

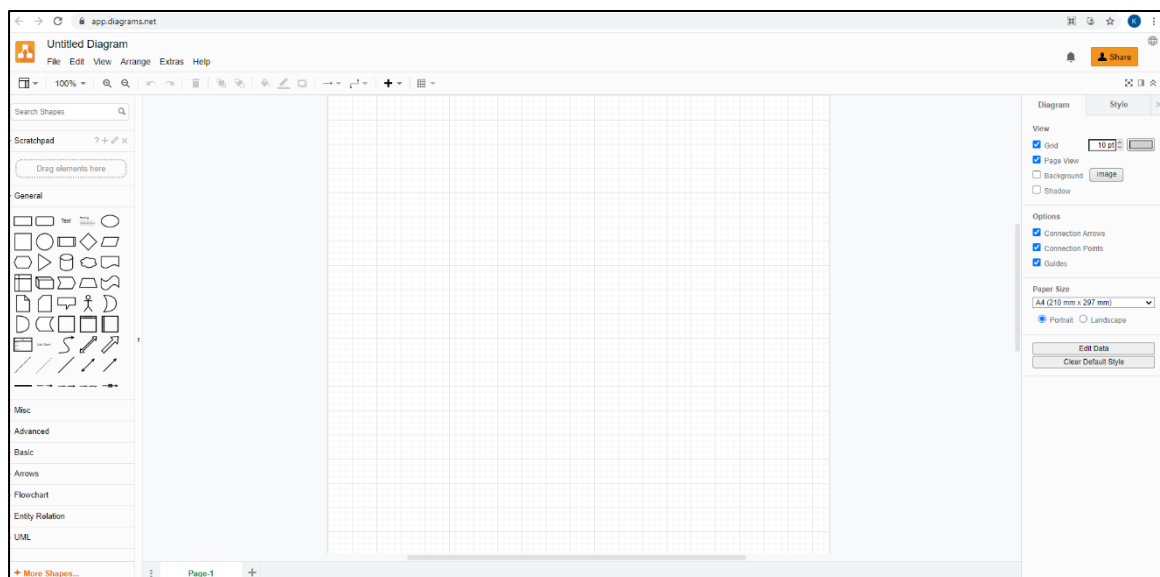
## Lab Manual (Logical and Physical Schema)

### Week-1

The contents of this lab are based on using tool to design and model Logical and Physical Schema. Here, we are using a free tool “io.draw” which is available online from anywhere. The learners will be exploring the ER-Diagram features as part of Logical Schema design and Data Modelling as part of Physical Schema design using this tool in today’s Lab.

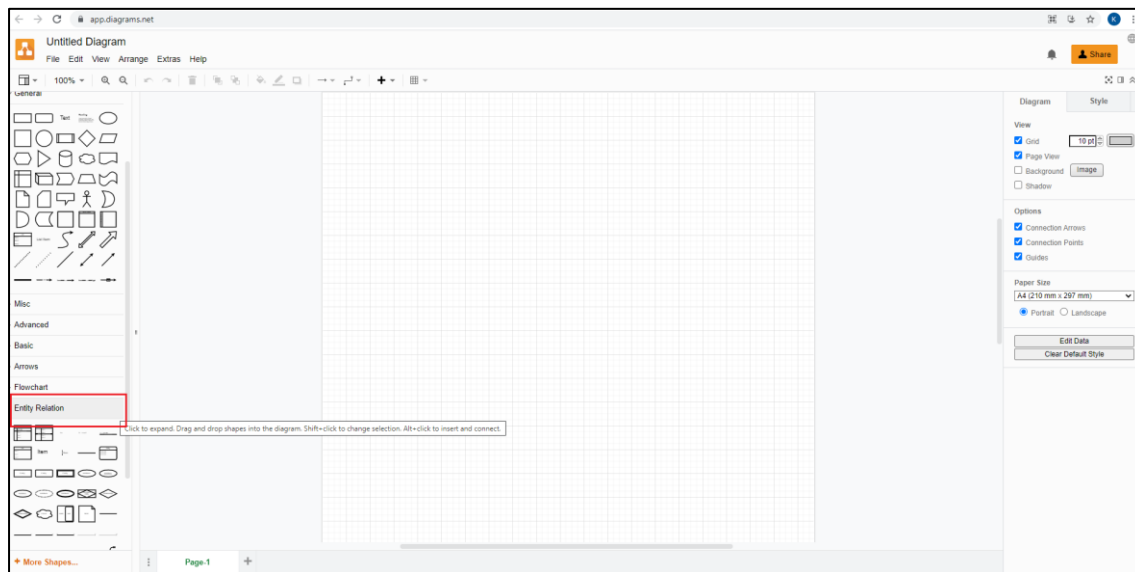
To begin with, you will follow this link: <https://app.diagrams.net/>

You will then see this interface on your browser:



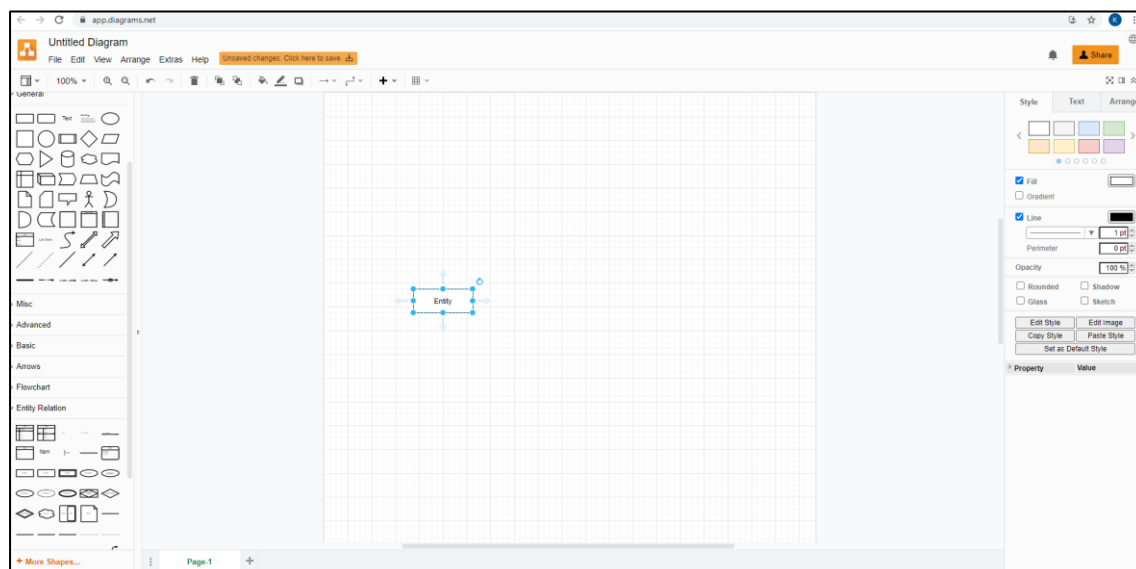
There are plenty different Design options available for you, however, for this module in this step you choose “Entity Relation” design option for designing your ER Diagram:

## Logical Schema Design:

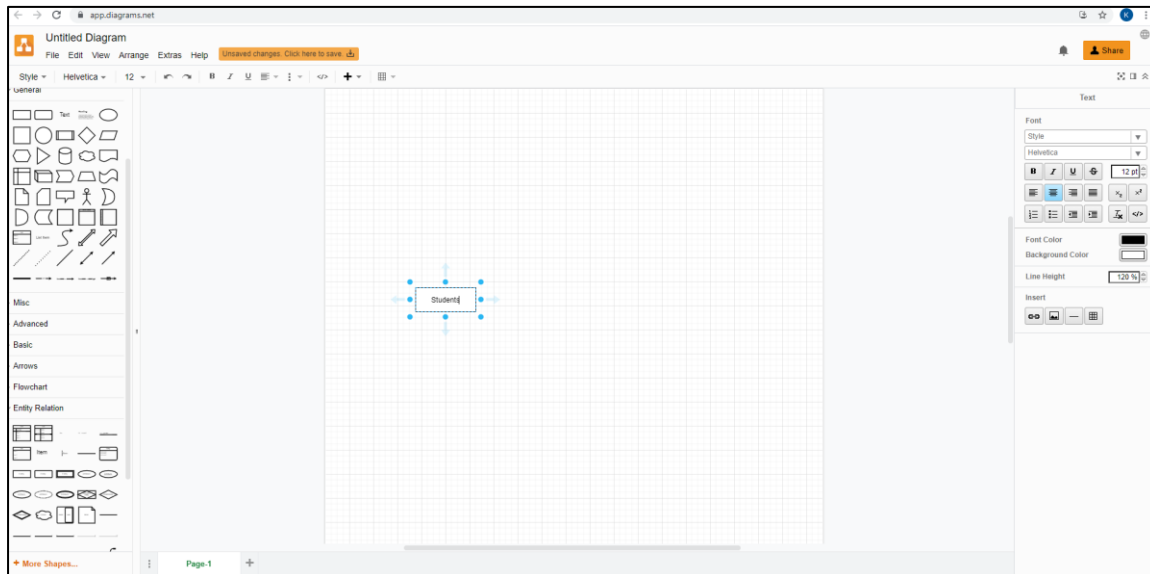


Simply by picking correct symbol(s) for the diagram and by dragging and dropping them to the work area of the tool you continue to proceed your design. You already know from the earlier discussion today that Entity is represented by “Rectangle”:

Selecting an Entity:

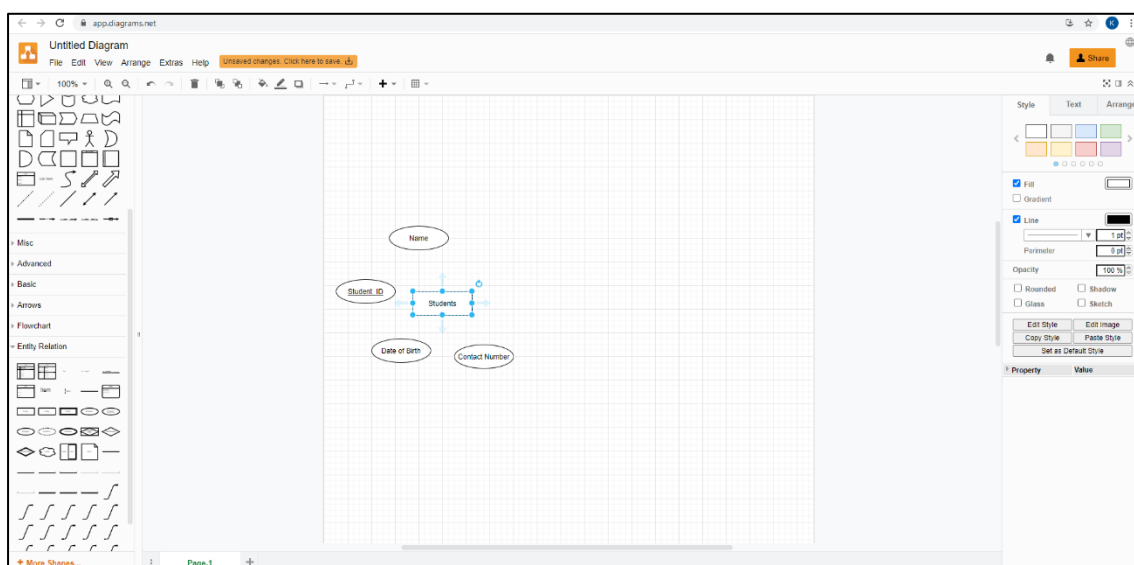


Naming the Entity with a proper and purposeful name for a Database:

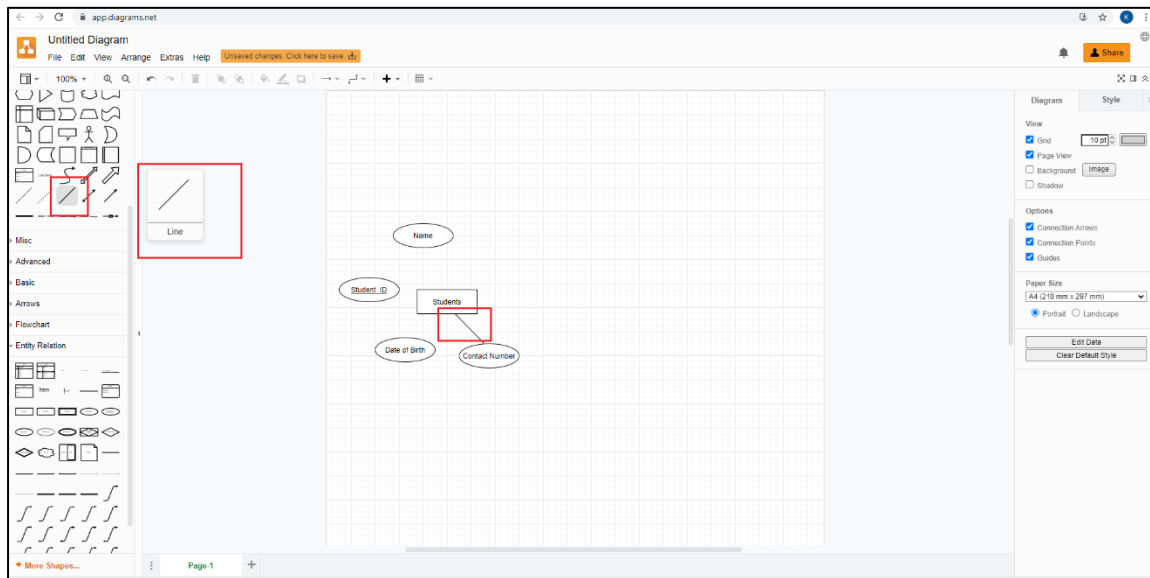


Now, we will add Attributes to the Entity “Students” by again dragging and dropping correct symbol of Attribute(s) in the ER-Diagram. Attribute in ER-Diagram is represented using the symbol of “Ellipse”:

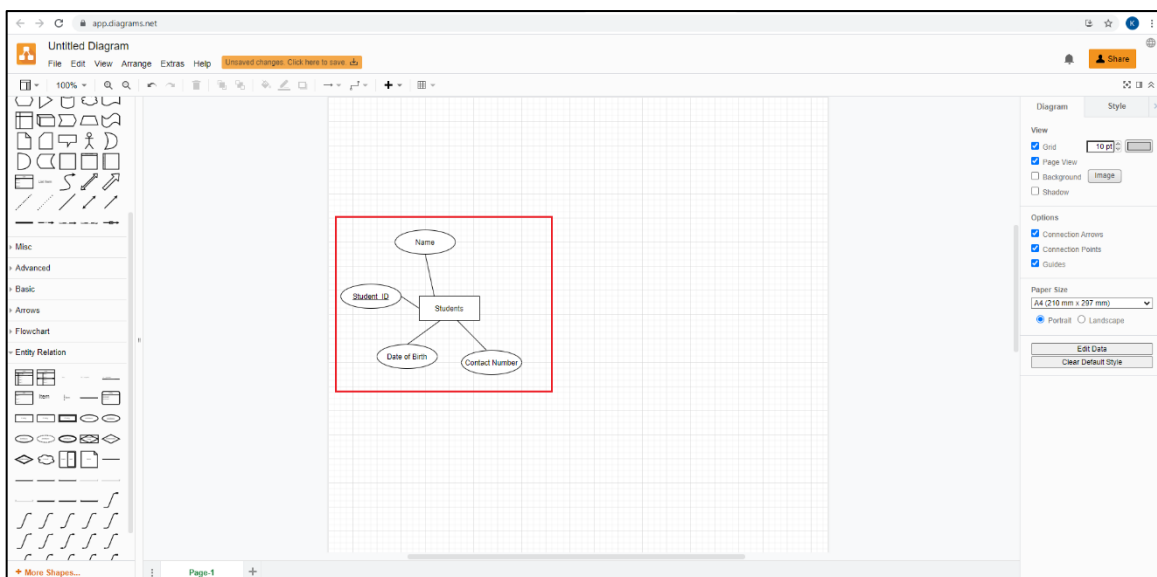
Here, we have dragged and dropped four attributes for the “Students” entity and one of them is a keyed attribute (with underline):



In this step we are connecting the attributes with the entity. Remember Attributes get connected with the entity with Straight line only. It's show below:

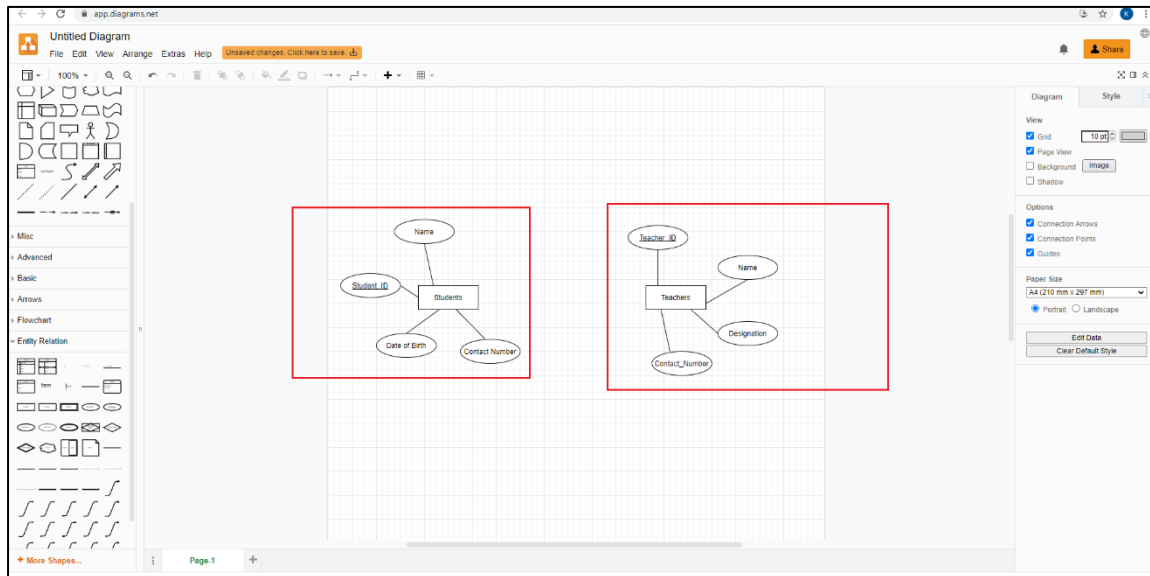


Similarly, you can connect all the attributes with the Entity “Students”:

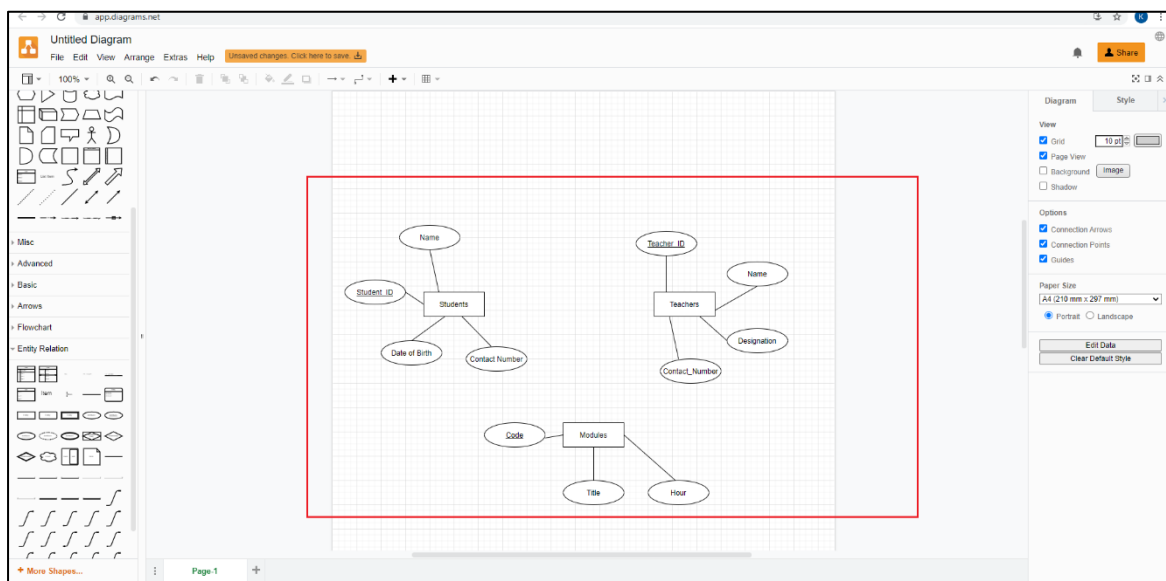


In an ER-Diagram, we usually have two or, more entities and then, based on the requirement these entities establish relationship(s) with the other entities.

We will now design another entity “Teachers” by following the same necessary steps:

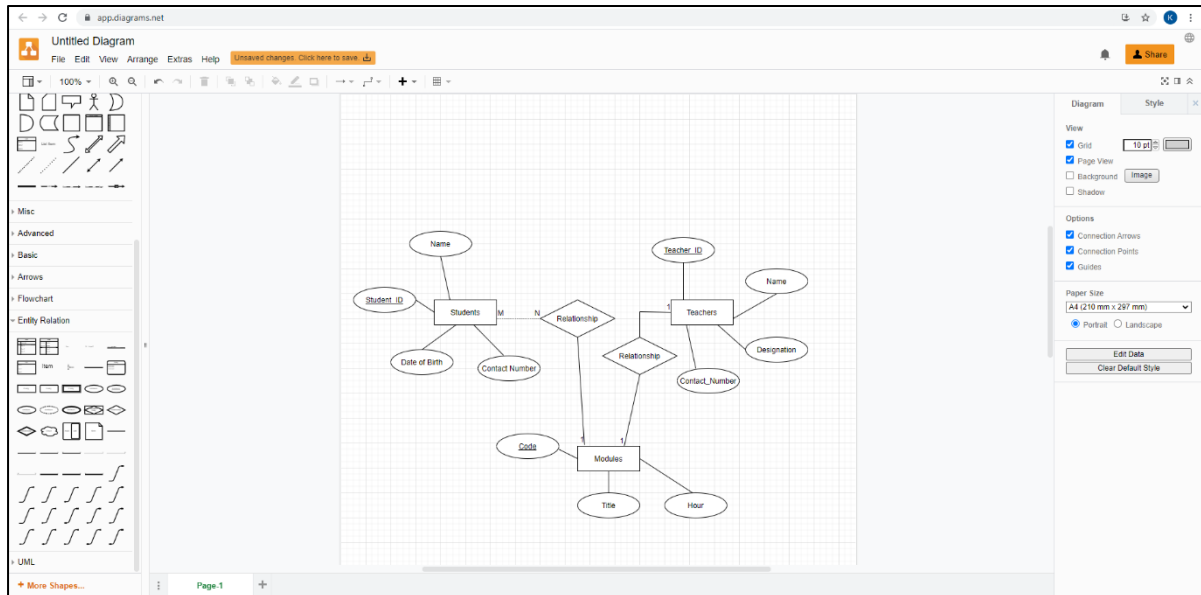


Why don't we include another entity in the ER-Diagram we designing? Let's now design another Entity, which we will name as "Modules":



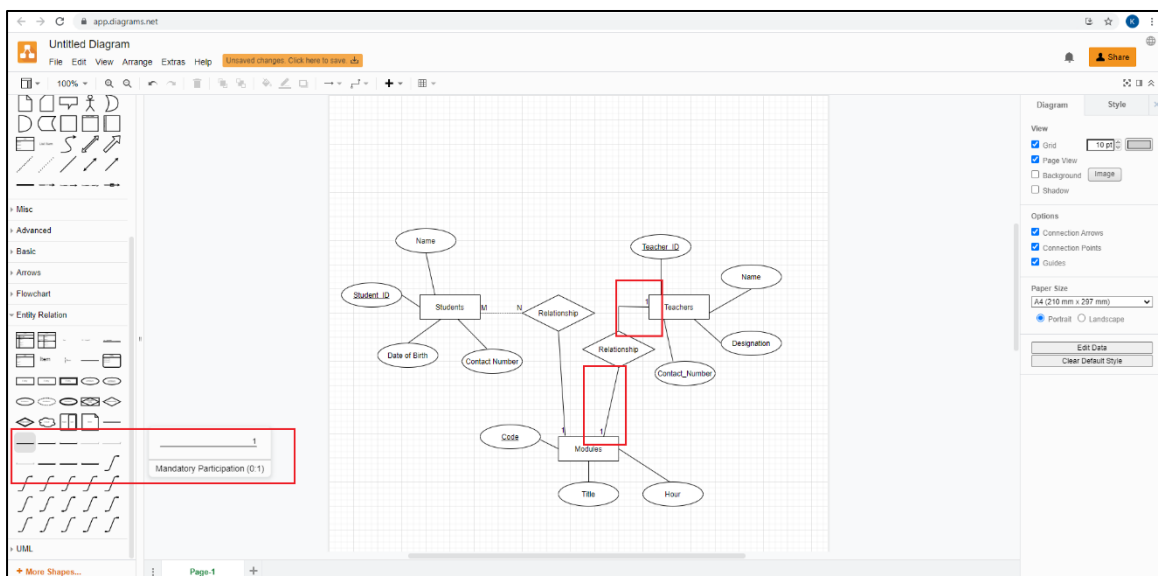
We now have three entities present but, none of them are related so, in this step we will provide relationships among the entities present. Assume the relationships among these three Entities are described as, "One teacher teaches only one module, and a module is learnt by many students".

The Relationships in the ER-Diagram are represented using “Diamond” symbol:

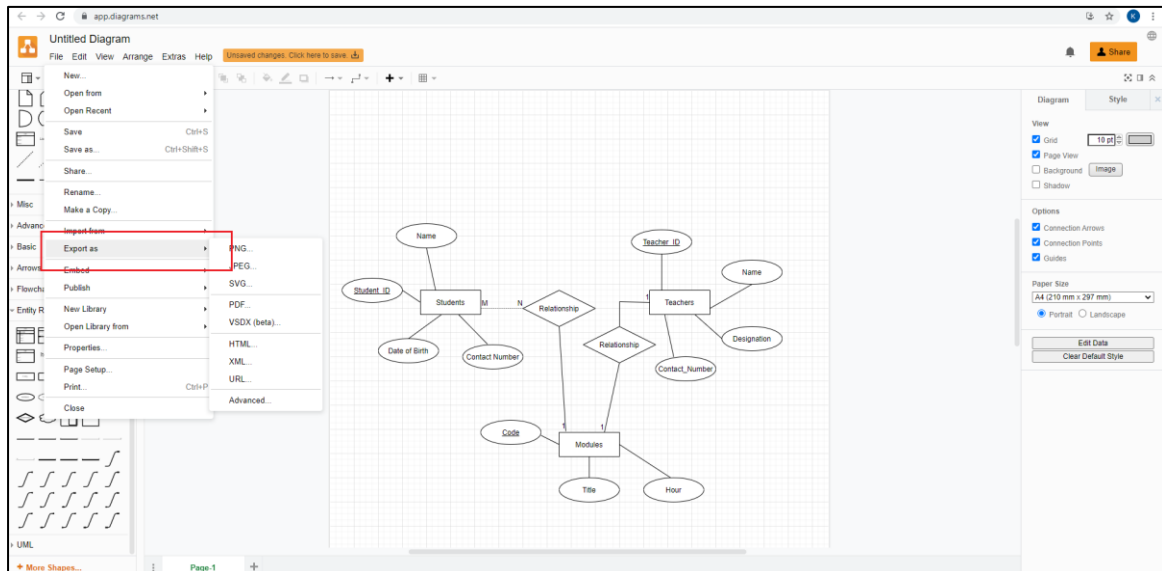


Here the “1....M”, “1.....1” with the relationships are called Cardinality mapping, which you have known from our discussion earlier today.

You can select the appropriate type(s) of mapping from the list and by dragging and dropping include the most suitable ones for the diagram while designing:



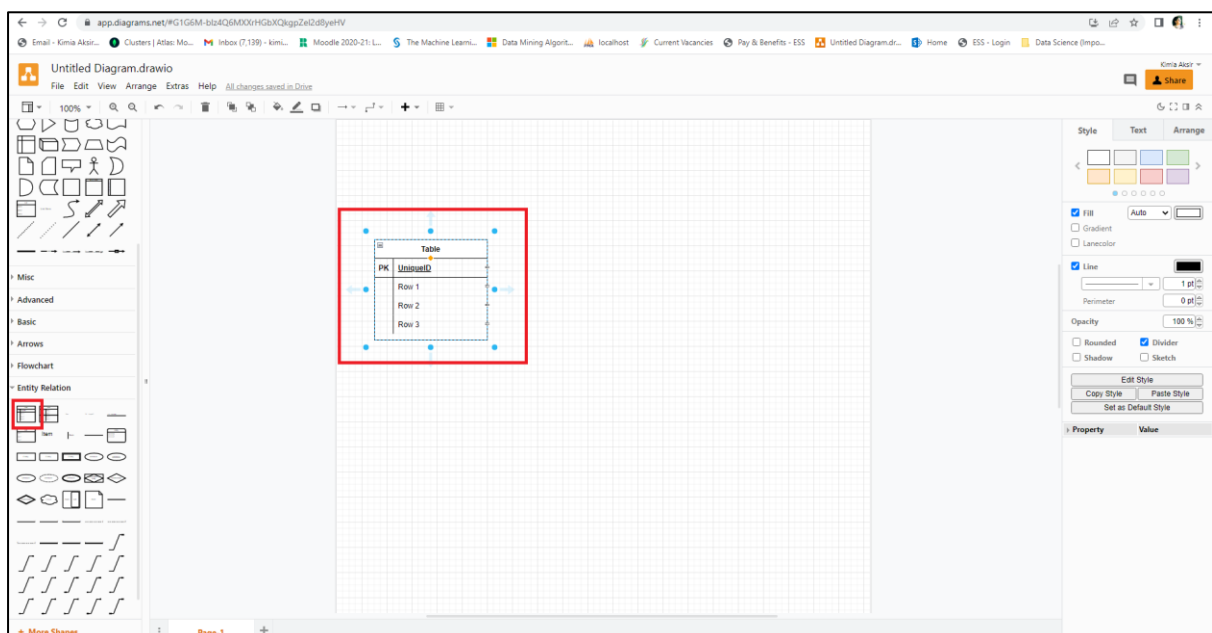
You can then save/export your completed work to your device:



## Physical Schema Design:

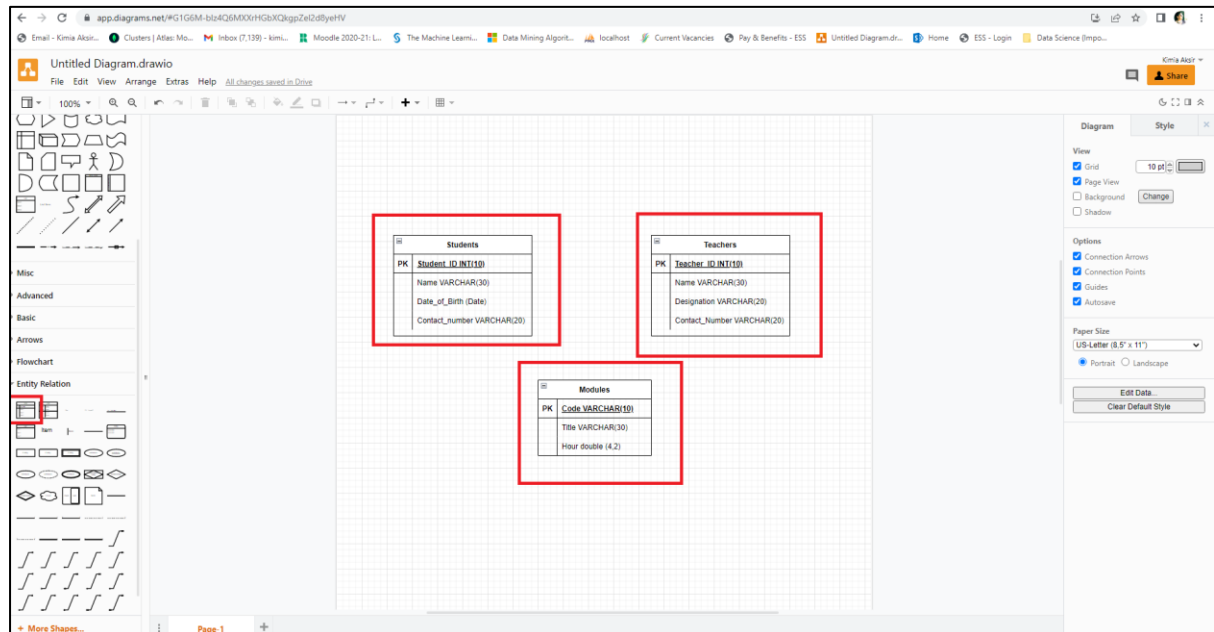
Again in this step you choose “Entity Relation” design option for designing the Physical Schema/Physical Data Model:

Now you can drag a suitable design option from “Entity Relation” category to represent each entity as a “database table” in the Data Model:



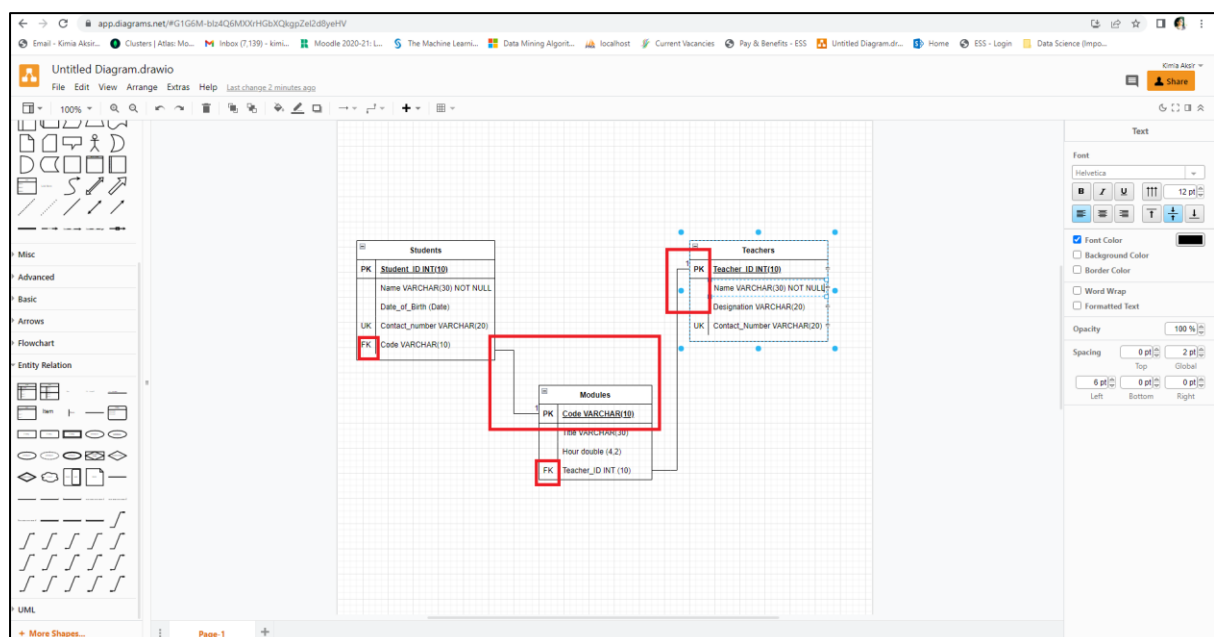
You are designing a Data Model based on the logical Schema you designed in the previous step. To represent Students, Teachers, and Modules, you will

need three objects as follows. You are now providing titles to these objects exactly as you want to title them as actual tables in the Database. Also, you will record the attributes accordingly as shown below:



The relationships/connections among the tables does not require to be of diamond shapes like in the Logical Schema, rather you can now do “Referencing” using Foreign Keys as shown below.

You can also add other constraints e.g, Unique Key (UK), NOT NULL etc., if you are given with such requirements.





Same as before you can now download the Data Model.

Now, You can start completing the Lab task for this week!