T.E.I. of Central Macedonia

Department of Informatics Engineering

VIDEO CLUB DB

Lab Project for “DATABASES I”

By Anestis Zioulis (4247)

Serres: 03/01/2017

Table of Contents

INTRODUCTION

NOTE: *For the database to be within project limits. There is only one writer, one director and one category per movie. Normally these would be many-to-many, and new tables had to be created in that case.*

This project was created for a local “Video Club” that rent movies in DVD form to its members. The project requirements were gathered after an interview with the owner of the club.

- The company rents the movies for a specific time to a specific price.
- The database must contain info about the members, movies and rentals.

So, eight (8) tables will be created for the requirements to be met:

- MEMBERS
- MOVIES
- RENTALS
- CATEGORIES
- WRITERS
- DIRECTORS
- ACTORS
- ACTORS_MOVIES

The **RENTALS** table is created because of the N:M relationship of the MEMBERS and MOVIES tables. As well as the **ACTORS_MOVIES** is created because of N:M of MOVIES and ACTORS tables.

Below we will see the database creation, the data manipulation and presentation.

Entity-Relationship Diagram

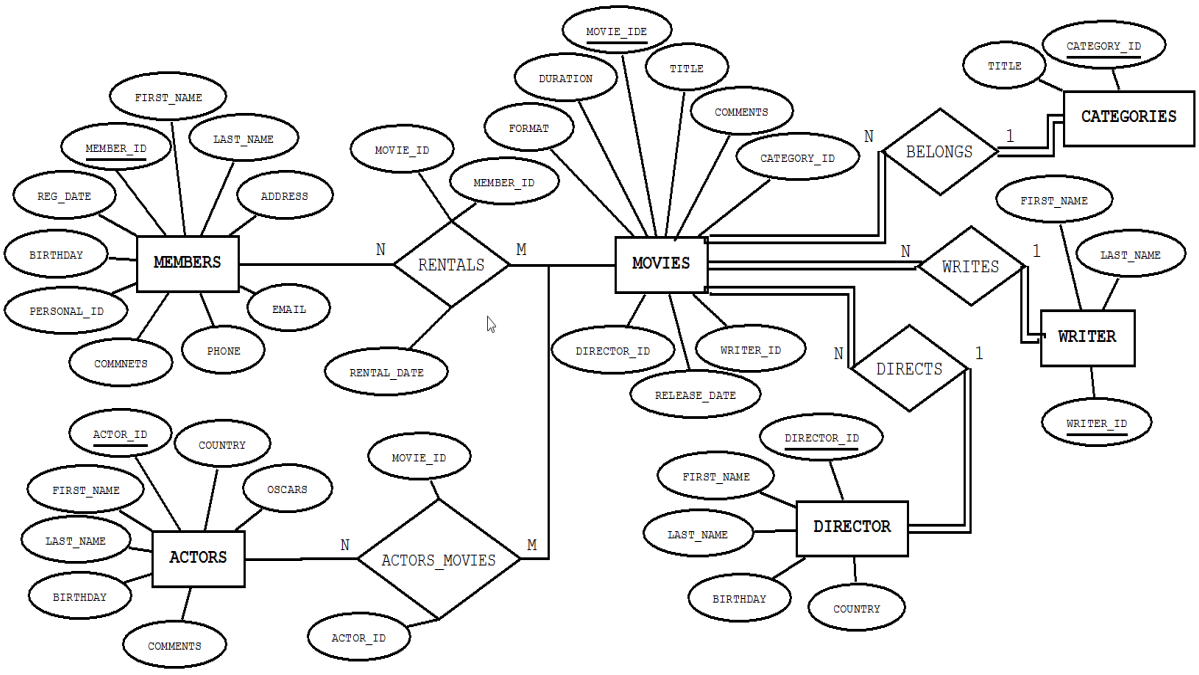


Figure 1 E.R. Diagram

E.R. to RELATIONAL MODEL CONVERTION

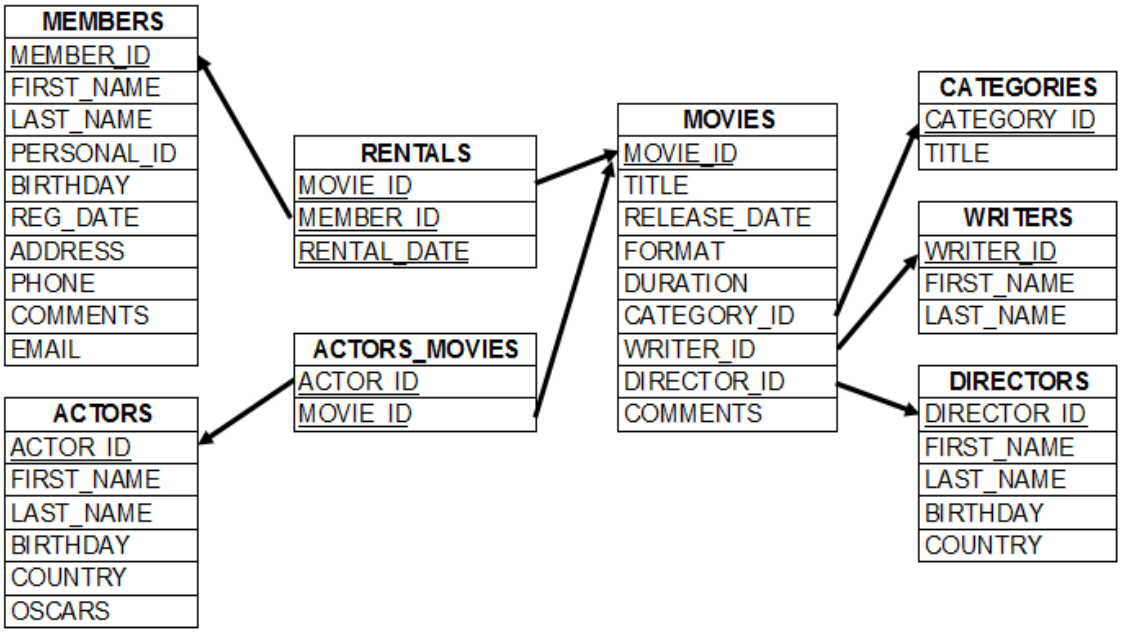


Figure 2 E.R. to Relational Conversion

CHAPTER 1 – CREATION of TABLES

The creation of the tables in MS ACCESS is through SQL queries. To create a query, we go to the “Create” menu and then “Query Design”. We close the popup window and then we press the “SQL” button on top left. We insert or write our query and then we press “Run” next to the SQL button. Creation should be in the following order to avoid conflicts.

The creation of table “MEMBERS”

```
CREATE TABLE MEMBERS(
MEMBER_ID AUTOINCREMENT(1,1),
FIRST_NAME VARCHAR(30) NOT NULL,
LAST_NAME VARCHAR(30) NOT NULL,
PERSONAL_ID VARCHAR(10) NOT NULL,
BIRTHDAY DATETIME NOT NULL,
REG_DATE DATETIME NOT NULL,
ADDRESS VARCHAR(30),
PHONE VARCHAR(20) NOT NULL,
COMMENTS VARCHAR(255),
EMAIL VARCHAR(255),
CONSTRAINT CMMEMBERS PRIMARY KEY(MEMBER_ID));
```

The creation of table “MOVIES”

```
CREATE TABLE MOVIES(
MOVIE_ID AUTOINCREMENT(1,1),
TITLE VARCHAR(50) NOT NULL,
RELEASE_DATE DATETIME NOT NULL,
FORMAT VARCHAR(10) NOT NULL,
DURATION INT NOT NULL,
CATEGORY_ID INT NOT NULL,
WRITER_ID INT NOT NULL,
DIRECTOR_ID INT NOT NULL,
COMMENTS VARCHAR(255),
CONSTRAINT CMMOVIES PRIMARY KEY(MOVIE_ID));
```

The creation of table “RENTALS”

```
CREATE TABLE RENTALS(
MOVIE_ID INT NOT NULL,
MEMBER_ID INT NOT NULL,
RENTAL_DATE DATETIME NOT NULL,
CONSTRAINT CRRENTALS PRIMARY
KEY(MOVIE_ID,MEMBER_ID,RENTAL_DATE));
```

The creation of table “CATEGORIES”

```
CREATE TABLE CATEGORIES(
CATEGORY_ID AUTOINCREMENT(1,1),
TITLE VARCHAR(30) NOT NULL,
CONSTRAINT CCCATEGORIES PRIMARY KEY(CATEGORY_ID));
```

The creation of table “WRITERS”

```
CREATE TABLE WRITERS(
WRITER_ID AUTOINCREMENT(1,1),
FIRST_NAME VARCHAR(30) NOT NULL,
LAST_NAME VARCHAR(30) NOT NULL,
CONSTRAINT CWWRITERS PRIMARY KEY(WRITER_ID));
```

The creation of table “DIRECTORS”

```
CREATE TABLE DIRECTORS(
DIRECTOR_ID AUTOINCREMENT(1,1),
FIRST_NAME VARCHAR(30) NOT NULL,
LAST_NAME VARCHAR(30) NOT NULL,
BIRTHDAY DATETIME NOT NULL,
COUNTRY VARCHAR(20),
CONSTRAINT CDDIRECTOR PRIMARY KEY(DIRECTOR_ID));
```

The creation of table “ACTORS”

```
CREATE TABLE ACTORS(
ACTOR_ID AUTOINCREMENT(1,1),
FIRST_NAME VARCHAR(30) NOT NULL,
LAST_NAME VARCHAR(30) NOT NULL,
BIRTHDAY DATETIME NOT NULL,
COUNTRY VARCHAR(20),
OSCARS INT,
COMMENTS VARCHAR(255),
CONSTRAINT CAACTORS PRIMARY KEY(ACTOR_ID));
```

The creation of table “MEMBERS”

```
CREATE TABLE ACTORS_MOVIES(
ACTOR_ID INT NOT NULL,
MOVIE_ID INT NOT NULL,
CONSTRAINT CAMACTORS_MOVIES PRIMARY
KEY(ACTOR_ID,MOVIE_ID));
```

We will now create constraints between the tables.

1. **RENTALS- MEMBERS Constraints**

```
ALTER TABLE RENTALS
ADD CONSTRAINT KRMERENTALS
FOREIGN KEY (MEMBER_ID)
REFERENCES MEMBERS(MEMBER_ID);
```

2. **RENTALS- MOVIES Constraints**

```
ALTER TABLE RENTALS
ADD CONSTRAINT KRMORENTALS
FOREIGN KEY (MOVIE_ID)
REFERENCES MOVIES(MOVIE_ID);
```

3. **MOVIES-CATEGORIES Constraints**

```
ALTER TABLE MOVIES
ADD CONSTRAINT KRMOCATEGORIES
FOREIGN KEY (CATEGORY_ID)
REFERENCES CATEGORIES(CATEGORY_ID);
```

4. **MOVIES-WRITERS Constraints**

```
ALTER TABLE MOVIES
ADD CONSTRAINT KRMOWRITERS
FOREIGN KEY (WRITER_ID)
REFERENCES WRITERS(WRITER_ID);
```

5. **MOVIES-DIRECTORS Constraints**

```
ALTER TABLE MOVIES
ADD CONSTRAINT KRMODIRECTORS
FOREIGN KEY (DIRECTOR_ID)
REFERENCES DIRECTORS(DIRECTOR_ID);
```

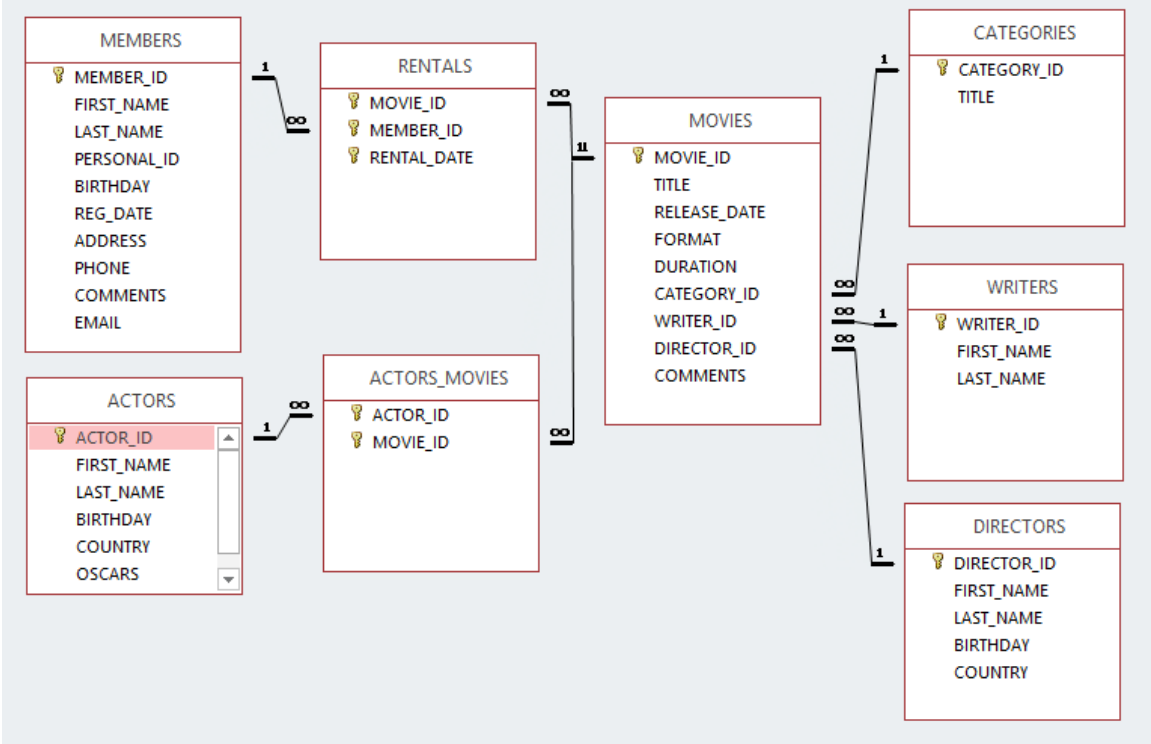
6. **ACTORS_MOVIES-MOVIES Constraints**

```
ALTER TABLE ACTORS_MOVIES
ADD CONSTRAINT KRAMMOVIES
FOREIGN KEY (MOVIE_ID)
REFERENCES MOVIES(MOVIE_ID);
```

7. **ACTORS_MOVIES-ACTORS Constraints**

```
ALTER TABLE ACTORS_MOVIES
ADD CONSTRAINT KRAMACTORS
FOREIGN KEY (ACTOR_ID)
REFERENCES ACTORS(ACTOR_ID);
```

MS Access Relational Model.



CHAPTER 2 – INSERT / UPDATE / DELETE DATA

INSERT

We insert data into the tables, in the following order to avoid conflicts. We insert data in CATEGORIES, WRITERS, DIRECTORS tables before we fill the table MOVIES.

Some dummy data are being provided below as example.

MEMBERS

1. INSERT INTO MEMBERS (FIRST_NAME, LAST_NAME, PERSONAL_ID, BIRTHDAY, REG_DATE, ADDRESS, PHONE)
   VALUES ("George","Papadopoulos","AE 12342", #02/01/1945#, #01/01/2009#, "Papadopoulou 30 Serres", "2321022312");
2. INSERT INTO MEMBERS (FIRST_NAME, LAST_NAME, PERSONAL_ID, BIRTHDAY, REG_DATE, ADDRESS, PHONE)
   VALUES ("Sunday","Takis","AA 15143", #23/06/1965#, #01/01/1999#, "Lucky 7 Salonica","23102321678");
3. INSERT INTO MEMBERS (FIRST_NAME, LAST_NAME, PERSONAL_ID, BIRTHDAY, REG_DATE, ADDRESS, PHONE)
   VALUES ("Alex","Englishman","ΩΩ 23112", #23/06/1997#, #01/01/1999#, "Sofokleous 1 Athens","2102789978");
4. INSERT INTO MEMBERS (FIRST_NAME, LAST_NAME, PERSONAL_ID, BIRTHDAY, REG_DATE, ADDRESS, PHONE)
   VALUES ("Rudy","Gandi","ΠΠ 112233", #23/06/1977#, #01/01/2009#, "Sayan 23 Bulba","2110785523");

CATEGORIES

1. INSERT INTO CATEGORIES (TITLE)

   VALUES ("DRAMA");
2. INSERT INTO CATEGORIES (TITLE)

   VALUES ("ACTION");
3. INSERT INTO CATEGORIES (TITLE)

   VALUES ("FANTASY");
4. INSERT INTO CATEGORIES (TITLE)

   VALUES ("HORROR");

WRITERS

1. INSERT INTO WRITERS (FIRST_NAME, LAST_NAME)

   VALUES ("BOBY","GREAT");
2. INSERT INTO WRITERS (FIRST_NAME, LAST_NAME)

   VALUES ("STEVEN","SABAT");
3. INSERT INTO WRITERS (FIRST_NAME, LAST_NAME)

   VALUES ("BORAT","NASGUL");
4. INSERT INTO WRITERS (FIRST_NAME, LAST_NAME)

   VALUES ("KIROV", "NAGASAKI");

DIRECTORS

1. INSERT INTO DIRECTORS (FIRST_NAME, LAST_NAME, BIRTHDAY, COUNTRY)
   VALUES ("SUZAN","LAMIANOF", #02/10/1966#, "ROMANIA");
2. INSERT INTO DIRECTORS (FIRST_NAME, LAST_NAME, BIRTHDAY, COUNTRY)
   VALUES ("GIORGIO","PANTESPANI", #15/06/1996#, "ITALY");
3. INSERT INTO DIRECTORS (FIRST_NAME, LAST_NAME, BIRTHDAY, COUNTRY)
   VALUES ("SIMON","PEREZ", #25/03/1936#, "GERMANY");
4. INSERT INTO DIRECTORS (FIRST_NAME, LAST_NAME, BIRTHDAY, COUNTRY)
   VALUES ("SYLVIA","MAYERS", #12/04/1977#, "AMERICA");

MOVIES

1. INSERT INTO MOVIES (TITLE, RELEASE_DATE, FORMAT, DURATION, CATEGORY_ID, WRITER_ID, DIRECTOR_ID, COMMENTS)
   VALUES ("BAYWATCH", #23/06/1988#, "DVD", 123, 1, 1, 1, NULL);
2. INSERT INTO MOVIES (TITLE, RELEASE_DATE, FORMAT, DURATION, CATEGORY_ID, WRITER_ID, DIRECTOR_ID, COMMENTS)
   VALUES ("POPE IN DOPE", #23/06/1966#, "DVD", 66, 3, 3, 3, NULL);
3. INSERT INTO MOVIES (TITLE, RELEASE_DATE, FORMAT, DURATION, CATEGORY_ID, WRITER_ID, DIRECTOR_ID, COMMENTS)

VALUES ("ONCE UPON A TRAIN", #23/06/1955#, "BLUE-RAY", 157, 2, 2, 2, NULL)

4. INSERT INTO MOVIES (TITLE, RELEASE_DATE, FORMAT, DURATION, CATEGORY_ID, WRITER_ID, DIRECTOR_ID, COMMENTS)
   VALUES ("EAT ME", #06/06/1999#, "BLUE-RAY", 157, 4, 4, 4, NULL);

ACTORS

1. INSERT INTO ACTORS (FIRST_NAME, LAST_NAME,BIRTHDAY, COUNTRY, OSCARS, COMMENTS)
   VALUES ("LOGAN","MCTREVOR", #11/11/1987#, "ENGLAND", 5, NULL);
2. INSERT INTO ACTORS (FIRST_NAME, LAST_NAME,BIRTHDAY, COUNTRY, OSCARS, COMMENTS)
   VALUES ("FRANSCESCA","FABIANI", #12/12/1990#, "ITALY", 15, NULL);
3. INSERT INTO ACTORS (FIRST_NAME, LAST_NAME, BIRTHDAY, COUNTRY, OSCARS, COMMENTS)
   VALUES ("SERGEY","VOLANOF", #01/01/1971#, "RUSSIA", 0, NULL);
4. INSERT INTO ACTORS (FIRST_NAME, LAST_NAME,BIRTHDAY, COUNTRY, OSCARS, COMMENTS)
   VALUES ("YURI","MOLOTOV", #05/06/1984#, "RUSSIA", 3, NULL);

ACTORS_MOVIES

1. INSERT INTO ACTORS_MOVIES (ACTOR_ID, MOVIE_ID)
   VALUES(1,1);
2. INSERT INTO ACTORS_MOVIES (ACTOR_ID, MOVIE_ID)
   VALUES (1, 3);
3. INSERT INTO ACTORS_MOVIES (ACTOR_ID, MOVIE_ID)
   VALUES (4, 1);
4. INSERT INTO ACTORS_MOVIES (ACTOR_ID, MOVIE_ID)
   VALUES (3, 3);

RENTALS

1. INSERT INTO RENTALS (MOVIE_ID, MEMBER_ID, RENTAL_DATE)
   VALUES (1, 3, #01/01/2017#);
2. INSERT INTO RENTALS (MOVIE_ID, MEMBER_ID, RENTAL_DATE)
   VALUES (1, 4, #02/01/2017#);
3. INSERT INTO RENTALS (MOVIE_ID, MEMBER_ID, RENTAL_DATE)
   VALUES (3, 5, #01/01/2014#);
4. INSERT INTO RENTALS (MOVIE_ID, MEMBER_ID, RENTAL_DATE)
   VALUES (4, 6, #01/01/2012#);

UPDATE

If we want to update some already inserted data, either because of a mistake or by the change of info (a member's address for example). We use one of the below examples applicable to our case.

MEMBERS

Update of the member with MEMBER_ID = 3 from George to Jack.

```
UPDATE MEMBERS SET FIRST_NAME = "Jack"

WHERE MEMBER_ID = 3;
```

MOVIES

Update of the movie title "EAT ME" to "EAT ME 2.5".

```
UPDATE MOVIES SET TITLE = "EAT ME 2.5"

WHERE TITLE = "EAT ME";
```

ACTORS

Update of the number of OSCARS of the actor "SERGEY VOLANOF" to 2.

```
UPDATE ACTORS SET OSCARS = 2

WHERE FIRST_NAME="SERGEY" AND LAST_NAME="VOLANOF";
```

DELETE

Some delete examples follow in case you need to delete data. Beware that you must follow an order here as well if it comes to tables with total participation.

MOVIES

```
DELETE *
FROM MOVIES
WHERE MOVIE_ID=2;
```

MEMBERS

```
DELETE *
FROM MEMBERS
WHERE LAST_NAME = "PAPADOPOULOS" AND COMMENTS = "BANNED";
```

ACTORS

```
DELETE *

FROM ACTORS

WHERE OSCARS <= 2;
```

CHAPTER 3 – QUERIES

SELECT

1. **Show the member with MEMBER_ID = 3.**

   SELECT *

   FROM MEMBERS

   WHERE MEMBER_ID=3;

2. **Show the name, surname and date of birth of the members that where born before or at January 1st, 1985.**

   SELECT FIRST_NAME, LAST_NAME, BIRTHDAY

   FROM MEMBERS

   WHERE BIRTHDAY<=#01/01/1985#;

3. **Show the movies that where released from January 1st, 1950 to January 1st, 1990.**

   SELECT *

   FROM MOVIES

   WHERE RELEASE_DATE BETWEEN #01/01/1950# AND #01/01/1990#;

4. **Show the movies that are available in blue-ray format and belong to category with the id = 2.**

   SELECT *

   FROM MOVIES

   WHERE FORMAT="BLUE-RAY" AND CATEGORY_ID= 2;

5. **Show the actors whom their country name doesn’t start with the letter “E” and order them in descending order by their last name.**

   SELECT *

   FROM ACTORS

   WHERE COUNTRY NOT LIKE "E*"

   ORDER BY LAST_NAME DESC;

6. **Show the actors whose name contains the letter “A” and have at least one Oscar.**

   SELECT *

   FROM ACTORS

   WHERE FIRST_NAME LIKE "*A*" AND

   OSCARS >=1;

AGGREGATION

1. **Show movies and their number of rentals if they were rented more than once. While grouping them by name.**

   SELECT MOVIE_ID, COUNT(RENTALS.MOVIE_ID)

   FROM RENTALS

   GROUP BY MOVIE_ID

   HAVING COUNT(RENTALS.MOVIE_ID)>1;

2. **Show drama movies.**

   SELECT MOVIES.MOVIE_ID, MOVIES.TITLE AS MOVIE, CATEGORIES.TITLE AS CATEGORY

   FROM CATEGORIES, MOVIES

   WHERE CATEGORIES.CATEGORY_ID=MOVIES.CATEGORY_ID

   AND

   CATEGORIES.TITLE="DRAMA";

3. **Show action movies that where released before 1980.**

   SELECT MOVIES.MOVIE_ID, MOVIES.TITLE AS MOVIE, CATEGORIES.TITLE AS CATEGORY, MOVIES.RELEASE_DATE AS RELEASE_DATE

   FROM CATEGORIES, MOVIES

   WHERE CATEGORIES.CATEGORY_ID=MOVIES.CATEGORY_ID

   AND

```
CATEGORIES.TITLE="ACTION" AND

MOVIES.RELEASE_DATE<#01/01/1980#;
```

4. **Show all the movies by category.**

```
SELECT MOVIES.MOVIE_ID, MOVIES.TITLE AS MOVIE,
CATEGORIES.TITLE AS CATEGORY

FROM CATEGORIES, MOVIES

WHERE CATEGORIES.CATEGORY_ID=MOVIES.CATEGORY_ID

AND

CATEGORIES.CATEGORY_ID>=0 ;
```