Movie Database Project

Jordan Chandler (Jordan.chandler@ichandler.net)

Challenge

The objective of this project is to design a relational database that portrays some aspect of your favorite TV show or movie. This database should contain at least 4 tables, 3 primary keys, and 2 foreign keys. Other database artifacts are encouraged if appropriate. Each table should serve a clear distinct purpose that adds value to another table or the database as a whole. Produce an ER diagram and explain how you would populate the database via web scraping from IMDB.

Database Design Approach

My design is centered around my love of movie trilogies or other extended series. My design approach as both the product owner and product manager is to:

1. Define a preliminary elevator pitch given the challenge requirements.
2. Research the field, existing products, and competitive landscape.
3. Identify key sources of inspiration and information.
4. Propose preliminary user stories.
5. Story map the stories.
6. Review the story map and stories with the product owner.
7. Choose as the initial story the story that will provide the most customer value.
8. Design to the initial story.
9. Verify design outcomes with prototypical end users.

# User Stories

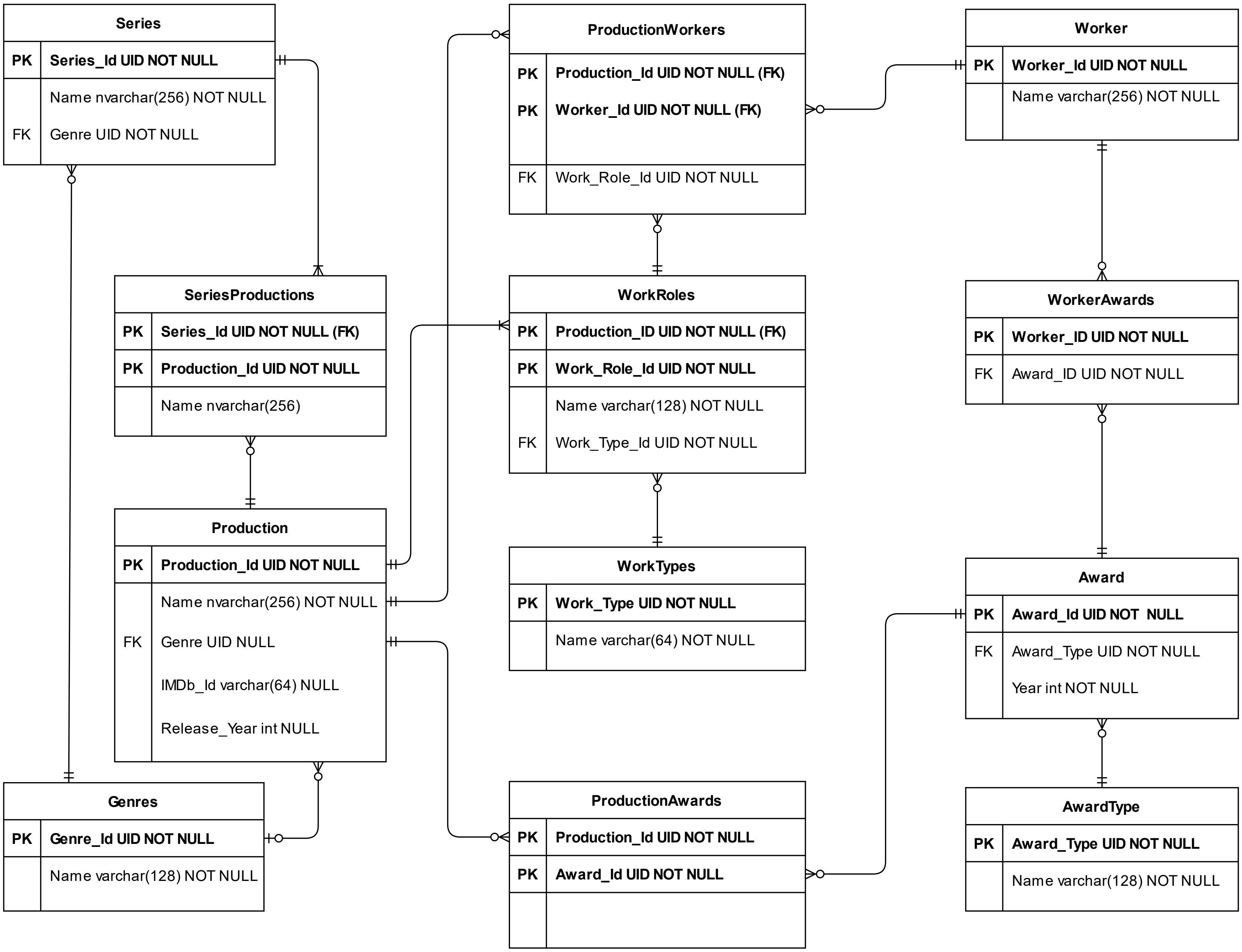
## Story 1

As a fan of movie trilogies and other extended movie series such as the original Star Wars trilogy and later related movies, I want to find movie series that with movies produced within a given date range, of a certain genre, that received an award or certain awards, that have a least a given number of sequels so that I can learn of movie series that match my interests or requirements.

## Story 2

As a fan of movie trilogies and other extended movie series such as the original Star Wars trilogy and other later movies, I want to find information about the series, the series' titles, casts, crews, awards.

Movie Series Database Entity Relationship Diagram



	Series
PK	**Series_Id UID NOT NULL**
	Name nvarchar(256) NOT NULL
FK	Genre UID NOT NULL

	SeriesProductions
PK	**Series_Id UID NOT NULL (FK)**
PK	**Production_Id UID NOT NULL**
	Name nvarchar(256)

	Production
PK	**Production_Id UID NOT NULL**
	Name nvarchar(256) NOT NULL
FK	Genre UID NULL
	IMDb_Id varchar(64) NULL
	Release_Year int NULL

	Genres
PK	**Genre_Id UID NOT NULL**
	Name varchar(128) NOT NULL

	ProductionWorkers
PK	**Production_Id UID NOT NULL (FK)**
PK	**Worker_Id UID NOT NULL (FK)**
FK	Work_Role_Id UID NOT NULL

	WorkRoles
PK	**Production_ID UID NOT NULL (FK)**
PK	**Work_Role_Id UID NOT NULL**
	Name varchar(128) NOT NULL
FK	Work_Type_Id UID NOT NULL

	WorkTypes
PK	**Work_Type UID NOT NULL**
	Name varchar(64) NOT NULL

	ProductionAwards
PK	**Production_Id UID NOT NULL**
PK	**Award_Id UID NOT NULL**

	Worker
PK	**Worker_Id UID NOT NULL**
	Name varchar(256) NOT NULL

	WorkerAwards
PK	**Worker_ID UID NOT NULL**
FK	Award_ID UID NOT NULL

	Award
PK	**Award_Id UID NOT NULL**
FK	Award_Type UID NOT NULL
	Year int NOT NULL

	AwardType
PK	**Award_Type UID NOT NULL**
	Name varchar(128) NOT NULL

Notes: UID is an Identity or GUID column, varchar columns are assumed to support UNICODE.

Database ETL

My proposed steps to populate the Movie Series database with 100 movies, their IMDb IDs, and their release years is as follows:

1) Identify sources of IMDb data available via the Internet. Of course, the primary source is IMDb but we should consider whether there are other sources available which might be easier to access or provide better legal terms.
2) Understand the copyright claims made by the source over the IMDb data. If the data can be legally obtained and used for the intended purpose without any contractual terms with the source, proceed to extract the data as outlined below.
3) If the data, needs to be licensed from the source, do so.
4) Decide the extraction strategy. If possible, extract the data using a source provided API.
5) If not possible, decide which web scraping strategy to use from the list below:
    a. Use a web scraping vendor such as Agenty (https://www.agenty.com), Mozenda (http://www.mozenda.com), etc. to configure and perform web scraping in the cloud.
    b. Use desktop web scraping tools such as:
        i. Web Scraper
        ii. ParseHub
    c. Use a web scraping browser extension such as Apify
    d. Use a web scrapping software library callable from a high-performance language such as C/C++/Rust.
    e. Use a web scrapping software library callable from a high-level language such as Python, Java,
    f. Performing local level web requests that return HTML and parse the data with RegEx, grep, or other generalized text search libraries.
6) The data will be obtain from IMDb and stored in a relation database such as PostGreSQL, MySQL, or SQL Server in the cloud or on premise is required.