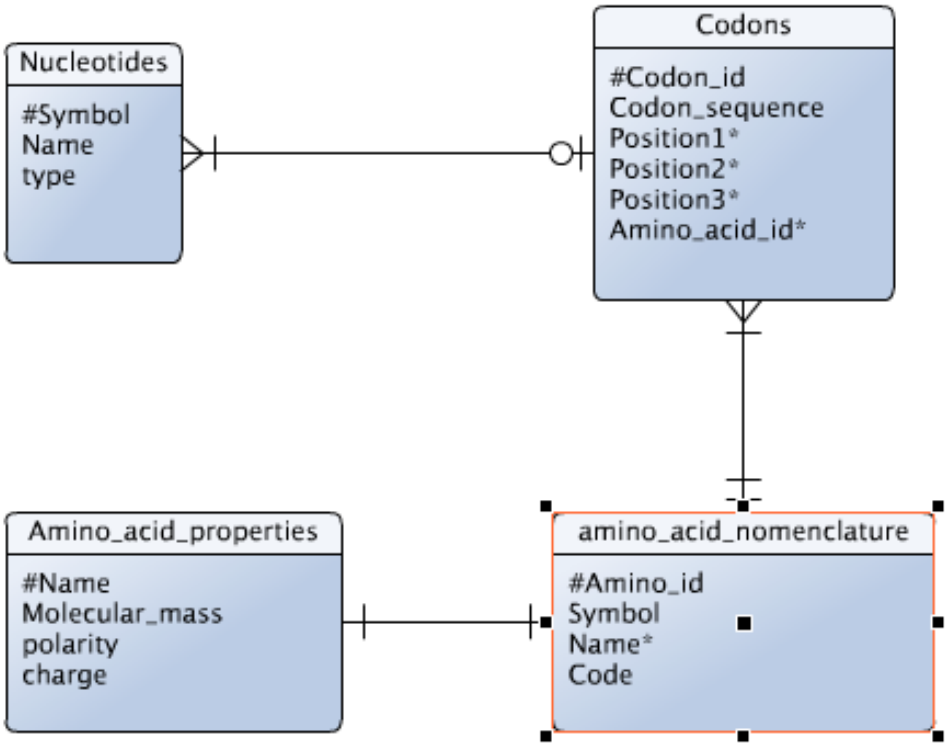
# INF115: Databases Assigment 2

by

*Hussam Samir Yousif*
*hyo064@student.uib.no*
*207460*

INF115 Databases Compulsory Assigment 2: Hussam Samir Yousif, hyo064, 207460

# Task 1.




Fairly simple ER-Diagram with 4 tables, from my understanding of the biology at hand.

Note I've highlighted the foreign keys with "*" after the attribute name.

# Task 2 I)

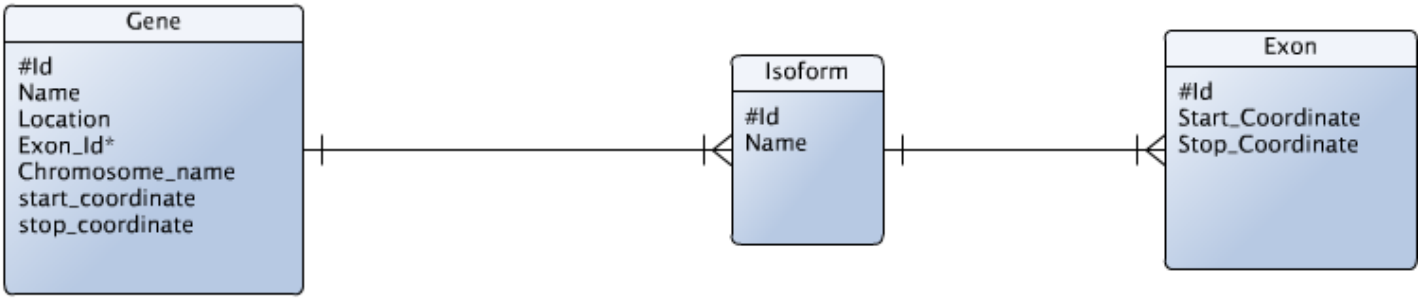
The entities are the following:

- Chromosome(#Id, Name)
- Gene_Location(#Gene_location, start, stop, chromosome_id*)
- Gene(#Gene_id, Name, Gene_location*)
- Exon(#Exon_id, Exon_location*)
- Exon_Location(#Exon_Location, Start_Coordinate, Stop_Coordinate, Isoform_name*)
- Isoform(#Isoform_id, Isoform_name)

Note: this is a diagram I made before seeing that the next task asks us for a naive diagram. By naive I mean one which does not conform to higher Normalization forms.

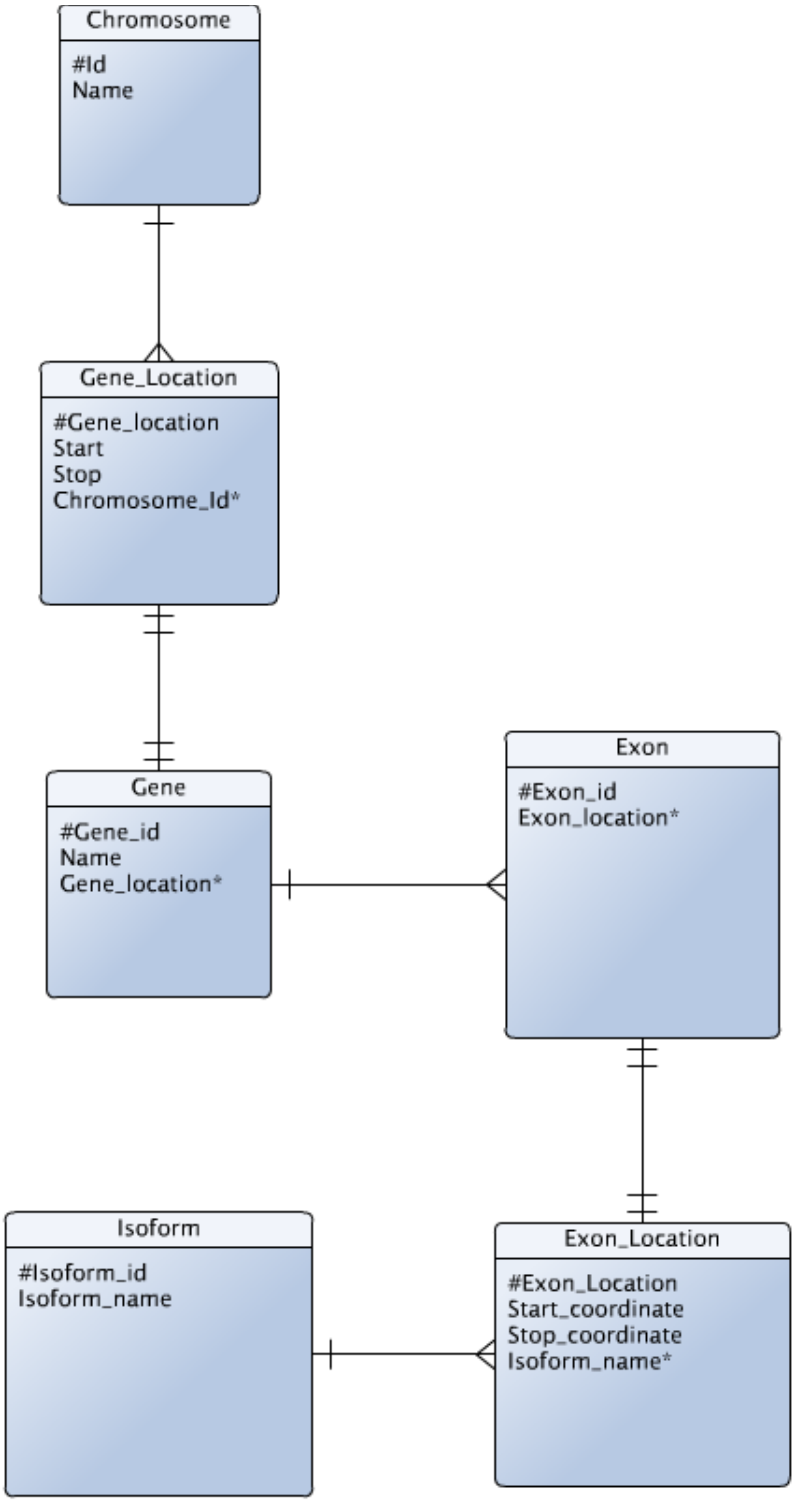
# Task 2 II)

The following is a naive graph which I tried to make violated the first normalization form on purpose such that it wouldn't conflict with task 2 III. For instance, I made it so that Gene contains Exon_id an attribute that will have multiple values. That one attribute will violate the first normalization form and as such give a "naive" database.




# Task 2 III)

INF115 Databases Compulsory Assigment 2: Hussam Samir Yousif, hyo064, 207460




This is the proper diagram for Task 2.

INF115 Databases Compulsory Assigment 2: Hussam Samir Yousif, hyo064, 207460

First Normalization form: I've structured the graph in a way such that no column would return two or more values. In mathemathical terms we say that it's one to one or injective.

Second Normalization form and Third Normalization form: There are no partial or transitive dependencies. This is the reason I have Gene_Location and Exon_Location. For instance the start and stop Coordinates doesn't depend on the Exon_Id attribute, but on the Exon_Location attribute and as such were given their own table.

# Task 3 I)

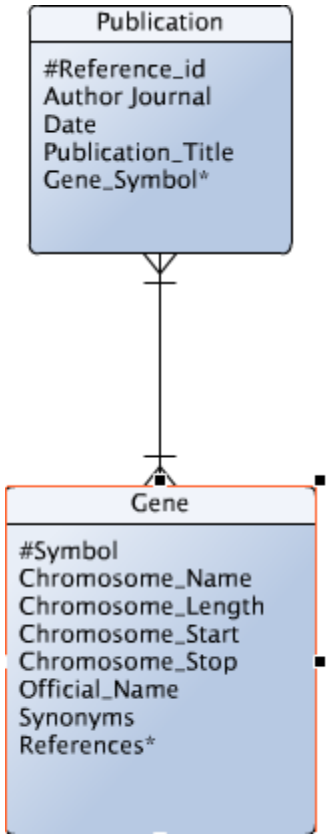
Again I'll write it Naively at first and then evolve the table as we get further in the task.

The naive and simple implementation would be to have two entities:

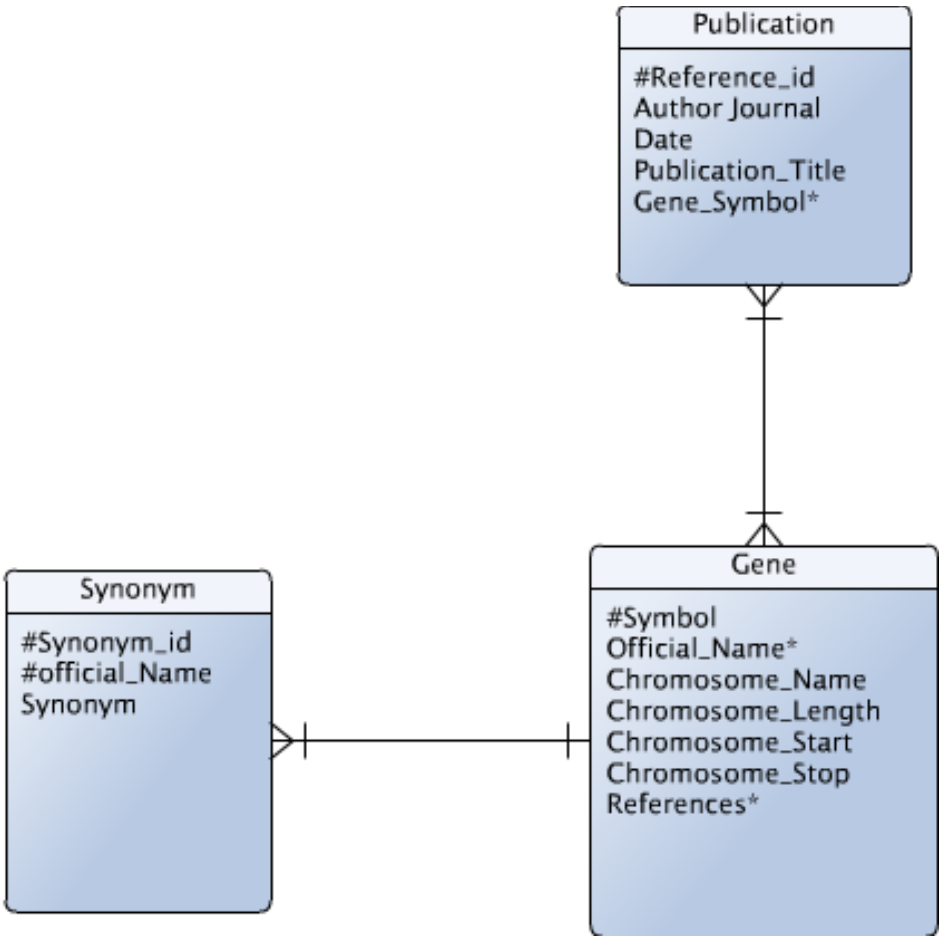
- Publication(#Referenence_id, Author, Gene_Symbol, Journal, Date, Publication_Title)
- Gene(#Gene_Symbol. Chromosome_Name, Chromosome_Length, Chromosome_Start, Chromosome_Stop, References*, official_Name, Synonyms)

To make it not conform to the first normal form I use the entry Synonyms which will contain multiple values according to the task at hand. Using Synonyms this way will make this Diagram not conform with the First Normalization Axioms.

# Task 3 II)



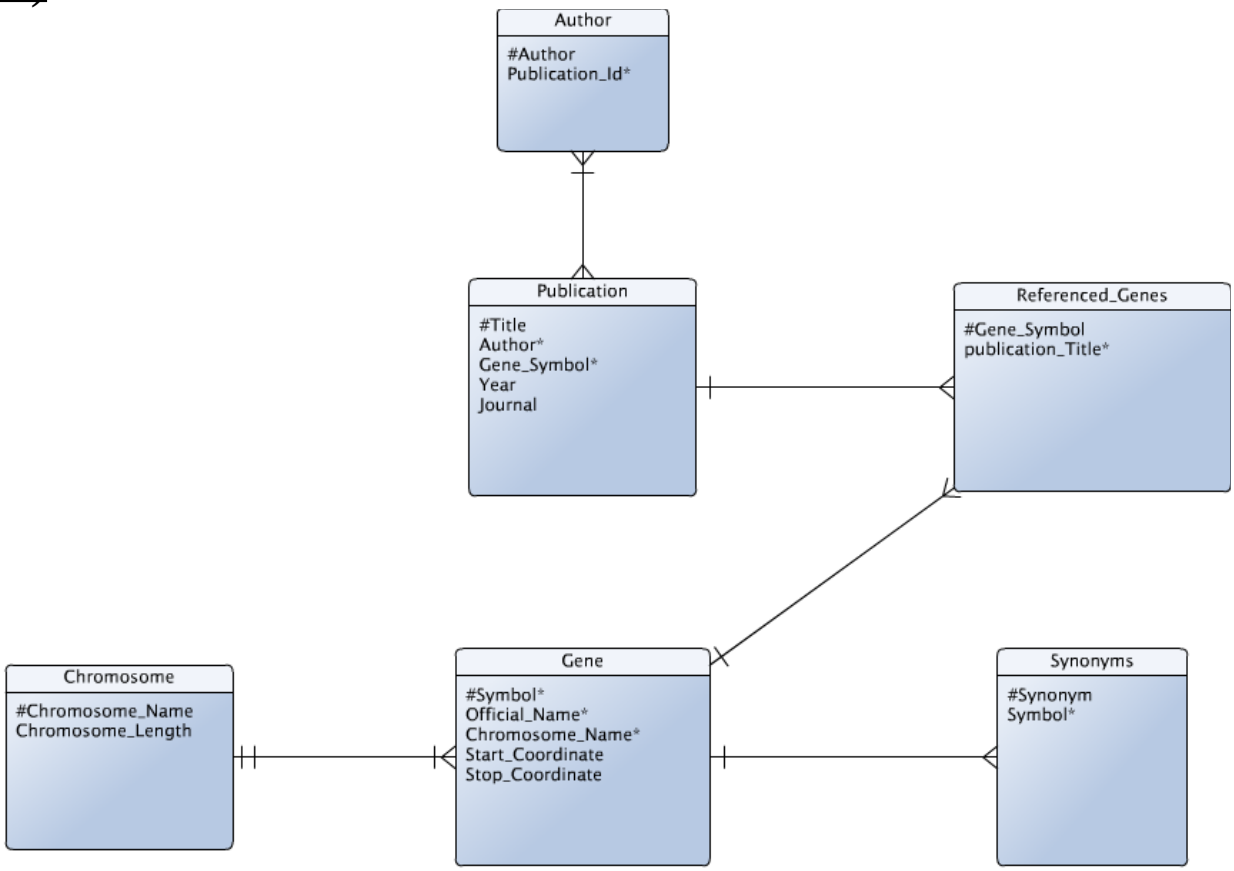
# Task 3 III)




This now conforms with the First Normalization form as Synonyms have their own table and as such will be injective (will not have multiple values in a column).

INF115 Databases Compulsory Assigment 2: Hussam Samir Yousif, hyo064, 207460

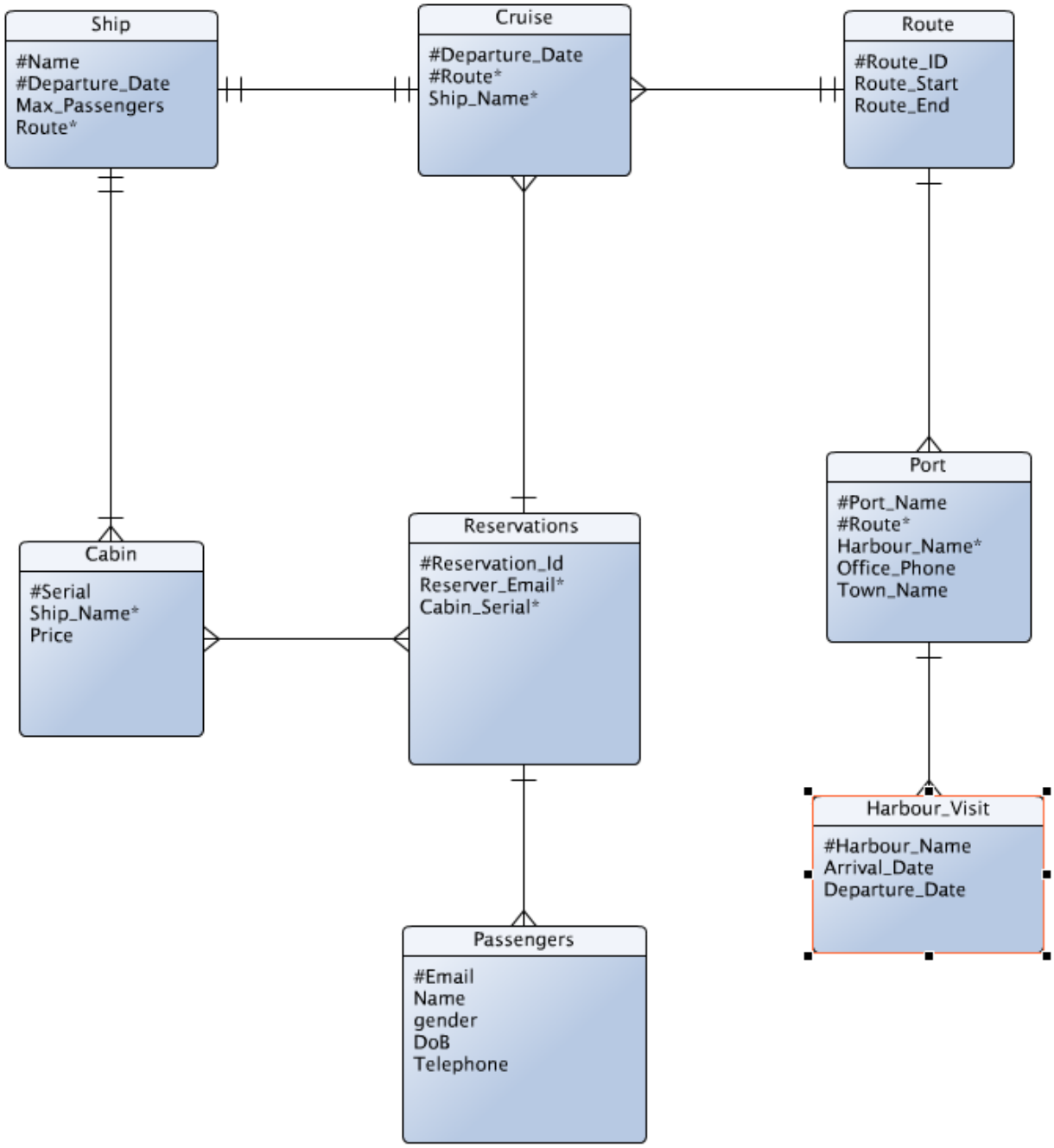
# Task 3 IV)




I've split all the dependencies and made it such that all the primary keys are super keys (BCNF). Furthermore there are no multiple entries in any column (again Synonyms takes care of that) and I've split of the entries that do not depend on the table. For instance Chromosome attributes are not dependent on Gene and as such was split.

INF115 Databases Compulsory Assigment 2: Hussam Samir Yousif, hyo064, 207460

# Task 4)




Note: The group leader told me that having a loop was ok in this task and as such I structured my database in this manner.

Honestly it’s a fairly simple diagram that follows the instructions of the assignment to the letter.

Everything revolves around the cruise. Which has a ship and a route. A route has a port and each port has multiple Harbour_visits. I didn't have a Harbour array because it wasn't necessary for the task at hand, our main concern is the cruise, not the port itself. A ship has cabins and the primary keys of cabins contains both the floor and cabin number. I made it so because the assignment provided a string example "4-11". Furthermore we have a reservation which is linked both to the cruise and passengers.

Note: DoB = Date of Birth.

# Task 5 I)

There are multiple things that make this designed problematic:

- The cubic quantity is not assigned
- Truck doesn't have a primary key
- Model can and should have a different array as the fields Model and Max_Weight doesn't depend on assignments or Registration_Year.

# Task 5 II)

The functional dependencies are fairly simple in this task. We see that Registration information fit together and Model information fit together.

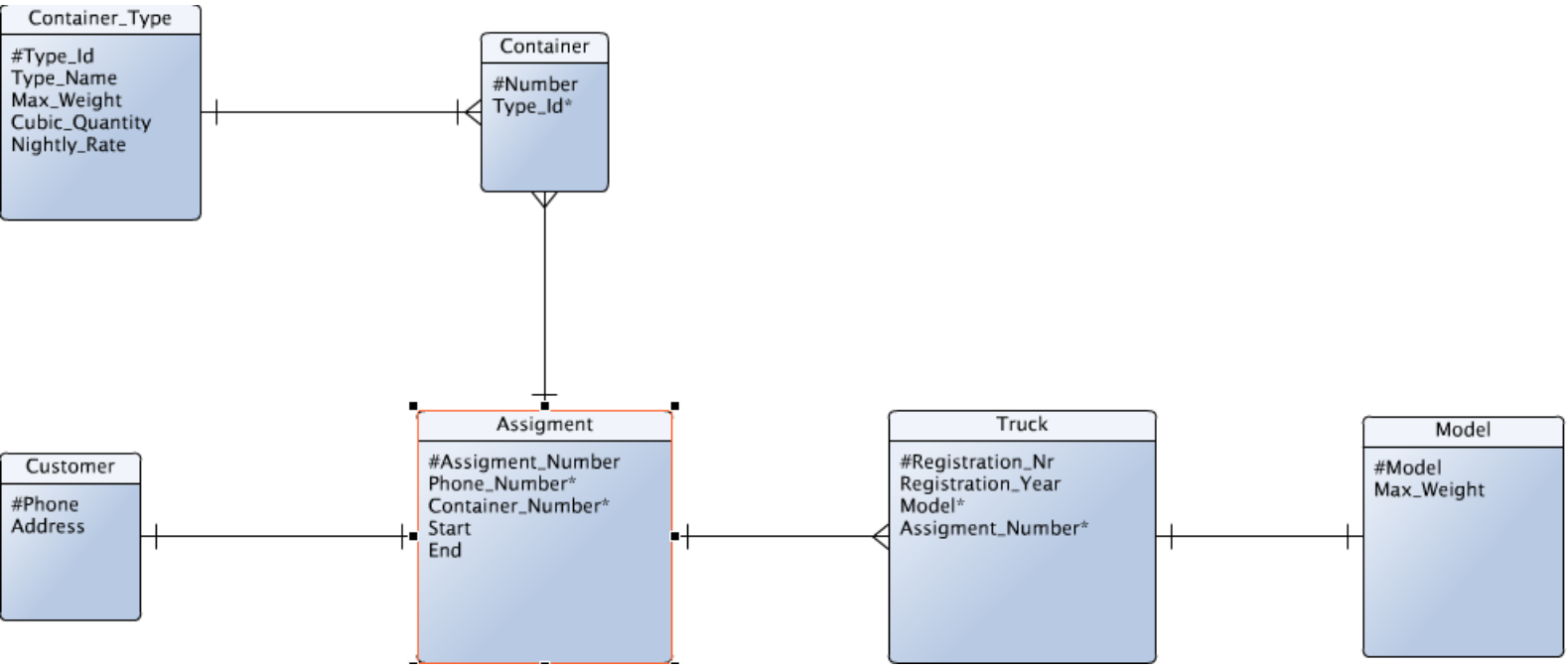
- Registration_Nr → Registration_Year
- Model → Max_Weight

# Task 5 III)

A candidate key is a key that determines other attributes. As such The most appropriate candidate keys are the following:

- Registration_Nr
- Model

# Task 5 IV)




The first given arrays were all good except for the truck one. I split off the model as I saw that the Max_Weight attribute transitively depends on Registration number. As such the graph has attained BCNF.