Assignment Description

Spotify is a leading global audio streaming company that serves millions of people worldwide with music, podcasts, and other audio material. Spotify, which has a large archive of music and a rising user base, is looking to improve its data management skills by developing a comprehensive and scalable database system.

Spotify wants to design and develop a robust and efficient database system to manage the extensive collection of music tracks, user, playlists, and other related metadata.

ERD

Based on the Use case below, Develop an Entity Relationship Diagram showing all entities, relationships, attributes, and keys and making several components generic so that the system can easily be expanded.

Use Case

Users: This table stores information about registered users of the platform, like username, email, password, and registered date e.t.c.

Songs: This table contains information about individual songs available on the platform, title, artist, duration, release date and genre

Album: This table stores information about albums released by artists, title, artist, release date, genre

Artists: This table holds details about music artists name, biography, origin, etc.

Playlists: This table represents user-created playlists title creation date, etc.

Playlist Songs: This table establishes the relationship between playlists and songs, indicating the songs included in each playlist, it also includes position to maintain the order of the song in the playlist.

User Likes: This table tracks the songs that users have liked and the liked date.

DBMS

Install Microsoft SQL Server as database provider and Microsoft Management Studio to access and manage the database. The database should be created by means of DDL queries and add the correct users, roles, and rights by means of DCL queries. Deliver both DCL and DDL queries in a .sql-file

Users, roles, and rights

The data analysts at Spotify are interested in the information that the data in the database provides, but they need to have access. Add the following users along with their corresponding roles and rights.

Name	Rights
Admin	Admin account
John Paul	Chief Data Analyst: can only perform DML queries
James Brown	Junior Analyst: Can perform DML queries with exception of User History and PlayList

SQL queries.

Create the following DML queries to extract important information from the database. Test data must be added to obtain information.

Information needs
The data analysts at Spotify are interested in the information that the data in the database provides. Pay attention to efficiency: JOINS and SUBQUERY'S, EXISTS (), and IN(), etc. Develop the following information needs as DML queries.

Information Needed

1. What is the total number of songs liked by each user.
2. List the usernames of users who have never liked any song.
3. List the usernames of users who have liked all songs in a specific playlist.
4. Retrieve the top 5 most liked songs and their corresponding like counts.
5. Which songs appear in more than one playlist.
6. Get the total number of songs liked by each user.
7. Which users have liked songs from the "Pop" genre and have listened to any album released by the artist named "Ed Sheeran".

Integrity

Primary Keys, Foreign Keys and Index
Provide the databases with primary and foreign keys. Pay attention to relationships and the referential integrity. Also provide the columns with an index wherever it is logical to do so.

Constraints
Provide the database with a few constraints. Implement NOT NULL, UNIQUE, and CHECK for columns where it is logical to do so at least once.

Tip: CASCADE and NO ACTION are not the only actions that can follow a constraint.

Views, Stored Procedures and Triggers

Realise two views, two stored procedures, and two triggers. Use views, stored procedures, and triggers for the **right** purposes.

Referential integrity: Implement referential integrity for all foreign key constraints for a logical use case at least once.

Database Recovery

Backup
Create a full, differential, and incremental backup and transaction log for the Spotify database by means of CREATE BACKUP statements.

Recovery
Use the RESTORE statement to restore the database with a full, differential, and incremental backup and transaction log. Implement a transaction for a logical use case.

Concurrency
Reduce the chance of concurrency problems to a minimum by means of a query, for instance with phantom reads and deadlocks. In addition, create two transactions for logical use cases.

Develop a program in C# that tests the speeds of ADO.NET and NoSQL by means of a Stopwatch. Make sure that all tests are combined in a single solution. Select one entity on which to perform CRUD operations. Do this for rows of 1, 1000, 100,000, and 1,000,000. Make sure that all tests within the various DBMSs are the same.

You must use MongoDB for the NoSQL test. To set up the MongoDB database, you do not need to redesign and implement the entire database, but instead select a use case comprising at least 3 tables.

Next, generate a report in PDF format. This is not an official document and does not need to contain a cover page, table of contents, etc. Make sure that the report meets the following requirements:

- Presents the speeds in a matrix (table) and in graphs
- For each comparison an accompanying explanatory text to explain results that are worthy of note.
- Description of measures taken to demonstrate that the information is reliable, along with, at any rate, a description of the specifications of the device on which the tests were run.