# DBAS 4002 - Assignment 1

# INNER/OUTER JOINS & Set Operations

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

Due Date: **See due date designated on the Assignment 1 dropbox on Brightspace.**

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

#### Required Databases:

* Chinook
* Bookstore
* Numbers
* Cars
* Lunches

#### Assignment Instructions:

Create or restore all five databases listed above as separate databases in SSMS. Besides Chinook, **DDL creation scripts for the other databases are available on Brightspace**.

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

For this assignment, you will be using five separate databases, Chinook, Bookstore, Numbers, Cars and Lunches. For the latter four, you will need to run the DDL scripts listed below to create the two databases required for this assignment.

#### Submissions:

When you are finished, upload your script to Brightspace as your submission for Assignment 3, using a file name similar to: **[YourName]\_DBAS\_Assignment1.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.

## SELECT Query Requirements

1. **(Chinook db)** Display the First Name, Last Name of each customer along with the First Name and Last Name of their support rep, sorted by customer last and first names. Give the support rep columns an appropriate alias. **(59 records)**
2. **(Chinook db)** Display the track name, genre name, and mediatype name for each track in the database, sorted by track name. **(3503 records)**
3. **(Chinook db)** Display the name of every artist and the total number of albums each artist has available for sale. Results should show the highest totals first. **(275 records)**
4. **(Chinook db)** Display the first name and last name of each customer along with a unique list of the types of media that they have purchased. **(128 records**)
5. **(Chinook db)** Display the first name and last name of the single customer who has purchased the most video tracks. **(1 record,)**
6. **(Chinook db)** Display the name of the artist and number of orders for the single artist who has had the highest number orders of his/her music placed. **(1 record)**
7. Q6 - Needed one more join, from InvoiceLine to Invoice. Then you can count the number of Invoices. Counting the InvoiceLines is getting the count of all lineitems for invoices that INCLUDE a track by the particular artist.
8. **(Chinook db)** Display the TrackID and Track Name of any tracks that have not yet been purchased. **(1519 records)**
9. **(Bookstore db)** Using the “b\_” tables, display the first and last names of all authors who currently do not have any books listed, sorted author last/first name. **(2 records)**
10. **(Bookstore db)** Using the “b\_” tables, display the Customer number, First name, and Last name of any customers who have yet to place an order, sorted customer last/first name. **(6 records)**
11. **(Cars db)** Using the Cars\_Car\_Types, Cars\_Number\_Of\_Doors and Cars\_Colors tables, create a query that returns every possible combination of the values of each table. (**Hint**: The result set should contain 24 rows.)
12. **(Lunches db)** List the employee ID, last name, and phone number of each employee with the name and phone number of his or her manager. Make sure that every employee is listed, even those that do not have a manager. Sort by the employee’s id number. **(10 records)**
13. **(Multiple dbs)** Create one full list of first names and last names of all customers from the Chinook tables, all authors from the Bookstore tables, all customers from the Bookstore tables, and all employees from the Lunches tables. Sort the list by last name and first name in ascending order. **(103 records) \*\*\* See note about this query below rubric \*\*\***
14. **(Numbers db)** Using the Numbers\_Twos and Numbers\_Threes tables, show the results of a query that only displays numbers that do not have a matching value in the other table. **(51 records)**
15. **(Numbers db)** Using the Numbers\_Twos and Numbers\_threes tables, show the results of a query that only displays numbers that have a matching value in the other table. Here’s the catch: **You are not permitted to use a WHERE clause or JOINs for this query. (17 records)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assignment 1 – Marking Rubric** | |  |  |  |
| **Criteria** | **Insufficient (0 pts)** | **Partially Correct (1 pt)** | **Correct (2 pts)** | **Mark** |
| **Queries 1 to 14**  **(Separate rubric criteria line for each)** | Query solution statement contains multiple errors, too many omissions, or was not attempted. | Query solution statement was mostly correct, but contains at least one error or omission. | Query solution statement returns the correct record results required by the question.  Solution includes all additional requested details such as sorting, column aliasing, etc. |  |
| **Readability & Formatting**  (At least 60% of query requirements must be complete) | SQL script is poorly formatted, hard to read, or contains many errors/omissions from requirements. | SQL script is generally well-formatted, but some improvement can be made. | SQL script file is well-formatted, clear and easily readable.  Script contains no syntax errors that would prevent the full script from being run.  All SELECT statements are numbered (with a comment) to indicate which query it is intended to solve. |  |
|  |  |  | **Total:** | **/30** |

\*\*\* Note about Query 12: You may run into a collation error on this one:

Error: Cannot resolve the collation conflict between "SQL\_Latin1\_General\_CP1\_CI\_AS" and "Latin1\_General\_CI\_AS" in the UNION operation.

To resolve: For any column from a **Chinook** table in the SELECT clause, put the following after EACH column – **COLLATE DATABASE\_DEFAULT**

Ex. SELECT GenreID COLLATE DATABASE\_DEFAULT, Name COLLATE DATABASE\_DEFAULT FROM Genre;