This code applies to SQL Server express - -a powerful, free Windows database engine from Microsoft. The concepts will be the same for other programs, but the syntax may be slightly different.

If you’re used to working in Excel, you’ll find that SQL can work as a spreadsheet-processing tool, only on steroids. It can save you time in running your digital interview. Because that’s what data is – just another source we will debrief.

Before we begin, make sure to download and install the SQL Server Express Advanced version – <http://www.microsoft.com/en-US/download/details.aspx?id=42299> – to get all of the tools you need. Now that we have our tool, we need an interviewee.

Let’s download our data, that is the source. SQL can process several file types. We will be working with a spreadsheet, an Excel file, which can be downloaded from a web platform called DropBox. Here is the link - [**http://bit.ly/1VB3gB5**](http://bit.ly/1VB3gB5). Best to download it on your desktop or a place from where you can retrieve it easily.

Next, we open Microsoft SQL Server Management Studio.

Before importing the data, we need to create a database, where to import the data into.

* Right+click on Databases
* New Database - {give it a name, say “Fish”} - click OK

​

Say you’re investigating the fishing industry. This is where we will import every table related to the project. Let’s start with our first source, which you’ve downloaded already.

* Right+click on the new database name - ​Task​s - ​Import Data​ - **Nex​t**
  + Data Source – Excel (note the file sources that import to SQL)
  + Browse - select the table from where you've saved it
  + Excel version - Excel 2007 (you can click on the drop-down and select it)
  + "First row has column names" has to be ticked - **Next​**
* Destination - SQL Server Native Client (when your laptop is your server, like it will have to be in this case) - **Next**​
* Select "Copy data from one or more tables or views" - **Next​**
* Select Source Tables and Views - tick only "CustomsSRCShrimp$" - **Next​**
* ​Under "Review Data Type Mapping"​ click **Next**
* Tick "Run immediately" – **Next - Finish​**

Sometimes you’ll see the tabe straight away in the data tree on the left, under Tables. But we’ll encourage it to show up.

* Right+click on your dbs. name – Refresh.

You should see your table under Tables.

You notice that your dbs features $ in its name. SQL adds it by default. In order for us to work smoothly, we must remove it.

* Right+click on your table’s name – Rename – delete $

We can start working now. Part of our data cleaning due dill, a standard check is to call up our data. Let us

1. **Count every row**:

SELECT \*

FROM [dbs name]

We could have also called up just a slice of our data. Like this:

1. **Call the first 1000 rows**:

Right+Click on the table’s name – Select Top 1000 Rows

This is for us to get some quality time with our source. Meaning, you want to count the number of total records; check out the variables you’ll be working with; the type of data you have to consider; start thinking of the questions you’ll be asking; but, most importantly, mapping the dirt to clean for precision journalism.

Each time you have to deal with names, you must consider bored data entry clerks. That’s code for “misspelling”/dirty data. Let’s have a quick look at the companies that ship our fish.

1. ***Group and order names alphabetically*** to spot various entries for the same company or catch misspelled words.

SELECT SHIPPER\_NAME

FROM CustomsSrcShrimp

GROUP BY [Shipper\_Name]

ORDER BY 1;

This is a full list with the names of each shipper company. Results show multiple entries for the same companies. In this case, we must decide which spelling we want to go for and update our table. To do this, we will first create a column where we will be working, so as to keep the original in case we might mess up.

***To create a new column:***

ALTER TABLE CustomsSrcShrimp

ADD MyShipper\_Name char(250);

***Copy the data*** from the original column to the mirror field you’ve just created:

UPDATE CustomsSrcShrimp SET MySHIPPER\_NAME = SHIPPER\_NAME;

Let’s clean up names, so that we have only one label for one company. Let’s execute our alphabetical list again. We have three different entries for our first company. Let’s take the first spelling and use it to overwrite the other two. We need an **Update with a Where clause**

UPDATE CustomsSrcShrimp SET MyShipper\_Name = 'Andaman Seafood Co Ltd'

WHERE Shipper\_Name = 'Andaman Seafood Co., Ltd.';

Result: 11 rows were affected.

Let’s do the second one

UPDATE CustomsSrcShrimp SET MyShipper\_Name = 'Andaman Seafood Co Ltd'

WHERE Shipper\_Name = 'Andaman Seafood Co.,Ltd.';

Result: 12 rows affected

Now let’s verify our work:

SELECT MySHIPPER\_NAME

FROM CustomsSrcShrimp

GROUP BY [MyShipper\_Name]

ORDER BY 1;

Result: only one entry for the same company.

Let’s do another one: Asian Seafoods Coldstorage vs. Asian Seafoods Coldstorage Public.

First and foremost, check that it’s the same company. Look at company’s registration number online, check addresses, phone numbers if you must etc. NEVER ASSUME ANYTHING. After you’re satisfied that it’s the same company, run your clean up query.

UPDATE CustomsSrcShrimp SET MyShipper\_Name = 'Asian Seafoods Coldstorage Public'

WHERE Shipper\_Name = 'Asian Seafoods Coldstorage';

Result: 20 rows affected

Keep updating names one by one, column by column.

You may also want try a **WHERE – LIKE** **clause** to speed up the process.

SELECT [Shipper\_Name]

FROM CustomsSrcShrimp

WHERE [Shipper\_Name] LIKE ('Anda%')

GROUP BY [Shipper\_Name]

**%** can sit before, after or at both ends of your search criteria. It stands for zero, one or multiple characters.

However attractive this may be to you, do not discount eyeballing.

To find more about quick searches based on criteria, here’s a good tutorial for you - <http://www.tutorialspoint.com/sql/sql-like-clause.htm>

**TIP\*\*\*** Make sure to keep the original columns as you add yours so that you can always compare and check your work as you progress.

**USING LOGIC TO TRANSFORM DATA**

Sometimes the data isn’t exactly as you need it to do your analysis and you need to transform it. For example, let’s pretend for a moment that the weight\_in\_kg column is wrong – some of those weights are actually in other units, and we need to use the weight unit field to check what we need to convert.

We can check what we’re dealing with using a query:

select weightunit, count(weightunit) from CustomsSrcShrimp

group by weightunit

Ah ha, we need to convert our LBS to kilograms to be consistent. Let’s add a column called new\_weight and use a bit of SQL Logic to standardize our weights.

ALTER TABLE CustomsSrcShrimp

ADD new\_weight float(8);

Ok, now we’ll use a SQL command called CASE… WHEN to populate the new field based on the information reported in weight unit

update CustomsSrcShrimp set

new\_weight=(Case when WeightUnit='LB' then weight\_in\_kg\*0.453592 else weight\_in\_kg end)

USING DATE FUNCTIONS

Every flavour of SQL has its own unique data functions. There are too many to go over in this short class but please refer to this reference for Microsoft’s date functions: <https://msdn.microsoft.com/en-us/library/ms186724.aspx#DateandTimeFunctions>

Let’s play around with a couple of them to see how powerful they can be. The data we’re looking at provides us with a date, but let’s say we’re interested in monthly data.

This can be done using the MONTH() function. For example, if we wanted to summarize our new\_weight field by month:

Select month([date]) as the\_month, sum(new\_weight) as the\_weight from CustomsSrcShrimp

group by month([date])

order by sum(new\_weight) desc

Notice how we had to wrap the field name date in brackets – date is itself a function and SQL Server would get confused otherwise!

Let’s say you want to see the name of the month, not the month number. Swap in the DATENAME function:

Select datename(month,[date]) as the\_month, sum(new\_weight) as the\_weight from CustomsSrcShrimp

group by datename(month,[date])

order by sum(new\_weight) desc

Lastly, let’s say you wanted to know how many days ago each transaction occurred. You can use the DATEDIFF function:

select DATEDIFF(day,[date],getdate() ) from CustomsSrcShrimp

Some notes on Dates:

Date functions only work if you have a valid date. For more on this topic, please refer to

<https://anubhavg.wordpress.com/2009/06/11/how-to-format-datetime-date-in-sql-server-2005/>

One example of this – let’s say you have dates in a text format and want to convert them to standard US format:

select convert(datetime, 'April 15, 2015 3:00 pm', 101)

2015-04-15 15:00:00.000

That output can then be used in any calculation you need to do with the Date functions.