Elena Ajayi

Database Management Systems I

12/05/23

Project Stage 4

Database Design Report: Just Watch Database.

streamingService(streaming_service_id pk,
streaming_service_name, URL)

This table keeps track of the streaming service details in the JustWatch database.

Columns:

- streaming_streaming_service_id (primary key, int, not null): unique identifier for a streaming service
- streaming_service_name (varchar): name of streaming service
- URL (varchar): URL linked to streaming service website, varchar because URLs can be varying lengths.

movies_price_quality(streaming_service_id, movieid, quality,
rent_price, buy_price)

This table tracks the price and quality available for a movie on a particular streaming service.

Columns:

streaming_streaming_service_id (primary key, foreign key, int, not null): unique identifier for a streaming service

movie_id (Primary Key, Foreign Key, int, not null): This primary key uniquely identifies each movie. It also serves as a foreign key, referencing the movie_id in the movies table.

quality (Enum: Free, SD, HD, 4K): An enumeration field specifying the quality of the movie.

Rent_price: A varchar field representing the price for renting the movie

buy_price: A varchar field representing the price for buying the movie·

·

shows_price_quality(streamingserviceid, show_id, quality, rent_price, buy_price)

This table tracks the price and quality available for a TV Show on a particular streaming service.

Columns:

· streaming_streaming_service_id (primary key, foreign key, int, not null): unique identifier for a streaming service

show_id (Primary Key, Foreign Key, int, not null): This primary key uniquely identifies each show. It also serves as a foreign key, referencing the show_id in the shows table.

quality (Enum: Free, SD, HD, 4K): An enumeration field specifying the show's quality.

Rent_price: A varchar field representing the price for renting the show
buy_price: A varchar field representing the price for buying the show

sporting_events_price_quality(streamingserviceid, event_id, quality, rent_price, buy_price)

This table tracks the price and quality available for a sporting event on a particular streaming service.

Columns:

streaming_streaming_service_id (primary key, foreign key, int, not null): unique identifier for a streaming service

event_id (Primary Key, Foreign Key, int, not null): This primary key uniquely identifies each sporting event. It also serves as a foreign key, referencing the event_id in the sporting_event table.

quality (Enum: Free, SD, HD, 4K): An enumeration field specifying the quality of the sporting event.

Rent_price: A varchar field representing the price for renting the sporting_event channel

buy_price: A varchar field representing the price for buying the subscription to a particular sporting channel

user(user_id pk, fname, lname, emailAddress, password, country)

This table keeps track of user details signed up for JustWatch.

Columns:

- user_id (primary key, int, not null): unique identifier for each user
  Fname (varchar) : fname used to sign up and sign in to JustWatch, varchar because names can be varying lengths
  Lname (varchar) : lname used to sign up and sign in to JustWatch can be varying lengths
- email_address (varchar): email address used to sign up and sign in to JustWatch, varchar because emails can be varying lengths.
- password (varchar): password used to sign up and sign in to JustWatch, varchar because emails can be varying lengths.
- country (varchar): country that user lives in, varchar because country names have varying lengths.

streaming_service_streams_Movie(streaming_service_id fk, movie_id fk)

This table keeps track of movies hosted by streaming services.

Columns

- streaming_service_id (foreign key, int, not null): foreign key that references that streaming service that hosts the movie. Integer because it will be a number and not null because both attributes together will make a key.
- movie_id (foreign key, int, not null): foreign key that references the movie being hosted by a streaming

service. Integer because it will be a number and not null because both attributes together will make a key.

streaming_service_streams_shows(streaming_service_id fk, show_id fk)

This table keeps track of tv shows hosted by streaming services.

Columns:

- streaming_service_id (foreign key, int, not null): foreign key that references that streaming service that hosts the tv show. Integer because it will be a number and not null because both attributes together will make a key.
- show_id (foreign key, int, not null): foreign key that references the tv show being hosted by a streaming service. Integer because it will be a number and not null because both attributes together will make a key.

streaming_service_streams_sporting_events(streaming_service_id fk, event_id fk)

This table keeps track of sporting events hosted by streaming services.

Columns:

- streaming_service_id (foreign key, int, not null): foreign key that references that streaming service that hosts the sporting event. Integer because it will be a number and not null because both attributes together will make a key.
- event_id (foreign key, int, not null): foreign key that references the sporting event being hosted by a streaming service. Integer because it will be a number and not null because both attributes together will make a key.

movie(movie_id pk, movie_title, synopsis, release_date, runtime,, age_rating, movie_rating)

This table tracks details of movies stored in the JustWatch database.

Columns:

- movie_id (primary key, int, not null): unique identifier for movies. Integer because number, not null because primary key
- movie_title (varchar): title of movie, varchar because movie titles have varying lengths.
- synopsis (tinytext): synopsis of movie, varchar because synopses have varying lengths.
- release_date(date): release date of movie, date because it is released on a date.
- age_rating (enum): age rating of movie, enum because it can only be values 'g', 'pg', 'pg-13', 'r', 'nc-17'
- movie_rating (varchar): rating given by imdb, varchar because it can be varying lengths.

shows(show_id pk, show_title, synopsis, release_date, runtime, age_rating, movie_rating)

This table tracks details of tv shows stored in Just Watch database.

Columns:

- show_id (primary key, int, not null): unique identifier of tv shows, integer because it will be a number, not null because primary key.
- show_title (varchar): title of tv show, varchar because titles of show have varying lengths.
- release_date (data): release date of tv show, date because it is represented by a date.
- age_rating (enum): age rating of movie, enum because it can only be values 'TV-Y, 'TV-Y7, 'TV-Y7 FV, 'TV-G, 'TV-PG', 'TV-14', 'TV-MA'

synopsis (tinytext): synopsis of movie, varchar because synopses have varying lengths.

- movie_rating (varchar): rating given by imdb, varchar because it can be varying lengths.

genre_movie(movie_id fk, genre)

This table tracks the genre of the movie.

Columns:

- movie_id (foreign key, int, not null): foreign key that references the movie that is being given a genre.
- genre (varchar): genre of movie, varchar because genre's can be varying lengths.

genre_show(show_id fk, genre)

This table tracks the genre of the tv show.

Columns:

- show_id (foreign key, int, not null): foreign key that references the shows that is being given a genre.
- genre (varchar): genre of shows, varchar because genre's can be varying lengths.

movie_language_subtitle (movie_id fk, language_subtitle)

This table tracks the subtitle languages of movies.

Columns:

- movie_id (foreign key, int, not null): foreign key that references the movie_id in movie table, int because number, not null it is a key.
- language_subtitle(varchar): This is the subtitle title language of the movie being referenced by the movie_id fk.

shows_language_subtitle(show_id fk, language_subtitle)

This table tracks the subtitle languages of tv shows.

Columns:

- show_id (foreign key, int, not null): foreign key that references the show_id in shows table, int because number, not null it is a key.
- language_subtitle(varchar): This is the subtitle title language of the show being referenced by the movie_id fk.

This table tracks the subtitle languages of episodes.

Columns:

- episode_id (foreign key, int, not null): foreign key that references the episode_id in shows table, int because number, not null it is a key.
- language_subtitle(varchar): This is the subtitle title language of the episode being referenced by the episode_id fk.

movie_language_audio(movie_id fk, language_audio)

This table tracks the audio languages of movies.

Columns:

- movie_id (foreign key, int, not null): foreign key that references the movie_id in movie table, int because number, not null it is a key.
- language_audio(varchar): This is the audio language of the movie being referenced by the movie_id fk.

shows_language_audio(show_id fk, language_audio)

This table tracks the audio languages of tv shows.

Columns:

- show_id (foreign key, int, not null): foreign key that references the show_id in shows table, int because number, not null it is a key.
- language_audio(varchar): This is the audio language of the show being referenced by the show_id fk.

episodes_language_subtitle(episode_id fk, language_subtitle)

This table tracks the audio languages of episodes.

Columns:

- episode_id (foreign key, int, not null): foreign key that references the episode_id in shows table, int because number, not null it is a key.
- language_audio(varchar): This is the subtitle title language of the episode being referenced by the episode_id fk.

movie_country_produced(movie_id fk, country)

This table tracks the production country of movies.

Columns:

- show_id (foreign key, int, not null): foreign key that references the movie_id in the movie table, int because number, not null it is a key.
- country(varchar): Country where the movie being referenced in movie_id is produced.

show_country_produced(season_id fk, country)

This table tracks the production country of tv shows.

Columns:

- show_id (foreign key, int, not null): foreign key that references the show_id in shows table, int because number, not null it is a key.
- country(varchar): Country where the tv show being referenced in show_id is produced.

Seasons(season_id pk, show_id fk, show_title, season_no, synopsis, runtime, release_date)

This table tracks details of seasons in the JustWatch database.

Columns:

- season_id (primary key, int, not null): identifier for season, int because it will be a number, not null because primary key.
- show_id (primary key, varchar, not null): identifier for the show_id primary key from the tvShoe table that references the tv show the season is from
- show_title (varchar): title of season, varchar because titles come in varying lengths.
- season_no (int): Season number, integer because it will be a number.
- Synopsis(tinytext): synopsis of season, varchar because synopses come in varying lengths.
- runtime (int): runtime of episodes in season, but one runtime will be shown for a season, int because it will be a whole number.
- release_date (date): release date of season, date because it represents a date.
-

Episode(episode_id pk, season_id fk, episode_name, episode_description, release_date, episode_no)

This table tracks details of episodes in the JustWatch database.

Columns:

- episode_id (primary key, int, not null): identifier of episode
- season_id (foreign key, varchar, not null): identifier of season_id from season table that references the season the episode is from.
- episode_name (varchar): name of episode
- Episode_description (varchar): description of episode
- release_date (date): release date of episode
- episode_no (int): number of episodes

sporting_event(event_id pk, event_name, event_description, competition_name, competition, airdate, venue, start_time, end_time)

This table tracks details of sporting events in JustWatch database.

Columns:

- event_id (primary key, int, not null): unique identifier of sporting event
- event_name (varchar): name of event
- event_description (varchar): description of event
- competition (varchar): name of competition
- airdate(date): airdate of event
- venue (varchar): venue of event

Start_time (time): start time of the event

End_time (time): end time of the event

Sporting_event_participants(event_id fk, participant_name)

This table keeps track of the details of the sporting_event participants.

Columns:

- event_id (Primary Key (1/2), Foreign Key, Integer, Not Null): This field serves as the initial part of the primary key, uniquely identifying each sporting event. It also acts as a foreign key, linking back to the event_id in the sporting_event table. As an integer, it captures the numerical essence of the identifier and is required (not null) to adhere to primary key constraints.

- Participants_name (Primary Key (2/2), Variable Character String, Not Null): This column represents the latter part of the primary key, uniquely identifying each participant associated with a sporting event. It employs a variable character string, accommodating names of different lengths. The not null constraint ensures its participation in upholding the integrity of the primary key.

cast_member(cast_id pk, fname, lname, birthday, bio)

This table keeps details of the cast.

Columns:

· cast_id (primary key, int, not null): unique identifier of cast member
· fname (varchar): first name of cast member
· lname (varchar): last name of cast member
· birthday (date): birthday of cast member
· bio (varchar): bio of cast member

cast_Stars_in_Movie(cast_id fk, movie_id fk):

This table keeps track of cast in movies.

Columns:

- cast_id (Foreign Key, INT, Not Null): Cast ID being referenced.
- movie_id (Foreign Key, INT, Not Null): Movie ID being referenced.
- PRIMARY KEY (cast_id, movie_id)

cast_Stars_in_Movie_role(cast_id fk, movie_id fk, role):

This table keeps track of cast role in movies.

Columns:

- cast_id (Foreign Key, INT, Not Null): Cast ID being referenced.
- movie_id (Foreign Key, INT, Not Null): Movie ID being referenced.
- role (VARCHAR, Not Null): Role of the cast member in the movie.
- PRIMARY KEY (cast_id, movie_id, role)

cast_Stars_in_season (cast_id fk, movie_id fk):

This table keeps track of cast in shows.

Columns:

- cast_id (Foreign Key, INT, Not Null): Cast ID being referenced.
- show_id (Foreign Key, INT, Not Null): Show ID being referenced.
- PRIMARY KEY (cast_id, show_id)

cast_Stars_in_season_role (cast_id fk, show_id fk, role):

This table keeps track of cast roles in shows.

Columns:

- cast_id (Foreign Key, INT, Not Null): Cast ID being referenced.
- show_id (Foreign Key, INT, Not Null): Show ID being referenced.
- role (VARCHAR, Not Null): Role of the cast member in the show.
- PRIMARY KEY (cast_id, show_id, role)

Updates made to Tables:

In the price quality tables, instead of having columns for each stream quality (bad idea, what happens when the service adds 5K?) instead, have a quality attribute with enum values SD, HD, 4K. Do the same for purchase type. Then all are in the key (streamingserviceid, movieid, quality, purchasetype)

- Updated and reflected in the report, Create statements, and inserts statements. Quality was converted to ENUM, yielding free, sd, hd, and 4k. These updates were made particularly for the following tables: movie_price_quality, shows_price_quality, and sporting_event_price_quality.

-keys

Many of your foreign keys are not showing up in the exported tables. It is because you are using incorrect syntax in those cases.

i.e. "episode_id INT REFERENCES episodes(episode_id).” MySQL is ignoring this. You did end up with foreign keys in some of the tables, like shows_language_audio, because the correct SQL is in the script: FOREIGN KEY (show_id) REFERENCES shows(show_id));

- Syntax was fixed, and all Foreign keys should be showing in the exported tables.

has enough data to test 10. looks ok. In the last stage, you may need to add some data here and there to demonstrate that specific queries work.

- Additional data was added. Therefore, each insert statement now contains 6 data entries to put in tables. In addition, I Corrected all my int(n) issues for correct SQL syntax. Moreover, I also added a Primary Key statement to tables with foreign keys so that the table has a primary key, and entries can't be put in with empty/null values.