A4B33DS – Database systems

Relational database technology

Relational database technology

Key: a set (possibly a singleton) of attributes (columns), that uniquely identifies the given entity.

Remarks:

1. An entity type (a table) may have multiple keys. E.g. (i) social security number (SSN), (ii) personal employee number (PEN), (iii) a synthetized key.
2. Every table has at least one key. A table of a (real) relational database cannot contain multiple identical rows.

PEN	SSN	Name	Surname	Date of birth
101	8811010033	Josef	Novák	01/11/88
	8811010033			
101				

Name	Surname	Date of birth
Josef	Novák	01/11/88
Josef	Novák	01/11/88

Relational database technology

Foreign key: a set (possibly a singleton) of attributes (columns), the tuple of their values is a value of the key of another (foreign) table.

Remarks:

1. Data type of each of the foreign key member attribute (column) has to be compatible with the data type of the corresponding primary key attribute of the referenced table.
2. The foreign key is not (need to be) a key of the table where it is defined.
3. Relationship between two entities implemented by the associattion foreign key -> **primary** key.
4. A table can have multiple keys => good practice is to choose one of them and use it systematically to implement relationship. **primary key**,

PEN	SSN	Name	Surname	Date of birth
101	8811010033	Josef	Novák	01/11/88

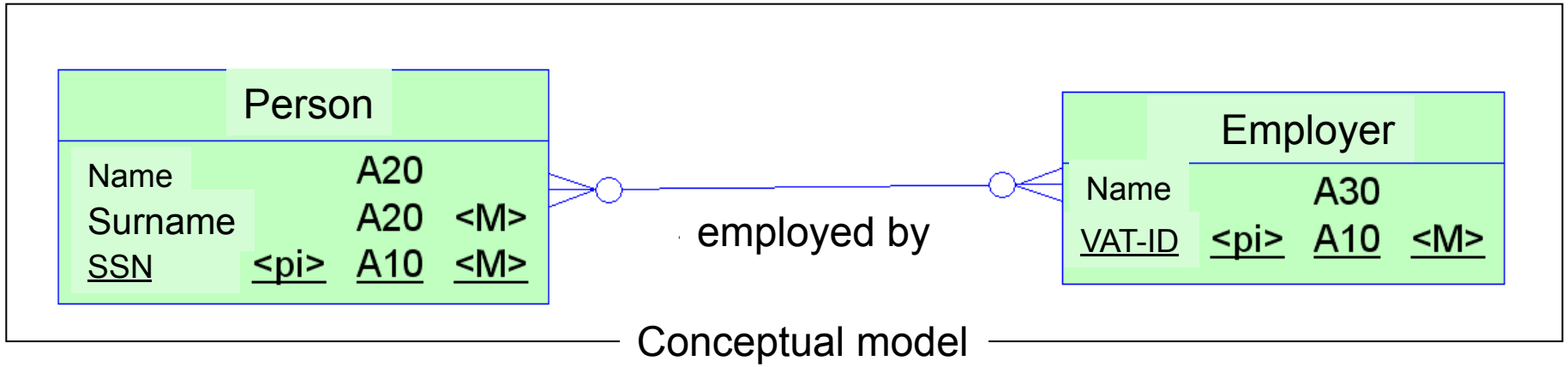
PEN	Description	Calibre
101	Kalašnikov	7,65

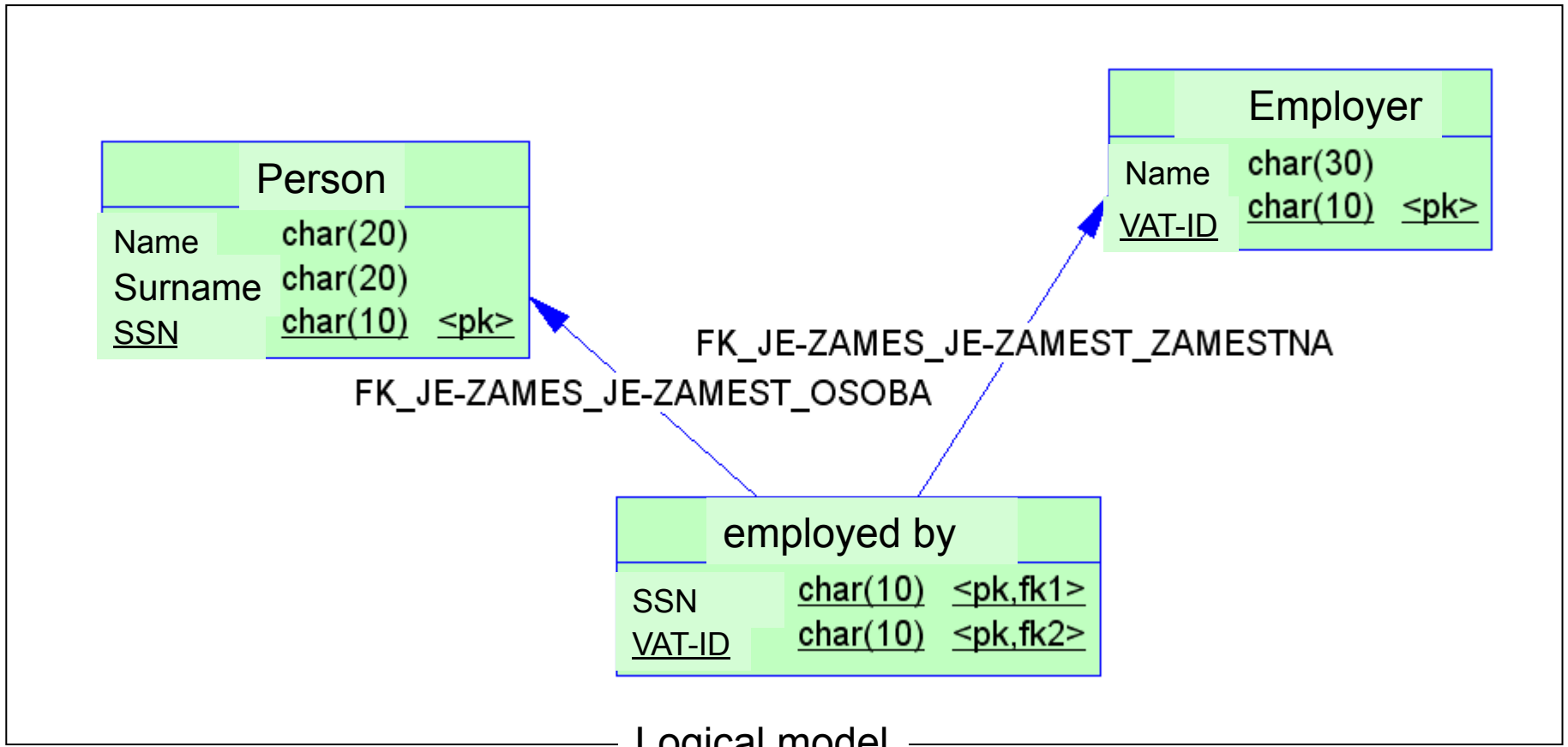
SSN	License table	Brand
8811010033	BE-04-30	Škoda

http://www.sybase.com/products/modelingmetadata/powerdesigner

Conversion conceptual model -> logical model

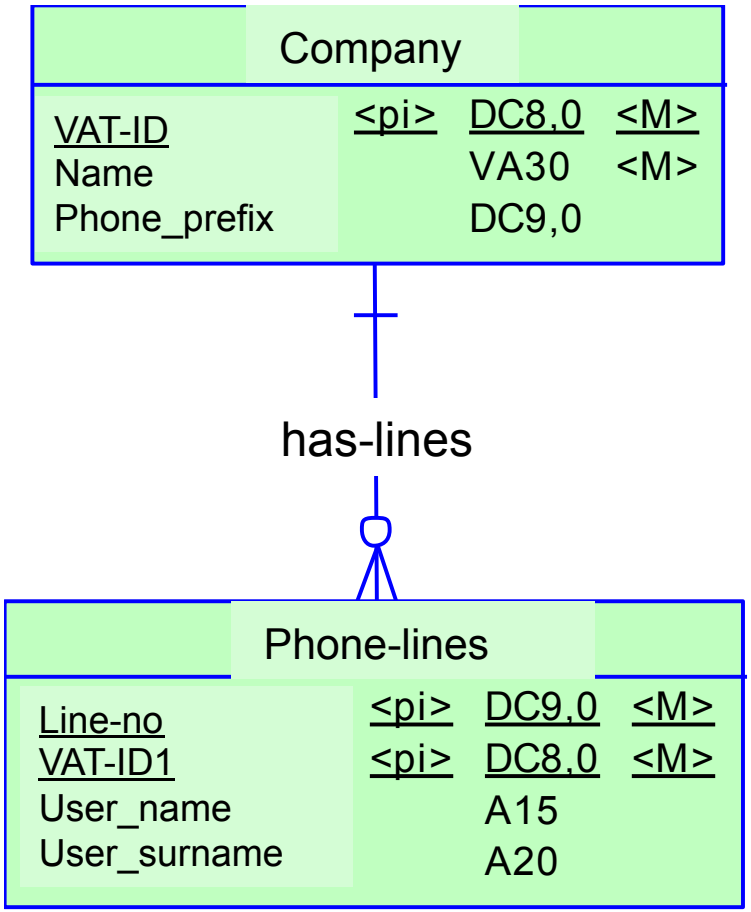
- Entity type -> a table
- Entity type attribute -> column (of a table)
- Relationship:
  - 1:1 or 1:N:
    - relationship -> foreign key on the side N of the relationship
    - attribute of a relationship -> column of the table on the side N of the relationship
  - N:M:
    - Relationship -> „reference“ table
    - Attribute of a relationship -> column of the reference table


Person
Name A20
Surname A20 <M>
SSN <pi> A10 <M>
employed by
Employer
Name A30
VAT-ID <pi> A10 <M>
Conceptual model


Person
Name char(20)
Surname char(20)
SSN char(10) <pk>
Employer
Name char(30)
VAT-ID char(10) <pk>
FK_JE-ZAMES_JE-ZAMEST_OSOBA
FK_JE-ZAMES_JE-ZAMEST_ZAMESTNA
employed by
SSN char(10) <pk,fk1>
VAT-ID char(10) <pk,fk2>
Logical model

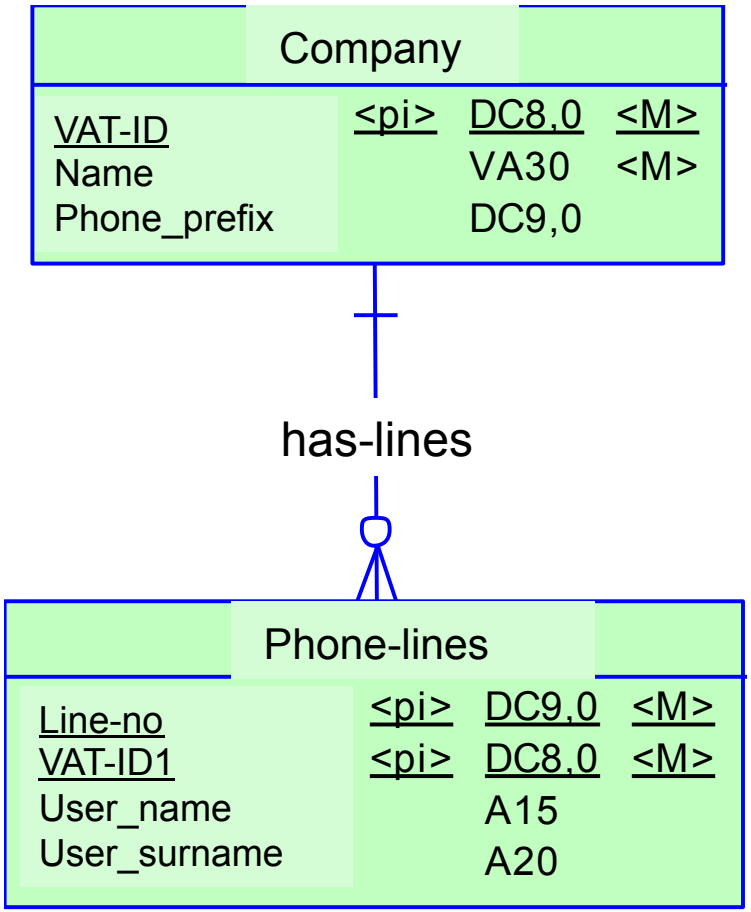
Weak entity type

Crow's Foot Notation



Weak entity type

Crow's Foot Notation



11111111	Nemocnice Motol	224 43
22222222	Honda Motol	234 09

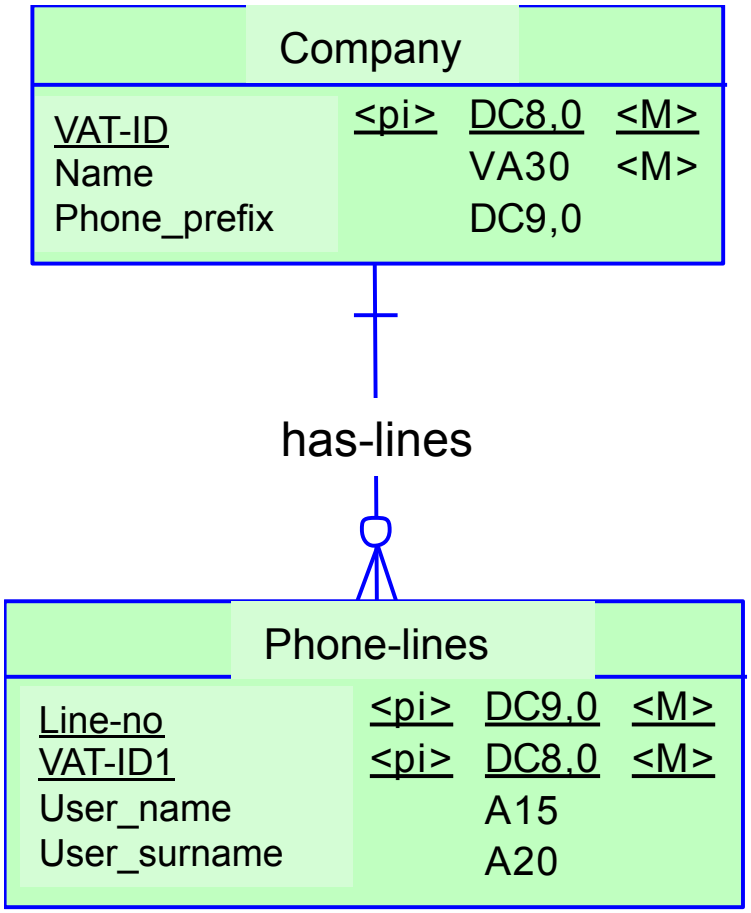
11111111	1111	Ústředna
11111111	2920	Kardiocentrum
11111111	2101	Novorozenecké oddělení
22222222	1111	Recepce
22222222	6690	Prodej nových vozů

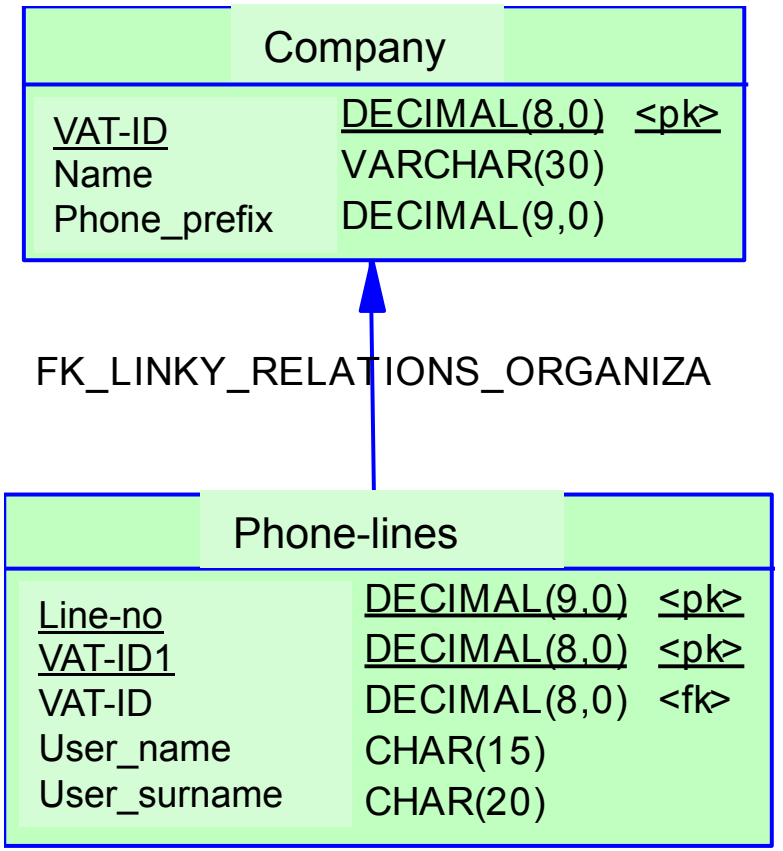
Cizí klíč

Primární klíč

Weak entity type

Crow's Foot Notation





11111111	Nemocnice Motol	224 43
22222222	Honda Motol	234 09

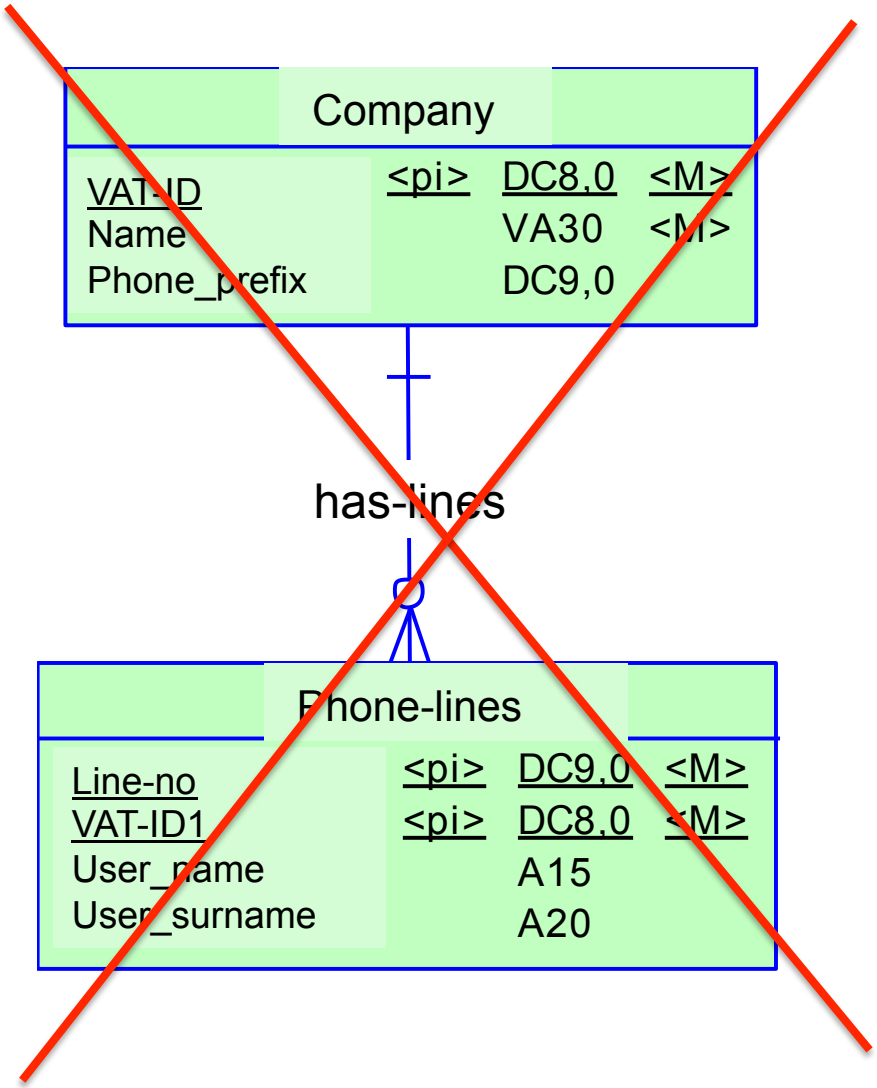
11111111	1111	Ústředna
11111111	2920	Kardiocentrum
11111111	2101	Novorozenecké oddělení
22222222	1111	Recepce
22222222	6690	Prodej nových vozů

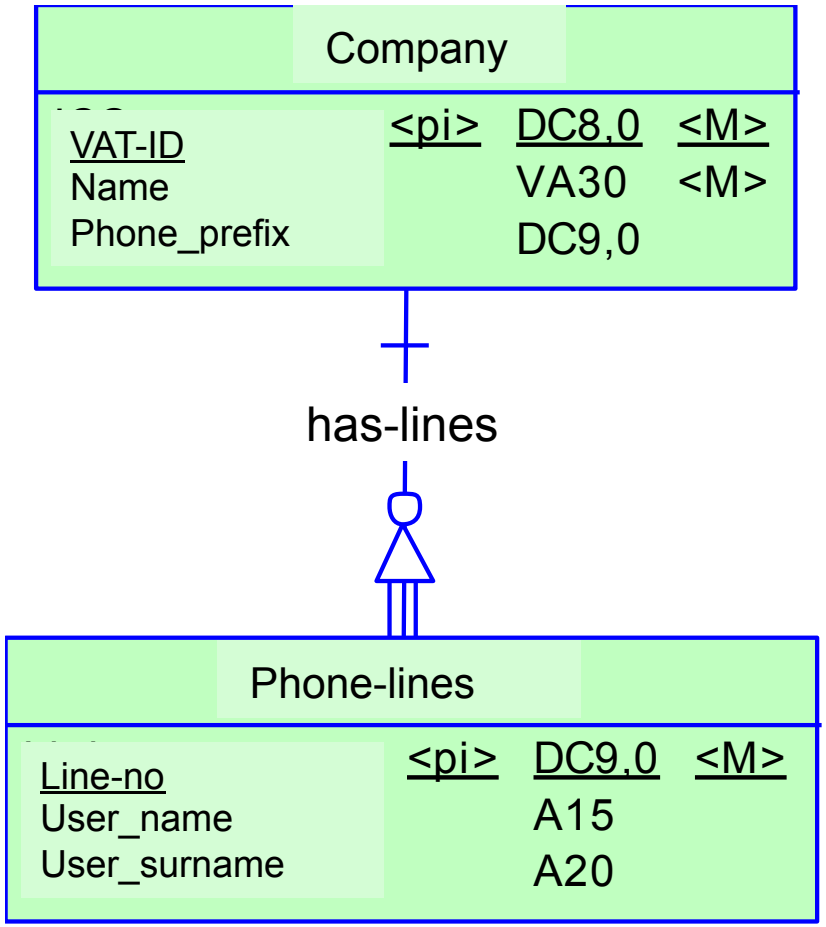
Cizí klíč

Primární klíč

Weak entity type

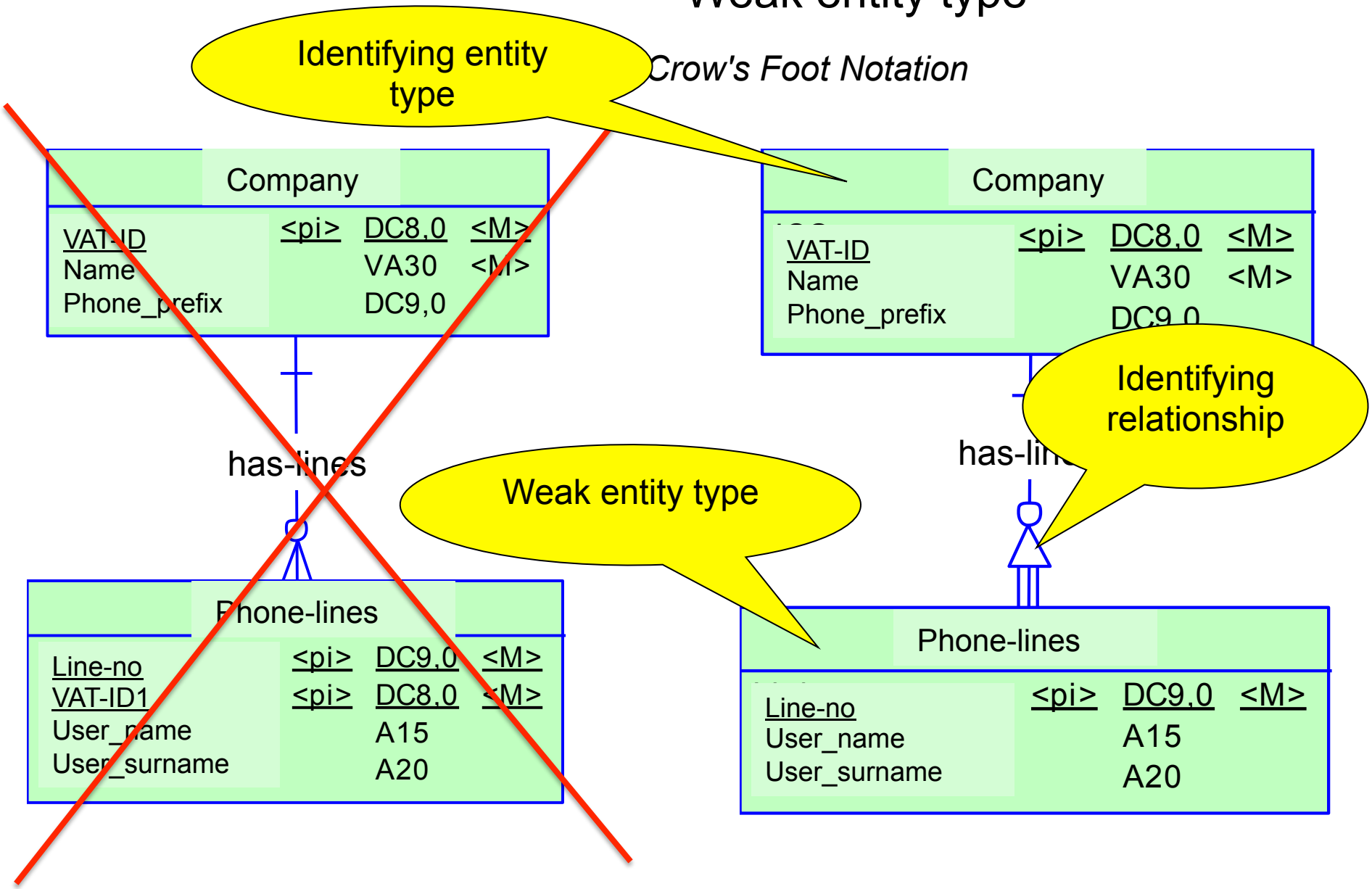
Crow's Foot Notation





Weak entity type

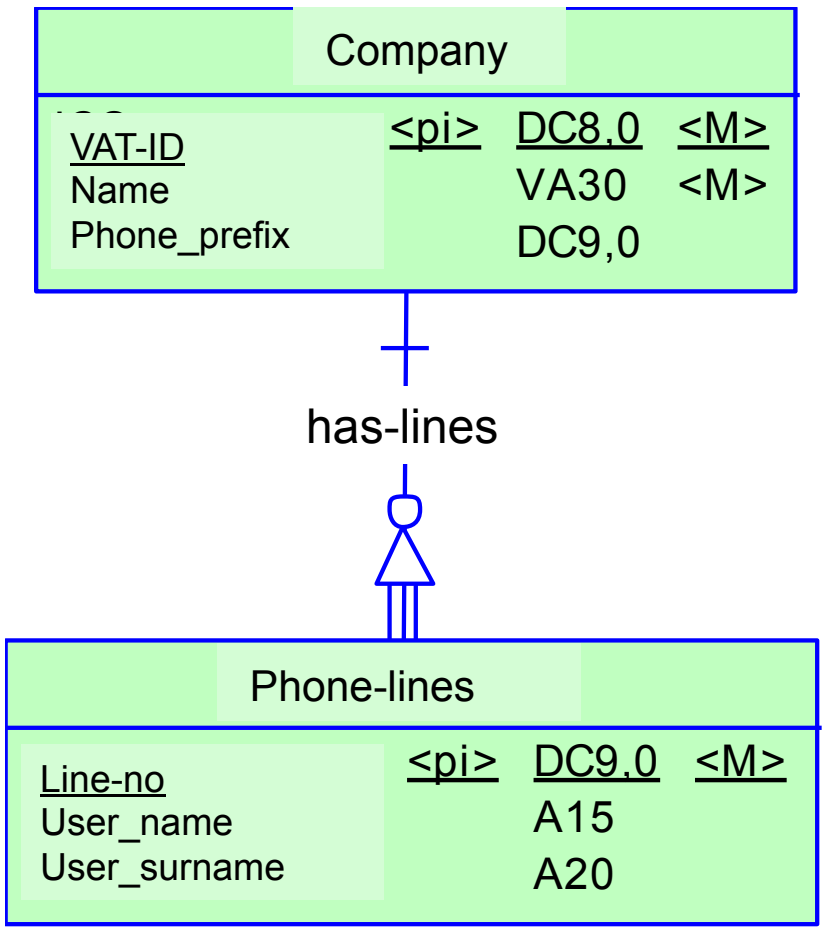
Crow's Foot Notation



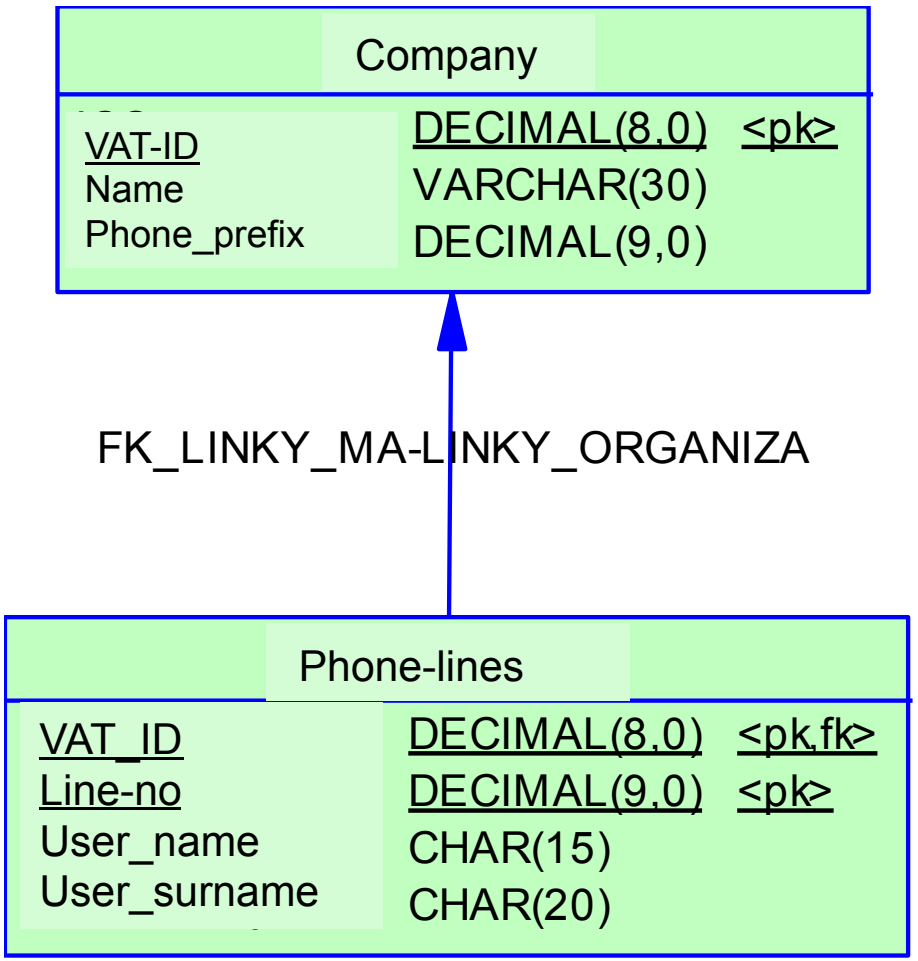
Weak entity type

Crow's Foot Notation

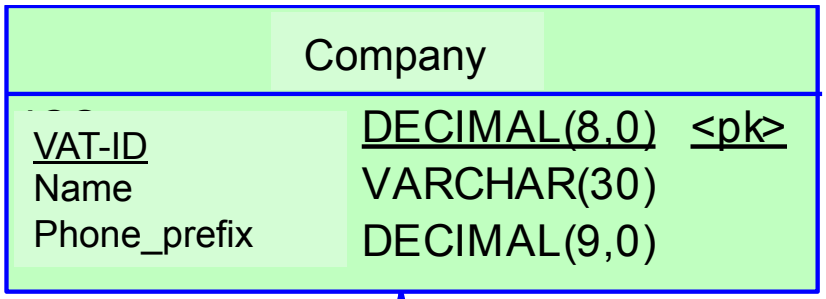
Conceptual model:

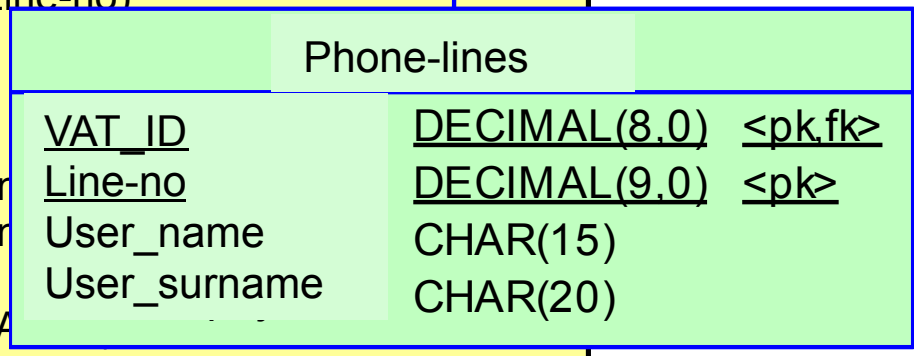


Logical model:



Physical model





FK_LINKY_MA-LINKY_ORGANIZA

```
create table Phone-lines (
        VAT-ID                  DECIMAL(8,0)        not null,
        Line-no                 DECIMAL(9,0)        not null,
        User_name               CHAR(15)            null,
        User_surname            CHAR(20)            null,
        constraint PK_LINKY primary key (VAT_ID, Line-no)
);

create table Company (
        VAT-ID                  DECIMAL(8,0)        not r
        Name                    VARCHAR(30)         not r
        Phone_prefix            DECIMAL(9,0)        null,
        constraint PK_ORGANIZACE primary key (VA
);

alter table Phone-lines
add constraint "FK_LINKY_MA-LINKY_ORGANIZA" foreign key (VAT-ID)
      references Company (VAT-ID)
      on delete restrict on update restrict;
```

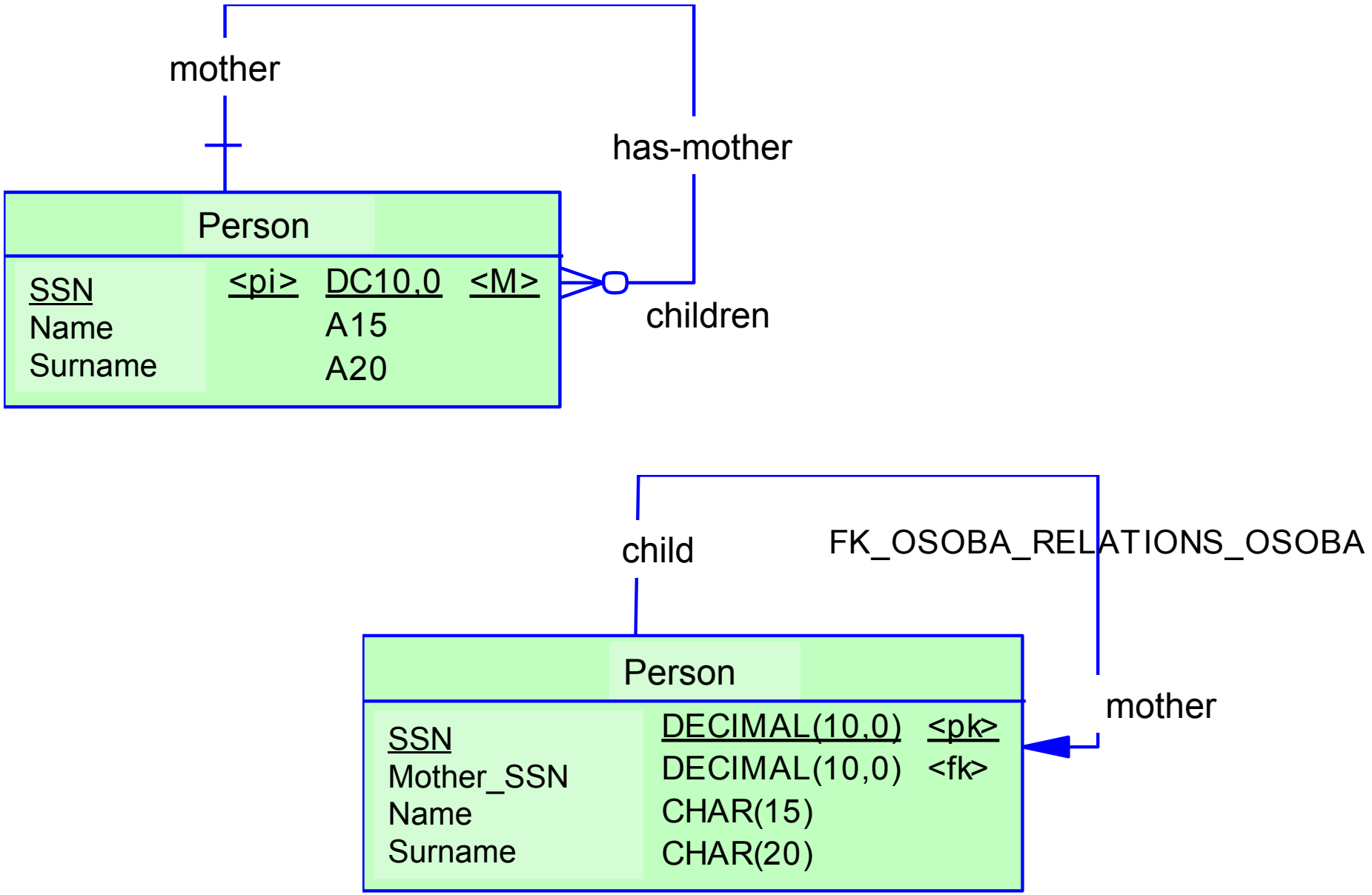
Physical model

```
create table Phone-lines (
        VAT-ID                  DECIMAL(8,0)        not null,
        Line-no                 DECIMAL(9,0)        not null,
        User_name               CHAR(15)            null,
        User_surname            CHAR(20)            null,
        constraint PK_LINKY primary key (VAT_ID, Line-no)
);

create table Company (
        VAT-ID                  DECIMAL(8,0)        not null,
        Name                    VARCHAR(30)         not null,
        Phone_prefix            DECIMAL(9,0)        null,
        constraint PK_ORGANIZACE primary key (VAT-ID)
);

alter table Phone-lines
add constraint "FK_LINKY_MA-LINKY_ORGANIZA" foreign key (VAT-ID)
      references Company (VAT-ID)
      on delete restrict on update restrict;
```

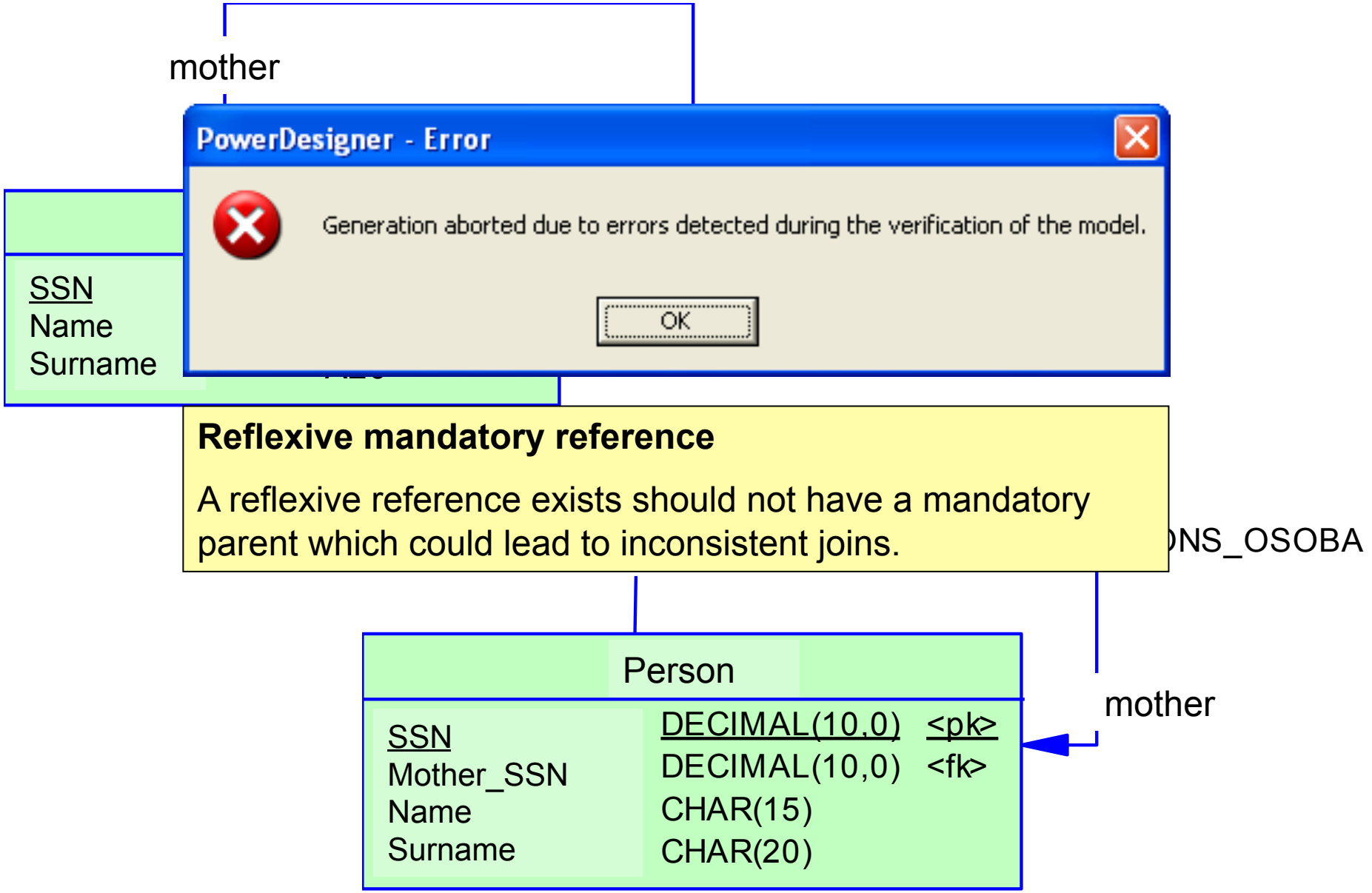
Physical model

```
create table Linky (
        ICO                     DECIMAL(8,0)        not null,
        Linka                   DECIMAL(9,0)        not null,
        Uzivatel-jmeno          CHAR(15)            null,
        Uzivatel-prijmeni       CHAR(20)            null,
        constraint PK_LINKY primary key (ICO, Linka)
);

create table Organizace (
        ICO                     DECIMAL(8,0)        not null,
        Nazev                   VARCHAR(30)         not null,
        Telefon_prefix          DECIMAL(9,0)        null,
        constraint PK_ORGANIZACE primary key (ICO)
);

alter table Linky
add constraint "FK_LINKY_MA-LINKY_ORGANIZA" foreign key (ICO)
      references Organizace (ICO)
      on delete restrict on update restrict;
```

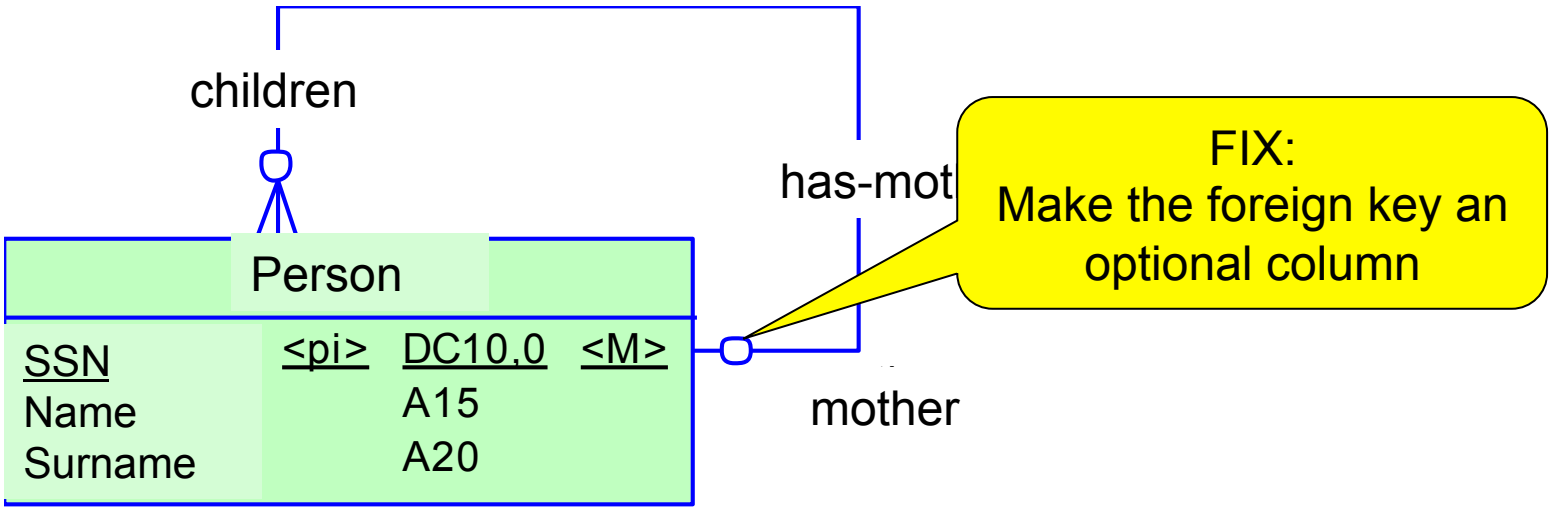
Reflexive (recursive) relationship

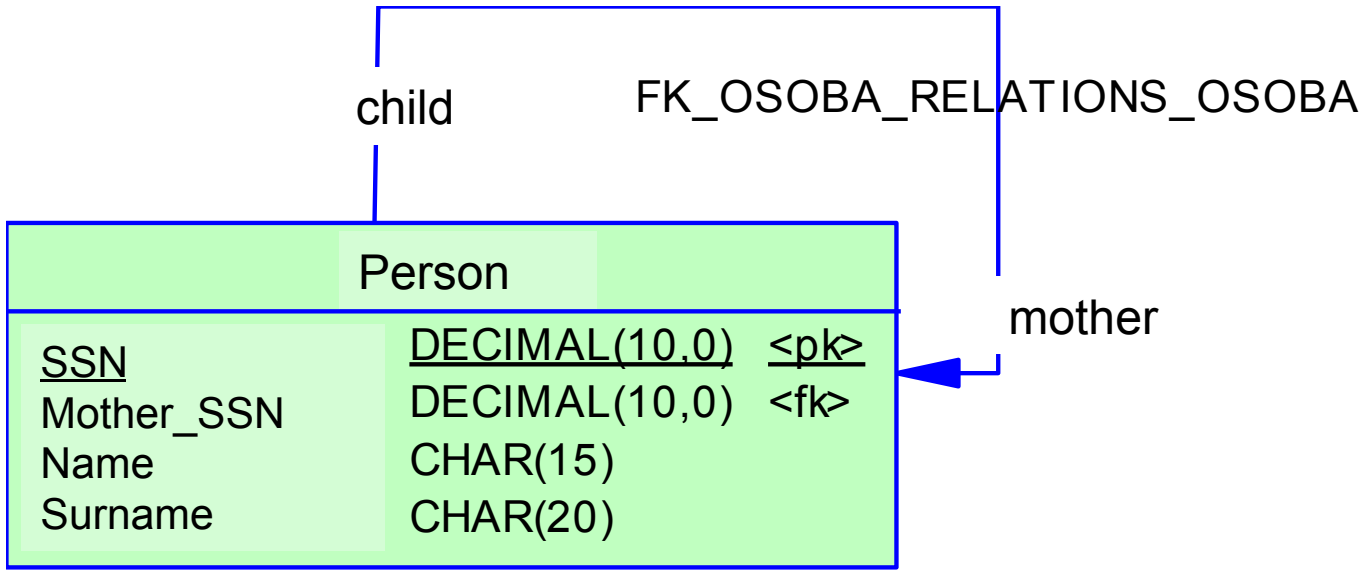


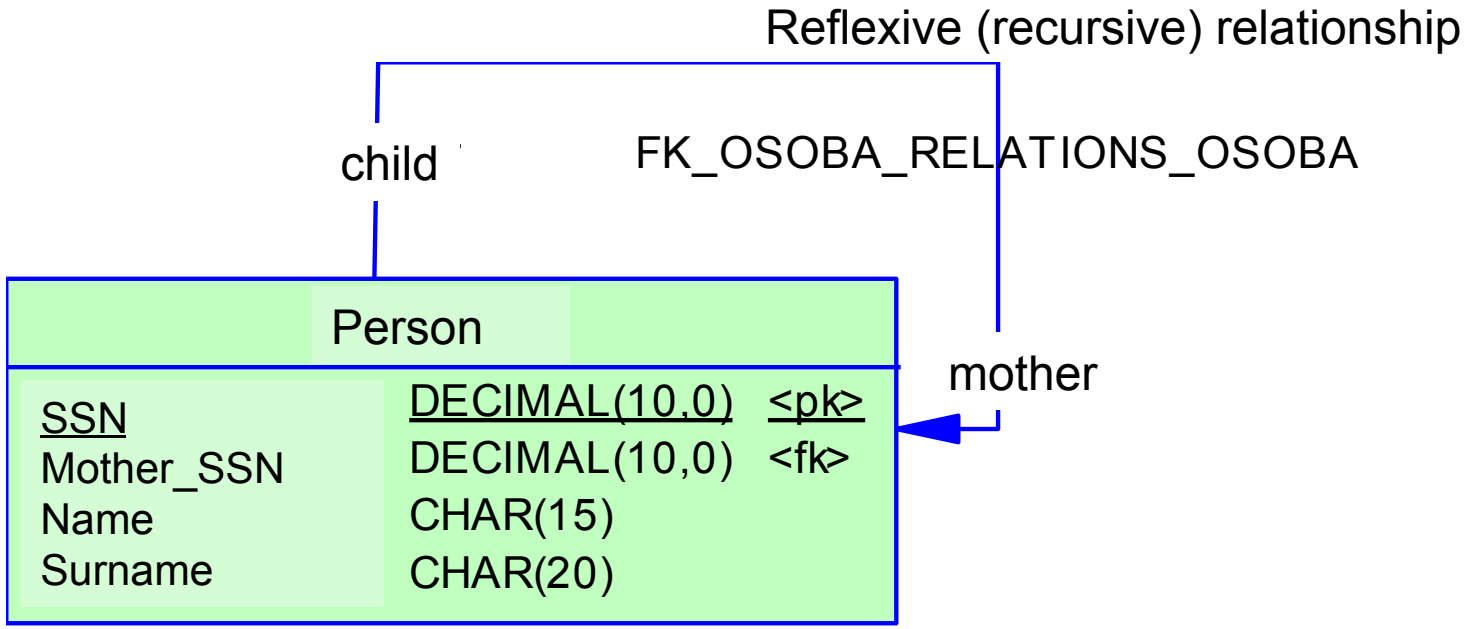
Reflexive (recursive) relationship



Reflexive (recursive) relationship







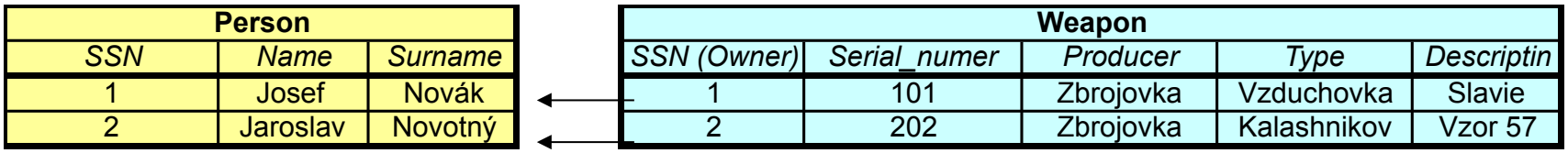

```
create table Person (
SSN                  DECIMAL(10,0)   not null,
Mother_SSN           DECIMAL(10,0)   null,
Name                 CHAR(15)        null,
Surname              CHAR(20)        null,
constraint PK_PERSON primary key (SSN)
);

alter table Person
   add constraint FK_OSOBA_RELATIONS_OSOBA foreign key (Mother_SSN)
      references Person (SSN)
      on delete restrict on update restrict;
```

Referential integrity 1

Let us have 2 tables A a B such that table B contains a foreign key referencing table A.

It must not happen that there is a row in table B that refrers to a non-existing row of table A.

Referential integrity



Person		
SSN	*Name*	*Surname*
1	Josef	Novák
2	Jaroslav	Novotný

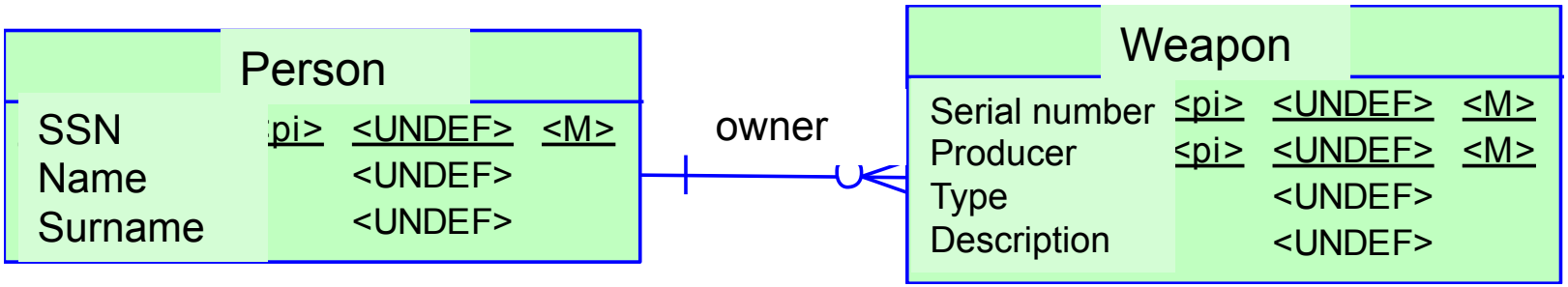
Weapon				
SSN (Owner)	*Serial_numer*	*Producer*	*Type*	*Descriptin*
1	101	Zbrojovka	Vzduchovka	Slavie
2	202	Zbrojovka	Kalashnikov	Vzor 57

How referential integrity can be violated?

a) We delete a record in table Perosn => foreign key of the respective row in Weapon table will point to a non-existing record in table Person.

a) We modify the value of the primary key of a record in table Person => foreign key of the respective row in Weapon table will point to a non-existing record in table Person.

Referential integrity 3




```
create table Weapon (
Serial_number        CHAR(10)     not null,
Producer             CHAR(10)     not null,
SSN                  CHAR(10)     not null,
Type                 CHAR(10)     null,
Description          CHAR(10)     null,
constraint PK_WEAPON primary key (Serial_number, Producer)
constraint FK_WEAPON_OWNER_PERSON foreign key (SSN)
     references Person (SSN)
     on delete restrict on update restrict;
);
```

Referential integrity 4