Schema Overview:

- User: Stores user information such as user_id, name, email, phone_no, and subscription_status.

- Content: Stores content information such as content_id, content_title, and content_type.

- Movie: Stores movie-specific information including content_id, title, description, release_date, duration, director, production, imdb_rating, pg_rating, country, poster_link, movie_link, and trailer_link.

- SubscriptionPlan: Stores subscription plan details such as plan_id, plan_name, duration, and price.

- WebSeries: Stores web series-specific information including content_id, title, description, release_date, director, cast, language, trailer_link, poster_link, and number_of_episodes.

- Seasons: Stores season information including season_id, content_id, and season_number.

- Episodes: Stores episode-specific information including episode_id, season_id, title, description, release_date, duration, thumbnail, and imdb_rating.

- Subscription: Stores subscription details including subscription_id, user_id, plan_id, start_date, and end_date.

- Watchlist: Stores user watchlist information including watchlist_id, user_id, content_id, episode_id, and added_date.

- Playback: Stores playback information including playback_id, user_id, content_id, season_id, episode_id, and playback_position.

- Rating: Stores user ratings and reviews including rating_id, user_id, content_id, episode_id, rating, and review.

- Genre: Stores genre information including genre_id and genre_name.

- Casts: Stores cast information including cast_id, actor_name, and content_id.

- ContentGenre: Stores the relationship between content and genre with content_id and genre_id.

Optimization Techniques Used:

1. Primary Keys: Proper primary keys are defined for each table to ensure uniqueness and improve indexing performance.

2. Foreign Keys: Appropriate foreign keys are established to maintain referential integrity and enforce relationships between tables.

3. Data Types: Optimized data types are used for columns to minimize storage requirements and improve query performance.

4. Indexing: Indexes are created on key columns and attributes that are frequently used in search and join operations. Indexes improve query performance by allowing faster data retrieval.

5. Compound Indexes: Compound indexes are created on multiple columns to enhance query performance for specific queries involving multiple attributes.

6. Partitioning: Partitioning is applied to large tables such as Episodes and Content based on relevant attributes. It helps distribute data across multiple filegroups or storage devices, improving query performance and manageability.