



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

DBS – Příklady – Procedurey a funkce

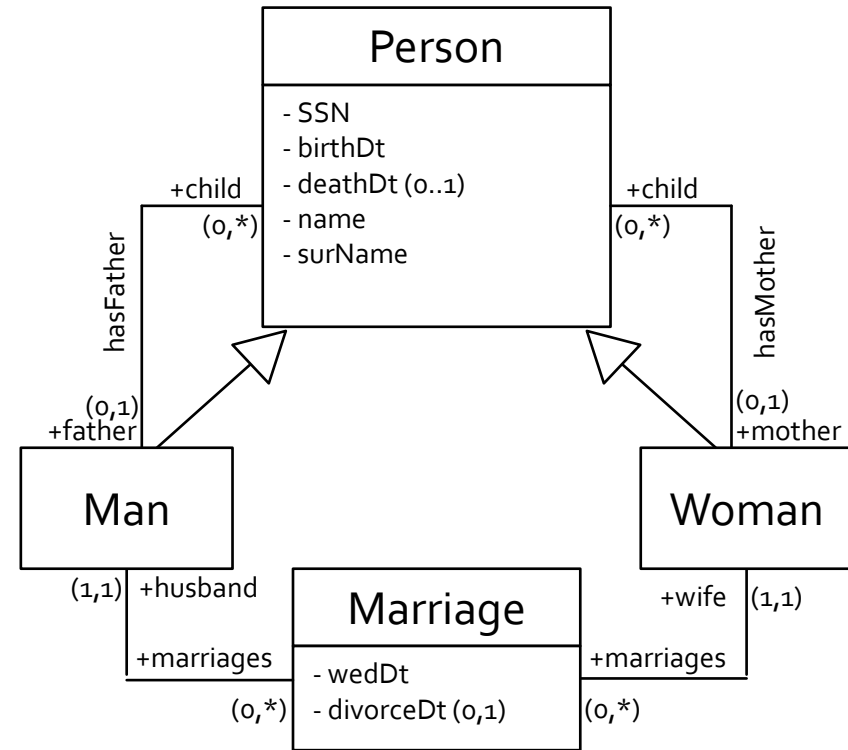
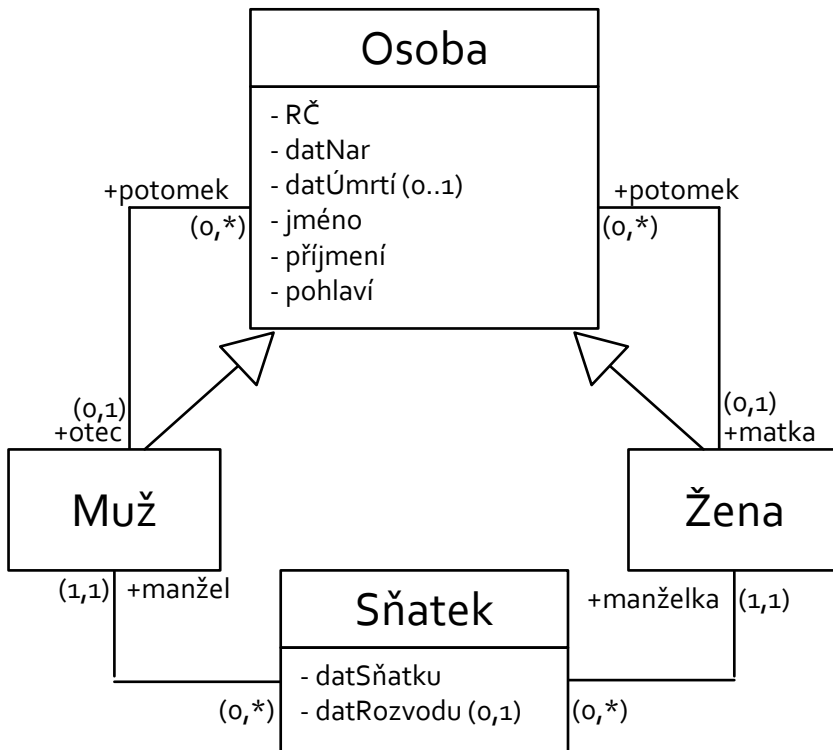
DBS – Examples – Procedures and functions

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)

Proc – Příklad 1 / Example 1

- Vložení nové osoby – muž

```
create procedure Nový_muž
  @rc varchar(11),
  @dn date,
  @du date,
  @jm varchar(30),
  @pr varchar(30)
as
-- jednu řádku do tab. Osoba
insert into Osoba(
  RČ, Jméno, Příjmení,
  DatNar, DatÚmrtí
) values(
  @rc, @dn, @du,
  @jm, @pr
);
-- a jednu řádku do tab. Muž
insert into Muž values(@rc);
```

- Insert of new person – Man

```
create procedure New_Man
  @ssn integer,
  @bdt date, -- birth date
  @ddt date, -- death date
  @name varchar(30),
  @surname varchar(30)
as
-- one row into tab. Person
insert into Person(
  SSN, name, surName,
  birthDt, deathDt
) values(
  @ssn, @name, @surname,
  @bdt, @ddt
);
-- one row into tab. Man
insert into Man values(@ssn);
```

Proc – Příklad 2 / Example 2

- Vložení nové **žijící** osoby

```
create procedure Nová_osoba
  @rc varchar(11),
  @dn date,
  @jm varchar(30),
  @pr varchar(30),
  @pohl char(1) -- M/Ž
as
insert into Osoba(
  RČ, Jméno, Příjmení,
  DatNar, DatÚmrtí
) values(
  @rc, @jm, @pr,
  @dn, null
);
-- a jednu řádku do tab. Muž, resp. Žena
if @pohl='M'
  insert into Muž values(@rc);
else
  insert into Žena values(@rc);
```

- Insert of new **living** person

```
create procedure New_Person
  @ssn integer,
  @bdt date, -- birth date
  @name varchar(30),
  @surname varchar(30),
  @gender char(1) -- M/F
as
insert into Person(
  SSN, birthDt, deathDt,
  name, Surname
) values(
  @ssn, @name, @surname,
  @bdt, null
);
-- one row into tab. Man, resp. Woman
if @gender='M'
  insert into Man values(@ssn);
else
  insert into Woman values(@rc);
```

Proc – Příklad 2 / Example 2

- Vložení nové **žijící** osoby alternativně, pokud mám procedury Nový_muž i Nová_žena

```
create procedure Nová_osoba
  @rc varchar(11),
  @dn date,
  @jm varchar(30),
  @pr varchar(30),
  @pohl char(1) -- M/Ž
as
if @pohl='M'
  exec Nový_muž
    @rc,@dn,null,
    @jm,@pr;
else
  exec Nová_žena
    @rc,@dn,null,
    @jm,@pr;
```

- Insert of new **living** person alternatively, if I have both procedures New_Man and New_Woman

```
create procedure New_Person
  @ssn integer,
  @bdt date, -- birth date
  @name varchar(30),
  @surname varchar(30),
  @gender char(1) -- M/F
as
if @gender='M'
  exec New_Man
    @ssn,@bdt,null,
    @name,@surname;
else
  exec New_Woman
    @ssn,@bdt,null,
    @name,@surname;
```

Proc – Příklad 3 / Example 3

- Vložení nové **žijící** osoby včetně informace o rodičích

```
create procedure Nová_osoba
  @rc varchar(11),
  @dn date,
  @jm varchar(30),
  @pr varchar(30),
  @pohl char(1), -- M/Ž
  @matka varchar(11),
  @otec varchar(11)
as
insert into Person values(...);
if @pohl='M'
  insert into Muž values(...);
else
  insert into Žena values(...);
if @matka is not null
  insert into Matka(RČ,RČŽ)
  values(@rc,@matka);
if @otec is not null
  insert into Otec(RČ,RČM)
  values(@rc,@otec);
```

- Insert of new **living** person including information about parents

```
create procedure New_Person
  @ssn integer,
  @bdt date, -- birth date
  @name varchar(30),
  @surname varchar(30),
  @gender char(1) -- M/F
  @mother integer = null,
  @father integer = null
as
insert into Person values(...);
if @gender='M'
  insert into Man values(...);
else
  insert into Woman values(...);
if @mother is not null
  insert into Mother (SSN,WSSN)
  values(@ssn,@mother);
if @father is not null
  insert into Father (SSN,MSSN)
  values(@ssn,@ father);
```


Proc – Příklad 4 / Example 4

- Vložení nového sňatku se změnou příjmení nevěsty

```
create procedure
Nové_manželství
  @rcm varchar(11),
  @rcz varchar(11),
  @ds date,
  @pr varchar(30)
as
insert into Manželství(
  RČM,RČŽ,DatSňatku
) values(@rcm ,@rcz, ds);
update Osoba set
  Příjmení = @pr
where RČ=@rcz;
```

- Insert of new Wedding including change of bride's surname

```
create procedure New_Marriage
  @mssn integer,
  @wssn integer,
  @wdt date,
  @surname varchar(30)
as
insert into Marriage(
  MSSN, WSSN, WedDt
) values(@mssn, @wssn,
  @wdt);
update Person set
  Surname = @surname
where SSN=@wssn;
```

Funcť – Příklad 5 / Example 5

- Funkce, vracující datum narození dané osoby

```
create function Dat_narození(  
    @rc varchar(11)  
    ) returns date  
as  
declare @ret date;  
select @ret=datNar  
    from Osoba  
    where RČ=@rc;  
return @ret;
```

- Function, returning birth date of given person

```
create function Birth_Dt(  
    @ssn integer  
    ) returns date  
as  
declare @ret date;  
select @ret=birthDt  
    from Person  
    where SSN=@ssn;  
return @ret;
```

Func – Příklad 6 / Example 6

- Funkce, vracející pohlaví dané osoby

```
create function Pohlaví(  
  @rc varchar(11)  
  ) returns char(1) -- M/Ž  
as  
if exists(  
  select * from Muž  
  where RČ=@rc  
  return 'M';  
if exists(  
  select * from Žena  
  where RČ=@rc  
  return 'Ž';  
return null;
```

- Function, returning gender of given person

```
create function Gender(  
  @ssn integer  
  ) returns char(1) -- M/F  
as  
if exists(  
  select * from Man  
  where SSN=@ssn  
  return 'M';  
if exists(  
  select * from Woman  
  where SSN=@ssn  
  return 'F';  
return null;
```

Func – Příklad 7 / Example 7

- Funkce, vracející počet dětí dané osoby

```
create function Poč_Dětí(  
    @rc varchar(11)  
    ) returns integer  
as  
declare  
    @pocM integer, @pocZ integer;  
select @pocM=count(*)  
    from Otec  
    where RČM=@rc);  
select @pocZ=count(*)  
    from Matka  
    where RČŽ=@rc);  
return @pocM+@pocZ;
```

- Function, returning number of children of given person

```
create function Child_Cnt(  
    @ssn integer  
    ) returns integer  
as  
declare  
    @cntM integer, @cntW integer;  
select @cntM=count(*)  
    from Father  
    where SSNM=@ssn);  
select @cntW=count(*)  
    from Mother  
    where SSNW=@ssn);  
return @cntM+@cntW;
```

Proc – Příklad 8 / Example 8

- Smazání muže a všech jeho mužských potomků

```
create procedure  
Smaž_Muže_Rekurzivně  
  @rcm varchar(11)  
as  
if exists(select * from Muž  
  where RČ=@rcm)  
begin  
  declare dite cursor for  
    select RČ from Otec  
    where RČM=@rcm;  
  declare @rcd varchar(11);  
  open dite;  
  fetch next dite into @rcd;  
  while @@FETCH_STATUS=0  
    begin  
      exec Smaž_Muže_Rekurzivně @rcd;  
      fetch next dite into @rcd;  
    end;  
  close dite;  
  deallocate dite;  
  delete from Osoba where RČ=@rcm;  
end;
```

- Deletion of a man and all his male descendants

```
create procedure  
Delete_Man_Recursively  
  @ssnm integer  
as  
if exists(select * from Man  
  where SSN=@ssnm)  
begin  
  declare child cursor for  
    select SSN from Father  
    where SSNM=@ssnm;  
  declare @ssnc integer;  
  open child;  
  fetch next child into @ssnc;  
  while @@FETCH_STATUS=0  
    begin  
      exec Delete_Man_Recursively @ssnc;  
      fetch next child into @ssnc;  
    end;  
  close child;  
  deallocate child;  
  delete from Person where SSN=@ssnm;  
end;
```