Exercise 2: TDT4145 - Data Modelling, Databases and Database Management Systems

alexaoh, jimoskar

February 2021

Task 1: Constraints and syntax with specialization

a)

- Disjointness: This means that each entity in the superclass may belong to either one of the subclasses, but not to several of them. In other words: The subclasses of a specialization must be disjoint sets.
- Completeness/totalness: This means that each of the superclass entities must be an entity in at least one of the subclasses. Whether the superclass is an entity in several of the subclasses is defined via the disjointness/overlapping definition, as discussed briefly in the above point.

b)

i) Disjoint and total: An example of a miniworld could be registering the biological sex of people: either one is a female or a male, but one cannot be both.

ii) Disjoint and partial: An example of a miniworld could be origins of car brands, where we only have American and Japanese in the model. Each of the cars in the superclass could be either American, Japanese or of some other origin that is not modelled here.

iii) Overlapping and total: An example of a miniworld could be registering people in a hospital. Either they are there because they are sick, because they work there or because they are relatives of someone that are sick. If they are in the hospital, they must be identifiable with at least one of these subclasses.

iv) Overlapping and partial: An example of a miniworld could be registering only the patients in the aforementioned hospital. In this model we only have two categories of diseases: cardiovascular and cancer. Each patient could be receiving treatment for one or both of these categories, as well as other types of diseases that are not modelled as subclasses here.

c)

The syntactically invalid ER models are figures 1 and 4. Firstly, figure 1 is invalid because the relationship class 'Prefers' is directly connected to a specialization with no superclass.

Deleting the relationship class 'Prefers' could correct the syntactical error. Secondly, figure 4 is invalid because there is no superclass connected to the two disjoint subclasses. Perhaps the entity class 'Possession' was supposed to be the superclass?

Task 2: ER model of a zoo

The proposed model is shown in figure 1.

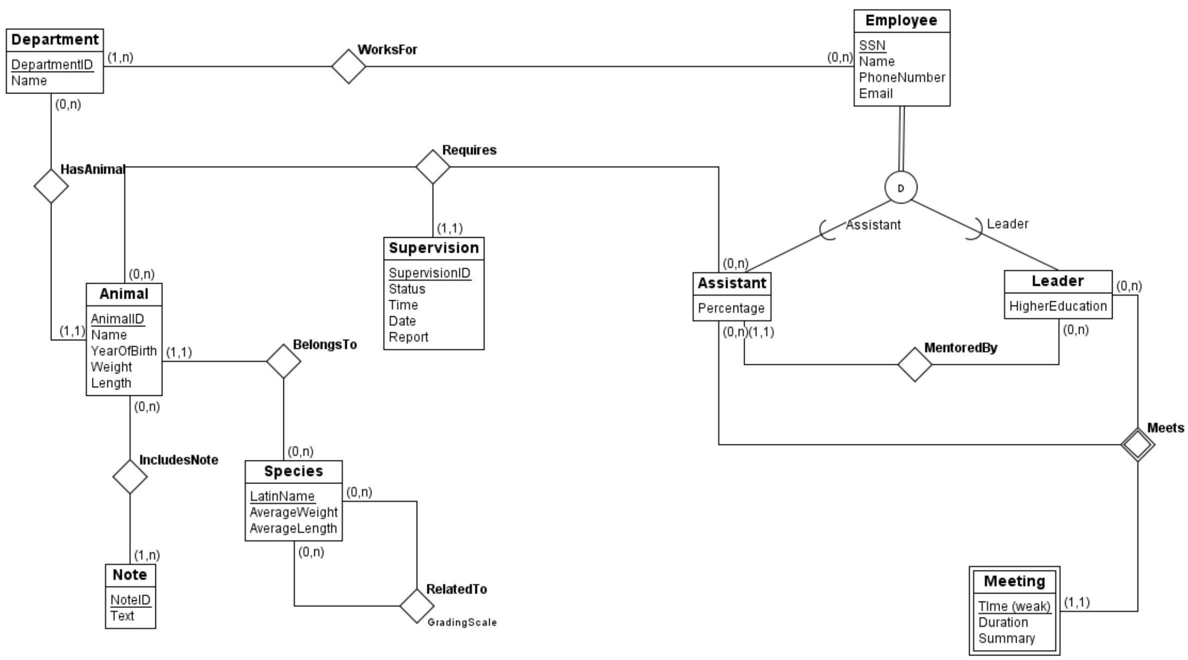

Figure 1: Zoo. Bedre å unngå tertiære relasjoner så langt det lar seg gjøre. Her kan man heller ha enkle relasjoner mellom Meeting og Assistant/Leader, samt mellom Supervision og Animal/Assistant, og heller bruke eksistensavhengighet (1,1) fra Supervision og Meeting til de andre entitetene.

Task 3: Important terms in the relational data model

The coherence between primary keys (PKs) and entity integrity can be explained as follows. The entity integrity asserts that all tuples in each relation can be identified in a unique way, by demanding a NOT NULL value in the primary key. If the primary key in a tuple is allowed to have NULL value, then it cannot be identified, which makes it useless. Moreover, one relation is a set of tuples, which means that one cannot have several identical tuples. This is another reason to why the PK is important for entity integrity.

Similarly, the coherence between foreign keys (FKs) and referential integrity can be explained as follows. The referential integrity asserts that each tuple in the referencing relation references an existing tuple in the referenced relation. The FK is important in this context, since this is the key in the referencing relation that is used to refer to the PK in the referenced relation. For the referential integrity to hold, the domains of the FKs and PKs in question have to match and the value of FK in a tuple in the referencing relation has to either occur as a value of the PK in a tuple in the referenced relation or be set as NULL. In this way, the FK is valid, and the referential integrity constraint is met.

Task 4: Conversion between models and relational algebra

a)

The relations are shown below

Exam(<u>ExamNo</u>, CourseCode, ExaminationAids)

Student(<u>StudentNo</u>, Name)

HasExam(<u>ExamNo</u>, <u>StudentNo</u>)

ExaminationLocation(<u>RoomNo</u>, Name, Capacity)

SetUp(<u>ExamLoc</u>, <u>ExamNo</u>, <u>StudentNo</u>, Date, StudentPlacement)

Table(<u>TableNo</u>, Type, ExaminationLoc)

Chair(<u>ChairNo</u>, Type, ExaminationLoc)

Burde vi skrive constraints her? Ja, det er lurt!

b)

1. $\pi_{HotelNo,Name}(Hotel)$
2. $\pi_{HotelNo,Name}(\sigma_{Area=Barcelona}(Hotel))$
3. $LargeRooms \leftarrow \sigma_{SquareMeterSize>100}(HotelRoom)$
   $Joined \leftarrow LargeRoom \star Hotel$
   $\pi_{RoomNo,Name}(Joined)$
4. $SmallRoom \leftarrow \sigma_{SquareMeterSize<8}(HotelRoom)$
   $SmallOrder \leftarrow \sigma_{Duration>7}(Order)$
   $Joined \leftarrow SmallRoom \star SmallOrder$
   $\Im_{COUNT\ OrderNo}(Joined)$

5. $Madrid \leftarrow \sigma_{Area='Madrid'}(Hotel)$
   $Joined \leftarrow Madrid \star (HotelRoom \star (Customer \star Order))$
   $\rho_{FullName}(\pi_{FirstName+' '+LastName}(Joined))$Spørre om denne projeksjonen er innafor? (Mulig.

6. $Hansen \leftarrow \sigma_{FirstName='Ole' AND LastName='Hansen'}(Customer)$
   $HansenDuration \leftarrow Hansen \star Order$
   $\tau_{Duration}(HansenDuration)$ Spørre om denne notasjonen? (Aldri sett, men mulig.)