201300180 Data & Information – Test 2 – Solutions

Question 1

a) The generalization is both covering and disjoint, which gives the following options:

- three tables `Friend, Friend1, Friend2`
- two tables `Friend1, Friend2` (both including attributes of 'Friend')
- a single table `Friend` with null values for attributes of 'Friend1' where appropriate

b) The option with tree tables is easiest to accomodate the associations, see variant 1.
Discarding the superclass is possible, but then we get two different assocations 'likes', one for each subclass. See variant 2.
If the subclasses are discarded and all friends are modelled in a single table, it becomes very very complicated to model the assocation 'Has_invited'. This is hardly a viable option.

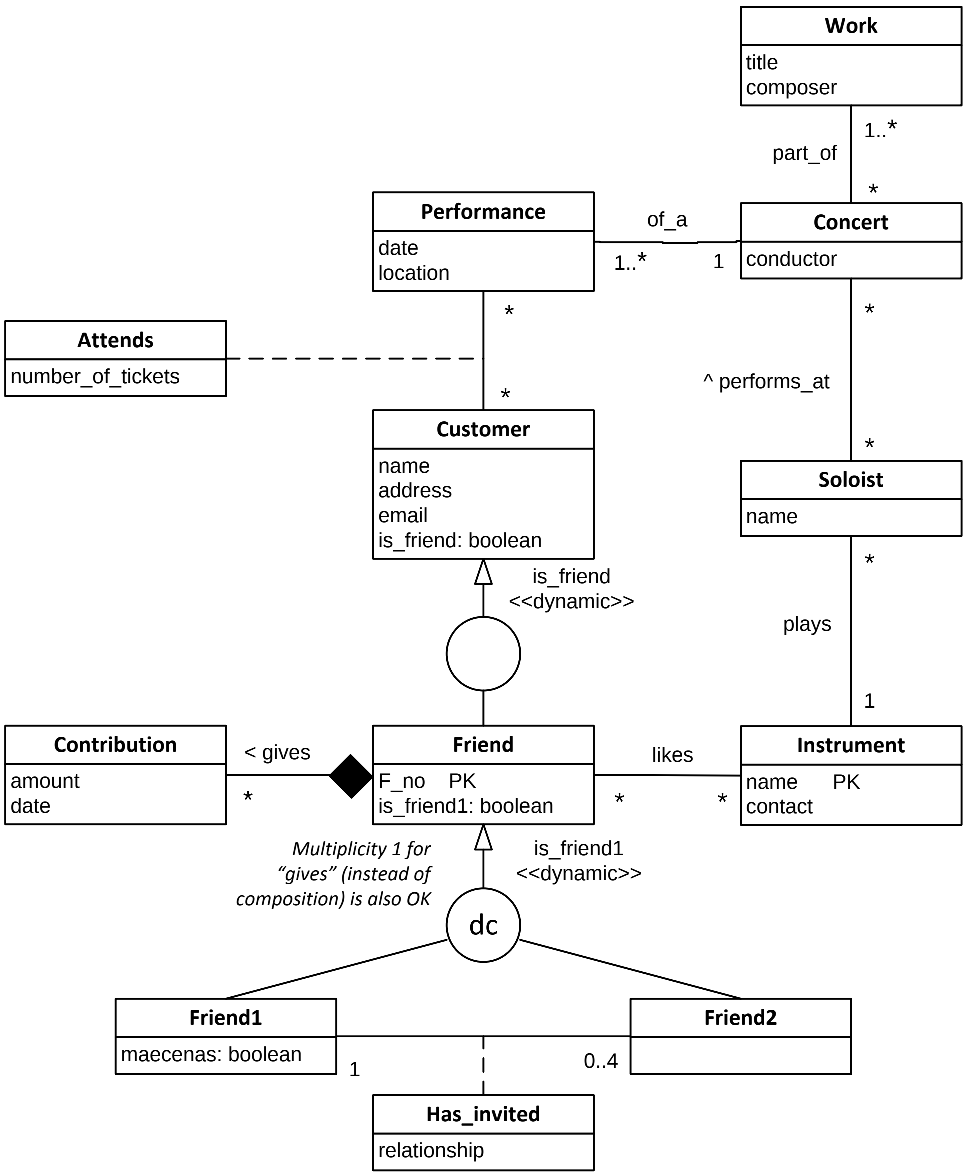
Variant 1

```
Friend(f_no, name, address, email, is_friend1,
      PK (f_no));
Friend1(f_no, maecenas,
      PK (f_no),
      FK (f_no) REF Friend,
      CHECK (4 <= (SELECT COUNT (*)
            FROM Friend2
            WHERE f_no = Friend2.f_1 )));
Friend2(f_no, f_1 NOT NULL, relationship,
      PK (f_no),
      FK (f_no) REF Friend,
      FK (f_1) REF Friend1 );
Likes(f_no, instr,
      PK (f_no, instr),
      FK (f_no) REF Friend,
      FK (instr) REF Instrument(name));
```

Variant 2

```
Friend1(f_no, name, addresss, email, maecenas,
      PK (f_no),
      CHECK (4 <= (SELECT COUNT (*)
            FROM Friend2
            WHERE f_no = Friend2.f_1 )));
Friend2(f_no, name, address, email, f_1 NOT NULL, relationship,
      PK (f_no),
      FK (f_1) REF Friend1 );
Likes1(f_no, instr,
      PK (f_no, instr),
      FK (f_no) REF Friend1,
      FK (instr) REF Instrument(name));
Likes2(f_no, instr,
      PK (f_no, instr),
      FK (f_no) REF Friend2,
      FK (instr) REF Instrument(name));
```

Question 2



Comments:

- Note the distinction between *Concert* and *Performance* of a concert. (You could also have called it *Concert type* and *Concert*).
- From the description it is not clear whether the location can differ across performances of the same concert. If you have location as attribute of Concert (type) this is also correct.
- For customer no further attributes are given. So it is a matter of choice which attributes of "Friend" you want to share with "Customer" (and then discard them in the subclass "Friend"). Name and address are a reasonable choice. Email could be specific for friends, or for any customer.
- Whether [Friend] *gives* [Contribution] is a composition is arguable.
  Pro: the contributions necessarily depends on a friend; without this friend no contribution.
  Con: but if the friend ever gets deleted from the system, the money remains to be donated.
- In many cases where multiplicity * is indicated, 1..* is also acceptable (e.g. someone is not a customer unless they visit at least one performance). The reverse is not true (e.g. a concert without works makes no sense).

Question 3a

i)	$E \longrightarrow P$	no	h: an employee can have different phone numbers
ii)	$L \longrightarrow C$	no	there could be different contracts for different cars with the same last day
iii)	$R \longrightarrow T$	yes	b: $R \longrightarrow C$; 2: $C \longrightarrow T$; therefore $R \longrightarrow T$
iv)	$R \longrightarrow AT$	yes	similarly, from a and e we find $R \longrightarrow A$, combined with iii) $R \longrightarrow AT$
v)	$DF \longrightarrow R$	yes	c: if the same driver has more rentals, they do not overlap. Therefore Driver and Final date uniquely identify a Rental.
vi)	$L \longrightarrow T$	no	from *ii)* we know $L \not\longrightarrow C$, so there is no way that M could depend on L
vii)	$CD \longrightarrow F$	no	a driver could have made different rentals for the same car
viii)	$DFE \longrightarrow T$	yes	*v)* + *iii)* yield $DF \longrightarrow T$. Then also $DFE \longrightarrow T$ (even though E has nothing to do with the involved attributes)
ix)	$R \twoheadrightarrow EP$	yes	E,P are completely independent of the attributes T,C,D,A,F,L.
x)	$C \twoheadrightarrow RL$	no	R,L are not independent of T,D,A,F,E,P, e.g. $R \longrightarrow F$.

Question 3b

1) In order to find out which FDs violate the BCNF condition, we first have to establish the candidate keys. Schema *R* has two candidate keys: *DEG* and *DFG*.
(You can find these by starting with *ABCDEFG* as a trivial superkey, and discard attributes that are fuctionally dependent. E.g. *ABCDEFG* is a superkey. Then, because $A \longrightarrow BC$, it also holds that *ADEFG* is a superkey. Because $DE \longrightarrow AF$ it also holds that *DEG* is a superkey. As there is no functional dependency between *D*, *E*, and *G*, we conclude that *DEG* is a candidate key. Similar for *DFG*.)

   All FDs in $\mathfrak{F}$ violate the BCNF condition, because all of them have a left-hand side that is not a superkey.

2) First, determine $\mathfrak{F}^+ = \{ A \longrightarrow BC, B \longrightarrow AC, DE \longrightarrow ABCF, FG \longrightarrow ABCE \}$
(where $A \longrightarrow BC$ is a shorthand for $A \longrightarrow B$, $A \longrightarrow C$)

   Start with (arbitrarily chosen) functional dependency $A \longrightarrow BC$.

   $(A)^+$ = ABC. Splitting over A we get
   - $R_1(A,B,C)$, with $\mathfrak{F}_1 = \{ A \longrightarrow BC, B \longrightarrow AC \}$
   - $R_2(A,D,E,F,G)$, with $\mathfrak{F}_2 = \{ DE \longrightarrow AF, FG \longrightarrow AE \}$

   Clearly, R_1 is in BCNF, candidate keys are *A* and *B*.

   For R_2 we have still have the two candidate keys *DEG* and *DFG*.
   R_2 is not in BCNF, both remaining FDs violate the condition.

   So we split R_2 (arbitrarily chosen) on $DE \longrightarrow AF$ and determine $(DE)^+$ = ADEF).
   This yields

   - $R_{21}(A,D,E,F)$, with $\mathfrak{F}_{11} = \{ DE \longrightarrow AF \}$
   - $R_{22}(D,E,G)$, with $\mathfrak{F}_{12} = \{ \}$

3) From the original functional dependencies, $FG \longrightarrow AE$ was lost in the decomposition in step 2.
The other FDs still exist in $\mathfrak{F}_1 \cup \mathfrak{F}_{21} \cup \mathfrak{F}_{22}$.

Alternatively,

If we had chosen $FG \longrightarrow AE$ as the basis for further decomposition of R_2, we would have obtained

- $R_{21}(A,E,F,G)$, with $\mathfrak{F}_{11} = \{ FG \longrightarrow AE \}$
- $R_{22}(D,F,G)$, with $\mathfrak{F}_{12} = \{ \}$

with dependency $DE \longrightarrow AF$ lost in the decomposition.

If we would have started with $DE \longrightarrow AF$ or $FG \longrightarrow AE$, eventually we would have obtained one of the above results.