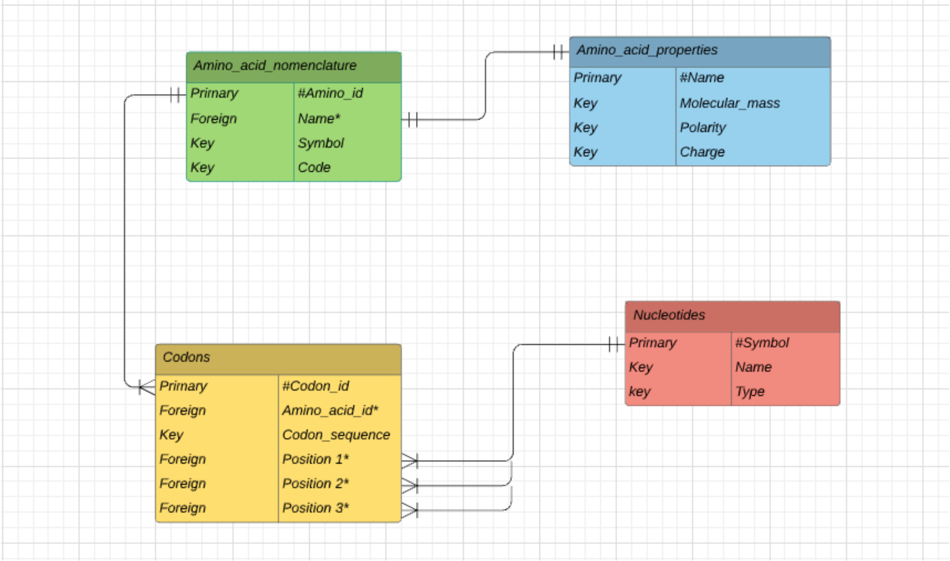
Noora Rehim Lindeflaten 24.03.20

Semesteroppgåve 2. Inf115

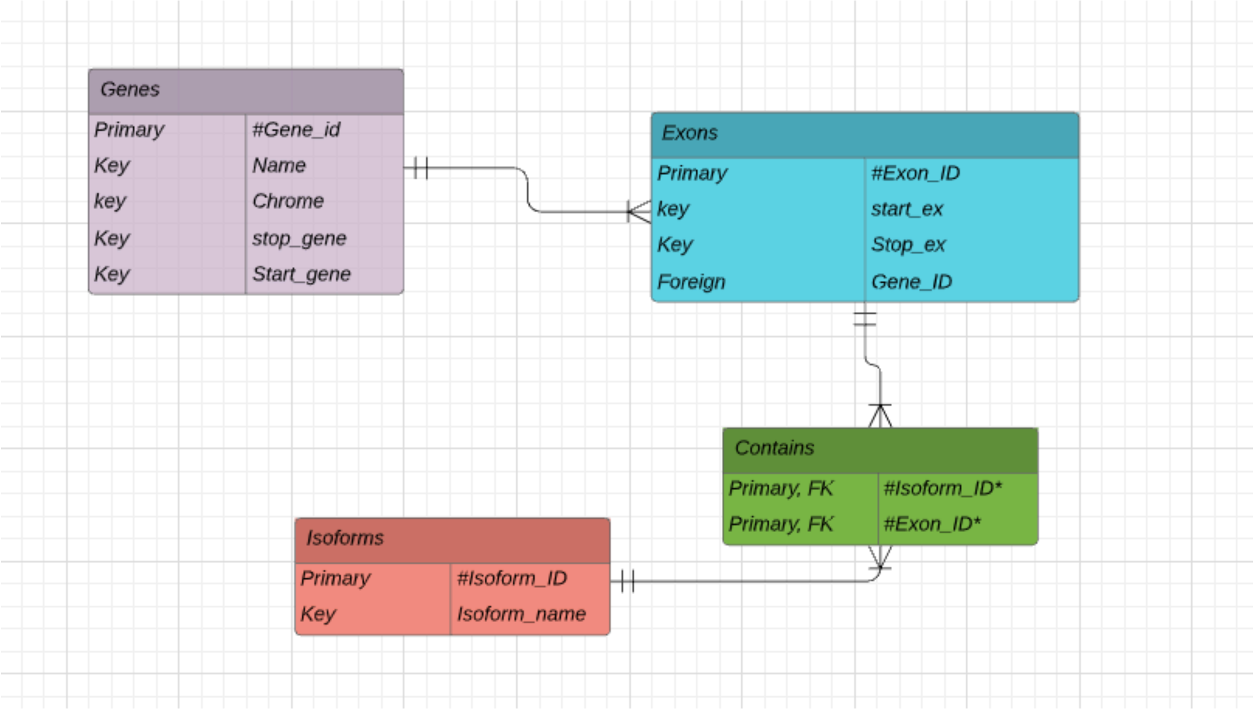
Problem 1.

This is the ER-diagram of the four tables from last assignment.



Problem 2.

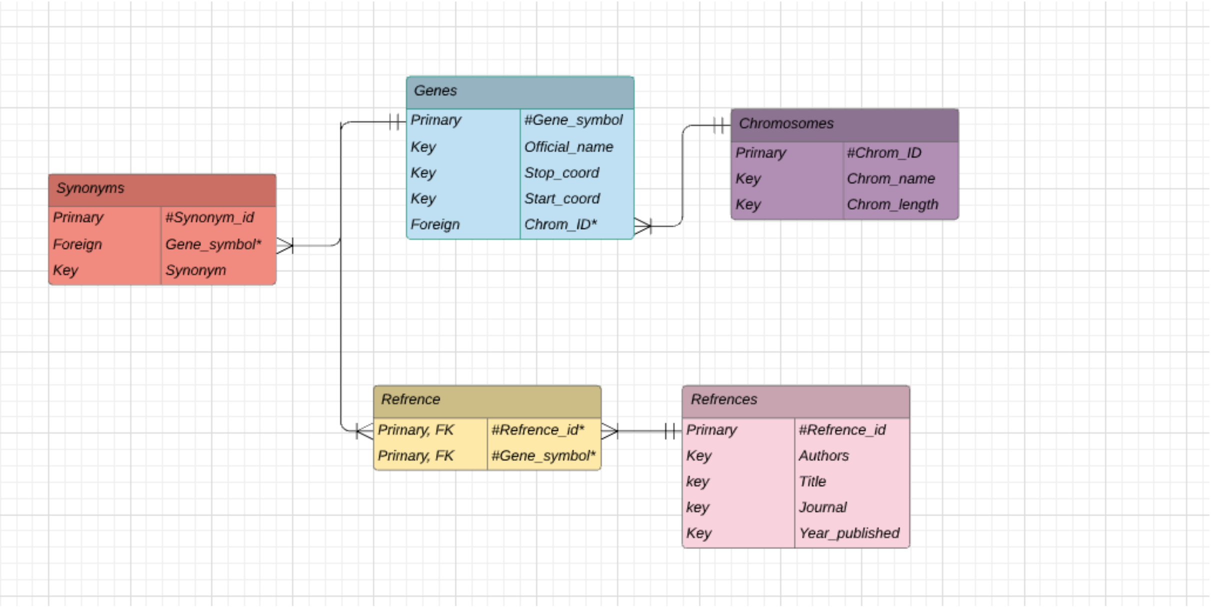
1. Entities are objects or things that we want to collect data on. In the first problem we see that these objects/things are: Genes, Exons and Isoforms
2.



3. Genes(#gene_ID, name*, Chrom_name, start_gene, stop_gene, location)
   Exons(#Exon_ID, start_ex, Stop_ex)
   Isoform(#Isoform_ID, Isoform_name)
   Contains(#Exon_id*, #Isoform_id*)

Problem 3.

1. Entities: Genes, Synonyms, refrences and Chromosomes
2.



3. The form **1NF** is the first step of normalization. A database table is considered 1NF if the domain of each attribute contains only atomic values and the value of each attribute contains only one value so that there is no repeated data.

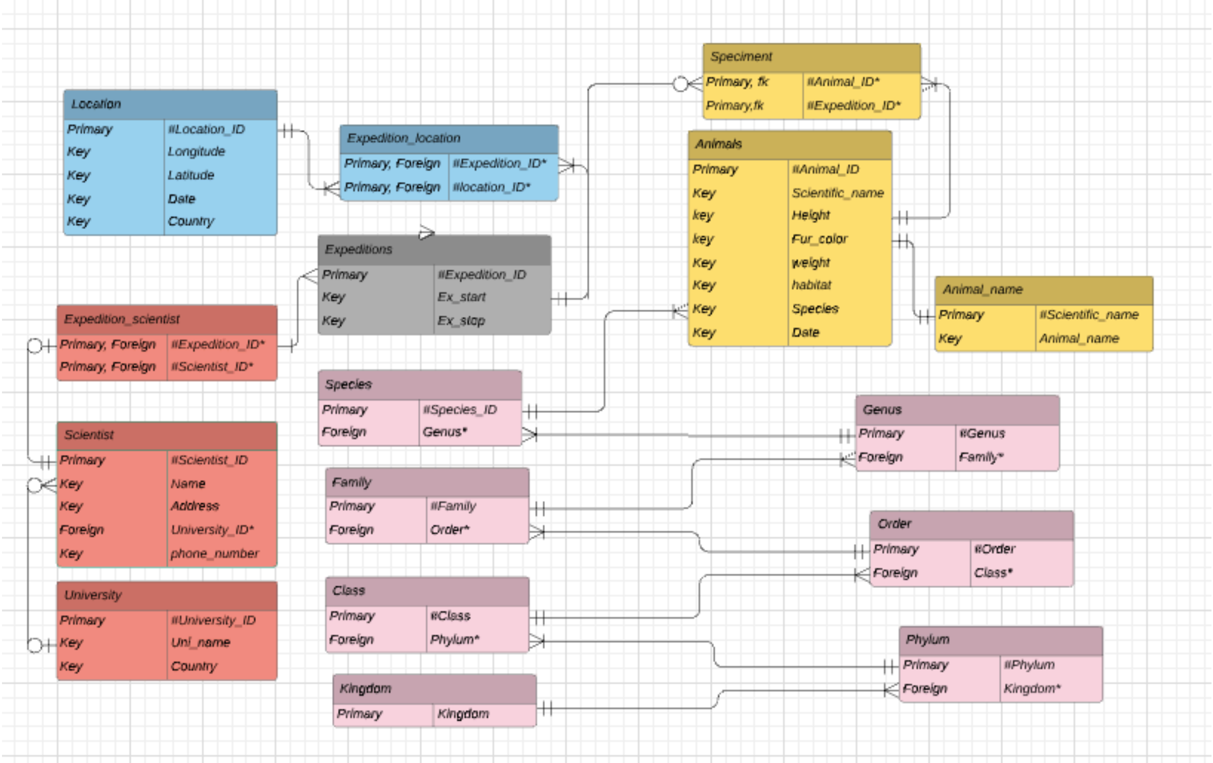
   2NF the second step of normalization. If the conditions in 1NF are present here and the table doesn't contain any attributes that are partially dependent on the primary key the table is considered to be 2NF.

   3NF. A database relation meets the third normalization criteria if the conditions above are present and the table also doesn't contain any attributes who are transitive dependency on another key.

4. Genes(#Gene_symbol, stop_coord, start_coord, Chrom_ID*)
   Genenames(#Gene_symbol, Official_name)
   Chromosomes(#Chrom_ID, Chrom_name, chrom_length)
   Synonyms(#Synonym_id, gene_symbol*, synonym)
   Refrences(#Refrence_id, authors, title, Journal, Year_published)
   Refrence(#refrence_id*, #gene_symbol*)

Problem 4.

Subproblem 1.



Location	
Primary	#Location_ID
Key	Longitude
Key	Latitude
Key	Date
Key	Country

Expedition_location	
Primary, Foreign	#Expedition_ID*
Primary, Foreign	#location_ID*

Expeditions	
Primary	#Expedition_ID
Key	Ex_start
Key	Ex_stop

Expedition_scientist	
Primary, Foreign	#Expedition_ID*
Primary, Foreign	#Scientist_ID*

Scientist	
Primary	#Scientist_ID
Key	Name
Key	Address
Foreign	University_ID*
Key	phone_number

University	
Primary	#University_ID
Key	Uni_name
Key	Country

Speciment	
Primary, fk	#Animal_ID*
Primary,fk	#Expedition_ID*

Animals	
Primary	#Animal_ID
Key	Scientific_name
key	Height
key	Fur_color
Key	weight
Key	habitat
Key	Species
Key	Date

Animal_name	
Primary	#Scientific_name
Key	Animal_name

Species	
Primary	#Species_ID
Foreign	Genus*

Genus	
Primary	#Genus
Foreign	Family*

Family	
Primary	#Family
Foreign	Order*

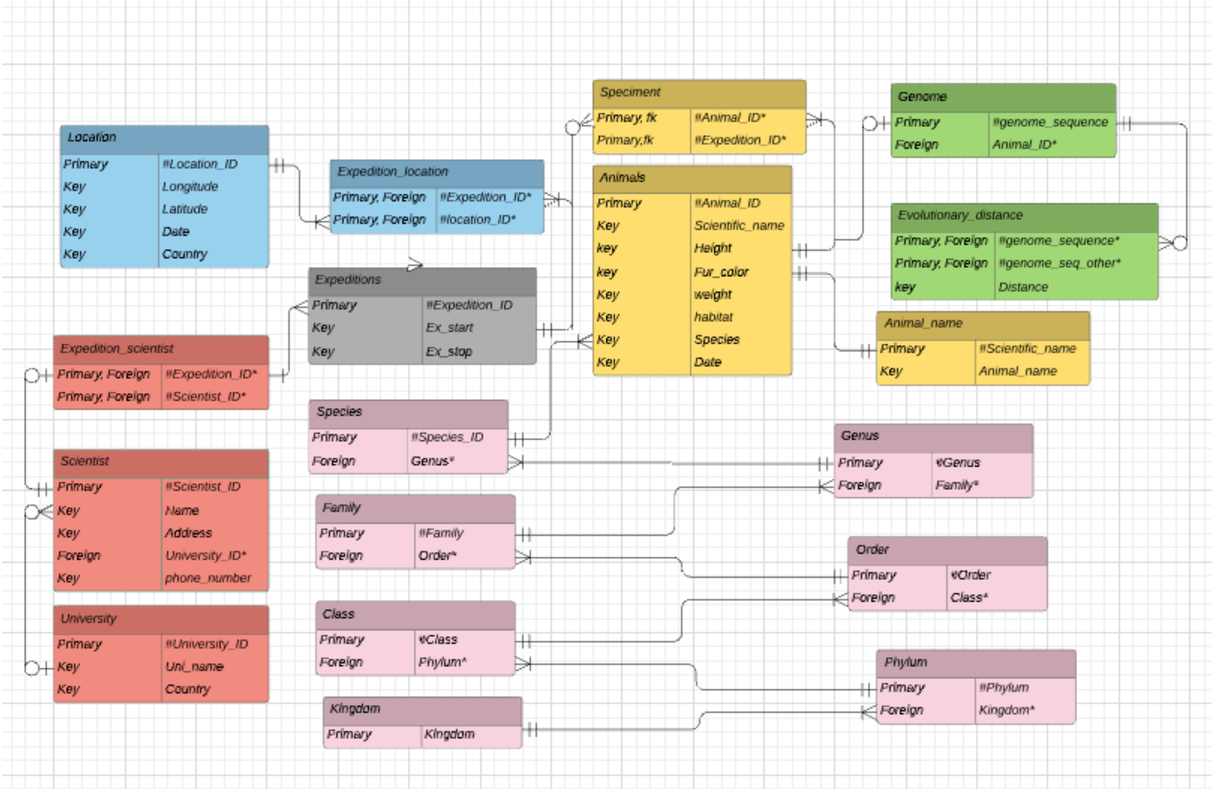
Order	
Primary	#Order
Foreign	Class*

Class	
Primary	#Class
Foreign	Phylum*

Phylum	
Primary	#Phylum
Foreign	Kingdom*

Kingdom	
Primary	Kingdom

Subproblem 2.



Speciment	
Primary, fk	#Animal_ID*
Primary,fk	#Expedition_ID*

Genome	
Primary	#genome_sequence
Foreign	Animal_ID*

Location	
Primary	#Location_ID
Key	Longitude
Key	Latitude
Key	Date
Key	Country

Expedition_location	
Primary, Foreign	#Expedition_ID*
Primary, Foreign	#location_ID*

Animals	
Primary	#Animal_ID
Key	Scientific_name
key	Height
key	Fur_color
Key	weight
Key	habitat
Key	Species
Key	Date

Evolutionary_distance	
Primary, Foreign	#genome_sequence*
Primary, Foreign	#genome_seq_other*
key	Distance

Expeditions	
Primary	#Expedition_ID
Key	Ex_start
Key	Ex_stop

Animal_name	
Primary	#Scientific_name
Key	Animal_name

Expedition_scientist	
Primary, Foreign	#Expedition_ID*
Primary, Foreign	#Scientist_ID*

Species	
Primary	#Species_ID
Foreign	Genus*

Genus	
Primary	#Genus
Foreign	Family*

Scientist	
Primary	#Scientist_ID
Key	Name
Key	Address
Foreign	University_ID*
Key	phone_number

Family	
Primary	#Family
Foreign	Order*

Order	
Primary	#Order
Foreign	Class*

University	
Primary	#University_ID
Key	Uni_name
Key	Country

Class	
Primary	#Class
Foreign	Phylum*

Phylum	
Primary	#Phylum
Foreign	Kingdom*

Kingdom	
Primary	Kingdom

Problem 5.

Subproblem 1.

- Patient(#PatientID, FirstName, SurName, Postcode, Address)
  The highest form of normalization level this table conforms to is BCNF. This is because it contains the Postcode attributer which is not dependent on address since addresses can have identical names but never the same postcode. Which means that the following is present here: A → B, A cannot be a non-prime attribute, if B is a prime attribute.

- Sample(#SampleID, SampleDate, PatientID*)
  The statement above is true for this one as well and the table reaches the level BCNF.

- Labtest(#TestID, TestName, SampleID*, ResitantToAntibiotic)
  In this Table we can see that one of the columns will store multiple values inside of one cell. This is problematic and it will result in a table with non-atomic cells, and as described earlier in this assignment a table needs to have only atomic cells to be on the form 1NF which means that this table doesn't even reach the first level of normalization.

- HospitalLocation(#LocationID, Region)
  This table also reaches the highest level of Normalization due to all the values being dependent on the primary key.

- PatientToLocation(#PatientID*, #LocationID*)
  BCNF, because this is a table connected to two other tables and it only contains Primary keys.

Subproblem 2.

1. The problem with the solution of the tables is that the OutbreakID is static and the problem with this is that you can't access previous outbreaks there is no place where the data of previous outbreaks are stored and you just recive the same value using this one.

2. Paitent_ID, Outbreak_ID and Feature_ID are all functionally dependet on Genom_ID because from the genome which belongs to the person you have the outbreak_id and it also contains a feature.
3. The candidate key of the genome_sequence is the Genom_ID as we can see in the tables
4. Patient(#PatientID, FirstName, SurName, Postcode, Address)
   Sample(#SampleID, SampleDate, PatientID*)
   Labtest(#TestID, TestName, SampleID*, ResitantToAntibiotic)
   HospitalLocation(#LocationID, Region)
   PatientToLocation(#PatientID*, #LocationID*)
   GenomeSequence(#GenomeID, PatientID*, OutbreakID, FeatureID*)
   ResistanceDeterminats(#FeatureID, AntibioticName)

   To get to the first step 1NF we know that we have to fix the table labtest. I then make a new table called resistance, and now we have a many to many relations between resistance and labtest so we need to add a contains table to break down the relations. This is how our tables look now: (the things I've altered are green)

Patient(#PatientID, FirstName, SurName, Postcode, Address)
Sample(#SampleID, SampleDate, PatientID*)
Labtest(#TestID, TestName, SampleID*)
Resistance(#Resistance_ID , AntibioticName)
Contains(#Resistance_ID*, #Test_ID)
HospitalLocation(#LocationID, Region)
PatientToLocation(#PatientID*, #LocationID*)
GenomeSequence(#GenomeID, PatientID*, OutbreakID, FeatureID*)
ResistanceDeterminats(#FeatureID, AntibioticName)

2NF
We can see that all our tables already meet the criteria of the 2nd normalization step, and so does the new tables we introduced.

3NF
Our tables are already on this form because of the connection tables resistance, and paitenttolocation.

BCNF
We can only have one candidate key in each table so here we will fix the problem we encounted in 5.2.1 with the Outbreak_ID. I have created a new table called Outbreaks which stores the data we were missing.

Patient(#PatientID, FirstName, SurName, Postcode, Address)
Sample(#SampleID, SampleDate, PatientID*)
Labtest(#TestID, TestName, SampleID*)
Resistance(#Resistance_ID , AntibioticName)
Contains(#Resistance_ID*, #Test_ID)
HospitalLocation(#LocationID, Region)
PatientToLocation(#PatientID*, #LocationID*)
GenomeSequence(#GenomeID, PatientID*, FeatureID*)
ResistanceDeterminats(#FeatureID, AntibioticName)
Outbreaks(#OutbreakID, Genome_ID*, Dato)