

# Homework 6 Solution:

(a) The problem to solve is the implementation of the two weak entity sets. We integrate the two relations into the corresponding tables. This gives us:

Studio(Name, Website, Address)

Actors(Name, Website)

Series(Title, Creator, Start\_Date, End\_Date)

Produces(Series Title, Studio\_Name)

Character(Name, Role, Series Title)

Plays(Actor Name, Character Name, Series Title)

Episodes(Episode Title, Series Title, Air\_date, Season, Description, URL)

PartOfEpisodesSeries(Episode Title, Series Title)

PartOfCharacter(Character Name, Series Title)

Staring(Actors Name, Series Title)

(b) The PartOfCharacter Table and the PartofEpisodesSeries tables are superfluous. We could merge the Series and Studio tables, and do away with the "Produces" table, but the result would not be in BCNF, since the dependencies on Studio\_Name would not have the left side being a key. Staring should be an independent table, as only few actors per series get to be a star, e.g. only Kirk and Spock.

(c) Here is my implementation with explicit foreign keys and non null constraints.

```
CREATE DATABASE IF NOT EXISTS startreck;
USE startreck;
```

```
DROP TABLE IF EXISTS starring;
DROP TABLE IF EXISTS episodes;
DROP TABLE IF EXISTS plays;
DROP TABLE IF EXISTS characters;
DROP TABLE IF EXISTS produces;
DROP TABLE IF EXISTS series;
DROP TABLE IF EXISTS actors;
DROP TABLE IF EXISTS studio;
```

```
CREATE TABLE studio (
    studio_name VARCHAR(50) PRIMARY KEY,
    website VARCHAR(100) NOT NULL,
    address VARCHAR(200) NOT NULL
);
```

```

CREATE TABLE actors (
    actor_name VARCHAR(50) PRIMARY KEY,
    website VARCHAR(100) NOT NULL
);

CREATE TABLE series (
    series_title VARCHAR(50) PRIMARY KEY,
    creator VARCHAR(100) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE DEFAULT '9999-12-31',
    CHECK (start_date < end_date)
);

CREATE TABLE produces (
    series_title VARCHAR(50),
    studio_name VARCHAR(50),
    PRIMARY KEY (series_title , studio_name),
    FOREIGN KEY (series_title)
        REFERENCES series (series_title)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (studio_name)
        REFERENCES studio (studio_name)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE characters (
    character_name VARCHAR(50),
    role VARCHAR(50) DEFAULT 'Crew Member',
    series_title VARCHAR(50),
    PRIMARY KEY (character_name , series_title),
    FOREIGN KEY (series_title)
        REFERENCES series (series_title)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE plays (
    actor_name VARCHAR(50),
    character_name VARCHAR(50),
    series_title VARCHAR(50),
    FOREIGN KEY (actor_name)
        REFERENCES actors (actor_name)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (character_name)
        REFERENCES characters (character_name)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (series_title)
        REFERENCES series (series_title)
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE episodes (
    episode_title VARCHAR(50),
    series_title VARCHAR(50),
    air_date DATE NOT NULL,

```

```
    season SMALLINT NOT NULL,  
    description TEXT,  
    url VARCHAR(100),  
    FOREIGN KEY (series_title)  
        REFERENCES series (series_title)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);  
  
CREATE TABLE starring (  
    actor_name VARCHAR(50),  
    series_title VARCHAR(50),  
    FOREIGN KEY (actor_name)  
        REFERENCES actors (actor_name)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (series_title)  
        REFERENCES series (series_title)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```