**Problem 2: Matrix Multiplication in SQL**

Recall from lecture that a *sparse* matrix has many positions with a value of zero.

Systems designed to efficiently support sparse matrices look a lot like databases: They represent each cell as a record (i,j,value).

The benefit is that you only need one record for every non-zero element of a matrix.

For example, the matrix

|  |  |  |
| --- | --- | --- |
| 0 | 2 | -1 |
| 1 | 0 | 0 |
| 0 | 0 | -3 |
| 0 | 0 | 0 |

can be represented as a table

|  |  |  |
| --- | --- | --- |
| row # | column # | value |
| 0 | 1 | 2 |
| 0 | 2 | -1 |
| 1 | 0 | 1 |
| 2 | 2 | -3 |

Take a minute to make sure you understand how to convert back and forth between these two representations.

Now, since you can represent a sparse matrix as a table, it’s reasonable to consider whether you can express matrix multiplication as a SQL query and whether it makes sense to do so.

Within [matrix.db](https://github.com/uwescience/datasci_course_materials), there are two matrices A and B represented as follows:

A(row\_num, col\_num, value)

B(row\_num, col\_num, value)

The matrix A and matrix B are both square matrices with 5 rows and 5 columns.

**(g) multiply:** Express A X B as a SQL query, referring to the class lecture for hints.

What to turn in: On the assignment site, turn in a text document, multiply.txt, which is value of the cell (2,3)

If you’re wondering why this might be a good idea, consider that advanced databases execute queries in parallel automatically.  So it can be quite efficient to process a very large sparse matrix in a database (millions of rows and columns)!

But a word of warning: In a job interview, don’t tell them you recommend implementing linear algebra in a database.  You won’t be wrong, but they won’t understand databases as well as you now do, and therefore won’t understand when this is a good idea.  Just say you have done some experiments using databases for analytics, then mention the papers in the reading if they seem incredulous!