# DBAS 4002 - Assignment 5

# Cursors – Discontinued Artists

Assignment Value: *12*% of overall course mark.

Due Date: **See due date designated on the Assignment 5 dropbox on Brightspace.**

Late submissions will receive the standard late submission penalty as stated in the course outline.

#### Required Databases:

* Chinook

#### Assignment Instructions:

**Restore a new copy of the Chinook database, with a Database Name of Chinook\_A5.**

Create a single .sql script file containing all SQL statements required to meet the requirements listed below.

#### Submissions:

When you are finished, upload your script to Brightspace as your submission for Assignment 3, using a file name similar to: **[YourName]\_DBAS\_Assignment5.sql**.

#### Evaluation:

To insure the greatest chance of success on this assignment, be sure to check the marking rubric contained at the end of this document or in Brightspace. The rubric contains the criteria your instructor will be assessing when marking your assignment.

## Assignment Scenario

The Chinook Company has hired you back on to do some data cleanup. With this renewed focus on their database, they’ve realized that some of the data in their database is unused, namely there are several artists that were entered into the system, but without any albums. Since they can’t make money off artists without albums to sell, they’ve decided on some data cleanup. They wish to add a process to the database that, when run, will automatically generate a report of all artists, their albums and the tracks on those albums, as well as remove any album-less artists from the main Artist table and move them to a new DiscontinuedArtists table.

## Requirements

Your solution should add the following to the new **Chinook\_A5** database:

1. A new table called **DiscontinuedArtists** will be added. **See ERD below**.
2. Create a new stored procedure called **uspCheckForDiscontinuedArtists**. This stored procedure will do all the processing to check through all artists, albums and tracks, generate the requested report, and move album-less artist records when required. **All this work MUST BE DONE USING A CURSOR.**
3. General stored procedure requirements:
   1. This stored procedure will be used to create and run the required cursor. Employees from Chinook should be able to re-run this stored proc as often as they want, should they wish to check for discontinued artists.
   2. Code statements inside the cursor should use proper transactions and error handling techniques for all statements that attempt to modify data in the database. (INSERTs, DELETEs, UPDATES, etc.). If no errors occur, data modifications should be permanently saved. If an error occurs, any pending data changes should be cancelled.
   3. During processing, the cursor should generate an “All Artists” report from all the records it processes. See details below.
4. Report Requirements
   1. The report must contain a list of ALL artists, their associated albums (if they have any), and any tracks listed for those albums. **Note:** Artists already added to DiscontinuedArtists table are not included in the report.
   2. The report should list the following data for each record:
      1. Record count (i.e.. A counter for every record processed in the cursor)
      2. Artist ID and Artist Name
      3. Album Title
      4. Track Name and Composer.
   3. A report title and column headings should be printed at the beginning of the report, prior to any record data has been printed. **Note:** Don’t spend hours on formatting. The “columns” of data don’t have to all line up perfectly.
   4. For any Artist records found that have no associated albums, print the artist’s ID and name, followed by a message – **NO ALBUMS OR TRACKS**.
   5. For any tracks found that have no composer listed, the report should display the message – **UNKNOWN COMPOSER**.
   6. For printing the report, the standard “X rows affected” message displayed by Management Studio when row data is changed should be suppressed. Research **NOCOUNT**.
5. Data Modification Requirements
   1. For any artist that has at least one associated album, no data modifications are required.
   2. For artists that have no associated albums, as the cursor processes that record, the following changes should be affected:
      1. Check whether the current artist being processed has already been added to the DiscontinuedArtists table. If so, don’t insert a duplicate artist.
      2. If not, create a new record for the artist in the new DiscontinuedArtists table, see ERD for all expected fields. The original artist PK value should be saved in the OriginalArtistID field, and the date/time the artist was discontinued should automatically be added to the record.
      3. Remove the existing record for the artist from the Artist table. Any artist record should only ever exist in either the main Artist table, or in the DiscontinuedArtists table.

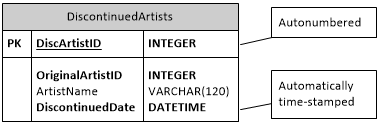
## Executing & Testing Statements

To ensure the accuracy of your work, your script should include testing statements that demonstrate that all your enhancements are functioning as expected and all work is completed to the requirements. Note that during tests that add new test records, hard-coded values are acceptable.

Your testing script should include:

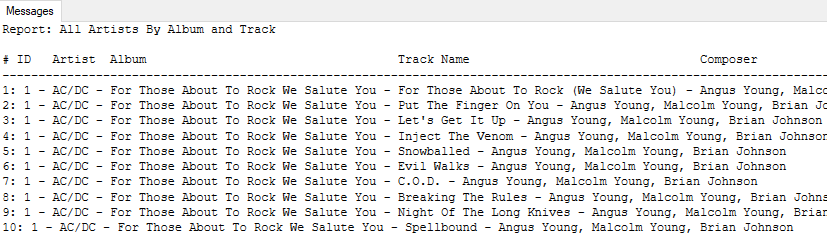
* A query that returns all records that the cursor will be processing – all artists, their associated albums (if any) and the tracks on those albums.
* A query that returns the contents of the DiscontinuedArtists table.
* **Test 1**: A statement that executes the stored procedure, to test it runs as expected. Success means:
  + The report was generated, with all expected data.
  + All album-less artists were removed from main Artist table and added to the DiscontinuedArtists table.
  + Include queries that verify that any modifications worked as expected.
* **Test 2**: Statements that test whether the stored proc will run correctly for any future artist records.
  + Statements to add 5 new test artist records, with no associated albums.
  + Statement to re-run the process.
  + Statement(s) to verify that the test records were removed from Artist and added to DiscontinuedArtists, as expected.
* Clean-up statements to remove all test records.

## ERD for Proposed Modifications

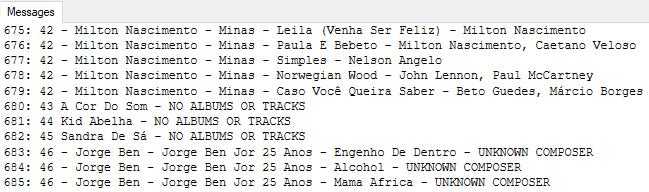


## Screenshots

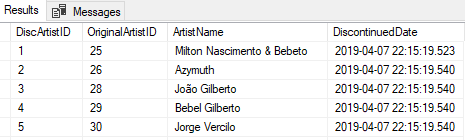
All Artists Report, showing report title, column headings and first 10 records.



All Artists Report, displaying examples of full records, album-less artist records and unknown composer records.



Query results showing first some records from DiscontinuedArtists table after running process.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Chinook – Discontinued Artists** | | | | | **Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| **Criteria** | | **Insufficient (0 pts)** | **Needs Development**  **(1-2 pts)** | **Sufficient**  **(3-4 pts)** | **Excellent (5 pts)** | **X** |
| **DDL** | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | New **DiscontinuedArtists** table successfully created and matches all requirements for fields, constraints and auto-numbering. Drop statement included. |  |
| **Stored Procedure** | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Script for **uspCheckForDiscontinuedArtists** stored procedure creates successfully, meeting all requirements, with no errors or omissions. |  |
| **Fetch Query** | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Query used to fetch all records required for processing correct, returning all expected records, with no record omissions or extra data. |  |
| **Cursor Use** | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Solution includes required cursor. Cursor is defined and used as expected, to correctly complete all required tasks. Cursor usage demonstrates a strong understanding of the concept. | 3 |
| **All Artists Report** | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Report is generated line by line by the cursor,and contains all expected records. Report has a title and column headings. No Albums/Tracks message displays correctly whenever expected. Unknown Composer message displays correctly whenever expected. Record numbering included and “Rows affected” messages are properly suppressed. |  |
| **Discontinued Artist Inserts** | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Cursor is used to add any album-less artist records to the DiscontinuedArtists table. All required data is inserted, with all expected auto-generated values. Works as expected, with no omissions or record duplication upon repeated execution of stored procedure. |  |
| **Artist Record Removal** | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Cursor is used to remove any album-less artist records from the main Artist table. Works as expected, with no omissions or record duplication upon repeated execution of stored procedure. |  |
| **Transactions & Error Handling** | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Transactions and error handling applied where required and applicable and their use demonstrates a strong understanding of the concepts. |  |
| **Testing Script** | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Statements are included to demonstrate all aspects of the required enhancements were successfully implemented. Testing coverage is comprehensive and clearly shows all enhancements work as expected. |  |
| **Comments &**  **Best Practices**  (Must complete at least 60% of functional reqs) | | Little to no effort was made or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Organizational or explanatory comments are used extensively, most are meaningful and easily understood. A consistent naming convention was used for most of the program and deviated very little. Source code was clean, consistently well-formatted and easy to read. |  |
|  |  | |  |  | **Total:** | **/60** |