**Marks 25%: Due 11th April**

Work to be submitted as SQL files via Moodle. The files should be called Question1.sql etc. There is no need to label the files.

Note: In the Moodle Tile for Week 1 there is a script which rewrites all of the tables in the AP database. This might be useful for you when testing your work.

Also submit a short word document with the following information for each question.

Whether you think you answer is correct and any problems you had writing the code.

**Question 1: (6 marks)**

Create a user defined function in SQL-Server called **fnEarliestUnpaidDate** which takes a VendorID as a parameter and returns the date of the earliest unpaid invoice from that vendor.

**Note:**

An invoice is unpaid if (invoiceTotal-PaymentTotal) >0.

The earliest date can be got from using MIN(INVOICEDATE).

Save the SQL for creating or altering the function into a file called question1.sql

**Question 2: (6 marks)**

Create a new table called VendorTermsArchive

create table vendorsTermsArchive (

vendorID int,

oldTermsID int,

newTermsID int,

changeDate smalldatetime)

Write a trigger called **Archive\_old\_Terms** so that:

When a VENDORS DefaultTermsID is updated a new entry is made in VendorTermsArchive table with the VendorID, the oldTermsID, the newTermsID and the date the change was made.

Save the SQL for creating or altering the trigger into a file called question2.sql

**Note:**

For this trigger you need to use the tables inserted and deleted to get the old and new values. You can use the system function getDate() to get the date.

**Question 3: (6 marks)**

1. **(2 marks)**

Write a query which counts the number of invoices for each vendorID.

The query should return the number of Invoices and the VendorId for each VendorID.

1. **(2 marks)**

Write a user defined function called **fnCountVendorInvoices** which takes a vendorID and returns the number of invoices for that vendorID.

1. **(2 marks)**

Write a stored procedure called **procNumberOfInvoices** which takes a vendorID.

If that vendorID does not exist in the database it **throws** an error, with error\_number 50001, message ‘vendor not found’ and state 1

If that vendorID exists but has no associated invoices it prints out **“No invoices”**

Otherwise it prints out the number of invoices. **“ x invoices”**.

Use the function fnCountVendorInvoices defined in part (b) in the stored procedure.

**Question 4: (7 marks)**

Write a stored procedure called procReissueInvoice.

The stored procedure should take an invoiceID and do the following.

Step 1: Create a new invoice with the same details, except that the InvoiceDate is today’s date and the InvoiceDueDate is 30 days from today.

Step 2: Update the invoiceLineItems table so any records with the old InvoiceID are updated to the new invoiceid (

Note: You can use the system function @@identity() to get the new ID)

Step 3: Delete the invoice with the original invoiceID from the invoices table.

The three steps should be done as a transaction.

If there are no invoices with the invoiceID then an error should be thrown with the message ‘no invoice with that id’ the status 5001, and the state 1.

If the transaction fails then there should be a rollback and an error should be thrown with a status of 5001, a message ‘failed due to transaction error’ and a state of 1.