# **COMP-1701 - Transferring Data to Databases**

## **Data Science & Machine Learning (DSML) - RRC Polytech**

Module E.1- Entity Relationship Diagram (ERD) – Draw.IO App Usage

In this database class, we use the application called Draw.IO to draw out our tables and our relationships. It is a common practice to draw out and create your Entity Relationship Diagram so you can see how the tables hook up the data, in effort to eliminate as much String duplication as possible.

REGARDING APPS/OS UPDATES/UPGRADES

During the term, if your laptop **apps** (ie: software you installed to your local hard drive) are **stable and running correctly**, then **do not run any app updates if/when prompted**, unless the instructors tells you otherwise (for this class’s DBeaver and Draw.IO, and likely in other classes’ apps/software you are taking). If unsure, check with your instructor. Updates should mostly only be ran, if there is a bug or security issue to remedy.

Your Operating System (OS) is a different story, always update (not upgrade) your current OS, just do not install a Major Version update. By Major Version, don’t upgrade from Windows 10 to Windows 11 during any term, as this may cause application chaos on your computer, and as your are IT professionals-to-be, you need to maintaining your systems’ state and are responsible for fixing any issues you might encounter.   
  
Major Version upgrades can issues that may require uninstallation/reinstallation of your apps/software or worse, complete hard drive/partition wipes and reinstallation of the OS, which may at that point, just be best to go back to the previous Major Version you were using.   
  
In either case, you should be backing up your Desktop, Documents, Download, and other important folders to OneDrive at this point anyways. If a hard drive gets wiped, you lose all your files. IT pros backup their works always.

**Download/Install draw.io Diagrams app**

* <https://github.com/jgraph/drawio-desktop/releases>
  + **Windows** >
    - 64-bit: draw.io-##.#.#.**msi**
  + **Mac** >
    - ARM 64: draw.io-**arm64**-##.#.#.**dmg**
    - 64-bit: draw.io-**x64**-##.#.#**.dmg**
    - Universal: draw.io-**universal**-##.#.#.**dmg**
  + **Linux** (click the **Show all *30* assets**…noting 30 is a changing count value) >
    - AMD 64: drawio-**amd64**-##.#.#.**deb (or .rpm)**
    - ARM 64: drawio-**arm64**-##.#.#.**deb (or .rpm)**

A screenshot of a computer

Description automatically generated

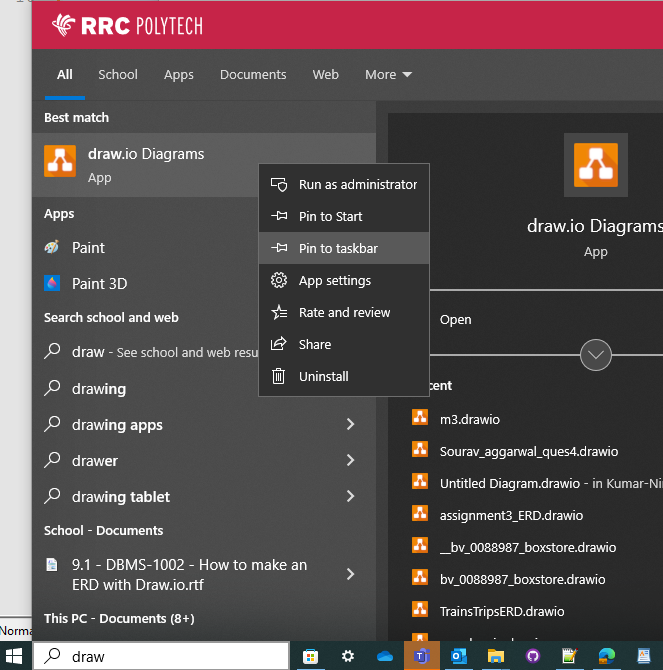
Please note Windows Users, you can also do the install thru **Microsoft Store** > *Search for*: **draw.io** and *install*: **draw.io Diagrams**

A screenshot of a computer

Description automatically generated

**Create a Shortcut to the Draw.IO Diagrams app**

Wise if you shortcut this app to your Taskbar, by right clicking on it from the Windows button/Search and Pin to Taskbar:



**Opening Draw.IO Diagrams app**

There are 3 critical actions you need to address when opening an existing diagram, or in this case, creating a new diagram.

A blue and white rectangular object with text

Description automatically generated

After clicking the appropriate option, Create New Diagram, mark down your filename with a similar naming convention as you \_boxstore.sql, only with a version in it after the word boxstore. That number will increment by 1 during different phases of the build, or while working on in, the minor version (second number) can increment while you are working on it, to keep previous versions. So copy/rename the file to 0.1 for continuing work on it. To start, here is what your file should be named: fl\_studentid\_boxstore\_v**0.0.drawio**

A screenshot of a computer

Description automatically generated

Click on **Entity Relationship Diagram**, then click CREATE at the bottom. There after just to a File > Save to your master SQL folder (where your \*\_boxstore.sql file is).

**Sample Draw.IO Diagram (do not follow this format)**

Initially you will get a sample diagram, which isn’t using formats we use when creating files, such as:

* We do not use rounded lines, we use **Sharp**
* We change the line formats to **Orthogonal**
* We resize the height of the table, to remove the spacing at the bottom of the table when complete.

So the current sample that looks like:

A screenshot of a graph

Description automatically generated

…in the end, should be in this format:

A screenshot of a computer

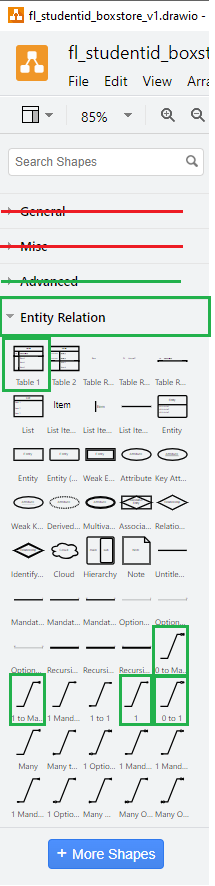
Description automatically generated

See the top image, the customers, orders and shipments tables all have a space below the last column identified, you can resize the table a smidge to remove those. The tables aren’t using the grid sizing very well either. The lines have rounded bends, and in general it is sloppy and does not align to the grid very well. You should be able to do better.

You will see a grid in the background, with large squares and within 4x4 small squares. If you are dealing with several table joins, you may want to separate the tables by 3 large squares. Also, the lines for the JOINs should be equidistant from each other, so noting the customer table, it starts and ends within 6 large blocks, the line going to orders is 2 large blocks in length, though the bend in the line is done at the 1 large block spot. These line formats can be tricky, though there is always a way to create these ERD diagrams to ensure lines from differing tables to not cross over another relationship line.

**Shapes (in the left index)**

Start your diagram with some spacing in mind, also ENSURE you are using Entity Relation objects from the left side (as ridiculously the app, even though you chose Entity Relation Diagram, has the General objects displayed, do not use those). Lower down, you will see Entity Relation, this is what you use. So click General to collapse(hide) it and click Entity Relation to expand(show) those objects…as seen in the image below)

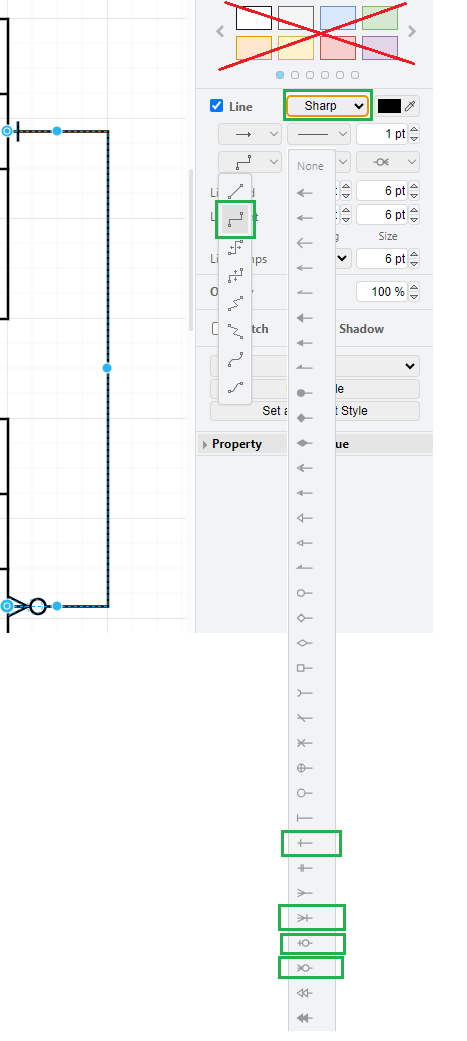


Within the Entity Relation area (image above), there will only be 1 TABLE object and 4 possible CARDINALITY RELATIONSHIP line ends you will ever use (this will be elaborated on more, when we talk about JOINs), the TABLE object is really your first need to understand:

* Table 1 (for every table)
* Cardinalities of:
  + 0 to Many (applied to FK)   
    
  + 1 to Many (most common JOIN, applied to FK)  
    
  + 1 (applied to FK, always on a PK)  
    
  + 0 to 1 (applied to FK, sometimes on PK when an FK table can have a record exist with the FK being NULL)  
    

The crow’s foot , if you twist it clockwise, would look like an M, for Many. The  and  are 1 and 0 respectively.

The following shows the options you need to select for any Relationship line, Sharp, Orthogonal, and the Cardinalities you are only allowed to use.



We will discuss more on cardinalities later.

**Adding The “people” Table To The ERD:**

The **people** table, should have 3 columns:

* **p\_id** **INT** **AUTO\_INCREMENT** and is the **PK** for the table
* **first\_name** and **last\_name**, both could be at **VARCHAR(40)** and **NOT NULL**
  + *following the full\_name column removal*

In Draw.IO, grab your only Table option (from the Shapes section) and place it on your diagram, it will appear as:

A table with numbers and dots

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated

**Select/Remove a Row **

Please make note of the equal sign, at the end of the column(row), if you click on , it selects the row, and you can hit your Delete button, it will delete that column(row).

**Add a New Row**

Also make not of the blue (very faint) arrow, next to Row 3. My mouse pointer was simply over top of that row, and by clicking that blue arrow, which will turn darker when you highlight it and it will add a new column(row) to the Table shape. This is the only way to properly add columns(rows), you can see it simply duplicates the previous row. You never need to do this for your PK, as it should always be one column.

So any of the current Row # have this blue arrow capability. If you do a Copy-Paste of a column(row), it may not work as you want it to, the only way to fix it, is to click Undo until the mis-formatted column(row) is removed, and then use the Hover Over Column(Row) and Click the Blue Arrow action.

Careful, if you are too close to another table, the blue arrow may also try to draw a relationship line to that other table, so create your tables below any existing tables, then move them up once you’ve completed the column(row) adds.

You can double click into any of the column(row) cells thereafter. In the ERD, the columns are drawn as rows, as within the master database, technically, all of your schema’s structure is stored as rows. So you just need to understand why we refer to them as column(row) here, technically they are columns rows.

In Class Activity

From your existing people table, using the Table shape you grabbed above. You need 3 column(rows) in your diagram, you need to also mark the people\_id as your PK (so just change the default value of UniqueID), and you should also specify the data types, though do them as abbreviations. Exclude full\_name if you still have it in your table, it will be removed shortly.  
  
A diagram of a person's name

Description automatically generated with medium confidence

Use abbreviations, for datatypes and NN (NOT NULL) for columns that have to have a value. In this case, last\_name does not require a last name thus NULL is the default and does not need to be entered to the diagram…as maybe you will can have Cher as an employee or Madonna as a Customer…with any luck.  
  
We still need to add some fields to the people, as we are storing the data for employees, customers, manufacturers’/distributors’ representatives and therefore require need the people table to have more fields.

Here is a legend you can work with:

A screenshot of a computer

Description automatically generated

**Next up…**

We will be adding more fields to the people table to make unique records and add people current metadata.

## 