

MOVIE STREAMING DATABASE

Ivan Stakhov



TOPIC

- This is a PostgreSQL relational database designed to model a real-world movie streaming platform like Netflix / Hulu / Disney+.
- This topic is medium difficulty, fits all course requirements, and is not taken by anyone else in your list.



REQUIREMENTS & USE CASES

- Manage users, subscription plans, movies, genres, watch history, and ratings
- Ensure data integrity via primary keys, foreign keys, and constraints
- Support queries for trending movies, popular genres, active users, churn risk
- Provide a normalized schema and ER-diagram, SQL scripts, transactions, and backup strategy

DATABASE DESIGN

01

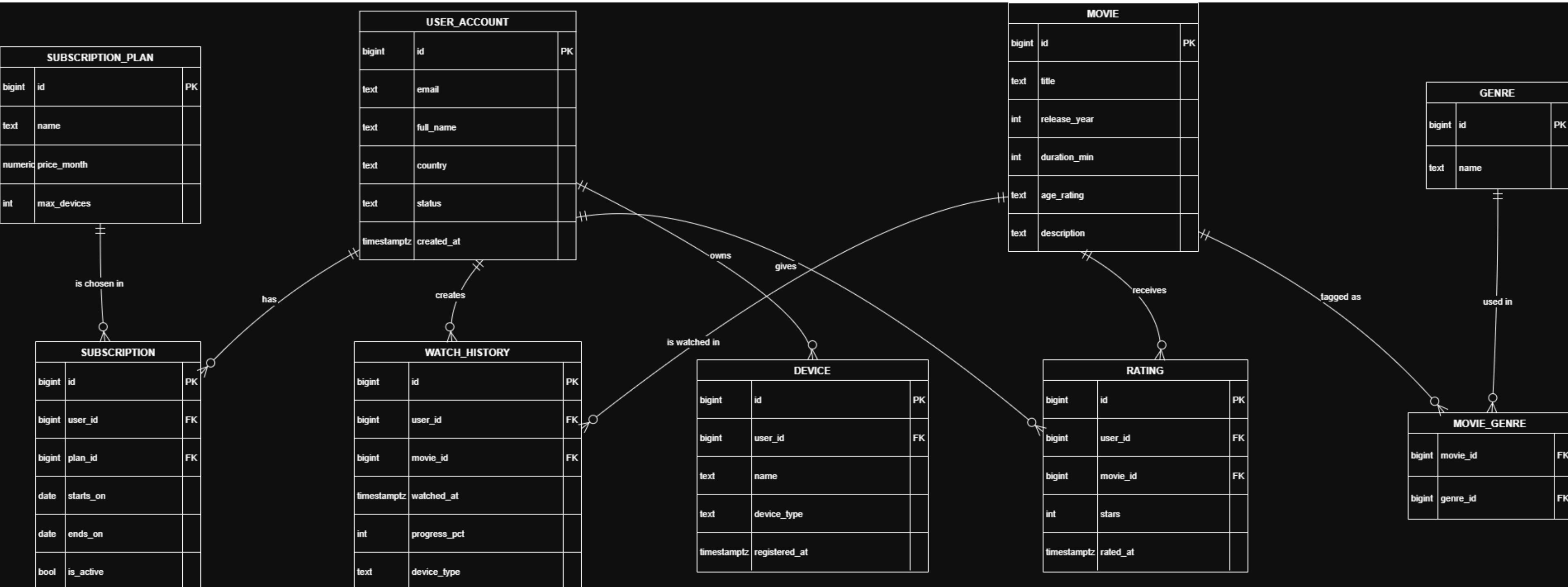
Tables: user_account, subscription_plan, subscription, movie, genre, movie_genre, watch_history, rating, device

Primary keys uniquely identify rows; foreign keys enforce relationships

02

Normalization to at least 3NF; separate tables handle many-to-many (movie–genre)

See ER-diagram (docs/erd.png) for relationships and cardinalities



OVERVIEW

01_schema_creation.sql: defines tables, constraints, and indexes

02_data_insertion.sql: populates tables with sample data

03_logic_and_views.sql: function, trigger, and reporting views

04_queries_basic.sql / 05_queries_advanced.sql: example queries

06_transactions_and_indexes.sql: transactions and index usage



RESULT

01

- Basic queries: SELECT, INSERT, UPDATE, DELETE, joins
- Advanced queries: GROUP BY, CTEs (WITH), window functions (RANK, ROW_NUMBER)
- Analytical reports: trending movies, popular genres, active users, average ratings

02

- Demonstrate ACID properties: BEGIN, COMMIT, ROLLBACK, SAVEPOINT
 - Example transaction: upgrade a user subscription safely
 - Example SAVEPOINT scenario: insert multiple watch events and undo one

03

- Indexes on watch_history (user/time), subscription (user/active), rating (movie)
- Use EXPLAIN ANALYZE to show index scans and performance impact
- Queries target normalization and performance using indexes and views

QUESTION TIME

THANK YOU

For your attention