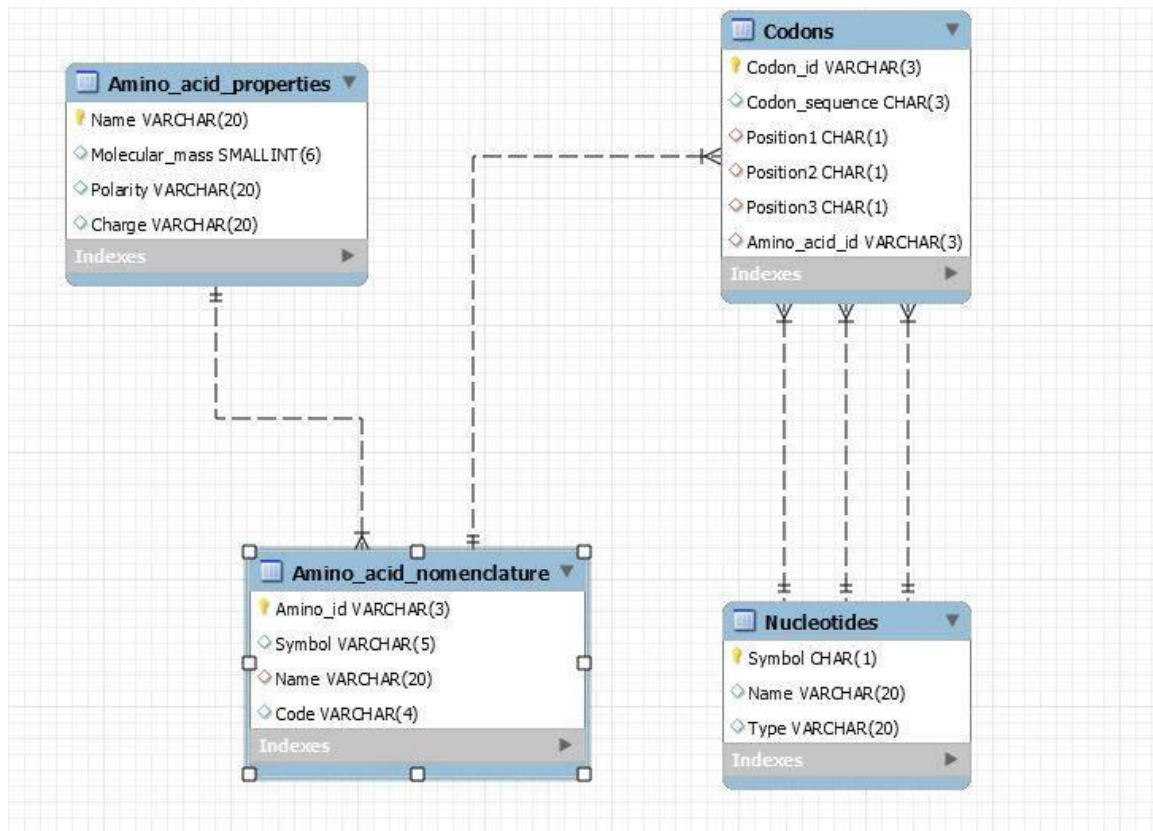


## INF115 Compulsory Exercise 2

### 1) (10%)

Using the four tables from compulsory exercise 1, reverse engineer an ER diagram to represent the entities and their attributes. Highlight primary keys, and select appropriate relationships between the entities.



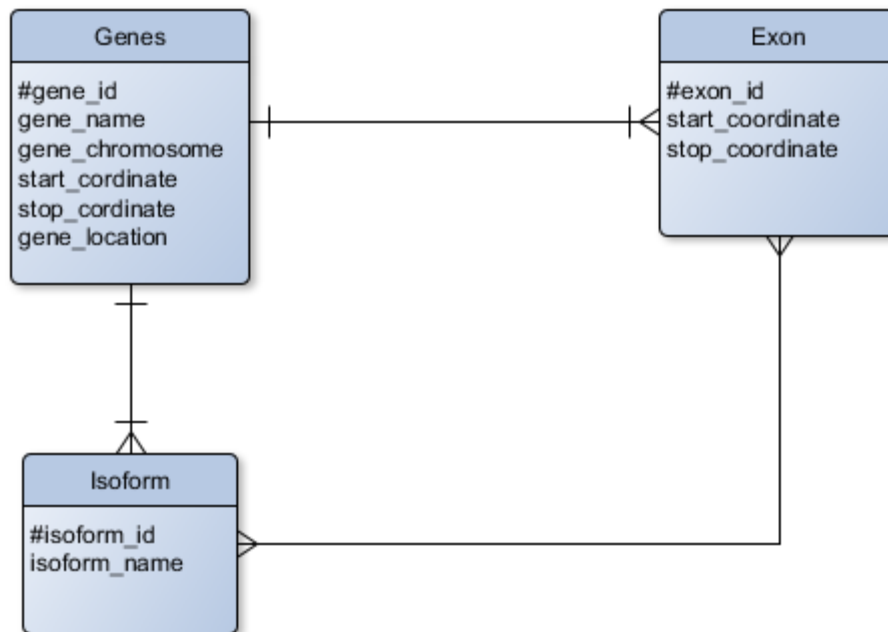
### 2) (20%)

A database stores the information about the relations between Genes, Exons and Isoforms. A Gene has a unique gene\_id and a name, the database will also store some information about the location of the gene, including the name of the chromosome on which it is located and the start and stop coordinate of the gene on the chromosome. Genes are always located on just one chromosome. Each gene contains one or more Exons, each with a unique exon\_id and a start and stop coordinate. Every gene has one or more Isoforms, each with a unique isoform\_id and an isoform\_name, each Isoform is created from one or more of the Exons of a particular gene.

**i) Identify the entities in the database description above.**

In the database description above, the entities are Genes, Exon and Isoforms.

ii) Produce an ER diagram showing the relations between these entities, highlight primary keys, and select appropriate relationships between the entities.



iii) Convert the ER diagram in 2.ii to a set of tables conforming to the third normal form, include appropriate primary and foreign keys.

Genes(#gene\_id, gene\_name, start\_coordinate, stop\_coordinate, gene\_location, gene\_chromosome\*)  
 Exon(#exon\_id, start\_coordinate, stop\_coordinate, gene\_id\*)  
 CombinationExonIsoform(#isoform\_id, exon\_id\* )  
 Isoforms(#isoform\_id, isoform\_name, gene\_id\*)

### 3) (25%)

Gene nomenclature can be confusing because the experimental nature of gene discovery can lead to the same gene being “discovered” and named a number of times by separate individuals. At some point the scientists will realise that they all discovered the same gene, and a standardised name will be agreed upon, along with a new identifier called the gene symbol. All of the other names will be recorded as official synonyms for this gene to allow backward compatibility with the literature that has previously been published.

A database will record the following information about the genome of a particular species:

Gene\_symbol, Official\_name, Synonyms, Chromosome\_name, Chromosome\_length, Start\_coordinate, End\_coordinate, References.

A typical example of this data is as follows:

Gene\_symbol: “SHH”  
 Official\_name: “sonic hedgehog”  
 Synonyms: “HHG1”, “MCOPCB5”, “SMMCI”, “TPT”, “TPTPS”

Chromosome\_name: "Chr5"

Chromosome\_length: 153573022

Start\_coordinate: 28456815

End\_coordinate: 28467256

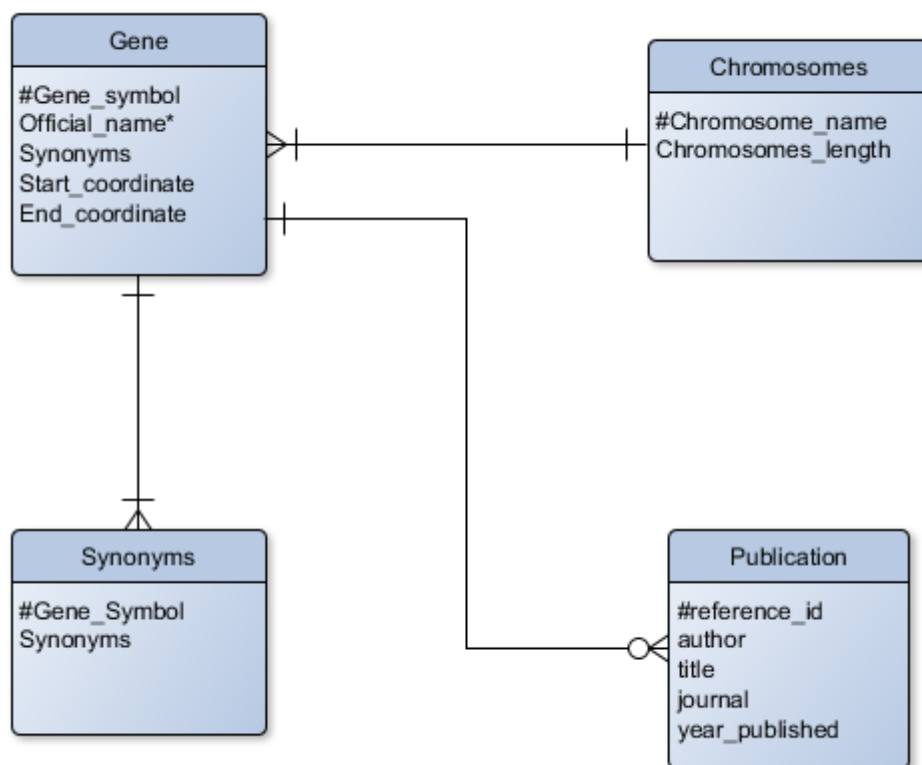
References: "J:227320 Holtz AM, et al., Secreted HHIP1 interacts with heparan sulfate and regulates Hedgehog ligand localization and function. J Cell Biol. 2015", "J:13476 Dickie MM, Hemimelic extra toes, Hx. Mouse News Lett. 1968"

The synonym field contains a list of all of the previously used names. The References field contains a list of all of the publications that have cited this gene (each composed of: reference id, authors, title, journal, year published). A publication can cite more than one gene. Assume for this exercise that all genes have at least one synonym.

**i) Identify the entities in the database description above.**

The entities in the database description above is Gene, Chromosome, Synonyms, Publication.

**ii) Create an ER diagram to represent the relations between the entities that you identified in question 3.i, specify the primary keys and select appropriate relationships between the entities.**



**iii) Convert the ER diagram in 3.ii to a set of tables conforming to the first normal form, but not the second normal form. Highlight primary keys and foreign keys.**

Gene( Gene\_symbol, Offical\_name, Synonyms, Start\_coordinate, End\_coordinate, References, #Chromosome\_name, Chromosome\_length )

Publication(#reference\_id, gene\_symbol , authors\*, title, journal, year\_published )

**iv) Further develop the table structure from 3.iii so that it conforms to Boyce Codd normal form (BCNF), again highlight primary and foreign keys.**

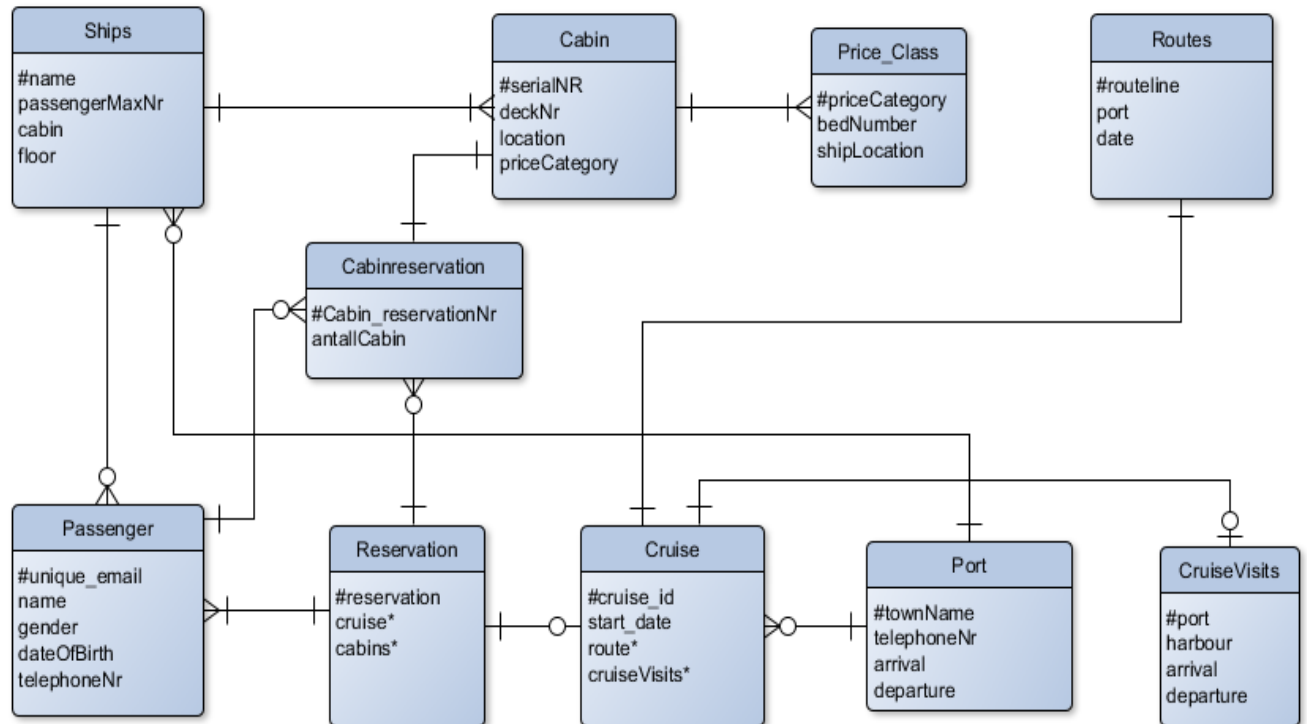
Gene(#Gene\_symbol, Official\_name, Start\_coordinate, End\_coordinate, chromosome\_name\*, reference\_id\* )  
 Synonyms(#Gene\_symbol, Synonyms)  
 Chromosome(#Chromosome\_name, Chromosome\_length )  
 Publication(#reference\_id, authors, gene\_symbol\*, title, journal, year\_published )  
 MultipleGene(#reference\_id\*, gene\_symbol)

#### **4) (20%)**

Global\_cruises offers luxury cruises worldwide, and is in need of a new system to handle ships, passengers and routes. The system must store information about each cruise ship. Each ship has a unique name. In addition, the maximum number of passengers will be stored for each ship. A cruise starts on a particular date from a particular port and follows a specific route.

The system must store which harbours a cruise visits on each day. It must be possible to find out which date a cruise arrives at and leaves a particular port. For every port the town name and telephone number of the port office should be stored. Every cruise ship has a number of cabins (rooms) in 4 to 8 decks (floors). A cabin is identified by a deck number and a serial number, for example "417" means cabin 17 on deck 4. The cabins are assigned to different price categories depending on the number of beds and location on the ship. The system must also store information about the passengers and their reservations. Every passenger gets a unique email address, name, gender, date of birth and a telephone number. A reservation always applies to one particular cruise, but multiple passengers can be entered on one reservation. Passengers can book a number of cabins in a single reservation.

**Create a data model (E / R diagram) for Global\_cruises, specify the primary keys and select appropriate relationships between the entities.**



## 5) (25%)

A company rents out containers, and maintains the following database. Primary keys are marked with a hash (“#”) and foreign keys are marked with an asterisk (“\*”):

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)

The company wishes to expand so that it can provide transport services for the containers. The company will purchase a number of trucks and the various assignments will be continuously distributed on the available transportation. Multiple trucks can be involved in the transportation of one assignment. The following table has been proposed to extend the database in order to handle the new transportation requirements:

Truck (Registration\_number, Registration\_year, Model, Maximum\_weight, Assignment\_number\*)

An example row is as follows:

( 'LY12345', 2012, 'Volvo XL', 8500, 3)

This means that the truck with registration number LY12345 was first registered in 201 and was used in assignment 3. This truck is the model Volvo XL, and this model has a maximum weight 8500 kg.

**i) Explain first why this solution proposed by the Truck table above is problematic.**

With multiple trucks on the same assignment will create duplicate information and redundancy.

The truck table hasn't specified the primary key yet, so the only candidate key Registration\_number can be used as the primary key. There can make problems when trucks are finished with the assignments and the truck information needs to be saved. So new trucks can move on to the next assignments.

**ii) Write down the functional dependencies of the Truck table.**

Functional dependencies:

Model -> Maximum\_weight  
 Registration\_number -> Registration\_year  
 Registration\_number -> Model  
 Registration\_number -> Maximum\_weight

**iii) Determine the candidate key(s) for the Truck table.**

Registration\_number is the only possible candidate key for the truck table.

**iv) Perform normalization to BCNF for the whole table (the original table expanded to incorporate transportation). Show primary keys and foreign keys in the final result.**

Container\_type (#Type\_id\*, Max\_weight, Cubic\_quantity, Nightly\_rate)  
 Container\_name(#Type\_id, Type\_name)  
 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\*)