**Lab 2**

**Group 1**

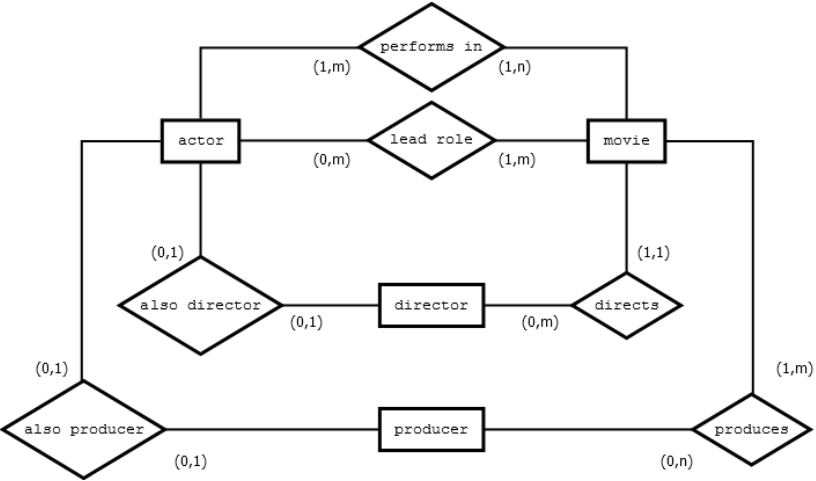
**Lab team:**

1. **Ievgenii Nudga**
2. **Aryan Sultan**

**Assignment 3: Movies Database**

A MOVIES database confirms to the constraints shown in the following ERM diagram (Actor is used

as a generic term and includes actress.):



**Assignment 5: Transform the ERM to a Relation Schema**

Please refer to the movie database of assignment 3.

Transform the ER-model of assignment 3 into a relational schema.

a) What are the relations?

b) Assuming artificial primary keys („id“) for every relation, add foreign keys needed to represent the relationship types. Mark the foreign keys with (FK).

c) Think about possible attributes for the entity types. Discuss some candidate keys.

The report shall contain the relation schema, including the attributes, the primary (PK), and foreign keys (FK). Describe the candidate keys in textual form. Also explain your design decisions for 'directs' and 'also director' by emphasizing the advantage of your solution in comparison to other options.

**Assignment 6 : Movie Database**

Please refer to the movie database of assignment 3.

A) Create SQL-DDL statements to implement your relation schema. Take special care on the constraints (PK, FK, NOT NULL, UNIQUE) and add constraints where appropriate! The script should be fully executable (no exceptions occur, the whole movie db is set up afterwards).

The report shall contain the SQL script. Please store the statements as a script (just below

each other) in the report, so that I can copy & paste it for testing purpose.

B) Execute your SQL script in Oracle.

The report shall contain the output.

C) Create SQL statements to drop the tables and execute the script.

The report shall contain the SQL script and the output.

**Answers for Assignment 6:**

**A) The SQL script for movie database is as follows:**

CREATE TABLE Actor(A\_id int CONSTRAINT a\_pk PRIMARY KEY,

A\_name varchar(20) not null,

DateofBirth char(10) not null);

CREATE TABLE Director (D\_id int CONSTRAINT d\_pk PRIMARY KEY,

a\_id\_fk int constraint fk\_2\_a references Actor not null unique,

D\_name varchar (20) not null,

Oscar\_won int,

Company varchar (50));

CREATE TABLE Producer (P\_id int CONSTRAINT prod\_pk PRIMARY KEY,

prod\_name varchar (20) not null,

Company varchar (50),

a\_id\_fk int constraint fk\_p\_2\_a references Actor not null unique);

CREATE TABLE Movie(M\_id int CONSTRAINT m\_pk PRIMARY KEY,

D\_id\_fk int constraint fk\_2\_d references Director not null unique,

title varchar(20) not null unique,

movie\_year char(4) not null,

budget numeric(20,2),

country\_of\_production varchar(20) not null,

movie\_genre varchar(20) not null);

CREATE TABLE performs\_in(a\_id\_fk int constraint fk\_perf\_2\_a references Actor not null,

M\_id\_fk int constraint fk\_perf\_2\_m references movie not null);

CREATE TABLE lead\_Role(a\_id\_fk int constraint fk\_ld\_2\_a references Actor not null,

M\_id\_fk int constraint fk\_ld\_2\_m references movie not null);

CREATE TABLE Produces(p\_id\_fk int constraint fk\_produces\_2\_producer references producer not null,

M\_id\_fk int constraint fk\_produces\_2\_m references movie not null);

**B) The output after creating the table:**



**C) SQL script for deleting the tables:**

drop table Produces;

drop table lead\_Role;

drop table performs\_in;

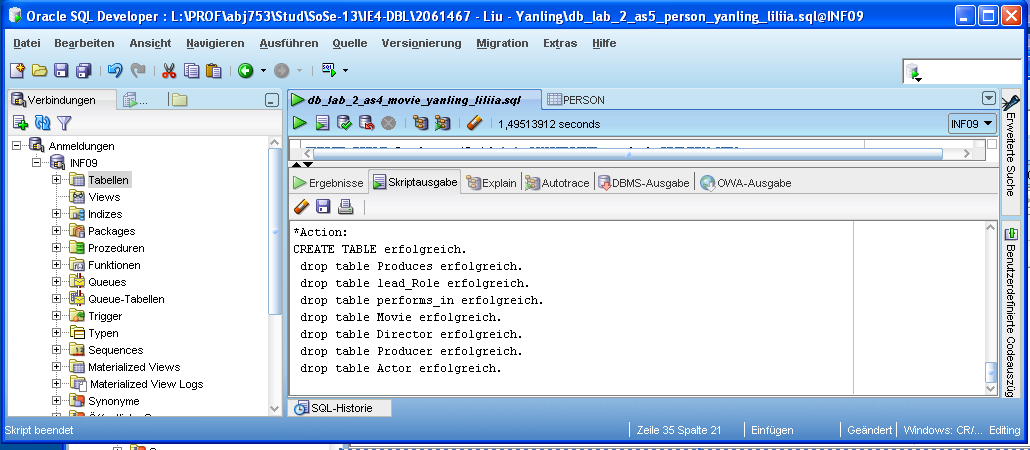
drop table Movie;

drop table Director;

drop table Producer;

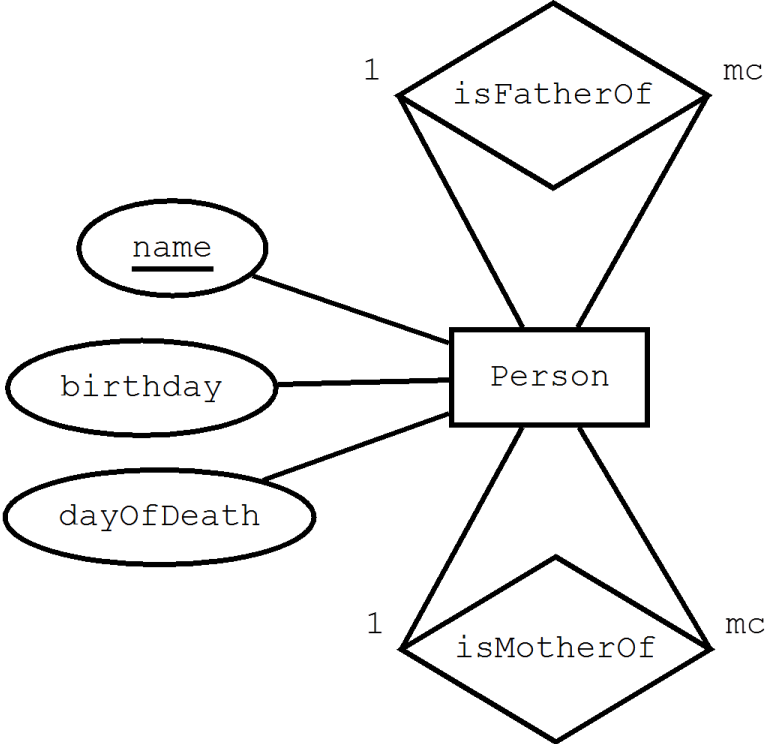
drop table Actor;

**The output after deleting the table:**



**Assignment 7 : Recursive Relationship**

Consider the following ER diagram:



**(A)** Convert the ERM to a relation schema!

The report shall contain the relation schema, including PKs and FKs.

**(B)** Create SQL-DDL statements (Oracle syntax) to implement your relation schema.

Execute your script in Oracle.

The report shall contain

– the script

– and the output from Oracle (copy from the result pane).

**(C)** For every column, give a justification why it is nullable or not.

**(D)** Insert the following members of one family into your database:

Children: Jane (b. 1.1.2006), Joe (2.2.2007)

Parents: Bob (b. 11.11.1965), Susan (b. 8.8.1966)

Grandparents: Rich (b. 19.4.1935), Sue (b. 13.5.1940, died 28.5.2002)

Walt (b. 10.3.1930, died 20.12.2007), Ellen (b. 9.9.1930)

In which order do you have to insert the data?

The report shall contain the script for inserting the family and a screenshot of the data in

the data editor.

**Answers for assignment 7:**

**A) Relational Schema:**

Person (name, birthday, dayOfDeath, Father(FK), Mother(FK)).

In this relational schema there is one primary key (PK), namely name. And we have two foreign keys (FK), for father and mother respectively as they belong to the same Person entity.

**B) Oracle Script and output:**

**DROP TABLE Person;**

**CREATE TABLE Person(**

**Name VARCHAR2(255) NOT NULL CONSTRAINT pk\_name PRIMARY KEY,**

**BD DATE NOT NULL,**

**DD DATE NULL,**

**Father\_of VARCHAR2(255) NULL,**

**Mother\_of VARCHAR2(255) NULL,**

**FOREIGN KEY (Father\_of) REFERENCES Person(Name),**

**FOREIGN KEY (Mother\_of) REFERENCES Person(Name)**

**)**

drop table person;

create table Person(P\_name varchar(20) primary key,

birthday date not null,

dayOfDeath date,

Father varchar(20) constraint father\_2\_person references Person,

Mother varchar(20) constraint mother\_2\_person references Person); **--creates table person**

insert into Person values('Walt', TO\_DATE('10-apr-1930', 'DD-MON-YYYY'),

TO\_DATE('20-dez-2007', 'DD-MON-YYYY'),

null,null); **--insert values in table person**

insert into Person values('Ellen', TO\_DATE('9-sep-1930', 'DD-MON-YYYY'),

null,null,null); **--insert values in table person**

insert into Person values('Rich', TO\_DATE('19-apr-1935', 'DD-MON-YYYY'),

null,null,null); **--insert values in table person**

insert into Person values('Sue', TO\_DATE('13-mai-1940', 'DD-MON-YYYY'),

TO\_DATE('28-mai-2002', 'DD-MON-YYYY'),

null,null); **--insert values in table person**

insert into Person values('Bob', TO\_DATE('11-nov-1965', 'DD-MON-YYYY'),

null,'Walt','Ellen'); **--insert values in table person**

insert into Person values('Susan', TO\_DATE('8-aug-1966', 'DD-MON-YYYY'),

null,'Rich','Sue'); **--insert values in table person**

insert into Person values('Jane', TO\_DATE('1-jan-2006', 'DD-MON-YYYY'),

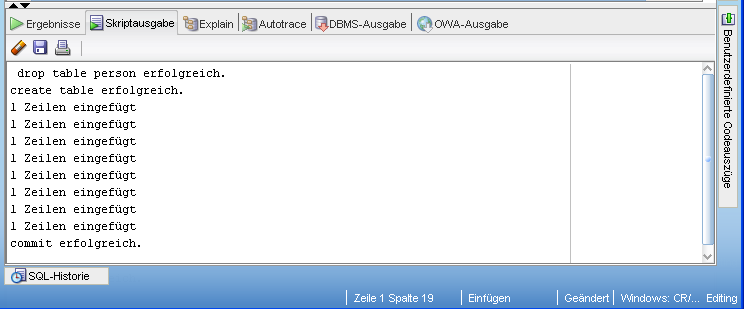
null,'Bob','Susan'); **--insert values in table person**

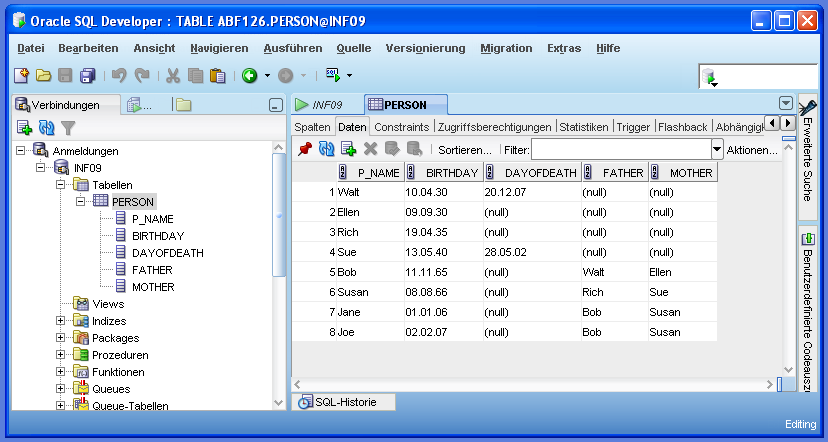
insert into Person values('Joe', TO\_DATE('2-feb-2007', 'DD-MON-YYYY'),

null,'Bob','Susan'); **--insert values in table person**

commit;

**The Screen Output after creating the table and inserting the values:**

****

****

In our code we have changed all names of months to German names, because the Oracle version was in German and it showed errors in the English names of months.

**C)**

Name column should be not null, every person has a name;

Birthday column should be not null, every person has a birthday;

DayOfDeath column should be nullable, every person is not dead;

Father(FK) and Mother(FK) columns should be nullable, we don’t have the information for Grandparents’ father and mother.

**D)**

First we enter the data for Grandparents, then parents and then children. The reason is the presence of foreign keys father and mother. These foreign keys contain the name of the person already present. So a parent should be present before we enter child’s data.