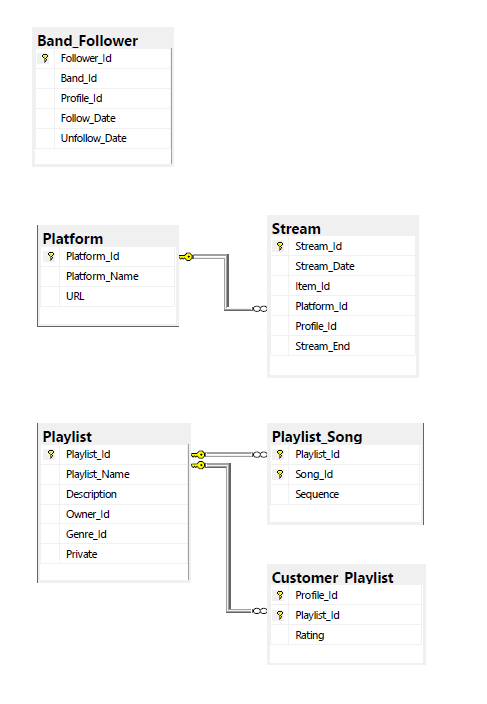
**Hallux Schema Additions**

The purpose of the Hallux schema changes is to serve as a sample solution to the following business requirement.

Hallux would like to expand their business to support social media and streaming.

The proposed schema changes shown below are designed to support the new business requirements. The first addition concerns “fans” of a band. The Band\_Follower table would store information about a band’s followers. The addition of the platform and stream tables will allow for the storing of information about the streaming of media. Lastly, the addition of the playlist set of tables facilitates the creation of playlists that can be shared and rated among Hallux users.



Appendix 1 – Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Column** | **Data Type** | **Description** |
| Platform | Platform\_Id | Integer | Unique Id |
| Platform | Platform\_Name | Varchar(30) | Streaming platform name |
| Platform | URL | Varchar(1000) | Link to platform |
|  |  |  |  |
| Stream | Stream\_Id | Integer | Unique Id |
| Stream | Item\_Id | Integer | Item streamed |
| Stream | Platform\_Id | Integer | Platform from which content was streamed |
| Stream | Stream\_Date | Datetime | Date content was streamed |
| Stream | Profile\_Id | Integer | Profile\_Id that streamed the content |
| Stream | Stream\_End | Datetime | Date / Time streaming ended |
|  |  |  |  |
| Playlist | Playlist\_Id | Integer | Unique Id |
| Playlist | Playlist\_Name | Varchar(30) | Display name of the playlist |
| Playlist | Description | Varchar(100) | Description of the playlist |
| Playlist | Owner\_Id | Integer | Profile\_Id that created / owns the playlist |
| Playlist | Genre\_Id | Integer | Genre of the playlist |
| Playlist | Private | Binary | Can the playlist be shared with other profiles? |
|  |  |  |  |
| Playlist\_Song | Playlist\_Id | Integer | Unique Id |
| Playlist\_Song | Song\_Id | Integer | Song |
| Playlist\_Song | Sequence | tinyint | Order in which to play the songs |
|  |  |  |  |
| Customer\_Playlist | Profile\_Id | Integer | Unique Id |
| Customer\_Playlist | Playlist\_Id | Integer | Playlist |
| Customer\_Playlist | Rating | tinyint | 1-10 rating of the playlist |
|  |  |  |  |
| Band\_Follower | Follower\_Id | Integer | Unique Id |
| Band\_Follower | Band\_Id | Integer | Band |
| Band\_Follower | Profile\_Id | Integer | Profile |
| Band\_Follower | Follow\_Date | Datetime | Date started following |
| Band\_Follower | Unfollow\_Date | Datetime | Date stopped following |
|  |  |  |  |

Appendix 2 – Data Definition Language

CREATE TABLE Platform

(

Platform\_Id int NOT NULL,

Platform\_Name varchar(30) NOT NULL,

URL varchar(1000) NOT NULL,

CONSTRAINT pk\_Platform PRIMARY KEY CLUSTERED (Platform\_Id)

)

go

CREATE UNIQUE INDEX idx\_Platform\_Name ON Platform (Platform\_Name)

go

CREATE TABLE Stream

(

Stream\_Id int NOT NULL,

Item\_Id int NOT NULL,

Platform\_Id int NOT NULL,

Profile\_Id int NULL,

Stream\_Date Datetime NOT NULL,

Stream\_End Datetime NULL,

CONSTRAINT pk\_Stream PRIMARY KEY CLUSTERED (Item\_Id, Platform\_Id, Stream\_Date)

)

Go

ALTER TABLE Stream ADD CONSTRAINT fk\_Stream\_Item FOREIGN KEY (Item\_Id) REFERENCES Item (Item\_Id)

go

ALTER TABLE Stream ADD CONSTRAINT fk\_Stream\_Platform FOREIGN KEY (Platform\_Id) REFERENCES Platform (Platform\_Id)

Go

ALTER TABLE Stream ADD CONSTRAINT fk\_Stream\_Profile FOREIGN KEY (Profile\_Id) REFERENCES Customer\_Profile (Profile\_Id)

Go

CREATE TABLE Playlist

(

Playlist\_Id int NOT NULL,

Playlist\_Name varchar(30) NOT NULL,

Description varchar(100) NULL,

Owner\_Id int NULL,

Genre\_Id int NULL,

Private binary NULL,

CONSTRAINT pk\_Playlist PRIMARY KEY CLUSTERED (Playlist\_Id)

)

Go

CREATE UNIQUE INDEX idx\_Playlist\_Name ON Playlist (Playlist\_Name)

go

ALTER TABLE Playlist ADD CONSTRAINT fk\_Playlist\_Genre FOREIGN KEY (Genre\_Id) REFERENCES Genre (Genre\_Id)

go

ALTER TABLE Playlist ADD CONSTRAINT fk\_Playlist\_Owner FOREIGN KEY (Owner\_Id) REFERENCES Customer\_Profile (Profile\_Id)

go

CREATE TABLE Playlist\_Song

(

Playlist\_Id int NOT NULL,

Song\_Id int NOT NULL,

Sequence tinyint NULL,

CONSTRAINT pk\_Playlist\_Song PRIMARY KEY CLUSTERED (Playlist\_Id, Song\_Id)

)

go

ALTER TABLE Playlist\_Song ADD CONSTRAINT fk\_Playlist\_Song\_Song FOREIGN KEY (Song\_Id) REFERENCES Song (Song\_Id)

go

ALTER TABLE Playlist\_Song ADD CONSTRAINT fk\_Playlist\_Song\_Playlist FOREIGN KEY (Playlist\_Id) REFERENCES Playlist (Playlist\_Id)

go

CREATE TABLE Customer\_Playlist

(

Profile\_Id int NOT NULL,

Playlist\_Id int NOT NULL,

Rating tinyint NULL,

CONSTRAINT pk\_Customer\_Playlist PRIMARY KEY CLUSTERED (Profile\_Id, Playlist\_Id)

)

go

ALTER TABLE Customer\_Playlist ADD CONSTRAINT fk\_Customer\_Playlist\_Profile FOREIGN KEY (Profile\_Id) REFERENCES Customer\_Profile (Profile\_Id)

go

ALTER TABLE Customer\_Playlist ADD CONSTRAINT fk\_Customer\_Playlist\_Playlist FOREIGN KEY (Playlist\_Id) REFERENCES Playlist (Playlist\_Id)

Go

CREATE TABLE Band\_Follower

(

Follower\_Id int NOT NULL,

Band\_Id int NOT NULL,

Profile\_Id int NOT NULL,

Follow\_Date Date NOT NULL,

Unfollow\_Date Date NULL,

CONSTRAINT pk\_Band\_Follower PRIMARY KEY (Follower\_Id)

)

go

CREATE UNIQUE CLUSTERED INDEX idx\_Band\_Follower ON Band\_Follower (Band\_Id, Profile\_Id, Start\_Date)

go

ALTER TABLE Band\_Follower ADD CONSTRAINT fk\_Band\_Follower\_Band FOREIGN KEY (Band\_Id) REFERENCES Band (Band\_Id)

go

ALTER TABLE Band\_Follower ADD CONSTRAINT fk\_Band\_Follower\_Profile FOREIGN KEY (Profile\_Id) REFERENCES Customer\_Profile (Profile\_Id