

# INF115 Databases and Modelling

## Obligatory Assignment 2 Solutions

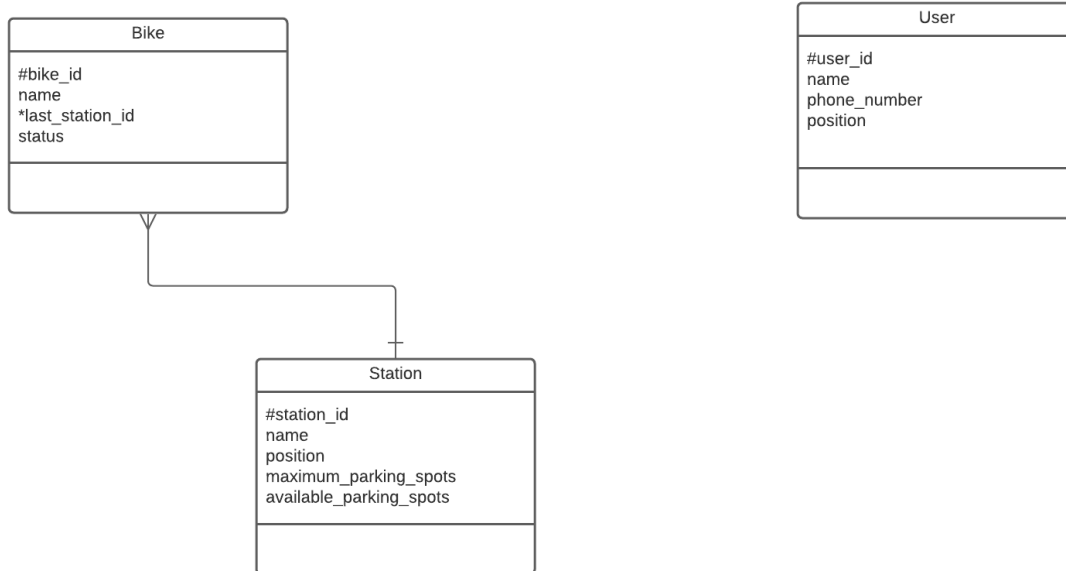
April 28, 2021

### Contents

<b>1 Problem 1 (10%)</b>	<b>2</b>
1.1 (6%) . . . . .	2
1.2 (4%) . . . . .	2
<b>2 Problem 2 (20%)</b>	<b>2</b>
2.1 (6%) . . . . .	2
2.2 (4%) . . . . .	2
2.3 (10%) . . . . .	3
<b>3 Problem 3 (25%)</b>	<b>3</b>
3.1 (4%) . . . . .	3
3.2 (10%) . . . . .	4
3.3 (5%) . . . . .	4
3.4 (6%) . . . . .	4
<b>4 Problem 4 (20%)</b>	<b>5</b>
4.1 (15%) . . . . .	5
4.2 (5%) . . . . .	6
<b>5 Problem 5 (25%)</b>	<b>6</b>
5.1 (2%) . . . . .	6
5.2 (5%) . . . . .	6
5.3 (6%) . . . . .	6
5.4 (2%) . . . . .	6
5.5 (10%) . . . . .	6

## 1 Problem 1 (10%)

### 1.1 (6%)



### 1.2 (4%)

There is no relation from user to either of bike and station, while bike is related to station

## 2 Problem 2 (20%)

### 2.1 (6%)

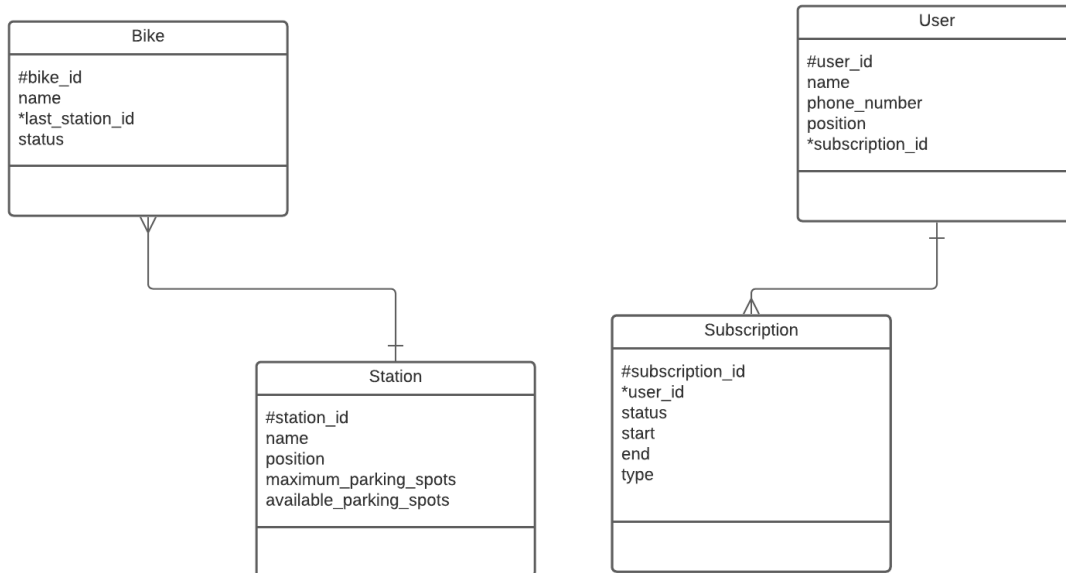
Status, Start, End and Type is dependant on Subscription ID. Also, SubscriptionID is dependant on UserID.

### 2.2 (4%)

You should create a new table Subscription with UserID as foreign key.

#SubscriptionID  
UserID  
Status  
Start  
End  
Type

## 2.3 (10%)



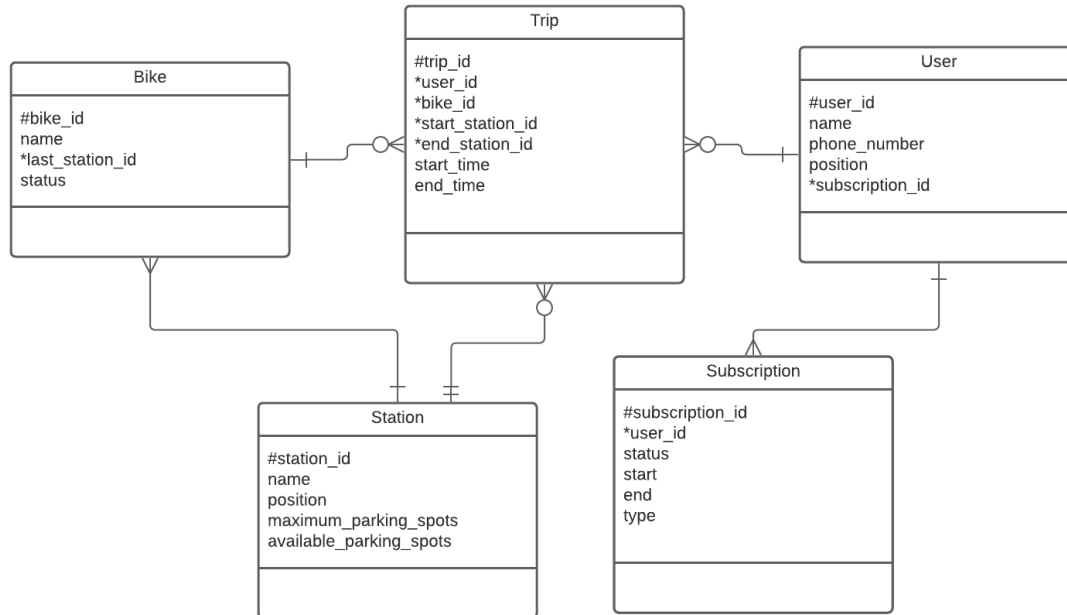
## 3 Problem 3 (25%)

### 3.1 (4%)

Creating table Trip.

#TripID  
\*UserID  
\*BikeID  
StartTime  
EndTime  
\*StartStationID  
\*EndStationID

### 3.2 (10%)



### 3.3 (5%)

See the reference for full examples of DBMS normalizations: (<https://beginnersbook.com/2015/05/normalization-in-dbms/>)

**First normal form (1NF)** As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

**Second normal form (2NF)** A table is said to be in 2NF if both the following conditions hold: Table is in 1NF (First normal form) No non-prime attribute is dependent on the proper subset of any candidate key of table.

**Third Normal form (3NF)** A table design is said to be in 3NF if both the following conditions hold: Table must be in 2NF Transitive functional dependency of non-prime attribute on any super key should be removed.

**Boyce Codd normal form (BCNF)** It is an advance version of 3NF that's why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every functional dependency  $X \rightarrow Y$ ,  $X$  should be the super key of the table.

### 3.4 (6%)

The most prominent normalization step is that Bike.Repairation Status is not atomic. Several users send in complaints to a single row. A way to solve this is to create 2 tables Complaint and Reparation.

#ComplaintID, \*UserID, \*BikeID, Description

#ReparationID, \*BikeID, \*ComplaintID (Could have other attributes, time, status etc)

There are other changes that also could be made, it is important that the student argues why they do the changes that they perform. Here are some examples of dependencies that could be further normalized.

- Station.Name/Position decides Station.maximumparkingspots
- User.UserID decides User.Name/phone\_number
- Subscription.Type and Subscription.Start decides Subscription.End

## 4 Problem 4 (20%)

### 4.1 (15%)

4 Possible points, one for every correct relationship highlighted in orange.

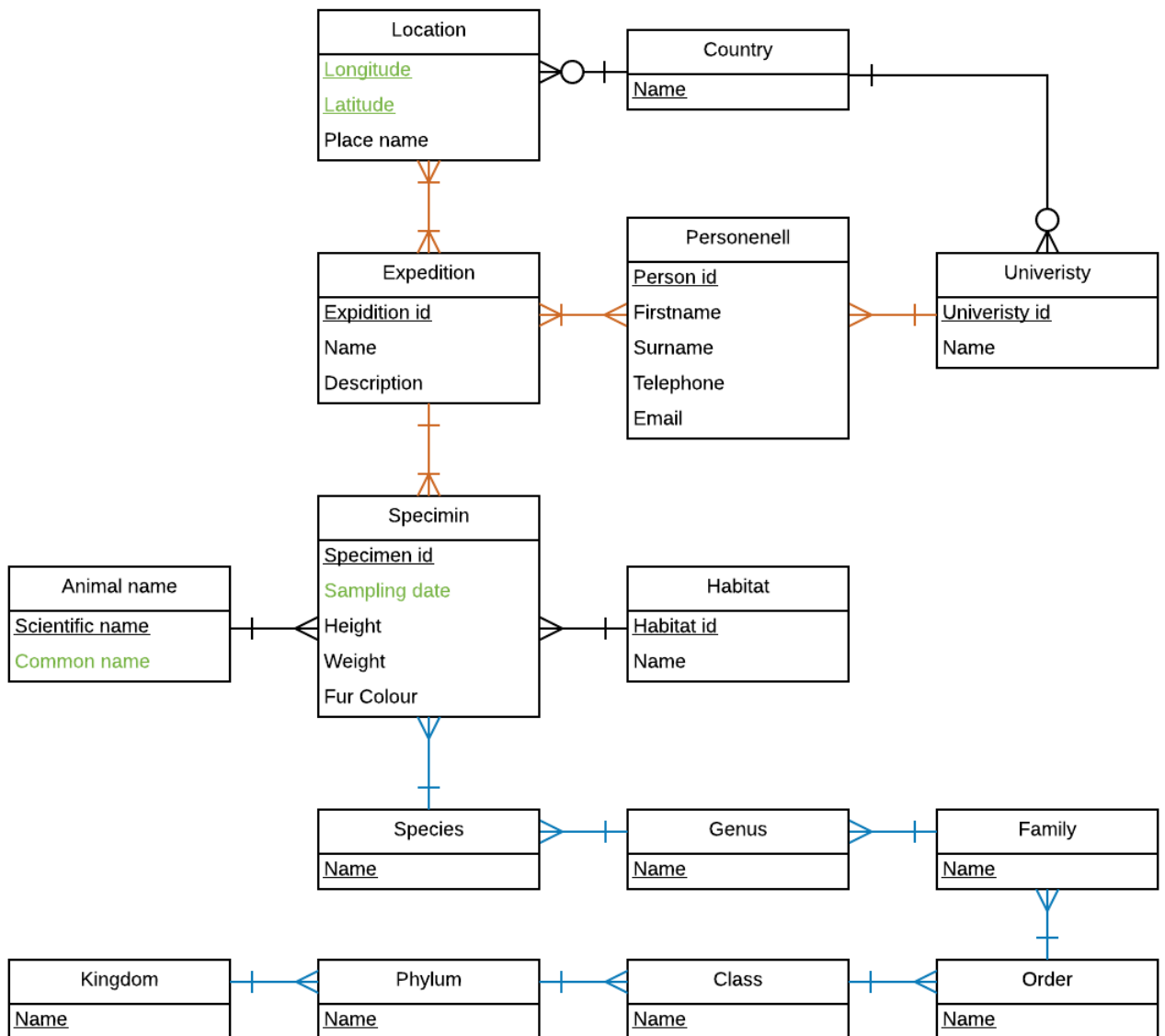
7 Possible points, one for every correctly specified relationship highlighted in blue.

(It is NOT possible/okay to put all levels of taxonomy in the same table.)

4 Possible points for correctly identifying the attributes in green.

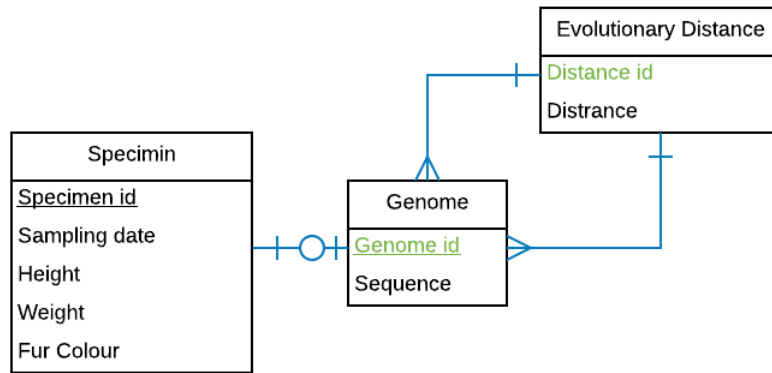
-1 Point for any table without a suitable primary key.

Clarifications: Be careful when reading names of entities and attributes in this task, as they are not necessarily the same for everyone. What is important, is that they represent the same thing.



## 4.2 (5%)

3 Possible points, one for each correctly specified relationship (Highlighted in blue). 2 Possible points, one for each suitable primary key (Highlighted in green)



## 5 Problem 5 (25%)

### 5.1 (2%)

2 Points if they identify that Labtest is not in 1NF.

Patient: BCNF

Sample: BCNF

Labtest: Not even 1NF

Resistance: not atomic

HospitalLocation: BCNF

PatientLocation: BCNF

### 5.2 (5%)

3 Points for identifying that OutbreakID and FeatureID is not atomic. 2 Extra points, for problems the student identify that you find reasonable.

### 5.3 (6%)

4 points if they find the functional dependencies:

*GenomeID OutbreakID*

*GenomeID FeatureID*

*GenomeID PatientID*

### 5.4 (2%)

2 Points if they find the appropriate candidate key: (GenomeID).

### 5.5 (10%)

Patient(PatientID, FirstName, SurName, Postcode, Address)

Sample(SampleID, SampleDate, PatientID\*)

Labtest(TestID, TestName, SampleID\*)

**ResistantTo(VaccineName\*, TestID\*)** Taken out of Labtest.

**Vaccines(VaccineName)** (To make it possible for both labtest, and ResistanceDeterminants to

find the correct VaccineName.)

HospitalLocation(LocationID, LocationName)

PatientLocation(PatientID\*, LocationID\*)

GenomeSequence(GenomeID, PatientID\*)

**GenomeToFeature(GenomeID\*,FeatureID\*)** Taken out of GenomeSequence because FeatureID is not Atomic.

**ResistanceDeterminats(FeatureID, VaccineName\*)** Taken out of GenomeSequence and GenomeToFeature, to be able to find the virus associated with the given featureID.

**GenomeToOutbreak(GenomeID\*,OutbreakID\*)** Taken out of GenomeSequence, because OutbreakID is not Atomic.

No more Normalization is needed.

Now that they are conforming to 1NF, the tables actually conforms to BCNF as well.