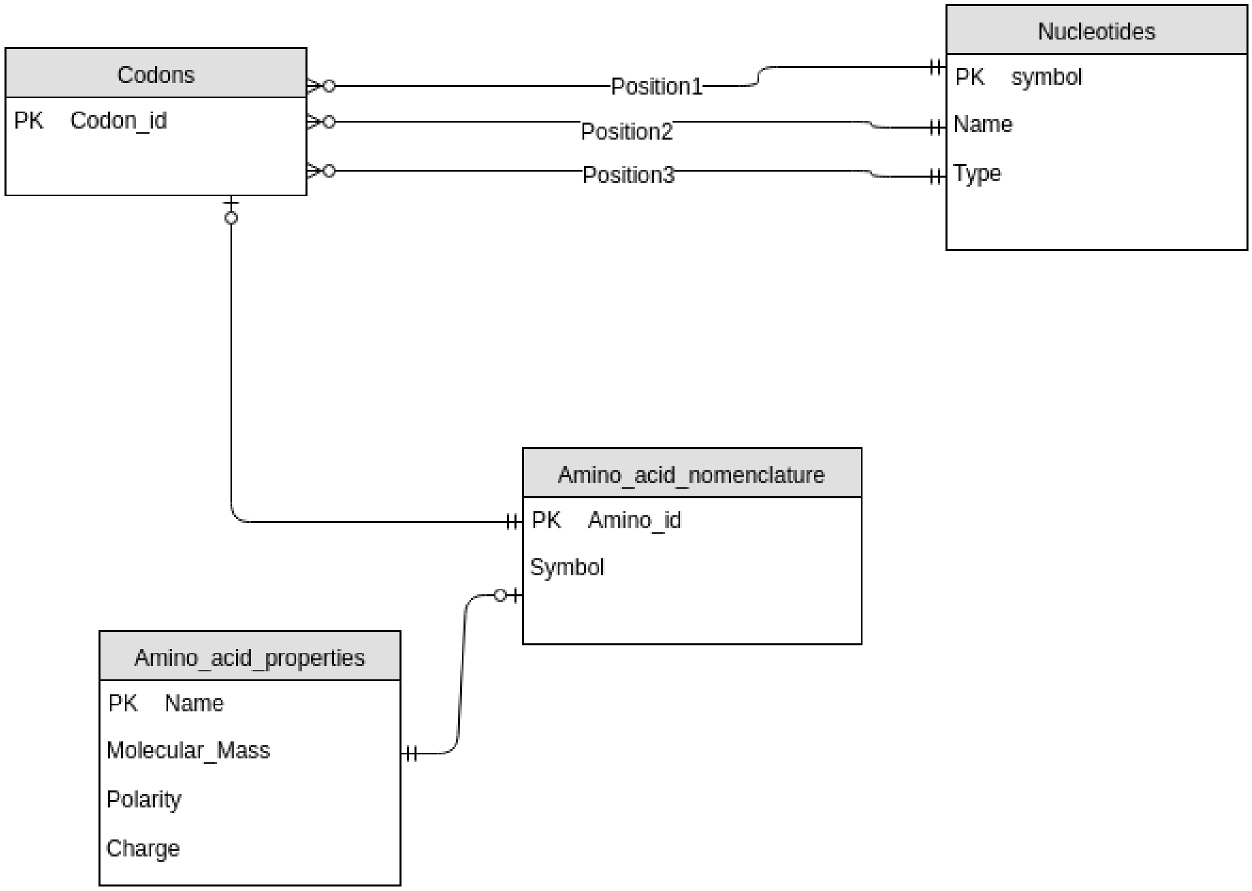
INF115 Compulsory Exercise 2

Task 1)

ER diagram of the tables in compulsory exercise 1



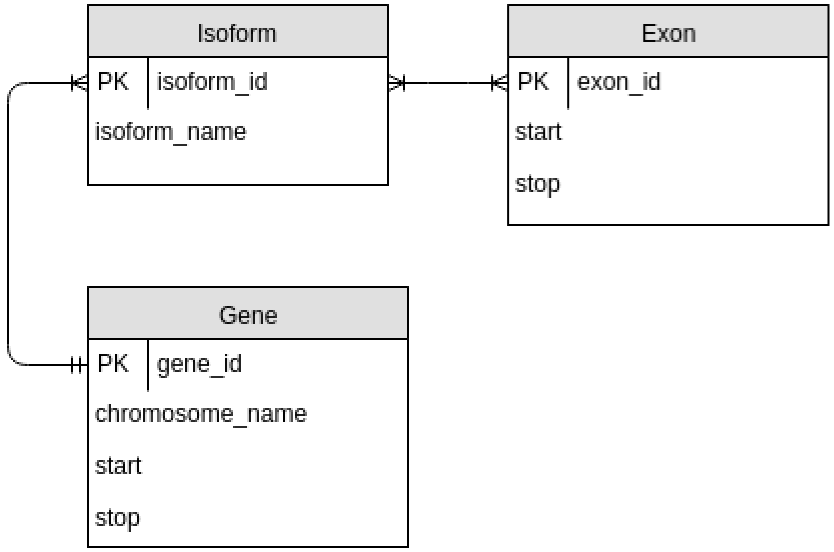
The primary keys are identified with the leading “PK”. Foreign keys in the tables are converted to relations. The codon_sequence attribute in the Codons table is also not here, as it just is a concatenation of the symbols in position 1, 2 and 3. Therefore it is omitted in this listing as a redundancy, but all the information is available in the ER-diagram.

Task 2)

i) Identify the entities in the database description

The entities in the database description are Gene, Exon and Isoform. As we only store the chromosome name it is not included as an entity, but instead stored in an attribute belonging to the gene entity.

ii) ER diagram of the identified entities



iii) ER diagram converted to a set of tables

Gene (#gene_id, chromosome_name, start, stop)
Isoform (#isoform_id, isoform_name, gene_id*)
Exon (#exon_id, start, stop)
Exon_in_isoform (#isoform_id*, #exon_id*)

The conversion is straightforward, with a linking table to model the exon_in_isoform relation. The set of tables is in 3NF because:

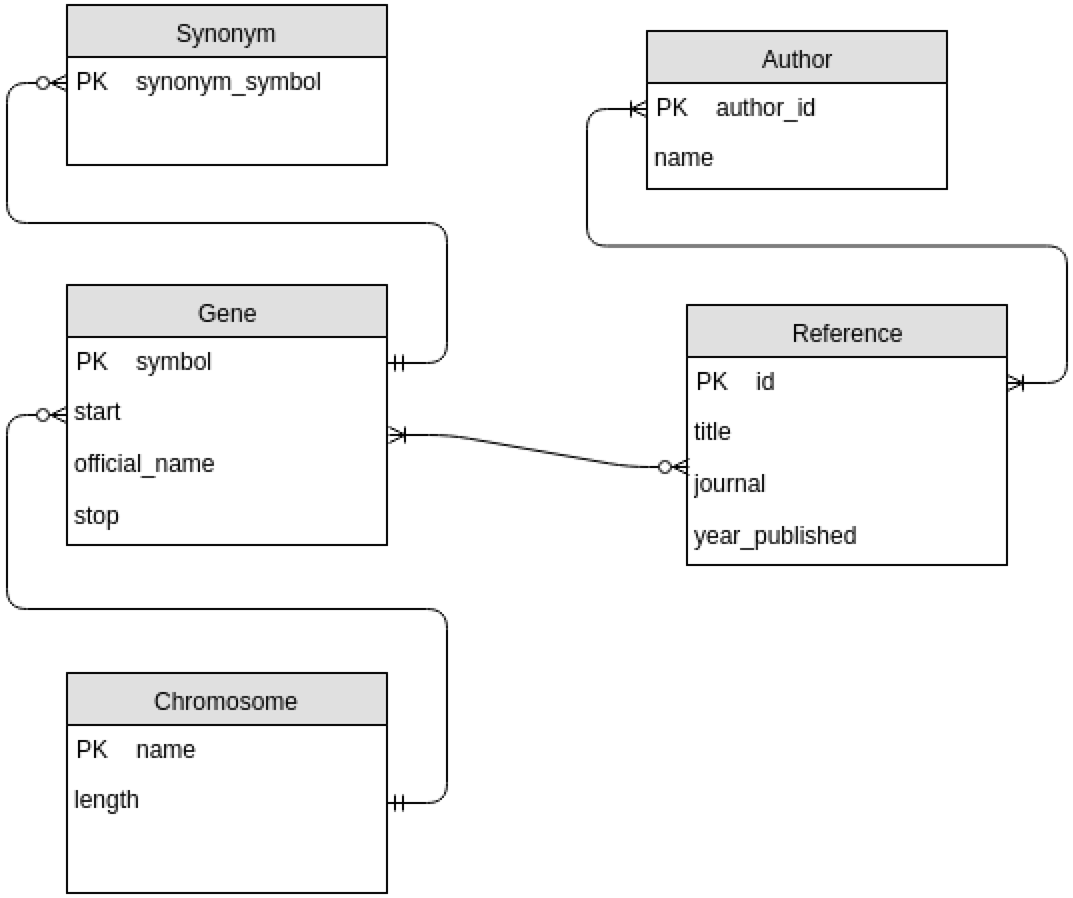
- Only atomic values makes it in 1NF
- No attribute is reliant on only part of the primary key (2NF)
- No attribute is reliant on less than the primary key (3NF)

Task 3)

i) The entities in the database description

The entities in the description is Synonym, Gene, Chromosome, Reference and Author. Author being a separate entity is not clear-cut from the description, but a field containing a list of authors is not atomic and would violate the first normal form.

ii) ER diagram of the database



When multiple genes are collected into one official gene, the non-official symbols of all the genes are entered as synonyms, and at the same time the references to the now synonym symbols are collected into the official gene.

From the description it seemed that Gene symbol was a unique identifier and would fit a primary key well, seeing as official_name is not necessarily unique. If symbol is not unique, a different primary key must be selected for the Gene table.

I couldn’t see if chromosome_name was a unique identifier in the text, if not a chromosome_id is needed as a primary key.

iii) ER diagram in tables not conforming to 2NF

Gene_with_reference_and_author (#symbol, #reference_id, #author_id, official_name, chromosome_name*, start, stop, title, journal, year_published, author_name)

Synonym (#synonym_symbol, gene_symbol*)

Chromosome (#name, length)

The tables above are not in 2NF, as the table gene_with_reference_and_author contains fields which are functionally dependent on only part of the primary key. One such example is reference_id, which determines journal and year_published.

The tables are in 1NF, as all the data types in the tables are atomic (no lists, tables, pointers etc as data types).

iv) ER diagram in tables conforming to BCNF

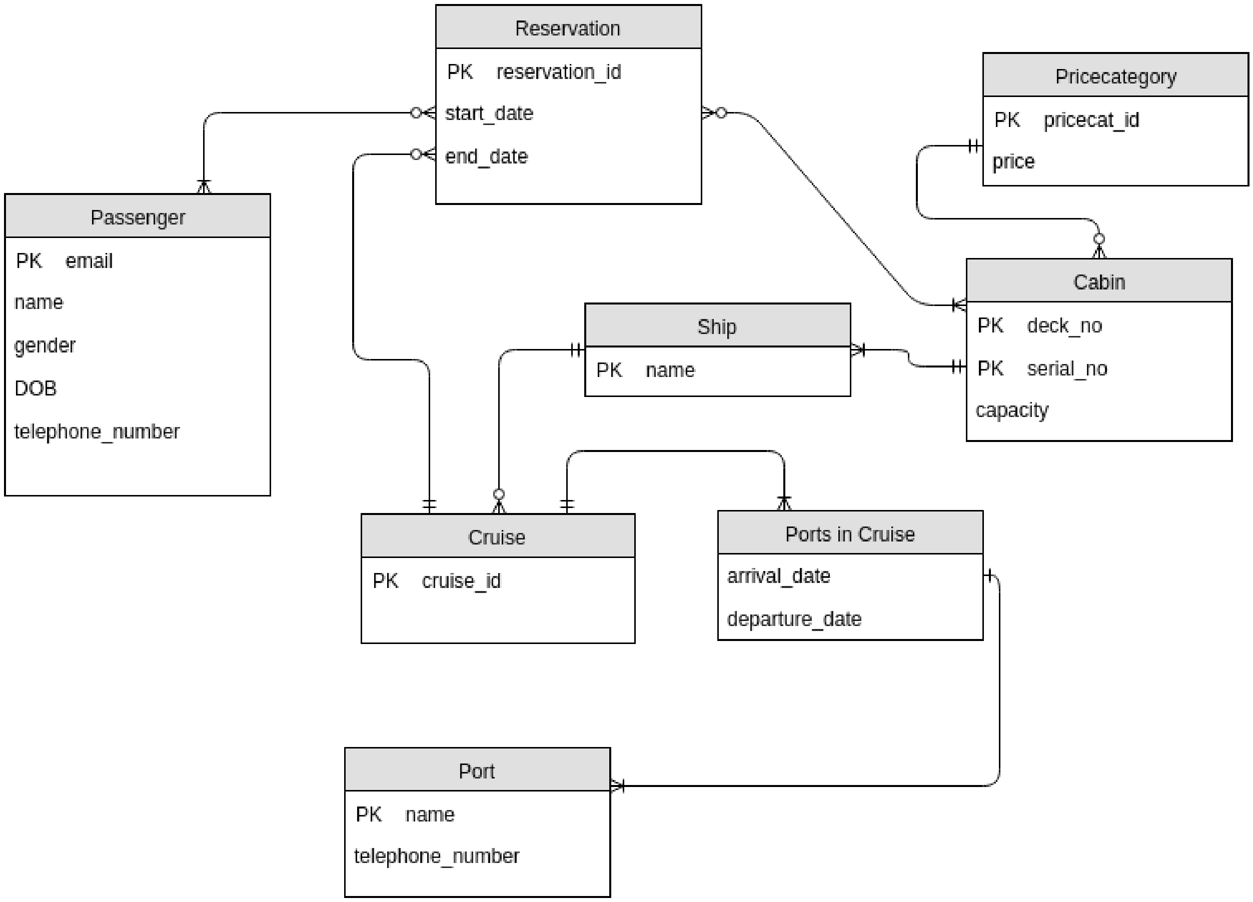
Gene_info (#symbol, official_name, chromosome_name*)
Gene_position(#symbol, start, stop)
Chromosome (#name, length)
Synonym (#synonym_symbol, gene_symbol*)
Reference (#id, title, journal, year_published)
Author (#author_id, name)
Reference_to_gene (#gene_symbol*, #reference_id*)
Author_of_reference (#author_id*, #reference_id*)

This is a set of tables in the database corresponding to the ER diagram in 3.ii, conforming to the BCNF. Two tables are added for many-to-many relationships in Reference_to_gene and Author_of_reference. Some information in Gene is split into two tables, to prevent it from having an alternate super key in chromosome_name, start and stop, which would uniquely identify the gene.

One could consider a split of the Reference table, to prevent different references in the same journal with the same title (and published year), but this might conflict with real life concerns like a monthly column with the same title.

Task 4)

ER diagram of the specified cruise database.



The required attributes of number of cabins and number of passengers for the ship entity is modeled through the relation with cabin, and cabin's capacity attribute. Reservations for parts of a cruise is handled with the reservation start and end date. The price category of a cabin is handled as a separate entity, with a price attribute.

Task 5)

i) Why is the Truck table problematic?

The most glaring problem with the proposed Truck table is the lack of a primary key. As I interpret the proposed table, there are two possible primary keys, depending on how the table is supposed to be used:

- Registration_number and assignment_number
    - If this is the intended primary keys, you can track all the assignments a truck has been or is part of. However by storing multiple assignments in this way will result in all the other attributes containing duplicate information for each assignment. Updating and maintaining the table is harder as a result, as you have to take great care to not store contradictory data
- Registration_number
    - If you just use registration_number as your primary key, the assignment_number will only contain the current or last assignment of a truck. You then have to update the Assignment_number of the truck

In either case, Trucks not assigned will have a null indicator in the assignment_number. Trucks and assignments are in practice a many-to-many relation, which requires a linking table to properly model the relation.

Unless the trucks are modified from the base model, the Maximum_weight should be the same for the same model. Therefore you store duplicate information in this field too.

ii) Non-trivial functional dependencies of the Truck table

Registration_number -> Registration_year, model, maximum_weight, (assignment_number if only registration_number as primary key)
Model -> maximum_weight

iii) Candidate key(s) for the Truck table

#Registration_number, #Assignment_number* is a candidate key, as Registration_number determines everything except assignment_number, and assignment_number determines itself.

iv) Normalization to BCNF

Initial tables:

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*, Container_number*, Start_date, End_date)
Truck (#Registration_number, Registration_year, Model, Maximum_weight, #Assignment_number*)

Normalized to BCNF:
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*, Container_number*, Start_date, End_date)
Truck (#Registration_number, Registration_year, Model*)
Truck_assigned_to_assignment(#Registration_number*, #Assignment_number*)
Truck_model_weight(#Model, Maximum_weight)

If Container(Type_name) is unique, a further split into Container_type_name(#Type_id*, Type_name) is required. If the Nightly_rate of a container is dependent on weight and cubic quantity, this attribute should also be split off (or computed).