

Introduction to SQL and ER to Relational Schemas

Discussion Session 3

Announcement

Please make sure your homework answers are **clear and legible**.

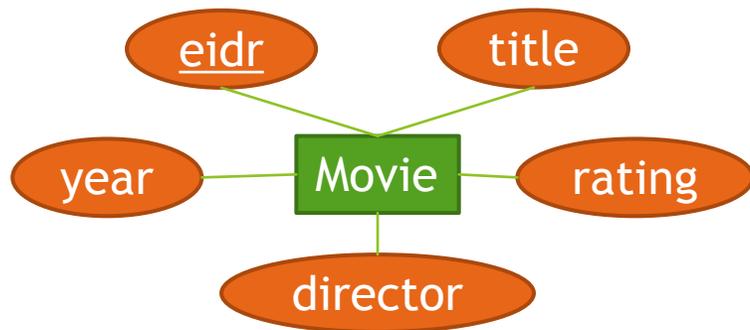
If in doubt, ask someone else to read the answers for you, and check if they understand what you wrote.

We will be **deducting** points from answers we can't understand.

Relational Model

We have been asked to help design a database for the entertainment industry. We are required to store information about **movies**, which have a *title*, *year* of production, *director's name*, *rating* (points out of 10), and a unique *entertaining identifier registry*. Assume that a director does not direct two movies with the same title in her lifetime.

► ER entity representation:



► Relation schema:

Movie(*eidr*: string, *title*: string, *director*: string, *year*: int, *rating*: real)

Relational Model

- ▶ Relation instance of the **Movie** relation schema:

eidr	title	year	director	rating
m01	It	2017	A. Muschietti	7.4
m02	Jurassic Park	1993	S. Spielberg	8.1
m03	It	1990	T. L. Wallace	7.5
m04	The Post	2017	S. Spielberg	7.2
m05	Child's Play	1988	T. Holland	6.6



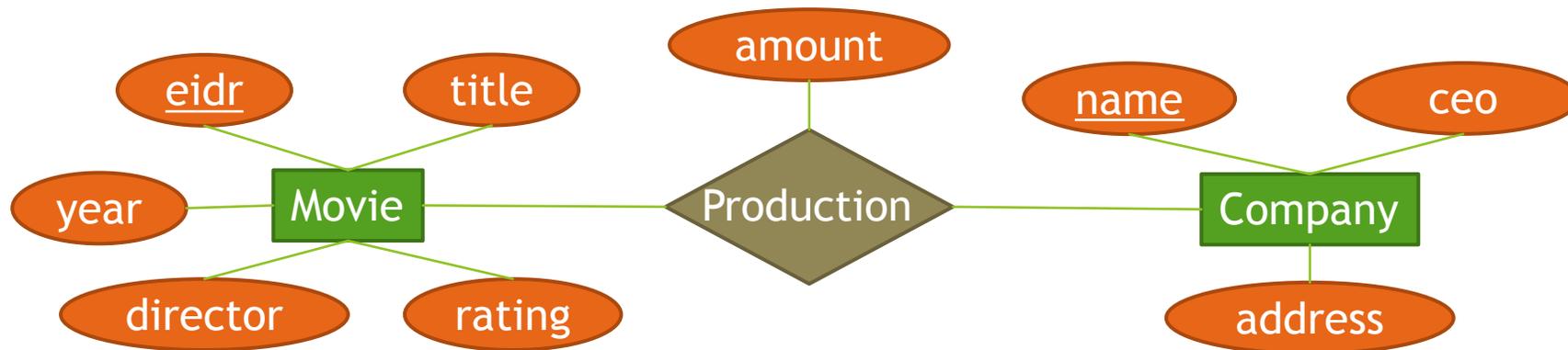
Key Constraints

eidr	title	year	director	rating
m01	It	2017	A. Muschietti	7.4
m02	Jurassic Park	1993	S. Spielberg	8.1
m03	It	1990	T. L. Wallace	7.5
m04	The Post	2017	S. Spielberg	7.2
m05	Child's Play	1988	T. Holland	6.6

- ▶ Given the *legal* **Movie** relation instance above:
 - ▶ What are the candidate keys of the **Movie** relation?
 - ▶ *<eidr>* and *<title, director>*
 - ▶ Can you give an example of a superkey?
 - ▶ *<eidr>*, *<eidr, title>*, *<title, director, rating>*, etc.
 - ▶ Is *<eidr, title, director>* a candidate key? Why?
 - ▶ No! It's not minimal.

A production **company** is an important part of the film industry as they finance movies. A company has a unique *name*, a *CEO*, and an *address*. When they **produce** a movie, they participate with a given *amount* of financial support. *Note, however, that there is possible to have independent movies (i.e. not produced by companies).*

- Provide the **ER diagram** involving **movies** and **production companies**. Give also their relational schemas.



```
Movie(eidr: string, title: string, director: string, year: int, rating: real)
Production(eidr: string, name: string, amount: real)
Company(name: string, ceo: string, address: string)
```

Foreign Key Constraints

```
Movie(eidr: string, title: string, director: string, year: int, rating: real)
Production(eidr: string, name: string, amount: real)
Company(name: string, ceo: string, address: string)
```

- ▶ Is (are) there any **foreign key constraint(s)** we must enforce to keep our DB consistent?
 - ▶ *<eidr>* and *<name>* in **Production** must refer to appropriate fields in **Movie** and **Company** relations, respectively.
- ▶ What actions would you enforce if either a **Movie** or a **Company** record is deleted?
 - ▶ Prevent the deletion.
 - ▶ Cascade the deletion.

SQL Time!



- ▶ Create the **Movie**, **Production**, and **Company** relations in SQL, enforcing the integrity constraints just described.

```
CREATE TABLE Movie(  
  eidr CHAR(3),  
  title CHAR(20),  
  year INT,  
  director CHAR(20),  
  rating REAL,  
  PRIMARY KEY (eidr),  
  UNIQUE (title, director)  
);
```

```
CREATE TABLE Company(  
  name CHAR(20),  
  ceo CHAR(20),  
  address CHAR(128),  
  PRIMARY KEY (name)  
);
```

```
CREATE TABLE Production(  
  eidr CHAR(3),  
  name CHAR(20),  
  amount REAL,  
  PRIMARY KEY (eidr, name),  
  FOREIGN KEY (eidr)  
    REFERENCES Movie(eidr)  
    ON DELETE CASCADE,  
  FOREIGN KEY (name)  
    REFERENCES Company(name)  
    ON DELETE CASCADE  
);
```

SQL Time!



- ▶ Suppose that we *do not* allow a **Movie** to be produced by *more than one* **Company**. Provide the SQL statement to create the relation **Production** with this new *key constraint*.

```
DROP TABLE Production;
```



```
CREATE TABLE Production(  
    eidr CHAR(3),  
    name CHAR(20) NOT NULL,  
    amount REAL,  
    PRIMARY KEY (eidr),  
    FOREIGN KEY (eidr)  
        REFERENCES Movie(eidr)  
        ON DELETE CASCADE,  
    FOREIGN KEY (name)  
        REFERENCES Company(name)  
        ON DELETE CASCADE  
);
```

SQL Time!



- ▶ We are getting very strict! Suppose we have been informed that *no independent movies should exist* in the DB. That is, each **Movie** must be produced by *one and only one* Company. How would you modify your relational schemas and SQL create statements?

```
Movie_Prod(eidr: string, title: string, director: string, year: int, rating: real, name: string, amount: real)
Company(name: string, ceo: string, address: string)
```

```
DROP TABLE Production;
DROP TABLE Movie;
```



```
CREATE TABLE Movie_Prod(
  eidr CHAR(3),
  title CHAR(20),
  year INT,
  director CHAR(20),
  rating REAL,
  name CHAR(20) NOT NULL,
  amount REAL,
  PRIMARY KEY (eidr),
  UNIQUE (title, director),
  FOREIGN KEY (name)
    REFERENCES Company(name) ON DELETE NO ACTION
);
```

SQL Time!



- ▶ Provide the SQL to insert two **Company** tuples:
 - ▶ *name*: Fox, *ceo*: Quagmire, *address*: Los Angeles
 - ▶ *name*: Universal, *ceo*: Griffin, *address*: Irvine

```
INSERT INTO Company(name, ceo, address)
VALUES('Fox', 'Quagmire', 'Los Angeles');

INSERT INTO Company(name, ceo, address)
VALUES('Universal', 'Griffin', 'Irvine');
```

- ▶ Provide the SQL to insert the movie *It*, identified by '*m01*', produced by *Universal*, who invested *\$1 million* in *1990*. The movie was directed by *Wallace* and received a rating of *7.5*.

```
INSERT INTO Movie_Prod(eidr, title, year, director, rating, name, amount)
VALUES('m01', 'It', 1990, 'Wallace', 7.5, 'Fox', 1000000);
```

SQL Time!



- Explain what happens if we try to delete the company *Fox*.

```
DELETE FROM Company WHERE name='Fox';
```



Error

SQL query:

```
DELETE FROM Company WHERE name='Fox'
```

MySQL said:

```
#1451 - Cannot delete or update a parent row: a foreign key constraint fails (`cs174a`.`movie_prod`,  
CONSTRAINT `movie_prod_ibfk_1` FOREIGN KEY (`name`) REFERENCES `Company` (`name`) ON DELETE NO  
ACTION)
```

Questions?