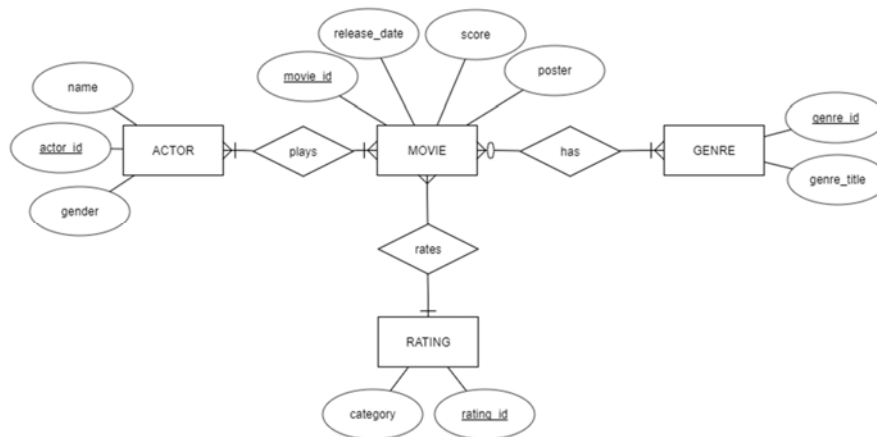# MOVIES 데이타베이스, 테이블 생성

```sql
use movies_db;

CREATE TABLE ACTOR (
  actor_id int IDENTITY(1, 1) PRIMARY KEY,
  name varchar(30) NOT NULL,
  gender char(1) NOT NULL
) ;

CREATE TABLE GENRE (
  genre_id int IDENTITY(1, 1) PRIMARY KEY,
  genre_title varchar(30) NOT NULL
) ;

CREATE TABLE RATING (
  rating_id int IDENTITY(1, 1) PRIMARY KEY,
  category varchar(16) NOT NULL,
);

CREATE TABLE MOVIE (
  movie_id int IDENTITY(1, 1) PRIMARY KEY,
  movie_title varchar(50) NOT NULL,
  release_date datetime NOT NULL,
  score float NOT NULL,
  poster varchar(30) NOT NULL,
  rating_id INT NOT NULL,
  FOREIGN KEY (rating_id) REFERENCES RATING(rating_id)
);
```

```sql
CREATE TABLE PLAYS (
  actor_id int REFERENCES ACTOR(actor_id),
  movie_id int REFERENCES MOVIE(movie_id),
  PRIMARY KEY (actor_id, movie_id)
);

CREATE TABLE HAS (
  movie_id int REFERENCES MOVIE(movie_id),
  genre_id int REFERENCES GENRE(genre_id),
  PRIMARY KEY (movie_id, genre_id)
);
```

Actor

- __actor_id : *int*
- __name : *str*
- __gender: *str*

+ __init__(self, id, name, gender)
+ __str__(): *str*
# *attribute에 대한 getter*
# *연관관계에 대한 method*

Genre

- __genre_id: *int*
- __genre_title: *str*

+ __init__(self, id, title)
# *attribute에 대한 getter*
# *연관관계에 대한 method*

1..n

1..n

0..n

1..n

Movie

- __movie_id: *int*
- __movie_title: *str*
- __release_date: *Date*
- __rating: *Rating*
- __players: [*Actor*]
- __genres: [*Genre*]

+ __init__(self, id, title, release_date)
# attribute에 대한 getter/setter
# *연관관계에 대한 method*

0..n

1

Rating

- __rating_id: *int*
- __category: *str*

+ __init__(self, id, category)
# *attribute에 대한 getter*