Danilo Zelenovic

501032542

A diagram of a company

Description automatically generated

**ENTITY TABLES:**

CREATE TABLE Author (

AuthorID INTEGER PRIMARY KEY AUTOINCREMENT,

Name VARCHAR(255) NOT NULL,

CONSTRAINT ck\_author UNIQUE (Name)

);

CREATE TABLE Novel (

NovelID INTEGER PRIMARY KEY AUTOINCREMENT,

Title VARCHAR(255) NOT NULL,

AgeRating VARCHAR(10),

ReleaseDate DATE,

Status VARCHAR(50),

HighestRating REAL,

LowestRating REAL,

ReadingStatus VARCHAR(50)

);

CREATE TABLE VoiceActor (

VoiceActorID INTEGER PRIMARY KEY AUTOINCREMENT,

Name VARCHAR(255) NOT NULL,

Gender VARCHAR(10)

);

CREATE TABLE Composer (

ComposerID INTEGER PRIMARY KEY AUTOINCREMENT,

Name VARCHAR(255) NOT NULL,

CONSTRAINT ck\_composer UNIQUE (Name)

);

CREATE TABLE Genre (

GenreID INTEGER PRIMARY KEY AUTOINCREMENT,

GenreName VARCHAR(255) NOT NULL,

CONSTRAINT ck\_genre UNIQUE (GenreName)

);

CREATE TABLE Platform (

PlatformID INTEGER PRIMARY KEY AUTOINCREMENT,

PlatformName VARCHAR(255) NOT NULL,

CONSTRAINT ck\_platform UNIQUE (PlatformName)

);

**RELATIONSHIP TABLES:**

CREATE TABLE PublishedBy (

NovelID INTEGER,

AuthorID INTEGER,

FOREIGN KEY (NovelID) REFERENCES Novel(NovelID),

FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID),

PRIMARY KEY (NovelID, AuthorID)

);

CREATE TABLE WrittenBy (

NovelID INTEGER,

AuthorID INTEGER,

FOREIGN KEY (NovelID) REFERENCES Novel(NovelID),

FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID),

PRIMARY KEY (NovelID, AuthorID)

);

CREATE TABLE VoicedBy (

NovelID INTEGER,

VoiceActorID INTEGER,

Gender VARCHAR(10),

FOREIGN KEY (NovelID) REFERENCES Novel(NovelID),

FOREIGN KEY (VoiceActorID) REFERENCES VoiceActor(VoiceActorID),

PRIMARY KEY (NovelID, VoiceActorID)

);

CREATE TABLE ComposedBy (

NovelID INTEGER,

ComposerID INTEGER,

FOREIGN KEY (NovelID) REFERENCES Novel(NovelID),

FOREIGN KEY (ComposerID) REFERENCES Composer(ComposerID),

PRIMARY KEY (NovelID, ComposerID)

);

CREATE TABLE BelongsToGenre (

NovelID INTEGER,

GenreID INTEGER,

FOREIGN KEY (NovelID) REFERENCES Novel(NovelID),

FOREIGN KEY (GenreID) REFERENCES Genre(GenreID),

PRIMARY KEY (NovelID, GenreID)

);

CREATE TABLE AvailableOn (

NovelID INTEGER,

PlatformID INTEGER,

FOREIGN KEY (NovelID) REFERENCES Novel(NovelID),

FOREIGN KEY (PlatformID) REFERENCES Platform(PlatformID),

PRIMARY KEY (NovelID, PlatformID)

);

**Description of each Table:**

Novel Table:

Columns: NovelID (Primary Key), Title, AgeRating, ReleaseDate, Status, HighestRating, LowestRating, ReadingStatus

Description: This table stores information about novels, including their titles, ratings, release dates, and reading statuses. It relates to the ER design by representing the entity "Novel."

Author Table:

Columns: AuthorID (Primary Key), Name

Description: This table stores information about authors of novels. It relates to the ER design by representing the entity "Author."

VoiceActor Table:

Columns: VoiceActorID (Primary Key), Name, Gender

Description: This table stores information about voice actors. It relates to the ER design by representing the entity "VoiceActor."

Composer Table:

Columns: ComposerID (Primary Key), Name

Description: This table stores information about composers. It relates to the ER design by representing the entity "Composer."

Genre Table:

Columns: GenreID (Primary Key), GenreName

Description: This table stores information about genres of novels. It relates to the ER design by representing the entity "Genre."

Platform Table:

Columns: PlatformID (Primary Key), PlatformName

Description: This table stores information about platforms where novels are available. It relates to the ER design by representing the entity "Platform."

**FOR RELATIONSHIPS:**

PublishedBy Table:

Columns: NovelID (Foreign Key), AuthorID (Foreign Key)

Description: This table represents the relationship between novels and authors, indicating which authors have published which novels.

WrittenBy Table:

Columns: NovelID (Foreign Key), AuthorID (Foreign Key)

Description: This table represents the relationship between novels and authors for scenario writing.

VoicedBy Table:

Columns: NovelID (Foreign Key), VoiceActorID (Foreign Key), Gender

Description: This table represents the relationship between novels and voice actors, indicating which voice actors have voiced which novels.

ComposedBy Table:

Columns: NovelID (Foreign Key), ComposerID (Foreign Key)

Description: This table represents the relationship between novels and composers, indicating which composers have composed music for which novels.

BelongsToGenre Table:

Columns: NovelID (Foreign Key), GenreID (Foreign Key)

Description: This table represents the relationship between novels and genres, indicating which genres each novel belongs to.

AvailableOn Table:

Columns: NovelID (Foreign Key), PlatformID (Foreign Key)

Description: This table represents the relationship between novels and platforms, indicating which platforms each novel is available on.