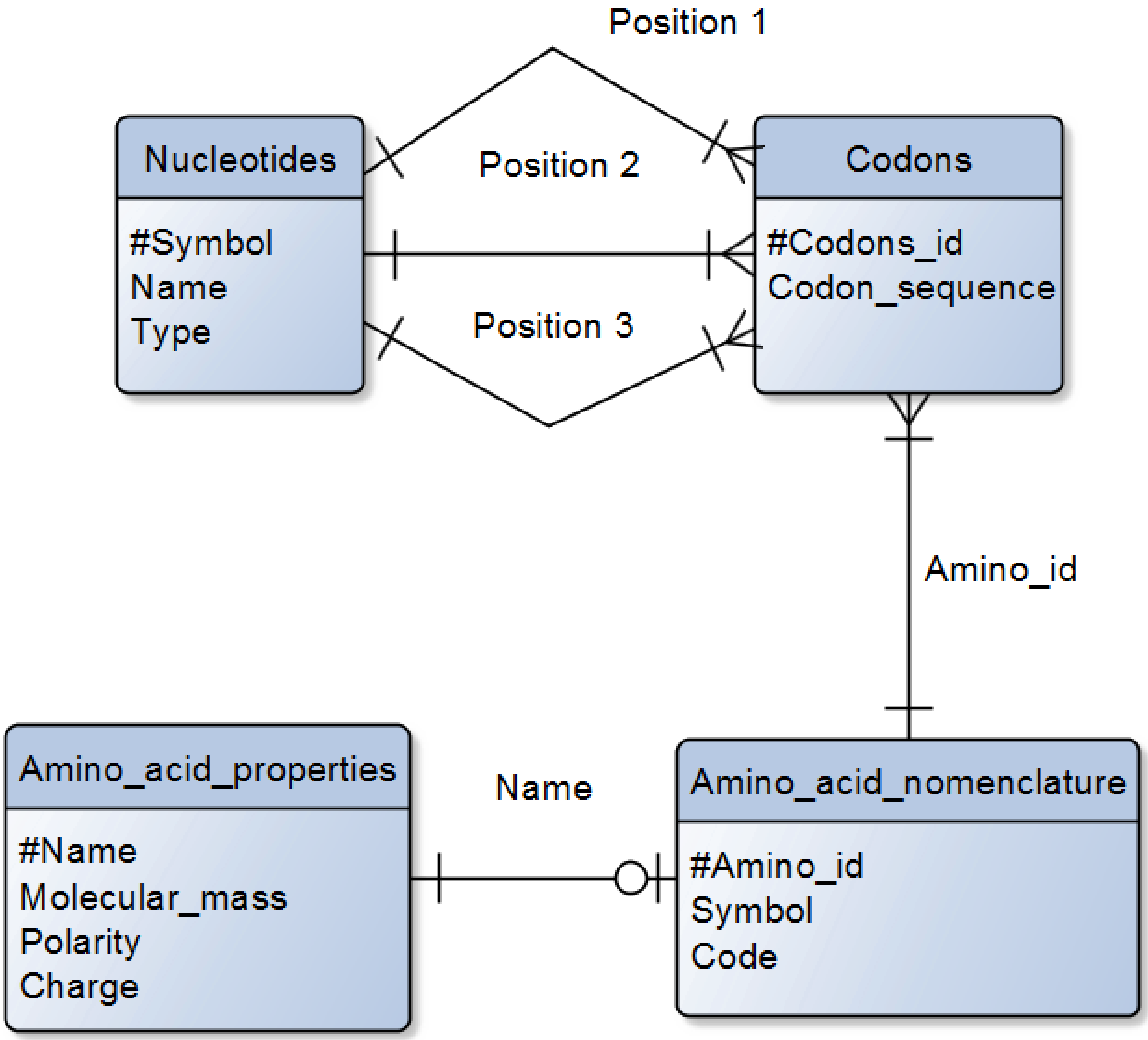
INF115 CA2

Task 1

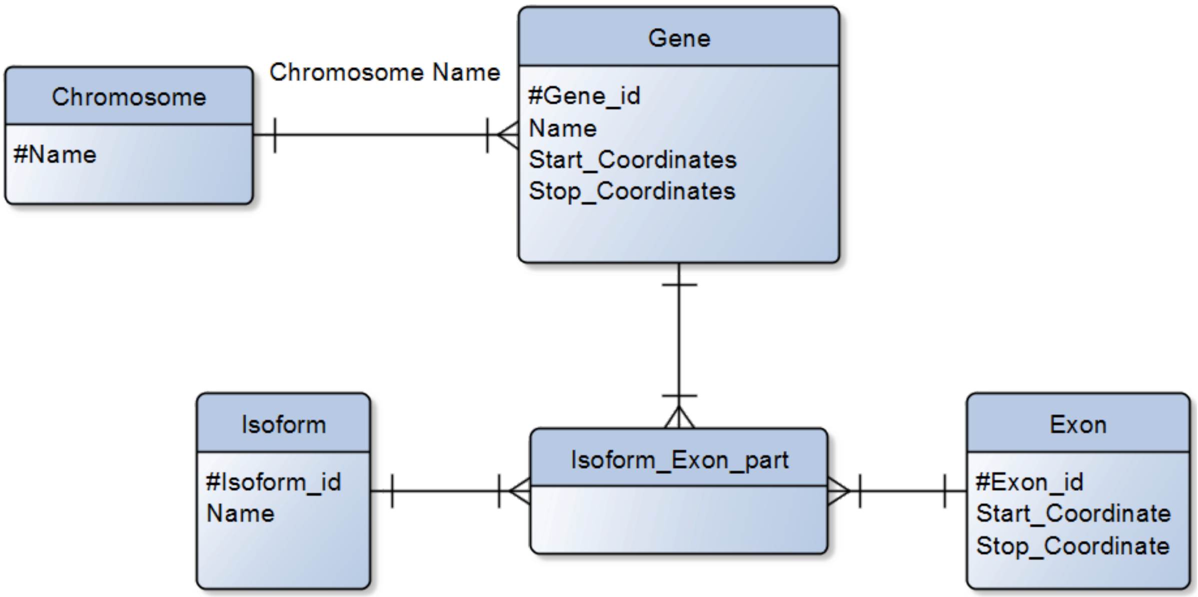


Task 2

i) Entities :

- Genes
- Exons
- Isoforms
- Chromosome

ii)



iii) 3NF Table

- Chromosome (#Name)
- Gene(#Gene_id, Name*)
- Gene_nomenclature(#Gene_name, Chromosome_name*, Start_coordinates, Stop_coordinates)
- Isoform_Exon_part(Gene_id*, Isoform_id*, Exon_id*)
- Isoform(#Isoform_id, Name)
- Exon(#Exon_id, Start_coordinate, Stop_coordinate)

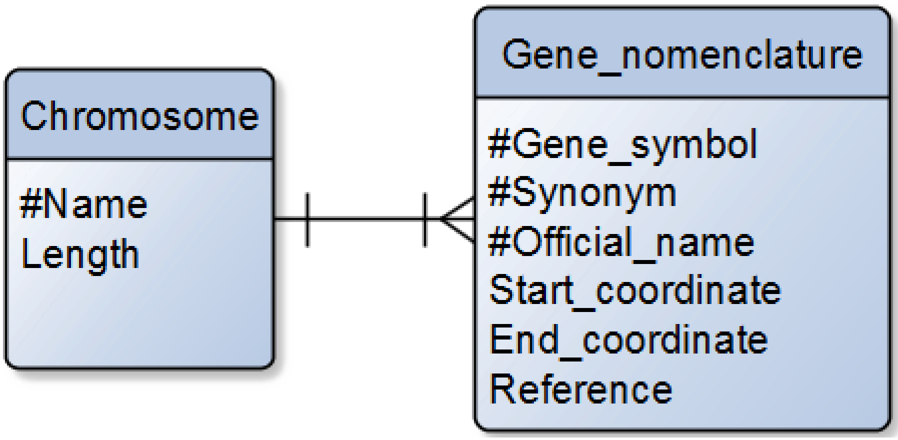
I think i could use Gene_id and Gene_name as a combinet primary key and avoid having two tables. Since having just Gene_id as a primary key and having the functionality : Gene_id -> Chromosome_name, but if i have gene_name in same table as gene_id i would have gene_name -> Chromosome_name as there is 1 unique name to 1 unique id. So Gene_id -> gene_name is also there, so i would think a A -> B -> C functionality would not break NF3 even if its just A that is a primary key and not also B since there is one B for each A.

Task 3

i) Entity :

- Gene nomenclature
- Synonym (will not be used in ER diagram, to later create a 1NF)
- Reference (not be in use for next task)
- Chromosome

ii)



Marked Gene_symbol, Synonym and Official name all as primary key to identify them as a unique key to get a unique row of a gene.

iii) 1NF

- Gene_nomenclature(#Gene_symbol, #Synonym, #Offical_name, Chromosome_name*, Start_coordinate, End_coordinate, reference)
- Chromosome(#Name, Length)

Using Gene_symbol, Official_name and synonym (we could identify this uniqely without Official_name.) to identify a unique row, we now have a functional dependency from Synonym -> Reference for backward compatibility which Gene_symbol or Official_name would not have same dependencie toward, and therefore we have a partial dependency.

iv)

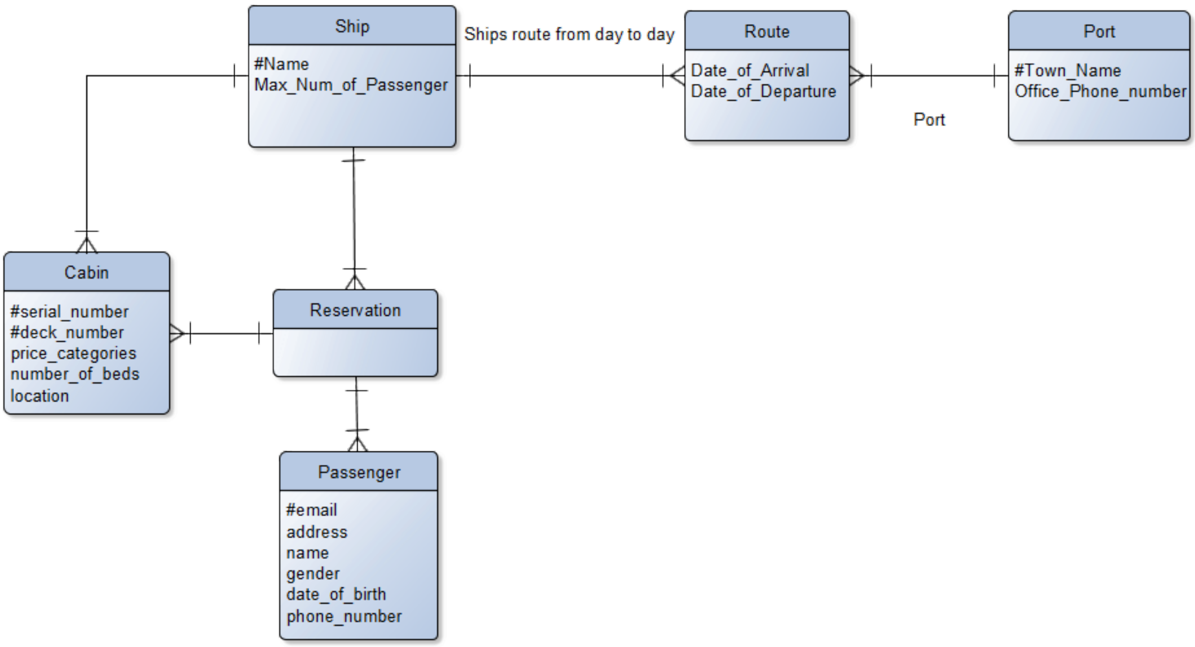
- Gene(#Offical_name, Gene_symbol*)
- Gene_nomenclature(#Gene_name, Chromosome_name*, Start_coordinates, Stop_coordinates)
- Gene nomenclature(#Gene_symbol, Chromosome_name*, Start_coordinate, End_coordinate)
- Reference(#Info, Gene_offical_name*, Synonym*)
- Synonym(#Synonym, Gene_offical_name*)
- Chromosome(#Name, Length)

Both gene_symbol and official_name are minimal candidate keys, and should not interfere in this beeing in BCNF. Reference is not in BCNF as there is a dependency from Synonym to Gene_official_name which makes it a determinant which is not a candidate key. So i think the easiest way to make it into BCNF is to split references into to tables : references and old_references, as i guess the old references were not updated to contain the official gene_name.

- Gene(#Offical_name, Gene_symbol*)
- Gene_nomenclature(#Gene_name, Chromosome_name*, Start_coordinates, Stop_coordinates)
- Reference(#Info, Gene_offical_name*)
- Old_Reference(#Info, Synonym*)
- Synonym(#Synonym, Gene_offical_name*)

- Chromosome(#Name, Length)

Task 4



The Serial_number and deck_number makes up for a combinet unique key to identify a cabin. A ship has multiple routes, which makes up its specific route it is following. A route without a Date_of_Arrival would be a ships start location, and a route without a Date_of_Departure would be a ships end destination for this tour.
I did not find out anyway to give the reservation a connection to having multiple Cabins and a Ship having multiple cabins which would not result in the circular E/R diagram shown.

Task 5

i) In the Truck table there is redundant information. Registration year will always be the same for a registration number, and the Maximum weight will always be the same for the model. For every Assignment the truck gets, the reduntant information will be added. If one would then for ex transfer the registration number to a new truck, one would need to update the whole database where it is needed in the future (Assume that one do not need to update information in allready done assignments.), i.e if you transfer the registration number from a Volvo XL to some other truck model, say Mercedes XXL, you would have to find every entry with this registration number and update its model and also its maximum weight. This is not a desired action, as this can lead to some rows not getting the update they need and the database will have some erroneous entries for assignements not finished.

ii) Functional dependencies

- Registration_number -> Registration_year
- Registration_number -> Model
- Model -> Maximum_weight

iii) Candidate keys

- Registration_number + Assignment_number

iv) BCNF Tables

Assume that each Assignement is for one customer only, and that each assignement can handle more than one container at a time, and use multiple trucks. And that trucks have multiple assignements, lets database handle that they dont assign a assignement to a truck that have a assignement with a End_date after the Start_date of the new assignement.

- Container_type (#Type_id, Type_name, Max_weight, Cubic_quantity, Nightly_rate)
- Container (#Container_number, Type_id*)
- Customer (#Telephone_number, Address)
- Assignment (#Assignment_number, Telephone_number*, Start_date, End_date)
- Assigned_container(Assignement_number*, Container_number*)
- Assigned_truck (Assignement_number*. Registration_number*)
- Truck(#Registration_number, Registration_year, Model*)
- Model(#Model, Maximum_weight)