Data & Information – Test 2: Solutions

24 May 2017, 13:45–15:15

Question 1 (Database Schema)

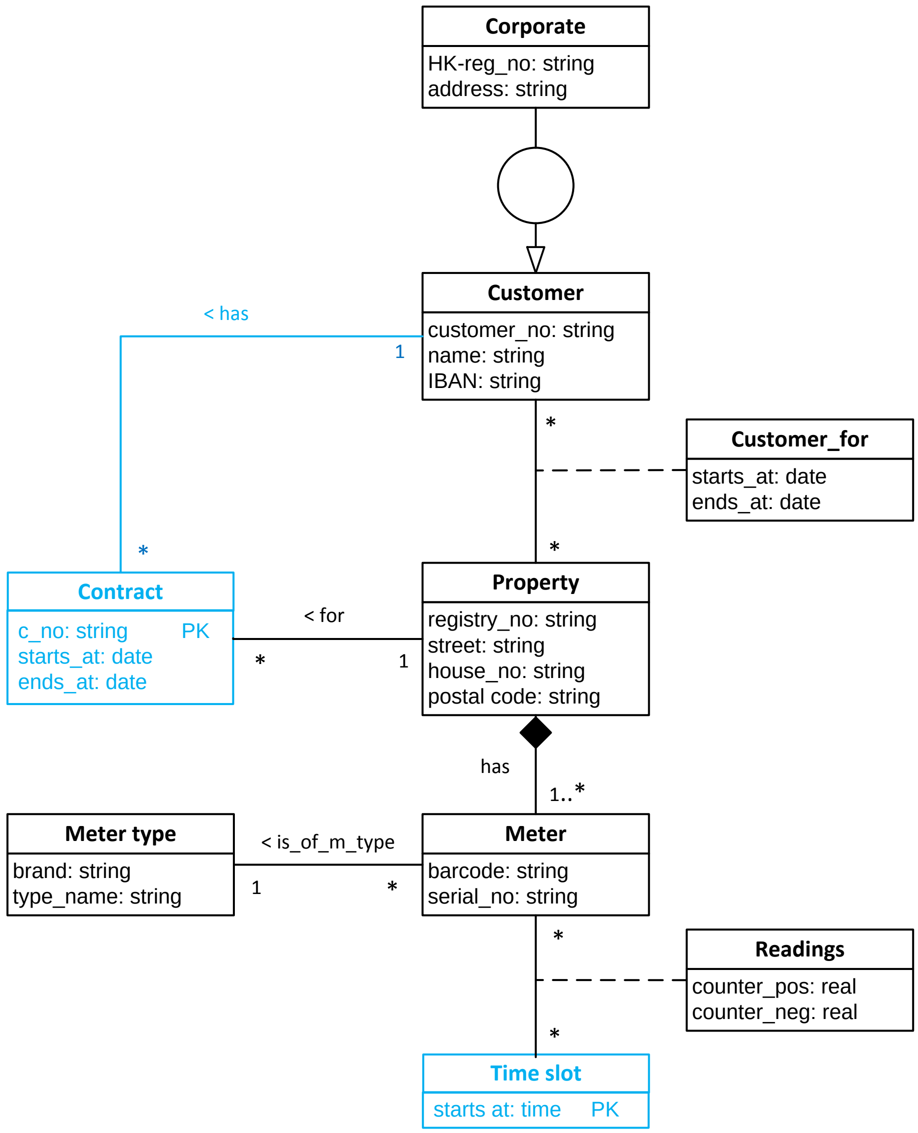
1a)

```
Contract_type(name, conditions,
     PK(name));

Flexible_contract_type(name,
     PK(name),
     FK(name) REF Contract_type (name)

Time_slot(start_at, price, name NOT NULL,
     PK(starts_at),
     FK(name) REF Flexible_contract_type(name));
```

1b) Alternative I: **only tables for the subclasses**
Difficulty: Contract should be associated with each of the subclasses individually, (complicated) CHECKs needed to make sure that a contract is associated with a single contract type.

Alternative II: **only a table for the superclass**
Difficulty: Time slots should be associated only with flexible contracts; can be done by including a CHECK in *Time_slot*.

Question 2 (Class Diagram)



Remarks:

- Another option for the generalization of Customer is to have subclasses for *Corporate* and *Private* customers. The generalication is then “dc”, *Private* has no attributes.
- Assocation *has* (indicated in blue) is OK, but rendant, the information can be retrieved from the attributes of *Contract* and *Customer_for* and the association *for*.
- Many multiciplicites that are given as “*” could have been more strictly defined as “1..*”. Both are regarded as correct.

Question 3 (Normalization)

3a)

a) $M\ T \longrightarrow R$ — **Yes**, from 6.

b) $R\ T \longrightarrow M$ — **No**. There could be different meters which have the same reading at the same moment

c) $Pp\ D \longrightarrow Cu$ — **Yes**, from 3.

d) $Co\ D \longrightarrow Pp$ — **Yes**. $Co \longrightarrow Pp$ follows from 1, then $Co\ D \longrightarrow Pp$ holds a fortiori

e) $Cu\ D \longrightarrow Pp$ — **No**. A customer could have different properties for which he has different contracts

f) $M \longrightarrow Co$ — **No**. If a property gets a new contract it does not imply that the meter is changed

g) $Co \longrightarrow Pt$ — **No**. There are many different (time slots with) prices in a contract.

h) $Co \longrightarrow I$ — **Yes**, from 1 ($Co \longrightarrow Cu$) and 2 ($Cu \longrightarrow I$)

i) $Co \twoheadrightarrow I$ — **Yes**. In general, $X \longrightarrow Y$ implies $X \twoheadrightarrow Y$

j) $Co \twoheadrightarrow Cu\ Pp$ — *(correct answer:)*
Yes. Statement 1 implies Co $\longrightarrow Cu\ Pp$ which implies $Co \twoheadrightarrow Cu\ Pp$
(original answer, also graded as correct:)
No. This would mean that *Cu Pp* on the one hand and *D I M Pt R T* on the other hand are completely independent. This is not the case, e.g. $M \longrightarrow Pp$ (4)

Ad j): Statement 1, “A contract is for one customer and for one property”, implies that both customer and property are functionally dependent on contract, i.e., $Co \longrightarrow Cu\ Pp$.
As an FD implies an MVD (see answer to subqestion i)), therefore the correct answer is “Yes”.
The argument given for the answer “No” is in itself correct: Indeed it is the case that *Cu Pp* is <u>not</u> completely independent from *D I M Pt R T.* (So the persons who answered this know what they are talking about and deserve points for j).) However, in the context of the functional dependency the argument is irrelevant.

How is it possible that a correct argument contradicts the correct solution? For those who want to know precisely, the paradox can be explained as follows.

If the multivalued dependency holds, then, for a given *Co*,
if $Co\ Cu_1\ Pp_1\ D_1\ I_1\ M_1\ Pt_1\ R_1\ T_1$ is a tuple in R
and $Co\ Cu_2\ Pp_2\ D_2\ I_2\ M_2\ Pt_2\ R_2\ T_2$ is a tuple in R,
is must be the case that
$Co\ Cu_1\ Pp_1\ D_2\ I_2\ M_2\ Pt_2\ R_2\ T_2$ is a tuple in R
and $Co\ Cu_2\ Pp_2\ D_1\ I_1\ M_1\ Pt_1\ R_1\ T_1$ is a tuple in R,

For <u>arbitrary</u> values of *Cu* and *Pp* this would not hold; e.g. $M \longrightarrow Pp$ would demand that Pp_2 in the last tuple is in fact Pp_1 , it must be the case then that $Pp_2 = Pp_1$.
But the values of *Cu* and *Pp* are not arbitrary. The functional dependency $Co \longrightarrow Cu\ Pp$ implies that for a given *Co* there is only a single value for *Cu* and a single value for *Pp*. In other words, it is given that $Pp_2 = Pp_1$ and that $Cu_2 = Cu_1$, hence the MVD condition trivially holds.

3b)

1) First, determine $\mathfrak{F}^+ = \{ C \longrightarrow ABEFG, D \longrightarrow G, F \longrightarrow ABEG \}$
   Schema *R* has one candidate key: *CD*.
   All FDs in $\mathfrak{F}$ violate the BCNF condition,
   because all of them have a left-hand side that is not a superkey.

For 2) and 3) the solution differs depending on which – arbitrarily chosen – FD you start with.

(i) Start with (arbitrarily chosen) functional dependency $C \longrightarrow ABEFG$.
$(C)^+ = CABEFG$. Splitting over *C* we get
$R_1(C,A,B,E,F,G)$, with $\mathfrak{F}_1 = \{ C \longrightarrow ABEFG, F \longrightarrow ABEG \}$
$R_2(C,D)$, with $\mathfrak{F}_2 = \{ \}$

Clearly, R_2 is in BCNF, and has candidate key is CD.

For R_1 we find candidate key *C* (all other attributes depend on *C*).
R_1 is not in BCNF, however, as $F \longrightarrow ABEG$ violates the condition.

So we split R_1 on $F \longrightarrow ABEG$ and determine $(F)^+ = FABEG$.
This yields
$R_{11}(F,A,B,E,G)$, with $\mathfrak{F}_{11} = \{ F \longrightarrow ABEG \}$
$R_{12}(F,C)$, with $\mathfrak{F}_{12} = \{ C \longrightarrow F \}$
R_{11} has candidate key *F* and is in BCNF,
R_{12} has candidate key *C* and is in BCNF.

3) From the original functional dependencies, $D \longrightarrow G$ was lost in the decomposition in step 1.
The other FDs still exist in $\mathfrak{F}_{11} \cup \mathfrak{F}_{12} \cup \mathfrak{F}_2$.

(ii) Start with (arbitrarily chosen) functional dependency $F \longrightarrow ABEG$.
$(F)^+ = FABEG$. Splitting over *B* we get
$R_1(F,A,B,E,G)$, with $\mathfrak{F}_1 = \{ F \longrightarrow ABEG \}$
$R_2(F,C,D)$, with $\mathfrak{F}_2 = \{ C \longrightarrow F \}$

R_1 has candidate key F and is in BCNF

For R_2 we find candidate key *CD*
R_2 is not in BCNF, however, as $C \longrightarrow F$ violates the condition.

So we split R_2 on $C \longrightarrow F$ and determine $(C)^+ = CF$.
This yields
$R_{21}(C,F)$, with $\mathfrak{F}_{21} = \{ C \longrightarrow F \}$
$R_{22}(C,D)$, with $\mathfrak{F}_{22} = \{ \}$
R_{21} has candidate key *C* and is in BCNF,
R_{22} has is clearly in BCNF and has candidate key CD

3) From the original functional dependencies, $D \longrightarrow G$ was lost in the decomposition in step 1.

(iii) Start with (arbitrarily chosen) functional dependency $D \longrightarrow G$.

$(D)^+ = DG$. Splitting over D we get
$R_1(D,G)$, with $\mathfrak{F}_1 = \{ D \longrightarrow G \}$
$R_2(D,A,B,C,E,F)$, with $\mathfrak{F}_2 = \{ C \longrightarrow ABEF, F \longrightarrow ABE \}$

R_1 is in BCNF, candidate key is D.

For R_2 we find candidate key CD.
R_2 is not in BCNF, both FDs violate the condition.

So we split R_2 on (arbitrarily chosen) $F \longrightarrow ABE$ and determine $(F)^+ = FABE$.
This yields

$R_{21}(F,A,B,E)$, with $\mathfrak{F}_{21} = \{ F \longrightarrow ABE \}$
$R_{22}(\mathrm{F},C,D)$, with $\mathfrak{F}_{12} = \{ C \longrightarrow F \}$

R_{21} has candidate key F and is in BCNF,
R_{22} has candidate key CD and is not in BCNF, as the FD violates the condition.

So we split R_{22} on $C \longrightarrow F$ and determine $(C)^+ = CF$.
This yields

$R_{221}(C,F)$, with $\mathfrak{F}_{21} = \{ C \longrightarrow F \}$
$R_{222}(C,D)$, with $\mathfrak{F}_{12} = \{ \}$

R_{221} has candidate key C and is in BCNF,
R_{22} has candidate key CD and is in BCNF.

3) From the original functional dependencies, $F \longrightarrow G$ was lost in the decomposition in step 1.

Note that in iii) the second and third step can be reversed, giving the same result.