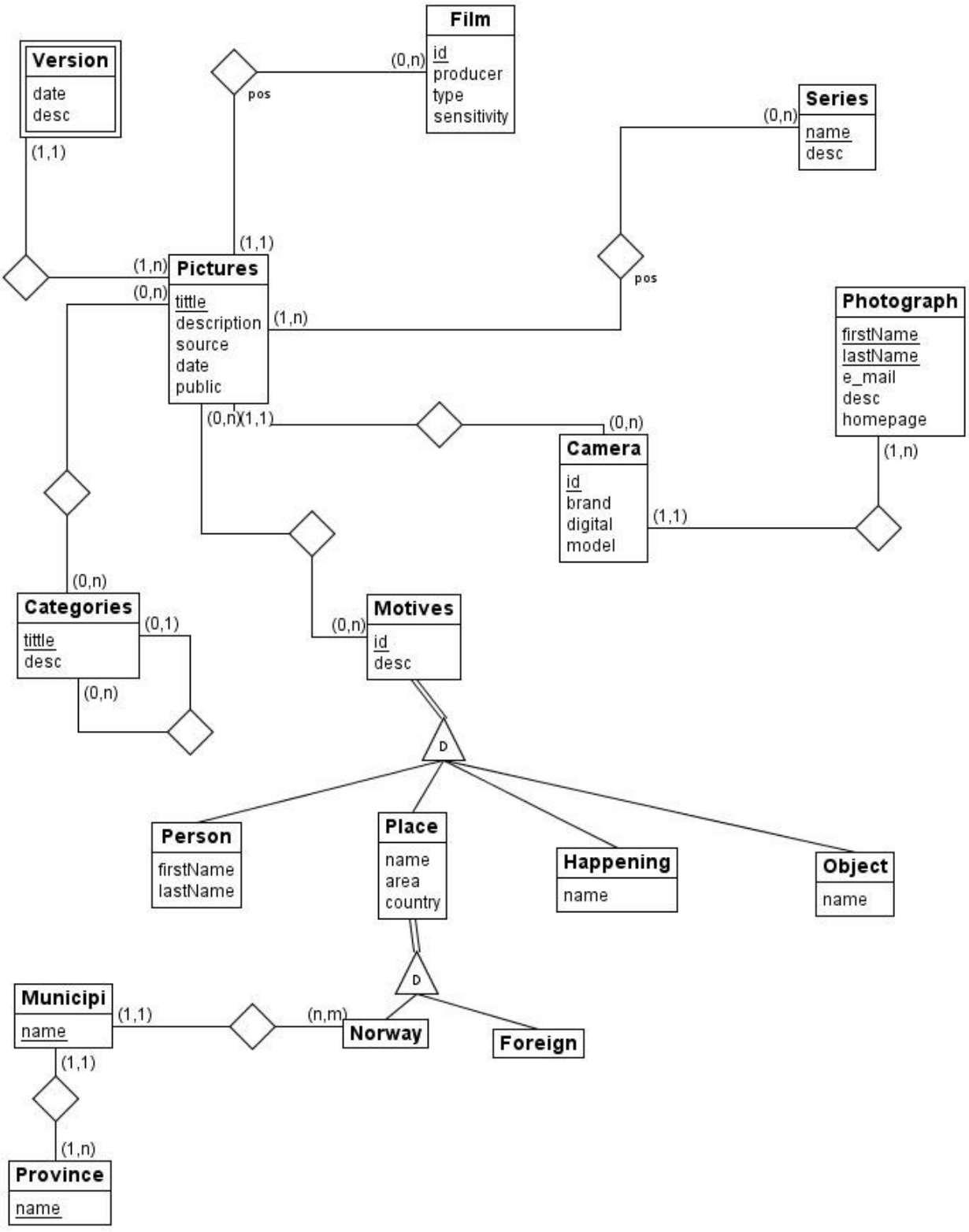
TDT4145: Data Modelling and Database Systems

Project: “Picture database”

[Exercise 2: “Logical database scheme”]

Group 17:
David Rozas Domingo
Miguel Bono Tur

ER Diagram


Film
id
producer
type
sensitivity
Version
date
desc
pos
(0,n)
(1,1)
Series
name
desc
(0,n)
(1,1)
(1,n)
Pictures
tittle
description
source
date
public
(0,n)
(1,n)
pos
Photograph
firstName
lastName
e_mail
desc
homepage
(1,n)
(0,n)
(1,1)
Camera
id
brand
digital
model
(1,1)
(0,n)
Categories
tittle
desc
(0,1)
(0,n)
Motives
id
desc
(0,n)
D
Person
firstName
lastName
Place
name
area
country
Happening
name
Object
name
D
Municipi
name
(1,1)
(n,m)
Norway
Foreign
(1,1)
(1,n)
Province
name

Compilable SQL-script and some comments

The SQL script has been created using MySQL. The version is mysql Ver 14.12 Distrib 5.0.45, for pc-linux-gnu (i486).

Up to now (12th March 2008) we have not received any feedback about the previous exercise, so we have decided to implement it using the ER Diagram which was delivered the last week.

In order to check quickly which is compilable, a digital version of the script can be downloaded in: http://folk.ntnu.no/davidro/sql/create_table_executable.sql

```
--Entity photographer
CREATE TABLE Photographer (
firstName CHAR(20),
lastName CHAR(20),
mail CHAR(20),
description TEXT,
homepage CHAR(30),
PRIMARY KEY (firstName,lastName));

--Entity Camera
CREATE TABLE Camera (
id INTEGER PRIMARY KEY,
brand CHAR(20),
model CHAR(20),
digital BOOL,
firstName CHAR(20),
lastName CHAR(20),
FOREIGN KEY (firstName,lastName) REFERENCES Photographer
        ON DELETE SET NULL
        ON UPDATE CASCADE);
--If a photographer is deleted, the camera will take a null (photographer unknown)

--Entity film
CREATE TABLE Film (
id INTEGER PRIMARY KEY,
producer CHAR(20),
type CHAR(20),
sensitivity CHAR(20));

--Entity serie
CREATE TABLE Serie (
name CHAR(20) PRIMARY KEY,
description TEXT);


--Entity Picture, including constraint for the source
CREATE TABLE Picture(
```

```
title CHAR(20) PRIMARY KEY,
description TEXT,
pdate DATE NOT NULL,
is_public BOOL,
camera INTEGER,
source CHAR(20),
FOREIGN KEY (camera) REFERENCES Camera
        ON DELETE SET NULL
        ON UPDATE CASCADE,
CHECK (source IN ('Digital camera', 'Film scanner', 'Flatbed scanner', 'Drum
scanner')));
--If a camera is deleted, the pictures will take a null (camera unknown)

--Version (weak entity)
CREATE TABLE Version (
storingDate DATE,
description TEXT,
idPicture CHAR(20),
FOREIGN KEY (idPicture) REFERENCES Picture
        ON DELETE CASCADE
        ON UPDATE CASCADE,
PRIMARY KEY (idPicture, storingDate));

--Relationship between film and picture
CREATE TABLE PictureInFilm (
idPicture CHAR(20) PRIMARY KEY,
idFilm INTEGER,
pos INTEGER,
FOREIGN KEY (idPicture) REFERENCES Picture
        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY (idFilm) REFERENCES Film
        ON DELETE CASCADE
        ON UPDATE CASCADE);

--Relationship between serie and picture
CREATE TABLE PictureInSerie (
idSerie CHAR(20),
idPicture CHAR(20),
pos INTEGER,
FOREIGN KEY (idPicture) REFERENCES Picture
        ON DELETE CASCADE
        ON UPDATE CASCADE,
FOREIGN KEY (idSerie) REFERENCES Serie
        ON DELETE CASCADE
        ON UPDATE CASCADE,
PRIMARY KEY (idPicture, idSerie));


--Entity categories
CREATE TABLE Category (
title CHAR(20) PRIMARY KEY,
description TEXT NOT NULL,
is_child_of CHAR (20),
```

```
FOREIGN KEY (is_child_of) REFERENCES Category);

--Motive
CREATE TABLE Motive (
id INTEGER PRIMARY KEY,
description TEXT NOT NULL);

--Person
CREATE TABLE Person (
firstName CHAR(20),
lastName CHAR(20),
id_motive INTEGER PRIMARY KEY,
FOREIGN KEY (id_motive) REFERENCES Motive
        ON DELETE CASCADE
        ON UPDATE CASCADE);

--Happening
CREATE TABLE Happening (
name CHAR(20),
id_motive INTEGER PRIMARY KEY,
FOREIGN KEY (id_motive) REFERENCES Motive
        ON DELETE CASCADE
        ON UPDATE CASCADE);

--Object
CREATE TABLE Object (
name CHAR(20),
id_motive INTEGER PRIMARY KEY,
FOREIGN KEY (id_motive) REFERENCES Motive
        ON DELETE CASCADE
        ON UPDATE CASCADE);


--Place
CREATE TABLE Place (
name CHAR(20),
id_motive INTEGER PRIMARY KEY,
area CHAR(20),
country CHAR(20),
FOREIGN KEY (id_motive) REFERENCES Motive
        ON DELETE CASCADE
        ON UPDATE CASCADE);

--Norwegian place
CREATE TABLE Norway (
id_motive INTEGER PRIMARY KEY,
FOREIGN KEY (id_motive) REFERENCES Place
        ON DELETE CASCADE
        ON UPDATE CASCADE);

--Non Norwegian place
CREATE TABLE ForeignCountry (
id_motive INTEGER PRIMARY KEY,
FOREIGN KEY (id_motive) REFERENCES Place
```

```
        ON DELETE CASCADE
        ON UPDATE CASCADE);

--Province
CREATE TABLE Province (
name CHAR(20) PRIMARY KEY);

--Municipi
CREATE TABLE Municipi (
name CHAR(20) PRIMARY KEY,
id_motive INTEGER,
province CHAR(20),
FOREIGN KEY (id_motive) REFERENCES Norway
        ON DELETE SET NULL
        ON UPDATE CASCADE,
FOREIGN KEY (province) REFERENCES Province
        ON DELETE SET NULL
        ON UPDATE CASCADE);
--On delete null on Norway, because if we delete a motive, we do not have to delete
the municipi
--On delete null on Province, taking into account that null will be "unknown"
```