*At*[*The Data Incubator*](http://www.thedataincubator.com/?utm_source=Blog&utm_medium=Link&utm_campaign=ProcessingDataPost)*, we run a free six week data science fellowship to help transition our Fellows from Academia to Industry. This post runs through some of the toolsets you’ll need to know to kickstart your Data Science Career*.

If you’re an aspiring data scientist but still processing your data in Excel, you might want to upgrade your toolset.  Why?  Firstly, while advanced features like Excel Pivot tables can do a lot, they don’t offer nearly the flexibility, control, and power of tools like [SQL](http://en.wikipedia.org/wiki/SQL), or their functional equivalents in [Python](https://www.python.org/) ([Pandas](http://pandas.pydata.org/)) or [R](http://www.r-project.org/) ([Dataframes](https://stat.ethz.ch/R-manual/R-devel/library/base/html/data.frame.html)).  Also, Excel has [low size limits](https://support.office.com/en-nz/article/Excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3), making it suitable for “small data”, not  “big data.”

In this blog entry we’ll talk about SQL.  This should cover your “medium data” needs, which we’ll define as the next level of data where the rows do not fit the 1 million row restriction in Excel.  SQL stores data in tables, which you can think of as a spreadsheet layout but with more structure.  Each row represents a specific record, (e.g. an employee at your company) and each column of a table corresponds to an attribute (e.g. name, department id, salary).  Critically, each column must be of the same “type”.  Here is a sample of the table Employees:

| **EmployeeId** | **Name** | **StartYear** | **Salary** | **DepartmentId** |
| --- | --- | --- | --- | --- |
| 1 | Bob | 2001 | 10.5 | 10 |
| 2 | Sally | 2004 | 20. | 10 |
| 3 | Alice | 2005 | 25. | 20 |
| 4 | Fred | 2004 | 12.5 | 20 |

SQL has many keywords which compose its query language but the ones most relevant to data scientists are SELECT, WHERE, GROUP BY, JOIN.  We’ll go through these each individually.

SELECT

SELECT is the foundational keyword in SQL. SELECT can also filter on columns.  For example

SELECT Name, StartYear FROM Employees

returns

| **Name** | **StartYear** |
| --- | --- |
| Bob | 2001 |
| Sally | 2004 |
| Alice | 2005 |
| Fred | 2004 |

WHERE

The WHERE clause filters the rows. For example

SELECT \* FROM Employees WHERE StartYear=2004

returns

| **EmployeeId** | **Name** | **StartYear** | **Salary** | **DepartmentId** |
| --- | --- | --- | --- | --- |
| 2 | Sally | 2004 | 20. | 10 |
| 4 | Fred | 2004 | 12.5 | 20 |

GROUP BY

Next, the GROUP BY clause allows for combining rows using different functions like COUNT (count) and AVG (average). For example,

SELECT StartYear, COUNT(\*) as Num, AVG(Salary) as AvgSalary FROM EMPLOYEES GROUP BY StartYear

returns

| **StartYear** | **Num** | **AvgSalary** |
| --- | --- | --- |
| 2001 | 1 | 10.5 |
| 2004 | 2 | 16.25 |
| 2005 | 1 | 25. |

JOIN

Finally, the JOIN clause allows us to join in other tables. For example, assume we have a table called Departments:

| **DepartmentId** | **DepartmentName** |
| --- | --- |
| 10 | Sales |
| 20 | Engineering |

We could use JOIN to combine the Employees and Departments tables based ON the DepartmentId fields:

SELECT Employees.Name AS EmpName, Departments.DepartmentName AS DepName

FROM Employees JOIN Departments

ON Employees.DepartmentId = Departments.DepartmentId;

The results might look like:

| **EmpName** | **DepName** |
| --- | --- |
| Bob | Sales |
| Sally | Sales |
| Alice | Engineering |
| Fred | Engineering |

We’ve ignored a lot of details about joins: e.g. there are actually (at least) [4 types of joins](http://en.wikipedia.org/wiki/Join_%28SQL%29), but hopefully this gives you a good picture.

Conclusion and Further Reading

With these basic commands, you can get a lot of basic data processing done.  Don’t forget, that you can nest queries and create really complicated joins.  It’s a lot more powerful than Excel, and gives you much better control of .  Of course, there’s a lot more to SQL than what we’ve mentioned and this is only intended to wet your appetite and give you a taste of what you’re missing.

* For additional tutorials on SQL, checkout this [interactive online tutorial](http://sqlzoo.net/wiki/Main_Page) or the [data-scientist-geared one](http://sqlschool.modeanalytics.com/) provided by [Mode Analytics](https://modeanalytics.com/). Also, check out this [comprehensive list of 59 websites to learn SQL](https://www.databasestar.com/learn-sql/).
* For additional resources on Python Pandas, checkout [a list of tutorials](http://pandas.pydata.org/pandas-docs/stable/tutorials.html) (complete with github repos!) or the [10 minute pandas tutorial](http://pandas.pydata.org/pandas-docs/stable/10min.html) for the impatient.
* For a tutorial about R’s Dataframes, checkout [this page](http://www.r-tutor.com/r-introduction/data-frame).

And when you’re ready to step it up from “medium data” to “big data”, you should [apply](http://www.thedataincubator.com/#apply) for a fellowship at The Data Incubator where we work with current-generation data-processing technologies like [MapReduce](http://en.wikipedia.org/wiki/MapReduce) and [Spark](http://en.wikipedia.org/wiki/Apache_Spark)!