# Exercise: Working with Error Handling IN T-SQL

## Using the Chinook DB:

1. Start off by writing a SELECT query to show all artists and their associated albums. Your results should include both artists WITH associated albums and artists WITHOUT albums.
2. From the resulting data, choose two artists, one with at least one associated album, and the second with no associated albums. Make note of each artist’s name and Artist ID.
3. First, for the artist with NO albums, use transactions and error handling structures to:
   * Attempt to delete the artist.
   * If no errors occur (which should be the case), commit the action.
   * If an error occurs, use error-handling techniques to display all relevant error information in the Results window.
   * If an error occurs, the delete attempt should be reverted.
   * Note: This delete should work, since it doesn’t break any FK constraints.
4. Write a simple SELECT query to verify whether the artist record was deleted and committed.
5. Next, using your artist that DOES have associated album(s), use transactions and error handling to:
   * Attempt to delete the artist.
   * If no errors occur, commit the action.
   * If an error occurs, (which it should!) use error-handling techniques to display all relevant error information in the Results window.
   * If an error occurs, the delete attempt should be reverted.
6. View the error data that should have been displayed when you ran the code from step 5.
7. Write another simple SELECT query to verify that the artist record was NOT deleted (ie. Was rolled back).
8. Finally, let’s see some error throwing, both with and without custom errors.
9. Write (or copy) the code from the previous delete attempt, using the artist with an album.
   * The delete and commit sections should be more or less the same.
   * Add code to ensure that if any errors occur, the data changes will be reverted.
   * First attempt: In your “uh-oh, an error occurred” section, start off with no code at all. Run it and study what happens. Are any errors displayed?
   * Second attempt: This time, add a THROW to the CATCH section. Run your code and observe what happens. If an error occurred, how is it displayed, and what kind of error info is shown?
   * Last attempt: Finally, add a THROW with your own custom message.
   * Note: The error number you use for custom messages must not conflict with existing error numbers, so it must be 50,000 or higher.
   * Run your statement blocks and view if/how your custom error messaging is displayed.
10. Test the data one last time in a simple SELECT query, to see that the artist record was NOT deleted (since we ran into, and then handled, the errors).