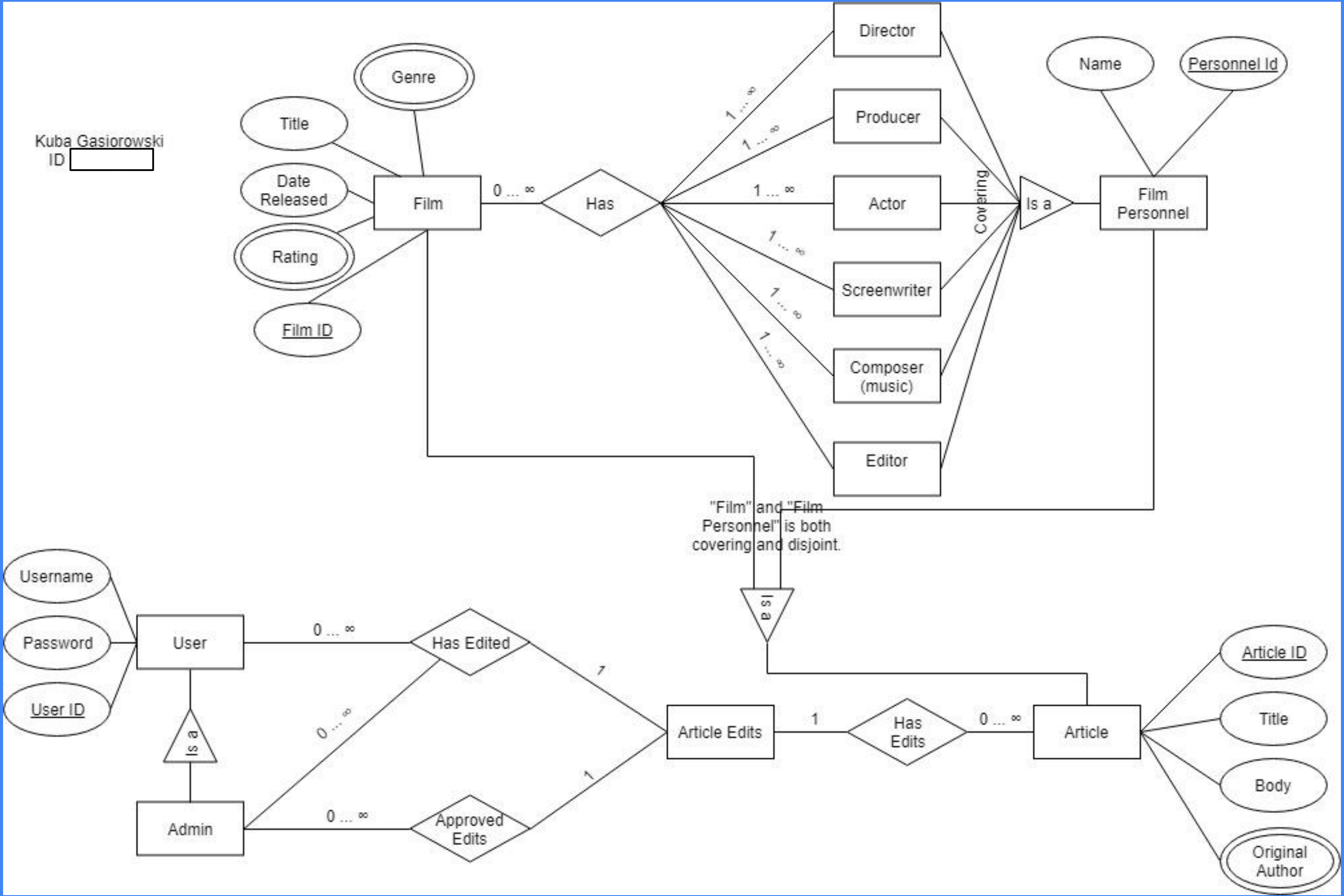
CSE305 Final Project - IMDBc

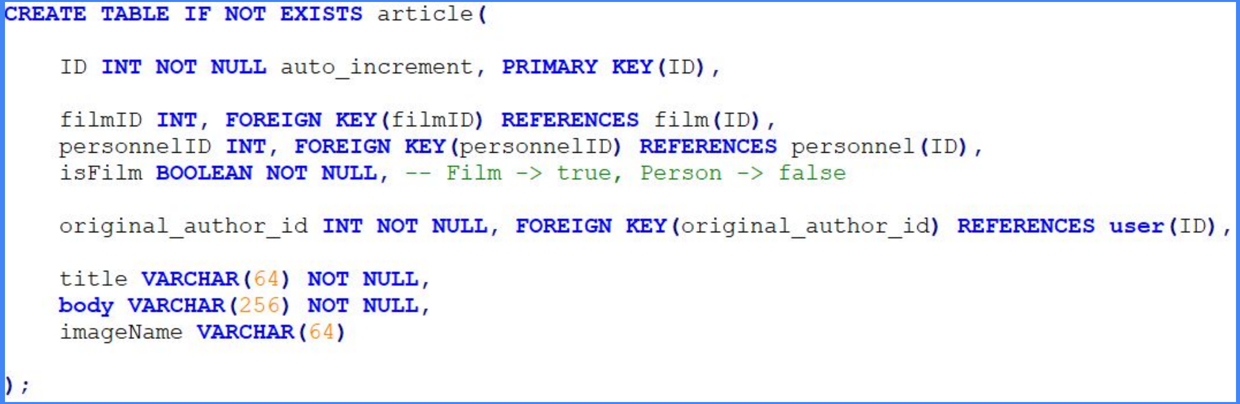
Author: Kuba Gasiorowski

Kuba Gasiorowski
ID

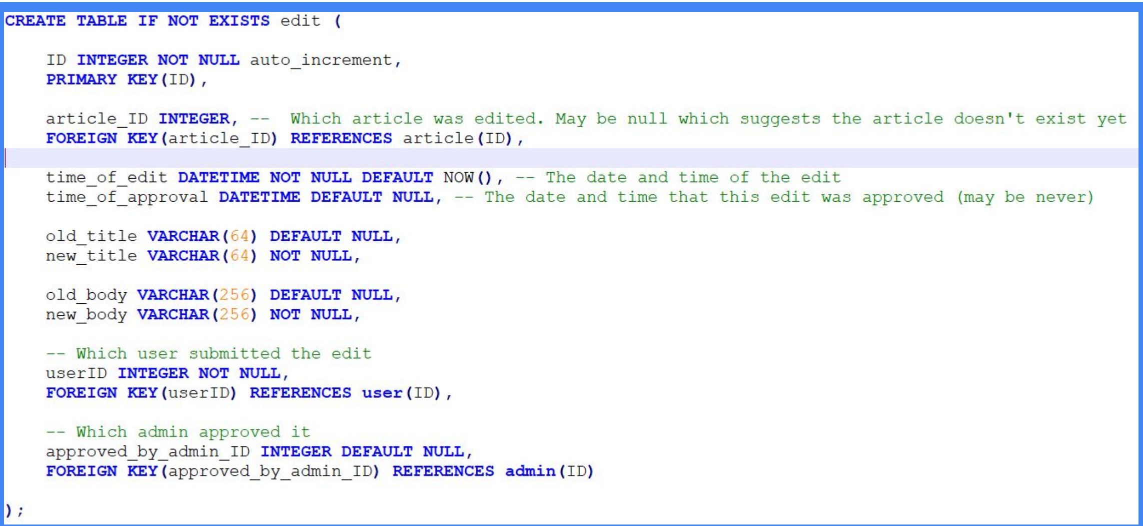


"Film" and "Film Personnel" is both covering and disjoint.

The user system (users, admins, articles, and edits)

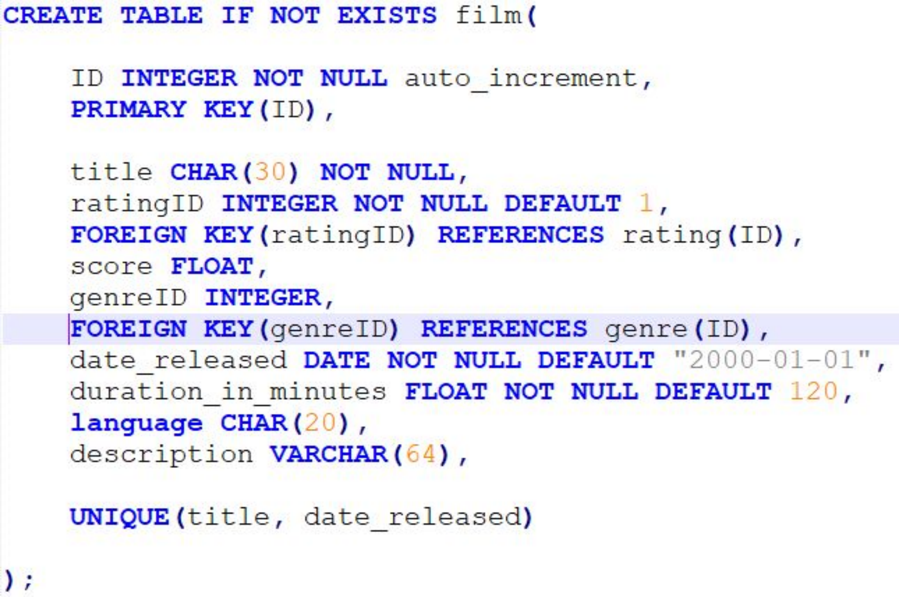


```
CREATE TABLE IF NOT EXISTS article(

    ID INT NOT NULL auto_increment, PRIMARY KEY(ID),

    filmID INT, FOREIGN KEY(filmID) REFERENCES film(ID),
    personnelID INT, FOREIGN KEY(personnelID) REFERENCES personnel(ID),
    isFilm BOOLEAN NOT NULL, -- Film -> true, Person -> false

    original_author_id INT NOT NULL, FOREIGN KEY(original_author_id) REFERENCES user(ID),

    title VARCHAR(64) NOT NULL,
    body VARCHAR(256) NOT NULL,
    imageName VARCHAR(64)

);
```

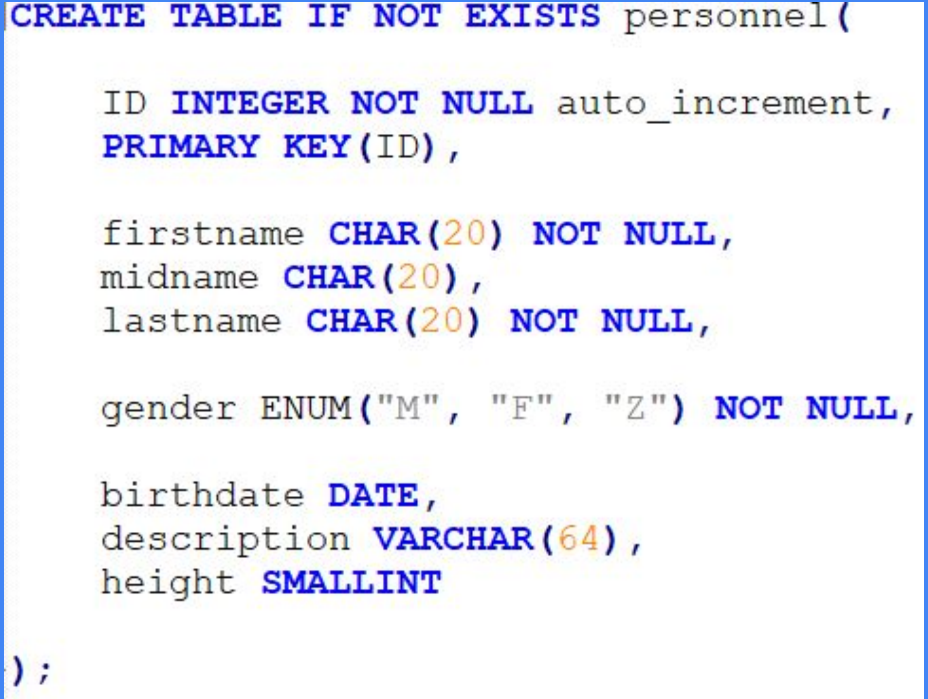


```
CREATE TABLE IF NOT EXISTS edit (

    ID INTEGER NOT NULL auto_increment,
    PRIMARY KEY(ID),

    article_ID INTEGER, --  Which article was edited. May be null which suggests the article doesn't exist yet
    FOREIGN KEY(article_ID) REFERENCES article(ID),

    time_of_edit DATETIME NOT NULL DEFAULT NOW(), -- The date and time of the edit
    time_of_approval DATETIME DEFAULT NULL, -- The date and time that this edit was approved (may be never)

    old_title VARCHAR(64) DEFAULT NULL,
    new_title VARCHAR(64) NOT NULL,

    old_body VARCHAR(256) DEFAULT NULL,
    new_body VARCHAR(256) NOT NULL,

    -- Which user submitted the edit
    userID INTEGER NOT NULL,
    FOREIGN KEY(userID) REFERENCES user(ID),

    -- Which admin approved it
    approved_by_admin_ID INTEGER DEFAULT NULL,
    FOREIGN KEY(approved_by_admin_ID) REFERENCES admin(ID)

);
```

The actual data system (films, personnel, credits)

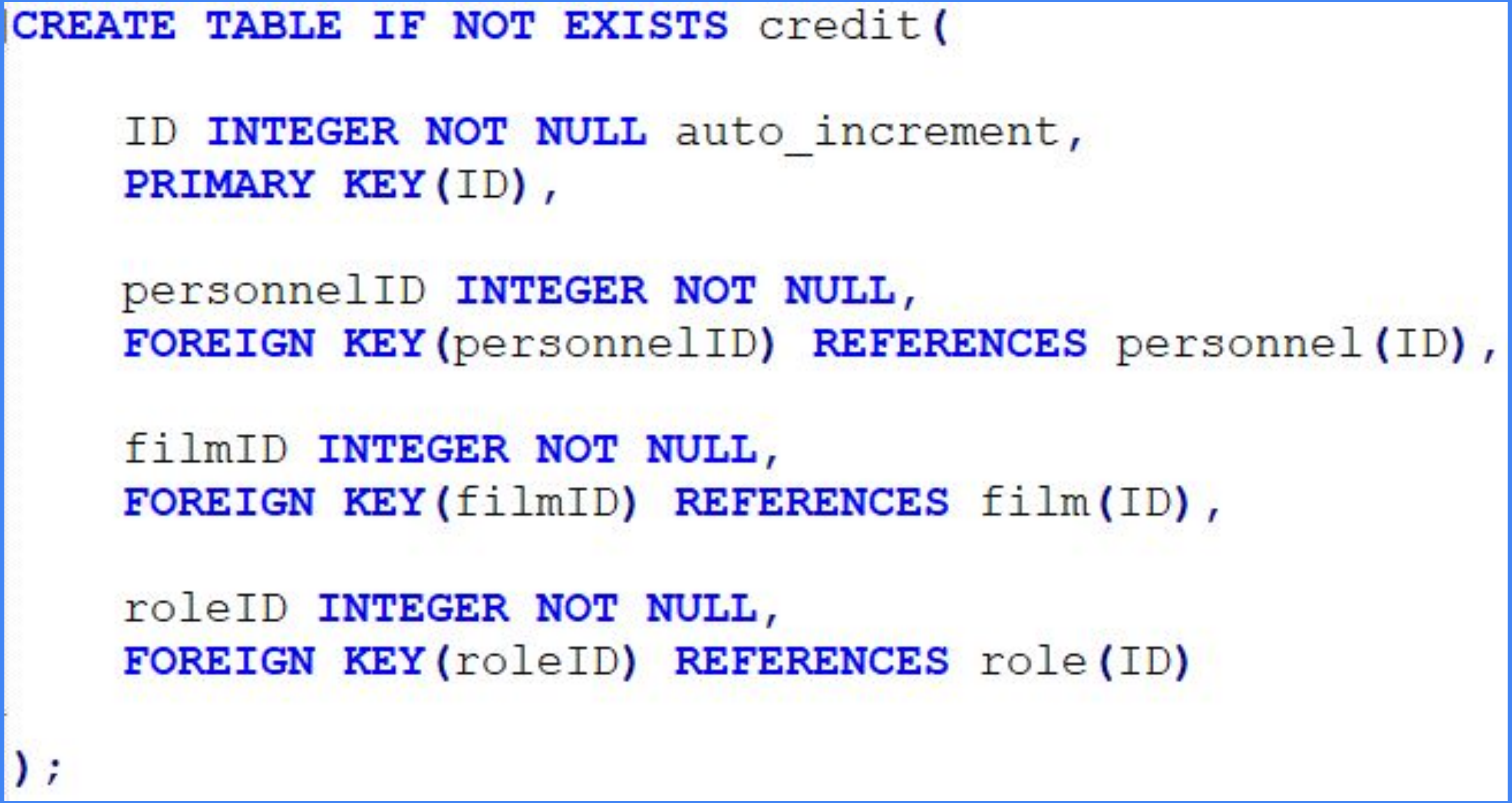


```
CREATE TABLE IF NOT EXISTS film(

    ID INTEGER NOT NULL auto_increment,
    PRIMARY KEY(ID),

    title CHAR(30) NOT NULL,
    ratingID INTEGER NOT NULL DEFAULT 1,
    FOREIGN KEY(ratingID) REFERENCES rating(ID),
    score FLOAT,
    genreID INTEGER,
    FOREIGN KEY(genreID) REFERENCES genre(ID),
    date_released DATE NOT NULL DEFAULT "2000-01-01",
    duration_in_minutes FLOAT NOT NULL DEFAULT 120,
    language CHAR(20),
    description VARCHAR(64),

    UNIQUE(title, date_released)

);
```



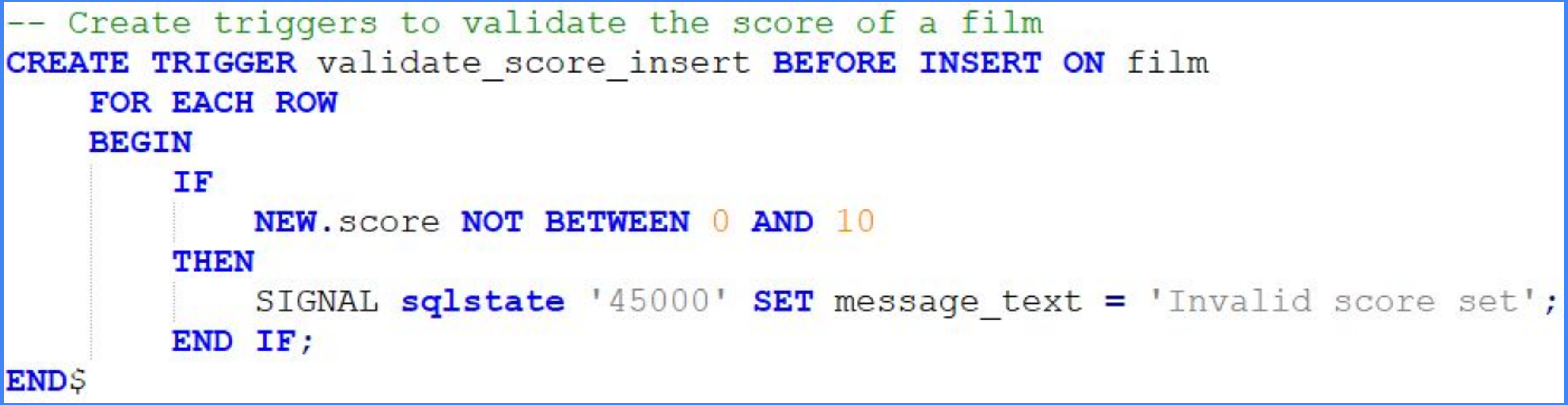
```
CREATE TABLE IF NOT EXISTS personnel(

    ID INTEGER NOT NULL auto_increment,
    PRIMARY KEY(ID),

    firstname CHAR(20) NOT NULL,
    midname CHAR(20),
    lastname CHAR(20) NOT NULL,

    gender ENUM("M", "F", "Z") NOT NULL,

    birthdate DATE,
    description VARCHAR(64),
    height SMALLINT

);
```

The actual data system (continued)



```
CREATE TABLE IF NOT EXISTS credit(

    ID INTEGER NOT NULL auto_increment,
    PRIMARY KEY(ID),

    personnelID INTEGER NOT NULL,
    FOREIGN KEY(personnelID) REFERENCES personnel(ID),

    filmID INTEGER NOT NULL,
    FOREIGN KEY(filmID) REFERENCES film(ID),

    roleID INTEGER NOT NULL,
    FOREIGN KEY(roleID) REFERENCES role(ID)

);
```

Constraints



```
-- Create triggers to validate the score of a film
CREATE TRIGGER validate_score_insert BEFORE INSERT ON film
    FOR EACH ROW
    BEGIN
        IF
            NEW.score NOT BETWEEN 0 AND 10
        THEN
            SIGNAL sqlstate '45000' SET message_text = 'Invalid score set';
        END IF;
END$
```

(they basically all look like this)