# Exercise: Premium Tracks

## Purpose: First time working with T-SQL’s programming aspects

## Description:

The Chinook company has been doing so well, they’ve decided it’s time for some price increases. At present, every track in their system costs only $0.99. After much market research, they’ve determined that customers are amenable to price increases based on the time length of tracks… longer tracks are rated higher and can demand a higher price. Chinook has decided to call these “Premium Tracks”.

Premium Tracks are defined as any track that is longer than 500,000 milliseconds in length.

## Details of New Pricing Scheme for Premium Tracks:

|  |  |  |  |
| --- | --- | --- | --- |
| Tier | Track Length (in milliseconds) | Rating | New Price |
| Tier 0 | **N/A – Not Premium** | **N/A** | **N/A** |
| Tier 1 | 500,000 to 1,000,000 | 1 | 1.99 |
| Tier 2 | 1,000,000 to 2,000, 000 | 2 | 2.99 |
| Tier 3 | 2,000,000 to 2,500,000 | 3 | 3.99 |
| Tier 4 | 2,500,000 to 3,000,000 | 4 | 4.99 |
| Tier 5 | Greater than 3,000,000 | 5 | 5.99 |

## Using the Chinook DB:

1. Start off by downloading the Premium Tracks DDL script and running it against the Chinook database. This will create the new PremiumTrack table.
2. The new table is based on the original Tracks table, but has the following changes:
   * GenreID, MediaTypeID and AlbumID fields have been removed
   * A new field called Rating has been added.
3. Once you’ve successfully created the new PremiumTrack table, it’s time to build a script that will:
   * Declare and set values for any variables that are required
   * Use a transaction and error handling to allow us to deal with any errors or commit the work if successful.
   * Loop through every track in the Track table and decide whether it qualifies as a Premium track. (has a track length (milliseconds) over 500,000.
   * For qualifying tracks, use the track length (milliseconds) to determine which tier it belongs to, set the appropriate rate and price values, and insert it into the new PremiumTrack table.
4. Start by declaring variables for:
   * A counter (for the loop)
   * The highest trackID in the Track table
   * The current track’s track length.
   * A rating variable
   * A newPrice variable
5. Set your counter to start at 1.
6. Use a SELECT statement to get the highest TrackID and set it as the value for the highest track ID variable.
7. Start your transaction
8. Start your try block.
9. Use a looping structure, which will go through each Track one at a time, starting at ID 1 and ending at the highest track ID. (Don’t hard-code it!)
10. For each iteration of the loop, your code should:
    * Determine and store the current track’s length
    * Determine if the current track qualifies as Premium. If it doesn’t, ignore it and move to the next track. If it does qualify, continue with the next steps:
    * Initialize the “default” values for rating and price
    * Determine which tier the current track belongs to and set the rating/price variable values accordingly.
    * Insert the current track record into the new PremiumTrack table, including the rating value and new price. Hint: Remember INSERT INTO… SELECT?
    * Don’t forget to increment your counter… we don’t want any infinite loops!
11. If no errors occur in the TRY block, commit the transaction.
12. Create a CATCH block, in case of errors. The CATCH block should:
    * Print a message to the console, indicating an error occurred.
    * Undo any changes that may have been made prior to the error being encountered.
13. Finally, no work is complete with some rudimentary testing. See the expected record counts table below to verify your work.
14. Testing Statements:
    * Write a SELECT against the new PremiumTrack table to see how many total records are in the newly-populated table.
    * Write a SELECT to see if any NON-premium tracks were mistakenly added.
    * Write five more SELECTs, one per tier, to check how many records exist per tier.
15. Expected Testing Results: (Note: If you’ve added track records in prior exercises, your numbers may be slightly off)

|  |  |
| --- | --- |
| Test | Expected record count |
| Total records in new PremiumTrack table | 335 records |
| Non-Premium tracks | 0 records |
| Tier 1 (500,000 to 1,000,000 ms) | 120 records |
| Tier 2 (1,000,000 to 2,000,000 ms) | 55 records |
| Tier 3 (200,000 to 2,500,000 ms) | 5 records |
| Tier 4 (2,500,000 to 3,000,000 ms) | 153 records |
| Tier 5 (over 3,000,000 ms) | 1. records |

1. After you’re done, upload your program script to the Brightspace dropbox for the Premium Tracks exercise. (**Note**: You do not have to upload the CREATE TABLE script)