

Projet SQL

L'Évolution de Netflix - De la Location DVD au Streaming



Contexte du Projet

Histoire

Au début des années 2000, Netflix était une entreprise de location de DVD par correspondance. La base de données que vous utilisez (DVD Rental) représente ce type d'activité. Aujourd'hui, Netflix est devenu un géant du streaming avec des millions d'utilisateurs dans le monde entier.

Votre Mission

En groupe de 4 à 6 étudiants, vous allez imaginer et concevoir **une nouvelle fonctionnalité** qui pourrait transformer cette base de données de location DVD vers un modèle de streaming moderne. Chaque amélioration doit être supportée par du code SQL fonctionnel.

Objectifs Pédagogiques

- Appliquer les connaissances en conception de bases de données relationnelles
- Maîtriser les concepts de base de SQL (SELECT, FROM, JOIN, WHERE, SUM, MIN, MAX, AVG, GROUP BY, HAVING)
- Créer des requêtes pour générer des rapports métiers

Idées de Fonctionnalités à Développer

Ci-dessous sont listées quelques idées de fonctionnalités que votre équipe pourrait développer. Cette liste est loin d'être exhaustive. Vous pouvez bien sûr travailler sur une idée qui n'est pas listée.

Chaque groupe doit sélectionner une seule fonctionnalité.

1. Système de Streaming et Visionnage en Ligne

- Ajouter la notion de "streaming" vs "location physique"
- Tracker le temps de visionnage par utilisateur
- Gérer les reprises de lecture (bookmarks)
- Historique de visionnage complet

Tables suggérées :

- `streaming_sessions` (session_id, customer_id, film_id, start_time, end_time, position)
- `viewing_history` (history_id, customer_id, film_id, watched_date, completion_percentage)

2. Système d'Abonnements à plusieurs niveaux

- Créer différents types d'abonnements (Basic, Standard, Premium)
- Gérer les limitations par abonnement (nombre d'écrans, qualité vidéo)
- Système de facturation récurrente
- Historique des abonnements

Tables suggérées :

- `subscription_plans` (plan_id, plan_name, price, max_screens, video_quality, simultaneous_streams)
- `customer_subscriptions` (subscription_id, customer_id, plan_id, start_date, end_date, status)
- `subscription_history` (history_id, customer_id, plan_id, change_date, action_type)

3. Système de Recommandations

- Analyser l'historique de visionnage pour recommander des films
- Créer des catégories de préférences par utilisateur
- Système de notation et d'avis
- Films "tendance" et "nouveautés"

Tables suggérées :

- `user_preferences` (preference_id, customer_id, category_id, preference_score)
- `film_ratings` (rating_id, customer_id, film_id, rating, review_text, rating_date)
- `recommendations` (recommendation_id, customer_id, film_id, recommendation_score, reason)

4. Contenu Original et Séries TV

- Ajouter des séries TV avec saisons et épisodes
- Gérer le contenu "Netflix Original"
- Tracker le visionnage par épisode
- Notifications de nouveaux épisodes

Tables suggérées :

- `tv_series` (series_id, title, description, release_year, status)
- `seasons` (season_id, series_id, season_number, episode_count)
- `episodes` (episode_id, season_id, episode_number, title, duration, release_date)
- `episode_views` (view_id, customer_id, episode_id, view_date, completion_status)

5. Profils Utilisateurs Multiples

- Plusieurs profils par compte (famille)
- Préférences par profil (contrôle parental, langues)
- Historique de visionnage séparé
- Recommandations personnalisées par profil

Tables suggérées :

- `user_profiles` (profile_id, customer_id, profile_name, is_kids_profile, avatar)
- `profile_preferences` (preference_id, profile_id, language, maturity_rating)
- `profile_viewing_history` (history_id, profile_id, film_id, view_date)

6. Système de Téléchargement Offline

- Gérer les téléchargements pour visionnage hors ligne
- Limites de téléchargements selon l'abonnement
- Date d'expiration des téléchargements
- Gestion de l'espace de stockage

Tables suggérées :

- `downloads` (download_id, profile_id, film_id, download_date, expiration_date, status)
- `download_limits` (limit_id, plan_id, max_downloads, retention_days)

7. Analytics et Métriques de Performance

- Tableau de bord des films les plus regardés
- Statistiques de visionnage par période
- Taux de rétention des clients
- Revenus et KPIs financiers

Tables suggérées :

- `viewing_metrics` (metric_id, film_id, total_views, total_hours, avg_completion_rate, period)
- `customer_metrics` (metric_id, customer_id, subscription_months, total_spent, churn_risk)

8. Système de Contenu Multilingue et Sous-titres

- Gérer les versions doublées et sous-titrées
- Préférences de langues par profil
- Accessibilité (sous-titres pour malentendants)

Tables suggérées :

- `audio_tracks` (track_id, film_id, language, audio_type)
- `subtitles` (subtitle_id, film_id, language, subtitle_type)
- `profile_language_preferences` (preference_id, profile_id, preferred_audio, preferred_subtitle)

Livrables Attendus

1. Un document unique à livrer par Groupe

Chaque groupe devra livrer un document unique, qui servira aussi bien de support pendant la présentation que de rapport final. Ce document est décrit ci-dessous.

Le document devra être **livré au plus tard le 12 janvier 2026 à 9h30** sur la plateforme Learn.

2. Présentation (15 minutes)

Structure suggérée :

- **Introduction** (2 min) : Présentation de la fonctionnalité choisie et contexte métier
- **Modélisation** (3 min) : Schéma des nouvelles tables et relations (diagramme ERD)
- **Démonstration SQL** (8 min) :
 - Chaque membre présente son code SQL (2 min/personne pour 4 membres)
 - Montrer les requêtes en action avec des résultats
- **Conclusion** (2 min) : Bénéfices métier et perspectives d'amélioration

Support visuel :

- Un document PDF, créé à partir de l'outil de présentation de votre choix (PowerPoint, Canva, Google Slides, ou autre)
- Le document comprendra 2 parties
 - La 1^{ère} partie comprendra :
 - Entre 5 et 10 pages
 - Une page avec les noms et photos de tous les élèves du groupe
 - La description de la fonctionnalité développée
 - Le diagramme de la base de données (ERD)
 - La structure des nouvelles tables
 - Les bénéfices métier et perspectives d'amélioration
 - La 2^{nde} partie comprendra :
 - 1 page par table créée
 - Chaque groupe devra créer entre **4 tables minimum et 10 tables maximum**
 - La structure de chaque table doit être décrite : liste des champs avec leur nom et type, indication si clef primaire
 - Un screenshot (impression d'écran) du contenu de la table est attendu, avec au moins 10 lignes créées par table. Les données

- doivent être cohérentes et suffisantes pour démontrer la fonctionnalité développée
- 1 page par requête SQL
 - Chaque groupe devra créer **1 requête « CREATE TABLE » par table créée**, soit entre 4 et 10 requêtes « CREATE TABLE »
 - Chaque groupe devra créer entre **10 et 15 requêtes « SELECT »**
 - Parmis les 10 à 15 requêtes « SELECT », au moins 4 devront utiliser des « **JOIN** » et au moins 3 supplémentaires utiliser des « **GROUP BY** ». Les requêtes peuvent bien entendu combiner des « JOIN » et des « GROUP BY »
 - Toutes les requêtes SQL créées (« CREATE TABLE » et « SELECT ») seront présentes avec leur code complet, leur nom et leur description
 - Un screenshot (impression d'écran) du résultat d'exécution de la requête est attendu

3. Code SQL

Chaque étudiant doit contribuer sur au moins une des composantes suivantes :

A. Requêtes « *CREATE TABLE* »

Exemples :

CREATE TABLE subscription_plans (...);

CREATE TABLE customer_subscriptions (...);

B. Requêtes « *SELECT* »

Requêtes « *SELECT* » pour exploiter la fonctionnalité

- Top 10 des films les plus regardés ce mois-ci
- Liste des clients à risque de churn
- Calcul du revenu par type d'abonnement

 Critères d'Évaluation

Critère	Points	Description
Pertinence de la fonctionnalité	10%	Innovation, utilité métier, réalisme
Qualité du modèle de données	25%	Normalisation, relations, contraintes d'intégrité
Qualité du code SQL	30%	Syntaxe, bonnes pratiques
Présentation orale	15%	Clarté, structure, timing
Documentation	10%	Complétude, clarté, schémas
Travail d'équipe	10%	Répartition équitable, collaboration

Bonus (jusqu'à 10%) :

- Vidéo de démonstration
- Tests de performance et optimisation avancée
- Créativité exceptionnelle, que ce soit sur la présentation, sur la documentation ou sur la solution technique

Ressources et Outils

Notions SQL

- Notions SQL vues en cours : <https://www.w3schools.com/sql/default.asp>

Structure initiale de la base de données DVD Rental

- Sample Database : <https://neon.com/postgresql/postgresql-getting-started/postgresql-sample-database>

PostgreSQL Sample Database Tables

There are 15 tables in the DVD Rental database:

- actor – stores actor data including first name and last name.
- film – stores film data such as title, release year, length, rating, etc.
- film_actor – stores the relationships between films and actors.
- category – stores film's categories data.
- film_category- stores the relationships between films and categories.
- store – contains the store data including manager staff and address.
- inventory – stores inventory data.
- rental – stores rental data.
- payment – stores customer's payments.
- staff – stores staff data.
- customer – stores customer data.
- address – stores address data for staff and customers
- city – stores city names.
- country – stores country names.

Documentation PostgreSQL

- PostgreSQL Documentation Officielle : <https://www.postgresql.org/docs/>
- PostgreSQL Tutorial : <https://www.postgresqltutorial.com/>

Outils de Modélisation

- dbdiagram.io : Création de diagrammes ERD en ligne
- draw.io : Diagrammes gratuits
- pgAdmin : Interface graphique PostgreSQL

Bonnes Pratiques SQL

- Nommer les tables au pluriel (ex: `subscriptions`)
- Utiliser snake_case pour les noms de colonnes
- Toujours définir une clé primaire
- Ajouter des commentaires pour toutes les requêtes (complexes ou non)
- Utiliser des transactions pour les opérations critiques

Exemple de Nomenclature

```
-- Tables : nom_au_pluriel  
CREATE TABLE subscription_plans (...);  
  
-- Colonnes : snake_case  
customer_id, first_name, created_at
```

FAQ

Q: Doit-on nécessairement utiliser PostgreSQL ?

R: Non, vous êtes libres du choix de la base de données.

Q: Peut-on modifier les tables existantes de la base DVD Rental ?

R: Oui, vous pouvez même partir d'une toute nouvelle base de données. Restez cependant sur le thème d'une base de données d'une société de location de DVD qui souhaite évoluer vers le streaming.

Q: Doit-on implémenter l'interface utilisateur ?

R: Non, ce n'est pas obligatoire. Le focus est sur la base de données et SQL.

Q: Peut-on s'inspirer d'autres plateformes que Netflix ?

R: Oui ! Disney+, Amazon Prime Video, HBO Max sont d'excellentes sources d'inspiration.

Q: Peut-on utiliser d'autres langages ou technologies que le SQL ?

R: Le SQL est obligatoire mais vous pouvez très bien enrichir votre solution avec d'autres langages ou technologies (IA, Python, PowerBI, etc).

Conseils pour Réussir

1. **Commencez simple** : Une fonctionnalité bien implémentée vaut mieux que plusieurs mal faites
2. **Testez régulièrement** : Validez chaque requête au fur et à mesure
3. **Documentez votre code** : Commentez vos requêtes SQL
4. **Répartissez le travail** : Chaque membre doit avoir une responsabilité claire
5. **Préparez votre présentation** : Répétez avant le jour J