Conception Phase

Concerning the habits tracker application from a backend perspective, the object of the portfolio examination requested by the course **DLBDSOOFP01**.

Student:

Abdelali BARIR

Matriculation Nº92127958

1. **Introduction**:

The project is a habit tracker designed to track the user's habits. It's a basic while loop to keep the program alive. The user can interact with the application either by typing or in most cases by selecting from multiple choice. This is achieved with the aid of the questionary library. The interface offers 7 options. The data collected from the user is stored in a database.

1.1 Analyse

This option gives the user the ability to track his consistency with habits.

1.1.1 Active habits

By choosing this option, all the active habits, that the user kept incrementing for a period are shown here in the form of a Pandas data frame.

1.1.2 Active habits with the same periodicity

Here the user is required to select a periodicity from his predefined habits resulting in a data frame of active habits with the same periodicity.

1.1.3 Longest streak ever made by the user.

Based on the biggest number of incrementations, the details of the habit's largest streak are shown. Those details include its id, name, periodicity, and ongoing streak.

1.1.4 Longest streak of a given habit.

Based on the habit's name, a search is conducted in both active and broken habit records to show the longest streak the user could keep with the habit.

For the rest of the options, **every** operation is mentioned in a table called tracking habits.

The changes made are instantly committed to the database. A message of the successful execution of an operation and the user can exit the application after.

1.2 Create

The user types in the name of the habit and select its periodicity. The details are passed as parameters to create a tracker object. This object holds its id, name, periodicity, date of creation, and streak. This class keeps track of the number of habits created using it, this is essential to make sure each habit has its unique id.

The object is passed into another function to insert the habit into the active table in our database.

1.3 Edit

The user is shown his active habits, limited to changing only the information he entered when creating the habit, selecting the new periodicity for his habit, or changing its name.

1.4 Increment

The user can increment a habit's streak by choosing its id while displaying in front of him the table of active habits.

1.5 Reset

The user can reset a habit's streak by choosing its id while displaying in front of him the table of active habits.

1.6 Delete

The user can delete a habit's streak by choosing its id while displaying in front of him the table of active habits. Before that happens, the function moves the habit to the broken habits table and specifies the date of the deletion as the day the streak was broken.

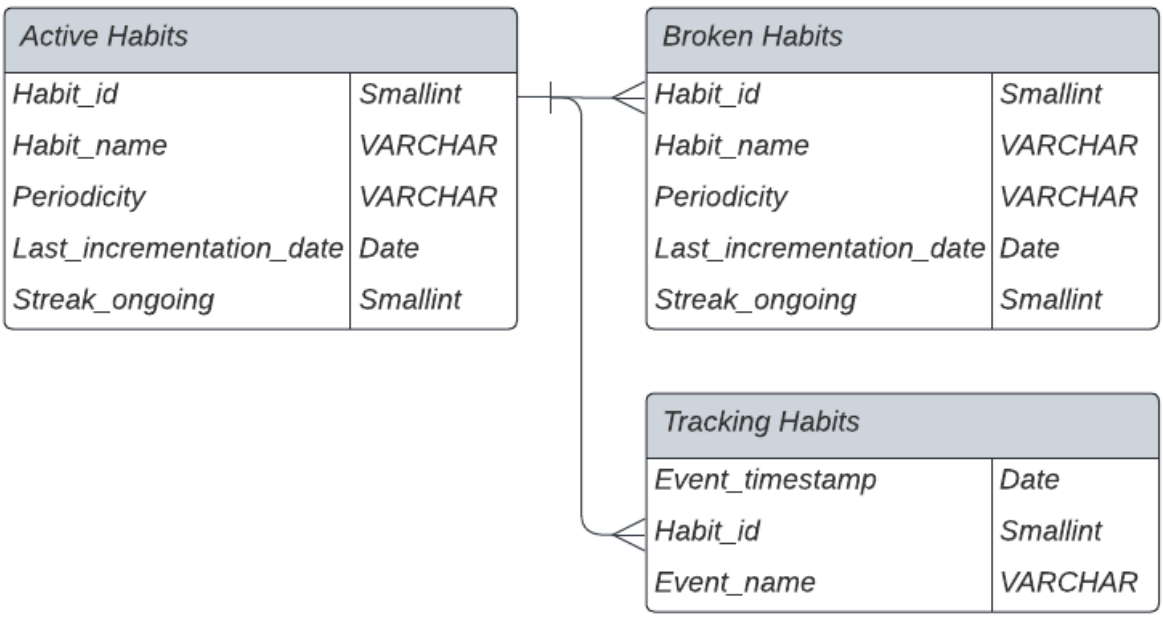
1.7 Exit

In case the user wants to exit the application from the first interface.

2. The broken habits check

This check is scheduled every day at midnight. Depending on each habit's periodicity, the check assesses whether the habit is incremented on time or not. If not, the delete function is triggered with an optional argument called broken set to True, this operation labels the deletion of this habit as due to being broken not being deleted in the tracking habits table.

3. The database's schema



4. The application’s schema