Get Geeky Homework

Here are some queries to help get to the answers of the questions you got for your seminar homework. I've left some white space near the end of each one in case you want to make some notes for your own ideas for further exploration.

1-Payroll Users: run a query to list all your processing groups and how many active employees are members of each. Confirm that those add up to all your active employees

This query counts the active employees in each group and displays the results:

```
SELECT Count(sEmployeeID) EmpCount, sPGCodeIDf ProcGrp
FROM tblPREE Emp
INNER JOIN tblPRProcessGrp PGrp on PGrp.sPGCodeID = Emp.sPGCodeIDf
WHERE Emp.sStatus = 'A'
GROUP BY sPGCodeIDf
ORDER BY ProcGrp
```

This query counts the active employees in the table:

```
SELECT COUNT(sEmployeeID)
FROM tblPREE
WHERE sStatus = 'A'
```

And you can satisfy yourself that those are the same value easily enough, but you have to do the addition which could be considerable for large numbers of employees and groups. So let's get a little fancy here by making a temporary table to hold our results and then running a count query against that table to do the addition for us:

```
SELECT Count(sEmployeeID) EmpCount, sPGCodeIDf ProcGrp
INTO #TempEmpCount FROM tblPREE Emp
INNER JOIN tblPRProcessGrp PGrp on PGrp.sPGCodeID = Emp.sPGCodeIDf
WHERE Emp.sStatus = 'A'
GROUP BY sPGCodeIDf
ORDER BY ProcGrp
```

```
SELECT SUM(EmpCount) FROM #TempEmpCount
```

We generally want to get rid of our own temporary tables, but behind the scenes I believe you will find that SQL Server will clean them up for you when you use the # in the name.

```
DROP TABLE #TempEmpCount
```

We can get fancier still here by including text in our results to lend some context to the bare numbers in our results. But combining text and numbers in query output requires us to learn a new keyword: CAST. Here's a quick example of its use:

```
SELECT 'Total Active Employees from tblPREE is: ' + CAST(COUNT(sEmployeeID) As Varchar(10))
FROM tblPREE
WHERE sStatus = 'A'
```

2-Everyone: write queries to produce a list of AP Invoice sessions you did each month of 2017 and how many invoice documents were in each session. Confirm that those 12 add up to the whole year's invoices

For the first individual month:

```
SELECT sSessionNumIDf Session, COUNT(sDocNum)
FROM tblDLDocument
WHERE sTransSourceIDf = 'API'
And dtmDocDate Between '1-1-2017' And '1-31-2017'
GROUP BY sSessionNumIDf
```

And then for each additional month you just modify the dates in the WHERE clause. With no other key word help, you would likely have to put the results in excel and add them up to get a total. Then use this one to get the annual total.

```
SELECT COUNT(sDocNum)
FROM tblDLDocument
WHERE sTransSourceIDf = 'API'
And dtmDocDate Between '1-1-2017' And '12-31-2017'
```

Once you learn more advanced SQL you can get quite elegant with problems like this. But this is a pretty good illustration of how effective you can be with basic knowledge.

3-The adventuresome: how many UDFs have you set up of each type?

Sure, we didn't talk about UDFs. But now that you've seen tables and column in MIP's database and learned how to find out about their contents with that handy Default Table Structure option on the Organization menu, you can poke around here a bit. The table that is going to be useful for us is, of course, tblUDFProperties. Here you'll see the sTypeID is the very first column. So our query is really pretty easy.

```
SELECT sTypeID, COUNT(sTypeID)
FROM tblUDFProperties
GROUP BY sTypeID
```

It will be a shame if you have none set up and get an empty result set. So try it against NTO if that happens to you.

4-Really adventuresome: how many reports has your organization set up of each Report Type

This one was just really a tricky question. I didn't ask how many reports there are of each type like we did with the UDFs. I asked you many your org has set up. So you have to have a way of ignoring all those default reports MIP provides to get the right answer here.

We never looked at the report tables in our webinar session, but if you tried this one you likely figured out quickly that the gold was going to be in tblRptSetup. So here's the query:

```
SELECT sRptType, COUNT(sRptType)
FROM tblRptSetup
WHERE sRptID Not Like '<%'
GROUP BY sRptType
```

You can make the output prettier with Aliasing, of course. But the volume of data in the report set may have been a surprise. If you've been a MIP user for a few years there may be hundreds of reports here! What may be even more surprising is how many samples MIP gives you. The NTO database has had over

350 of them for some time now! Many organizations lose control a bit here and you can find anywhere from a few hundred reports to more than a thousand. Perhaps your query has given you some ideas about increasing efficiency where you work? And the query you use here shows you very well which ones are needed the most. Among the ones used most often in most orgs are GLStd, TBNorm, RECustom, Posted, and GLExp. It’s not unusual to see more than 20 of each of these. Some, sadly, have literally 100 or more of **each**!