# DBAS 4002 - Assignment 2

# Basic Programmability & Stored Procedures

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

Due Date: **See due date designated on the Assignment 2 dropbox on Brightspace.**

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

#### Required Databases:

* Chinook

#### Assignment Instructions:

It is recommended you start with a fresh restore of the Chinook database, to reduce chance of problems resulting from past database modifications.

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

#### Submissions:

When you are finished, upload your script to Brightspace as your submission for Assignment 2, using a file name similar to: **[YourName]\_DBAS\_Assignment2.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 Music Company has identified some problems with their invoicing system. While their current system tracks the amount of money owed per invoice, it does not account for whether an invoice has been paid or not. In addition, invoice payments are occasionally done in installments, so are not fully paid all at once.

They would like to add functionality to their current database that will improve their invoice accounting and track any financial transactions, so they have a record of any payments received against an invoice. This will lead to better accounting practices, less errors, and therefore greater revenue.

The improvements they would like to add should be robust and protected from unexpected errors… nobody likes losing money due to software glitches!

## Requirements

Your solution should add the following to the existing Chinook database:

1. A new column called Paid should be added to the existing Invoice table. It will be marked as True only when an invoice has been fully paid. When a new invoice is created, the new field should always be set to False until full payment has been received.
2. A new table called InvoicePayment will be added. **See ERD below**.
3. Four new procedures should be added to the database.
   1. Each procedure must use transactions where appropriate and include basic error handling. If part of the procedure fails, the entirety of its tasks should be reverted.
   2. **Proc 1:** A procedure named **uspAddNewInvoice**, which will be used to add new invoice records to the Invoice table. Note: For this assignment, an Invoice will only ever have a single InvoiceLine item associated with it. This proc will receive all values for the new Invoice and its line item and add them as new, related records to the Invoice and InvoiceLine tables. The Invoice Date should be auto-generated by the system, and the Invoice Total field should be calculated (Price \* Quantity) instead of passed as a parameter.
   3. **Proc 2:** A procedure named **uspAddInvoicePayment**, which will be used to add a new record to the new InvoicePayment table. Moving forward, this proc will be used to process new payment entries. The Payment Date should be auto-generated by the system.
   4. **Proc 3:** A procedure called **uspRunInvoices**. The intent of this procedure is to be a one-time proc that will create invoice payment records for all existing records. (Moving forward, Invoices and Payments will use the other new stored procs.) This procedure will assume all existing Invoice have been paid in full. It will go through every existing Invoice record and use data from the existing Invoice records to create new InvoicePayment records. After gathering the appropriate data for an Invoice record, this procedure will call the **uspAddInvoicePayment** procedure to add the new Payment records.
   5. **Proc 4:** A procedure called **uspCheckPaidInvoices**, which will iterate through every Invoice record and compare it to its associated Payment record. If the amount in both records is the same, the procedure will update the Paid field in the Invoice table to True.

## Executing Procs & Testing Statements

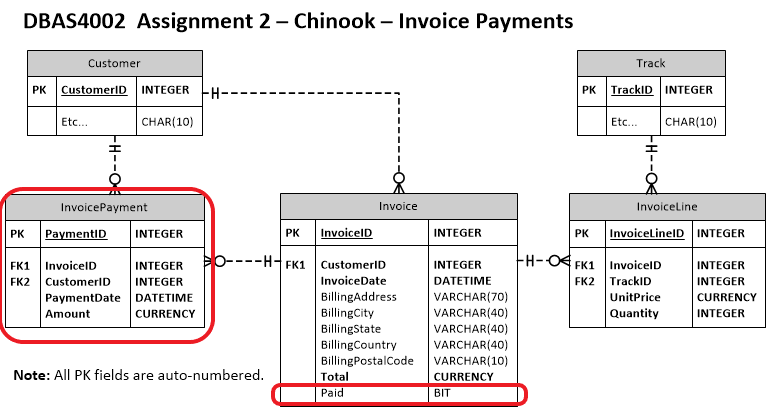
To ensure the accuracy of your work, your script should include executing calls to the new stored procedures, in the proper order, and include any supporting statements that, when run together, ensure all work is completed to the requirements. Note that during tests that add new test records, hard-coded values are acceptable.

Your testing script should include:

* A statement to add a new Invoice/InvoiceLine record (only one InvoiceLine record is required), using your uspAddNewInvoice procedure.
* A statement to add a new invoice payment for the new Invoice added in Test 1, using your uspAddInvoicePayment procedure.
* A statement to revert ALL Invoice records to unpaid status.
* A statement to remove all existing records from the InvoicePayment table.
* A statement to re-create all InvoicePayment records, using your uspRunInvoices procedure.
* A statement to re-set all applicable Invoices to Paid or not, using your uspCheckPaidInvoices procedure.

## Chinook ERD for Proposed Modifications

#### Invoice Payments Enhancement



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Chinook-Invoice Payments** | | |  |  |  |  |  |
| **Criteria** | | **Insufficient (0 pts)** | **Needs Development**  **(1-2 pts)** | **Sufficient (3-4 pts)** | **Excellent (5 pts)** |  | **X** |
| **DDL for Invoice & InvoicePayment** | | 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. | Modifications to existing Invoice table completed as expected.  New InvoicePayment table successfully created and matches all requirements for fields, constraints, auto-numbering.and relationships. |  |  |
| **Proc 1:**  **Add New Invoice** | | 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. | uspAddNewInvoice works as expected to add new Invoice and InvoiceLine records. All required parameters are included, have the correct datatype, and are used correctly in the procedure. InvoiceDate is system-generated, and Total is calculated |  |  |
| **Proc 2:**  **Add Invoice Payment** | | 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. | uspAddInvoicePayment works as expected to add new InvoicePayment records. All required parameters are included, have the correct datatype, and are used correctly in the procedure. PaymentDate is system-generated. |  |  |
| **Proc 3: Run Invoices** | | 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. | uspRunInvoices works as expected to add new InvoicePayment records for all existing Invoices. Procedure contains an appropriate call to uspAddInvoicePayment. |  |  |
| **Proc 4: Check Paid Invoices** | | 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. | uspCheckPaidInvoices works as expected to compare all Invoice and InvoicePayment records to determine whether they are paid. Procedure marks all appropriate Invoices as Paid. |  |  |
| **Executing & 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 correctly execute all required stored procedures, in the proper order, to complete the overall task. Statements to reset all Invoices as Unpaid and to clear the InvoicePayment table for testing are included. |  | 2 |
| **Transactions 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. | Use of transactions shows a strong understanding of the concept, and transactions are correctly applied in all reasonably appropriate places in the procedures. |  |  |
| **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. | Use of error handling shows a strong understanding of the concept, and error handling techniques are correctly applied in all reasonably appropriate places in the procedures. |  |  |
| **Programming** | | 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. | Use programming aspects (variable declaration and assignment, BEGIN-END, looping and conditional structures) are correctly applied in all reasonably appropriate places in the procedures. |  |  |
| **Script Usability** | | 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 demonstrates proper batching (GO) whenever needed. Script runs without errors. |  |  |
| **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** |