INF115 Databases and Modelling
Obligatory Assignment 2

Deadline: Friday 25th of March, (18.00) 25.03.2022

Contents

Submission

This assignment deals with the use of entity relationship (ER) diagrams and normalization in database design. You should submit your answers as a pdf file before the deadline. We recommend using Lucidchart (`https://www.lucidchart.com`), to make the ER diagrams. Note that you should enable the UML diagram tool (`https://www.lucidchart.com/pages/examples/uml_diagram_tool`). You can download individual diagrams as png images that can then be combined into a single pdf document. Primary keys should be marked with "#" (see Fig. 1).

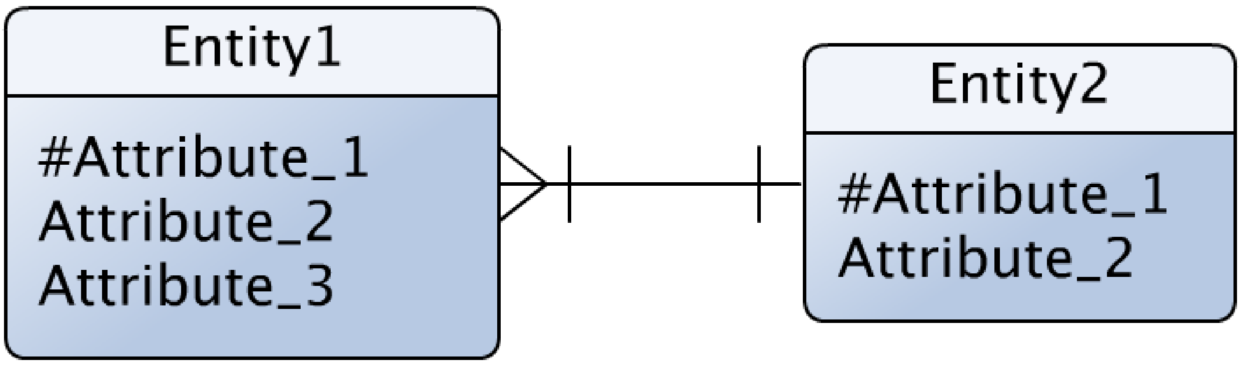


Figure 1: Example ER diagram. Attribute_1 of Entity1 and Entity2 are primary keys.

For some problems you will be asked to create tables. We give an example below to show the format you should use. Mark primary keys with "#" and foreign keys with "*".

- Table_1(#Attribute_1, Attribute_2, Attribute_3)
- Table_2(#Attribute_1, *Attribute_3)

Important! If you work together in groups on this hand-in you must name all your colleagues that you worked with on the document that you hand-in.

You may use various sources to help you solve the problems, but remember to cite the sources in the code (as comments) when significant parts of the code are taken from sources.

Deliver a clearly written document, with headings for each question answered, example:
4.a
answer or figure...

4.b
answer or figure...

Points will be deducted if formatting is unclear.

Good luck!

Introduction

You have been hired to create a database design for the leading bike commuting company in Bergen, Bergen Bysykkel. They need you to provide specific ER diagrams to developers such that they can implement your design. They give you the following information on the elements of the Bysykkel operation. You must assess how each entity of the operation is to be represented in database design (ER-diagram). Following is the entities given, and their respective properties:

Bergen Bysykkel

Bike

- Each bike must have a unique ID-number
- The bikes name, a random name which are not necessarily unique (it could be Arne or Fatima)
- The last station it was registered at. If the bike is active, it is the station it left from.
- Current status. What status should the bike have? For example, "Active", "Parked" or "Missing"
- Reparation status (Users can send in complaints about the bikes, hence the reparation status can either be NULL or any number of complaints from a defined list of 12 possible complaints: ("flat tire", "breaks not working", "gear not working"). So this element can contain multiple values ("flat tire", "flat tire", "flat tire") given from 3 different users). When a mechanic has fixed the complaints given, the element is set back to NULL.

User

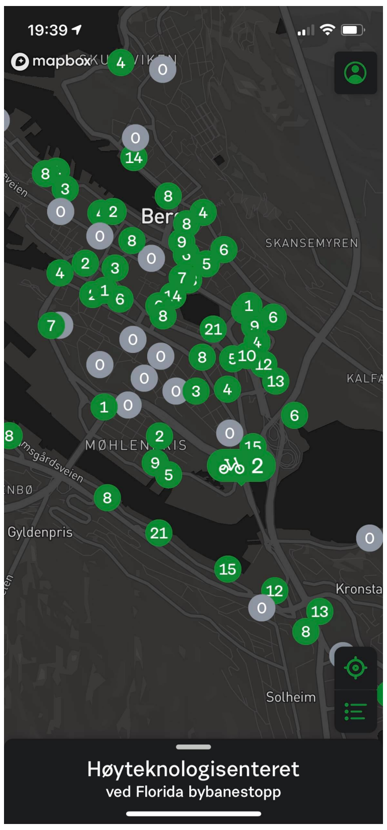
- A uniquely ID-number
- A name
- Their telephone number (Can several users have the same phone number?)
- Their GPS position (This is used to see where the nearest bicycle station is and is stored in latitude and longitude. For example in SQL:
  - Latitude FLOAT(10,6)
  - Longitude FLOAT(10,6)
  - *If you would like to further research how coordinates are stored and used in databases check out the Google Maps documentation: (we learn about PHP and XML in the next assignment) (`https://developers.google.com/maps/documentation/javascript/mysql-to-maps#createtable`)*

Station

- A unique ID-number
- A station name
- Its GPS coordinates
- Maximum parking spots
- Available parking spots

Problem 1 (10%)

1. Create an ER diagram from the description given above. The diagram should represent the entities and their attributes. Mark primary and foreign keys, and include relational arrows.
2. Given the current three entities, do all three tables contain a relationship. Example: does a user own a bike?



Høyteknologisenteret
ved Florida bybanestopp

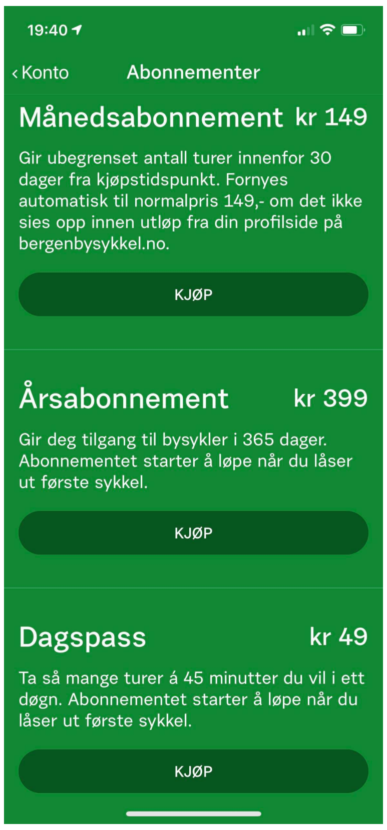
The Bysykkel app

Problem 2 (20%)

Bergen Bysykkel ha now figured out that they want further improvements to their database design. Until now the design has not included handling of subscription. This must be implemented to keep track of who has currently got a membership and who have had it before. A user can have more than one membership (the 1 active and older inactive ones). A fellow junior developer proposes that we add the following attributes to the User entity:

- SubscriptionID
- Status
- Start
- End
- Type

1. What is the problem with the proposed solution? Are there unwanted dependencies if we add the attributes that the junior developer proposes? Identify the problem.
2. Create table / tables to solve the problem. Mark primary keys, and foreign keys.
3. Based on the tables created, produce the extended ER diagram. Include appropriate primary, foreign keys and relations.


19:40

‹ Konto Abonnementer

Månedsabonnement kr 149

Gir ubegrenset antall turer innenfor 30 dager fra kjøpstidspunkt. Fornyes automatisk til normalpris 149,- om det ikke sies opp innen utløp fra din profilside på bergenbysykkel.no.

KJØP

Årsabonnement kr 399

Gir deg tilgang til bysykler i 365 dager. Abonnementet starter å løpe når du låser ut første sykkel.

KJØP

Dagspass kr 49

Ta så mange turer á 45 minutter du vil i ett døgn. Abonnementet starter å løpe når du låser ut første sykkel.

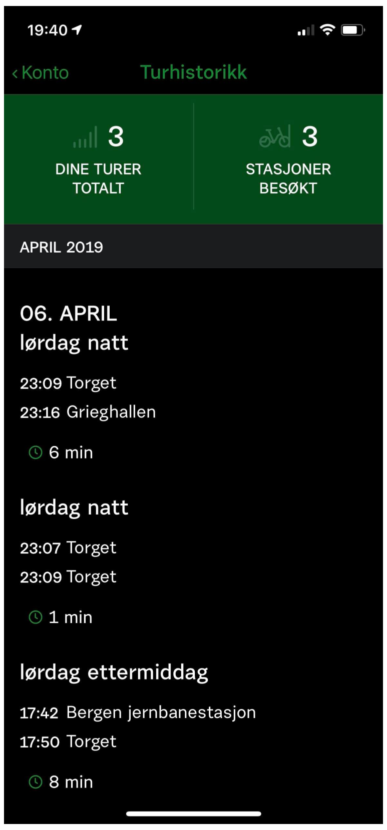
KJØP

Subscripton options

Problem 3 (25%)

Great! Directors at Bergen Bysykkel like your solution to the Subscription problem, and have decided that you are promoted to Lead Developer, and they have given you the responsibility of the next big problem to tackle with building their app. The design, in its current state, does not register who uses each bike. This is a problem and you now have to design the **Trip** entity to take this into account. A Trip has to keep track of the user, the bike, the start time, the end time, the start station and the end station.

1. Identify the attributes in the database description above and create a table.
2. Extend your ER diagram with this new entity and represent the relations between the entities that you identified in the previous question. Specify the primary keys and choose appropriate relationships between the entities.
3. The current database design is taking too much storage (some of your tables contains redundant information). Luckily you learned in INF115 lectures that there are 4 normalization forms to reduce size of a database system. First, explain the four normalization forms(1NF, 2NF, 3NF, BCNF).
4. Your boss wants you to implement the highest normalization form BCNF. Describe what changes you would have made to the current design to achieve this, and reduce redundancy. Then create tables to your suggestion.(You don't need to extend your ER-diagram) Mark primary keys and foreign keys. Explain the changes you make. Example: Is Reparation Status atomic (1NF)?



3 DINE TURER TOTALT

3 STASJONER BESØKT

APRIL 2019

06. APRIL

lørdag natt

23:09 Torget
23:16 Grieghallen
6 min

lørdag natt

23:07 Torget
23:09 Torget
1 min

lørdag ettermiddag

17:42 Bergen jernbanestasjon
17:50 Torget
8 min

Trips

Problem 4 (20%)

After the great success of the Bysykkel App, a group of scientists wants you to design a database for their expeditions around the world to catalogue all of the animals that they can find at a particular location. The scientists propose storing the information from each expedition in a database that will allow them to explore the evolutionary connections between the animals and locations.

The system must store information about each expedition, including

- which locations (longitude and latitude) were visited in which countries.
- the dates of those visits and the personnel who were on the expedition.

Note that the length and scale of an expedition may vary. Some expeditions visit just one location in one country, whereas others involve multiple locations in multiples countries over the course of a few weeks.

A minimum of two scientists will be present on each expedition, their names and contact information should also be stored in the database along with their university affiliation and the country that the university is from. The expedition will collect information about the animals from each of the locations that they visit, storing the sampling date and the type of habitat where each specimen was found in. A number of traits will be recorded for each specimen, (such as height, weight and fur colour) which will be used to compare the specimens after the expeditions. When analyzing the data it should be possible to organise the specimens by common name or scientific name, and also by any level of the taxonomy hierarchy (kingdom, phylum, class, order, family, genus and species).

Subproblem 1

Create an ER diagram for tracking this information. Specify the primary keys and choose appropriate relationships between the entities.

Subproblem 2

Modern DNA sequencing technology allows scientists to capture the entire genetic code of an organism. Comparing the generic codes of various organisms is particularly useful to evolutionary biologists, as the number of differences (mutations that have occurred over time) between two evolutionary related sequences can provide an accurate measure of the evolutionary distance between those organisms: examples are (human - chimpanzee (98.5%), human - dog (85%), human - yeast (25%))

Expand the ER diagram you created in the previous Subproblem to include genetic information. Note that you need to hand in both diagrams separately and thus have to first make a copy of the previous ER diagram. For the purposes of this exercise, assume that we can efficiently store the genome sequence of each specimen in a table. Each species has a single genome but due to the costs involved not all of the specimens will have their genomes sequenced. For each genome sequence we will calculate and store the evolutionary distance to all other genome sequences in the database.

Problem 5 (25%)

You are now a well renowned database architect in the research community, and The Norwegian government wants you to design their database for tracking coronavirus infections that are resistant to vaccines across the country. Some mutants of the virus are more resistant to certain vaccines, and when an outbreak is detected the government needs to be able to quickly identify the regions and individuals involved in order provide the correct treatment, and limit the extent of the outbreak. Keep in mind that there are many kinds of vaccines and that an infection may be resistant to several of them.

Subproblem 1

The following database is already in place to keep track of the tests that have been performed on patients, and the locations that the patients have visited. The lab test results can be either negative (in which case there is no resistance), or positive (in which case resistance to one or more vaccine has been detected). Primary keys are marked with "#" and foreign keys with "*".

- Patient(#PatientID, FirstName, SurName, Postcode, Address)
- Sample(#SampleID, SampleDate, PatientID*)
- Labtest(#TestID, TestName, SampleID*, ResitantToVaccine)
- Hospital(#LocationID, Region)
- PatientLocation(#PatientID*, #LocationID*)

Are there any problems with these tables. What is the highest normalization level that each table conforms to (1NF, 2NF, 3NF or BCNF)?

Subproblem 2

Sequencing technologies are also useful for outbreak tracking. Viral genomes are much smaller than human genomes and can be sequenced relatively quickly. By comparing the genome sequences of viruses we can determine if they are related ,i.e., part of the same outbreak (known in the media as Wuhan, British, Brazilian, Follo outbreaks). Furthermore, when outbreaks occur we can determine the evolutionary history and transmission paths of the virus, i.e., seeing who infected who, to determine the source of the outbreak. We can also perform in silico testing (experiment on the computer) (rather than time consuming laboratory testing), by scanning the genome sequences for genes and mutations that are known to be associated with vaccine resistance (collectively called resistance determinants).

The government would like to expand this system to store the genetic information of the viruses that are infecting each patient. When a patient is suspected of having a resistant infection, a sample will be taken from which the genome sequence of the virus is determined. The genome sequence will be in silico tested for resistance determinants, and will be assigned to an outbreak if a related genome is found in the database. Multiple samples can be taken from a patient at different time points. It is possible that (particularly unfortunate) patients could be infected with coronavirus' from multiple outbreaks (British and Brazilian variant at the same time).

The following tables have been proposed to incorporate the new requirements:

- GenomeSequence(#GenomeID, PatientID*, OutbreakID, FeatureID*)
- Resistance(#FeatureID, VaccineName)

An example row would be:

- GenomeSequence(GID0001,PID0007,OUTBREAK0001,FEATURE0001)

- Resistance(FEATURE0001, Pfizer)

This means that a genome with the ID GID0001 has been isolated from patient PID0007, that the genome found is associated with outbreak OUTBREAK0001 (Wuhan_december_2022), and that it contains a feature predicted to make it resistant to the vaccine Pfizer.

Given this information, answer the following questions

1. Why might the solution consisting of the additional tables above be problematic?
2. Give the functional dependencies of the GenomeSequence table.
3. Determine the candidate key(s) for the GenomeSequence table.
4. Perform step by step normalization from first normal norm (1NF) to Boyce-Codd normal form (BCNF) for the whole database (the original tables, with any changes that you think might be appropriate, plus the additional tables with genome information). Mark primary keys and foreign keys in the tables.