu jsem si přidal i datum narození tchána:

MANZELJ	MANZELR	TCHANJ	TCHANR	NAROZEN	VekTchana
Aj-tün	ZHOU	Khoa	KUEJ	02.11.1626	47,67
Če-ming	ZHOU	Keng	XI	24.04.1629	45,19
Chinh	LUO	Liu-taj	ZHOU	23.12.1627	46,53
Liu-taj	ZHOU	Čheng-ta	CHU	24.12.1608	65,52
Liu-taj	ZHOU	Saj	WU	07.10.1609	64,74
Liu-taj	ZHOU	Šeng	TI	04.05.1607	67,17
Tai	RONG	Liu-taj	ZHOU	23.12.1627	46,53
Tu	CHEN	Xang	WANG	19.01.1605	69,45
Tuan	XI	Liu-taj	ZHOU	23.12.1627	46,53
Ťüan	ČOU	Saj	WU	07.10.1609	64,74
Wei	CHEN	Sü	RONG	14.07.1606	67,97

Odfiltrovat jen vypravěče prologu je pak hračka:

```

/* T05d_Moji_Tchanove */
select distinct
  MANZELE.MANZELJ, MANZELE.MANZELR,
  MANZELE.OTECJ as TCHANJ, MANZELE.OTECR as TCHANR,
  TCHANOVE.NAROZEN,
  Round((#7/4/1674#-TCHANOVE.NAROZEN)/365.25,2) as VekTchana
from
  OSOBY as MANZELE left join OSOBY as TCHANOVE
    on MANZELE.OTECJ = TCHANOVE.JMENO
where
  not IsNull(MANZELE.MANZELJ) and
  not IsNull(MANZELE.OTECJ) and
  MANZELE.MANZELJ = "Liu-taj"
order by MANZELE.MANZELJ

```

MANZELJ	MANZELR	TCHANJ	TCHANR	NAROZEN	VekTchana
Liu-taj	ZHOU	Čheng-ta	CHU	24.12.1608	65,52
Liu-taj	ZHOU	Saj	WU	07.10.1609	64,74
Liu-taj	ZHOU	Šeng	TI	04.05.1607	67,17

A stejně jednoduché je pak vybrat ty stejně staré nejstarší - alespoň jeden to je, ale může jich být více:

```

/* T05e_Moji_Nejstarsi_Tchanove */
select *
from T05d_Moji_Tchanove
where
  VekTchana = (select Max (VekTchana) from T05d_Moji_Tchanove)

```

MANZELJ	MANZELR	TCHANJ	TCHANR	NAROZEN	VekTchana
Liu-taj	ZHOU	Šeng	TI	04.05.1607	67,17