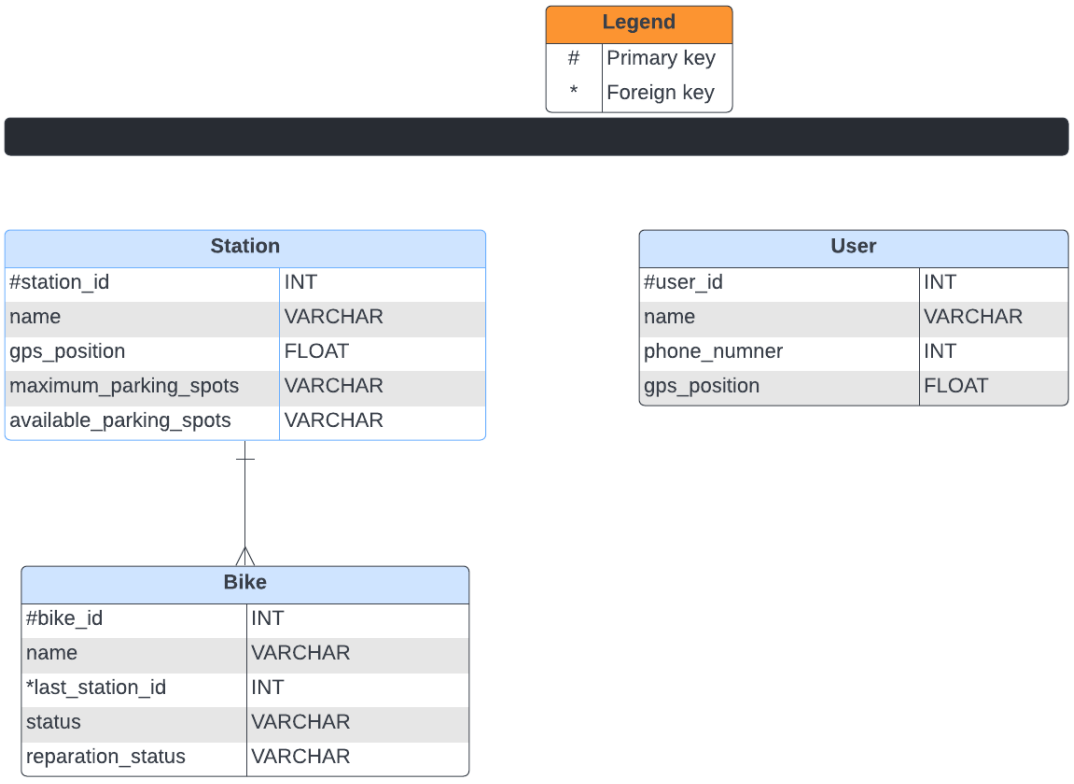
INF115 Obligatory 2 – V22

By Noel Santillana Herrera

Problem 1

1.1)



1.2)

There is a one-to-many relation from Station to Bike. Given the current three entities, the user is not has not relation to Bike or Station.

Problem 2

2.1)

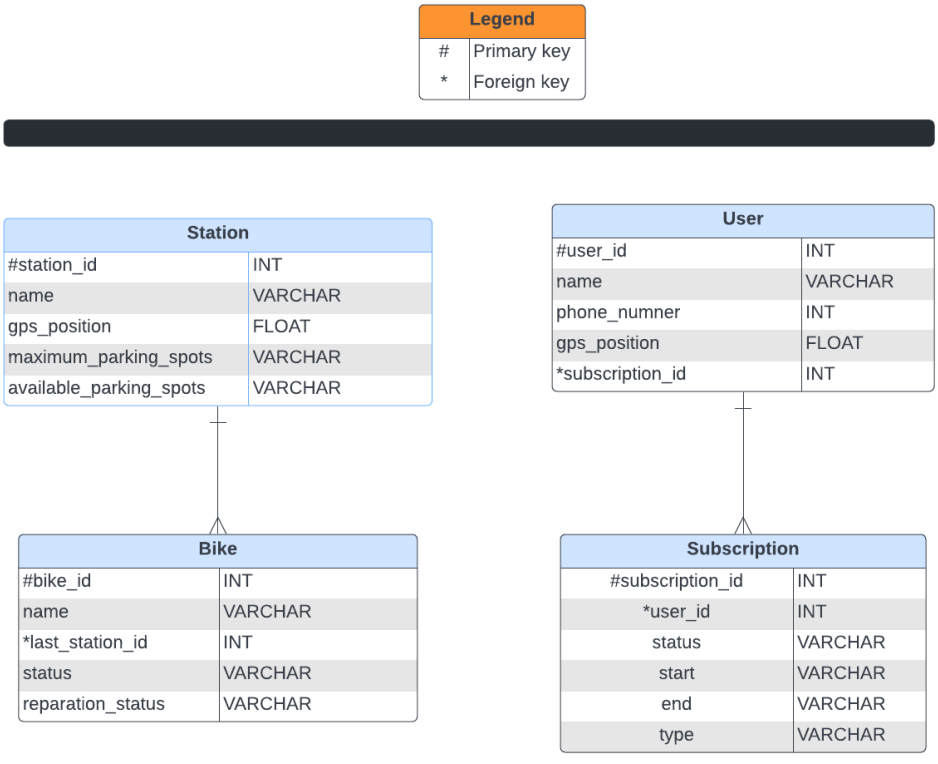
The problem with this proposed solution is that Status, Start, End and Type will be dependent on Subscription ID. Subscription ID is dependent on User ID.

2.2)

The Subscription table.

Subscription	
#subscription_id	INT
user_id	INT
status	VARCHAR
start	VARCHAR
end	VARCHAR
type	VARCHAR

2.3)



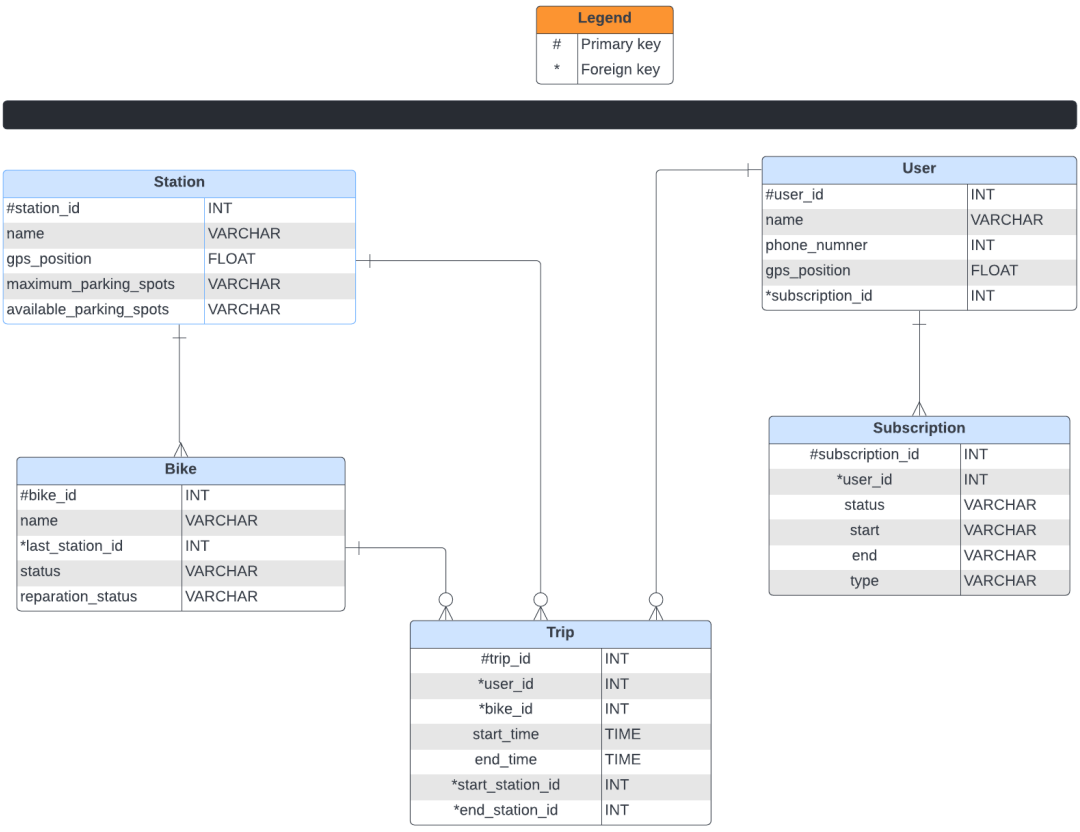
Problem 3

3.1)

The Trip table.

Trip	
#trip_id	INT
*user_id	INT
*bike_id	INT
start_time	TIME
end_time	TIME
*start_station_id	INT
*end_station_id	INT

3.2)



3.3)

The 4 normalization forms:

1NF – The values in a column of a table should not contain the same multiple values. The values should be atomic values, meaning that the values in the column cannot be divided.

2NF – A table has taken the second normalization form if it satisfies the 1NF condition and non-key are not dependent on a subset of a primary key.

3NF – A table has taken the third normalization form if it satisfies the 2NF conditions and non-key attributes are not transitively dependent on any of the super keys. For example, if A is dependent on B and B is dependent on C, then C is transitively dependent on A via B. If this is the case, then it should be removed.

BCNF – Stands for "Boyce-Codd Normal Form" and is a stricter version of 3NF. A table is in BCNF if it satisfies the 3NF conditions and, for each functional dependency (A -> B), B should be a super key.

Overall, the point of these normalizations is to avoid the redundancy and anomalies.

3.4)

Reparation_status in Bike table is not atomic because a bike can be ridden by multiple others users and get sent in complaints before it is sent to the workshop. The reparation_status lane will therefore have redundant values that says ("flat tire", "flat tire", etc..). We can make a different table for the list of complaints to get inserted to instead.

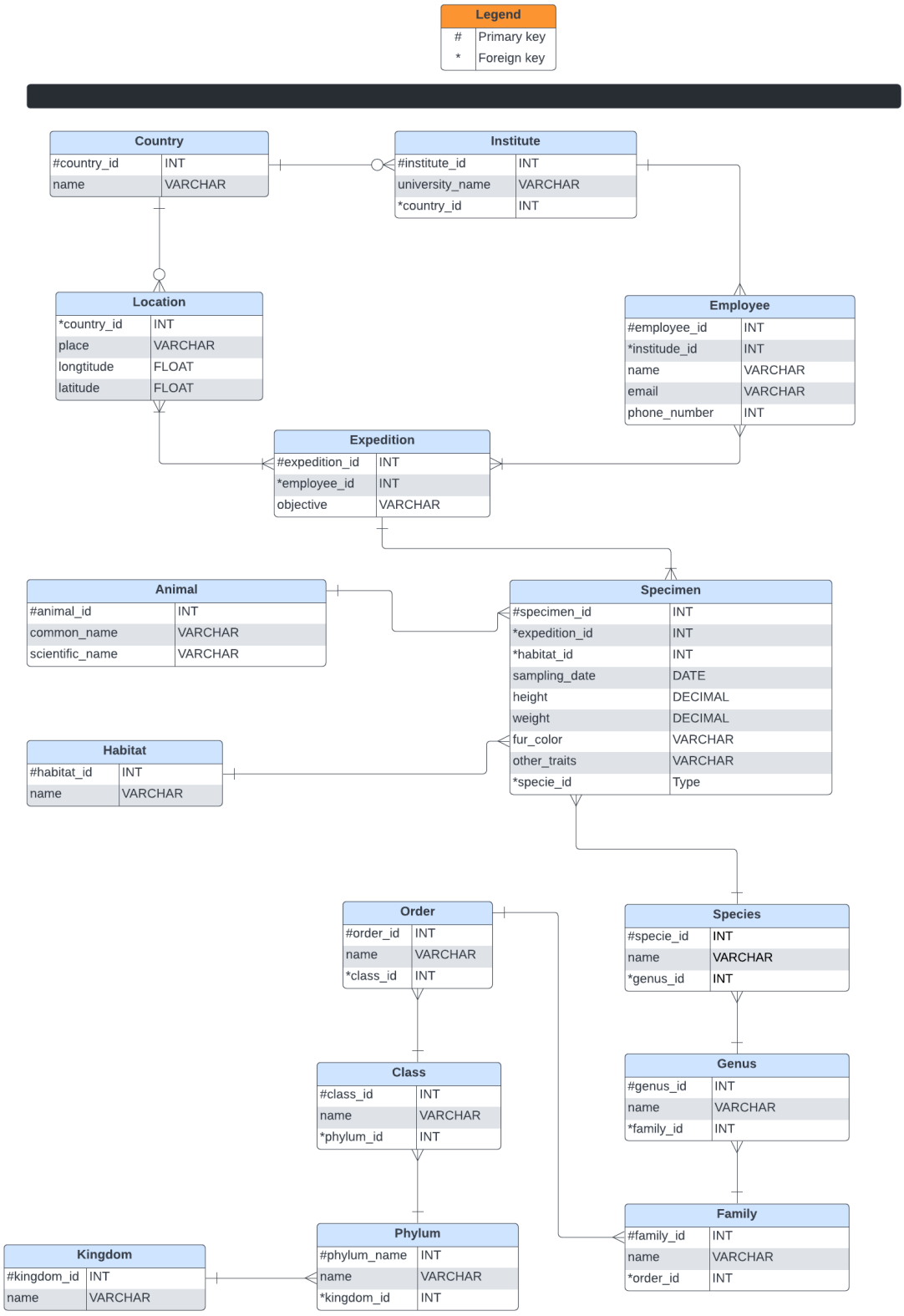
#complaint_id, * user_id, *bike_id, description

#reparation_id, *complaint_id, *bike_id, date

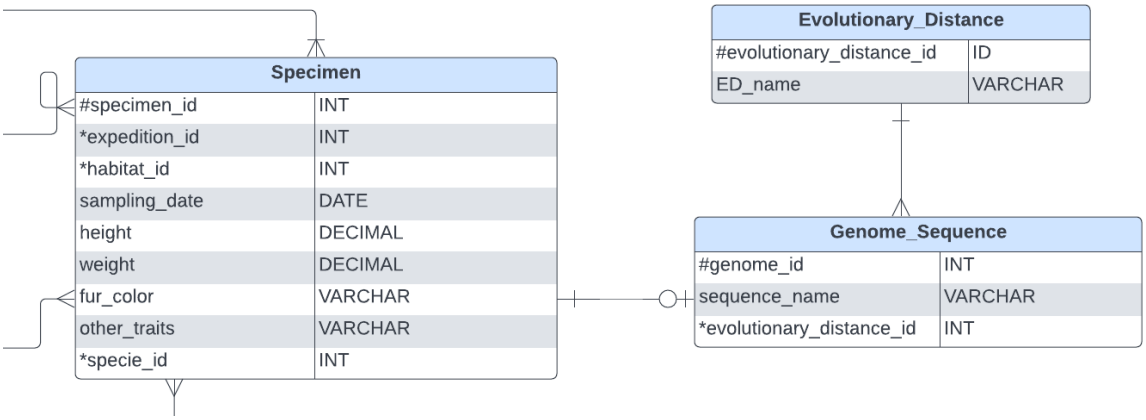
Another change to maybe make it more of BCNF is to split the names in the user table by adding first_name and surname instead of just name.

Problem 4

Subproblem 1)



Subproblem 2)



Problem 5

Subproblem 1

Patient table is BCNF.

Sample table is BCNF.

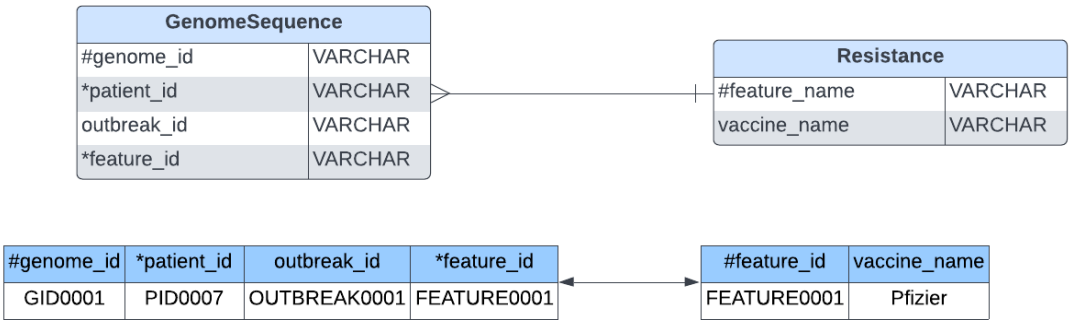
Labtest table is not in any forms because resistance is not atomic.

Hospital table is BCNF.

PatientLocation table BCNF.

Subproblem 2

To imagine these problems a little better I made the tables to look at.



1)

I believe the problem with this solution is that feature_id will have redundancy, meaning it is not atomic.

2)

Functional dependencies:

genome_id and patient_id

genome_id and feature_id

3)

The candidate key in GenomeSequence table is #genome_id

4)