RNDr. Jakub Lokoč, Ph.D.
RNDr. Michal Kopecký, Ph.D.
Katedra softwarového inženýrství
Matematicko-Fyzikální fakulta
Univerzita Karlova v Praze

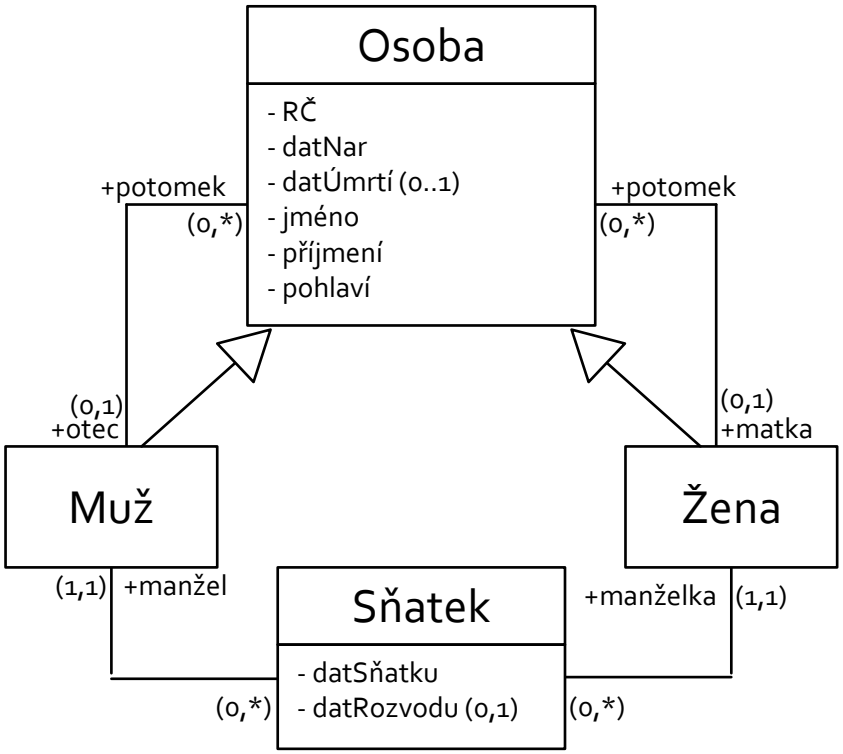
DBS – Příklady – Triggery
DBS – Examples – Triggers

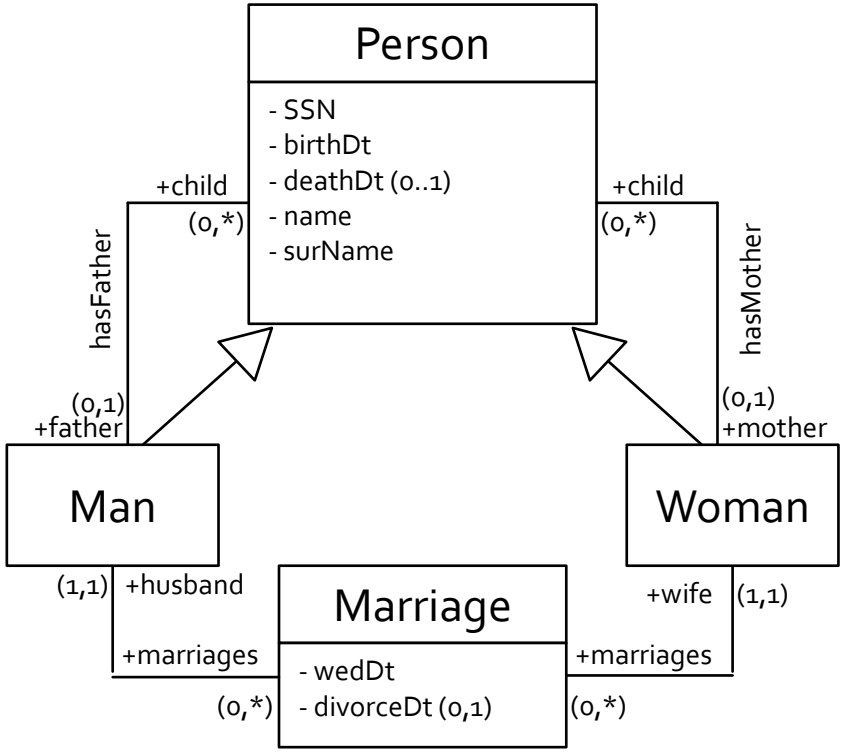
Triggery / Triggers

Matrika / Personal register

UML model

Poslední úpravou bude vytvoření hierarchie osob dle pohlaví, tedy přidání dvou entit Muž a Žena. To umožní detailněji specifikovat typy vztažených entit ve vztazích





SQL – Logický datový model
SQL – Logical Database Model

- Osoba(**RČ**,
  Jméno, Příjmení,
  DatNar, DatÚmrtí)
- Muž(**RČ**)
- Žena(**RČ**)
- Otec(**RČ**, RČM)
- Matka(**RČ**, RČŽ)
- Sňatek(**RČM,RČŽ,DatSňatku**,
  DatRozvodu)

- Osoba(**SSN**,
  Name, Surname,
  BirthDt, DeathDt)
- Man(**SSN**)
- Woman(**SSN**)
- Father(**SSN**, MSSN)
- Mother(**SSN**, WSSN)
- Marriage(**MSSN,WSSN,WedDt**,
  DivorceDt)

OCL – Příklad 1 / Example

- Matka musí být vždy starší, než její děti

  context o:Osoba **inv**
  o.matka.datNar
  <
  o.datNar

  nebo bezpečněji

  context o:Osoba **inv**
  o.matka→size()>o
  implies
  o.matka.datNar
  <
  o.datNar

- Mother has to be older than her children

  context p:Person **inv**
  p.mother.birthDt
  <
  p.birthDt

  or more safe

  context p:Person **inv**
  p.mother→size()>o
  implies
  o.mother.birthDt
  <
  p.birthDt

SQL – Příklad 1 / Example 1

- Matka musí být vždy starší, než její děti – nutno mít více triggerů, jedním z nich je ten, kontrolující vložení do tabulky Matka

```
create trigger check_matka_vek
on Matka
after insert
as
if exists(
   select *
   from inserted as I
     inner join Osoba as D -- dite
       on (D.RČ=I.RČ)
     inner join Osoba as M -- matka
       on (M.RČ=I.RČŽ)
   where D.datNar<=M.datNar)
begin
 RAISERROR('Dítě musí být mladší, než matka!', 16, 1);
 rollback transaction;
end;
```

- Mother has to be older than her children – it is necessary to have more triggers, one of them is the trigger, checking the insertion into Mother table

```
 create trigger check_mother_age
on Mother
after insert
as
if exists(
   select *
   from inserted as I
     inner join Person as C -- child
       on (C.SSN=I.SSN)
     inner join Person as M -- mother
       on (M.SSN=I.SSNW)
   where C.birthDt<=M.birthDt)
begin
 RAISERROR('Child has to be younger, than mother!', 16, 1);
 rollback transaction;
end;
```

OCL – Příklad 2 / Example 2

- Polygamie je zakázaná – žádný muž nemůže být oženěn dvakrát zároveň, tj. intervaly jeho sňatků se nemohou překrývat.
  Pro jednoduchost předpokládejme, že nerozvedené sňatky mají v datu rozvodu uvedeno *+infinity*

  context s1, s2: Sňatek **inv**
  s1.manžel=o2.manžel
  and s1<>s2
  implies
  s1.datRozvodu<s2.datSňatku
  or s2.datRozvodu<s1.datSňatku

- Polygamy is not allowed – no man is allowed to be married twice at the same time, i.e. intervals of his marriages cannot overlap.
  Let suppose for simplicity that not divorced marriages has set the divorce date to *+infinity*

  context m1, m2: Marriage **inv**
  m1.husband=o2.husband
  and m1<>m2
  implies
  m1.divorceDt<m2.wedDt
  or m2.divorceDt<m1.wedDt

SQL – Příklad 2 / Example 2

- Polygamie je zakázaná

```
create trigger check_polygamie
on Sňatek
after insert, update
as
if exists(
   select *
   from inserted as S1
     inner join Sňatek as S2
       on (S1.RČM=S2.RČM or
S1.RČŽ=S2.RČŽ and S1.DatSňatku<>
S2.DatSňatku
      and not (S1.DatRozvodu <
S2.DatSňatku or S2.DatRozvodu <
S1.DatSňatku))
begin
 RAISERROR('Polygamie je zakázána!',
16, 1);
 rollback transaction;
end;
```

- Polygamy is not allowed

```
create trigger check_polygamy
on Marriage
after insert, update
as
if exists(
   select *
   from inserted as M1
     inner join Marriage as M2
       on (M1.SSNM=M2.SSNM or
M1.SSNW=M2.SSNW)
      and M1.WedDt<> M2.WedDt
      and not (M1.DivorceDt <
M2.WedDt or M2.DivorceDt <
M1.WedDt))
begin
 RAISERROR('Polygamy is not
allowed!', 16, 1);
 rollback transaction;
end;
```

SQL – Příklad 3 / Example 3

- Žena nemůže být zároveň mužem

```
create trigger check_zena_muz
on Žena
after insert, update
as
if exists(
   select *
   from inserted as I -- zena
    inner join Muž as M -- muz
      on (M.RČ=I.RČ)
begin
 RAISERROR('Žena už je vedena
jako muž!', 16, 1);
 rollback transaction;
end;
```

- Woman cannot be already a man

```
create trigger check_woman_man
on Woman
after insert, update
as
if exists(
   select *
   from inserted as I -- woman
    inner join Man as M -- man
      on (M.SSN=I.SSN)
begin
 RAISERROR('Woman cannot be
simultaneously a man!', 16, 1);
 rollback transaction;
end;
```