# Data & Information – Test 2 (1.5 hours)

**24 May 2017, 13:45–15:15**

**Program: Technical Computer Science / Business & IT**
**Module: Data & Information (201300180)**
**Module Coordinator: Klaas Sikkel**

Please note:

- Please answer every question on a different sheet of paper (the answers will be distributed to different person for grading).
- You are not allowed to bring any study materials to the test; essential excerpts from the study materials are available as appendices. You do not need a calculator.

Grade = #points/10

## Introduction: Energy Demand Management

*Questions 1, 2, and 3a relate to this case description*

Germany has made substantial investments in renewable energy. The installation of photovoltaic systems (solar energy) and wind turbines has grown spectacularly over the last decade.
However, when a substantial share of electricity production depends on these sources, this gives the complication that the production capacity varies with the weather conditions. One of the major technological challenges for the next decade is how to efficiently store and retrieve energy that is produced at peak hours.
Another way of dealing with peak levels in energy production and consumption is to induce citizens to use less electricity when supply is low / demand is high, and to shift electricity consumption to periods when supply is high / demand is low. This is called *energy demand management.*

The basic idea is very simple. The price of electricity varies across the day. For each time slot of (say) 30 minutes, the price (€/kWh) will be made available 24 hours in advance. On average, this flexible price will be a bit lower than the standard price; citizens can further reduce their energy bill by shifting some of their electricity consumption to time slots with a lower price.
For owners of an electrical car, energy demand management should be very attractive. With a simple interface you can indicate when you plan to use the car again (e.g. tomorrow 8:00), then the car will charge when electricity is cheapest.
Producers of washing machines and dishwashers are currently developing models that can ask the electricity company for the price predictions and autonomously decide when it's best to run.

A large pilot project for energy demand management is being set up in the town Neustadt an der Ruhr. The pilot is conducted with support of the North-Rhine Westfalian and Federal Ministries for Economy and Energy. Also involved is the local electricity supplier Ruhrstrom AG, which has to disseminate the prices per timeslot and collect all the data about how much electricity was used by whom in which time slot. Neustadt is a suitable location for the pilot because all homes now have smart electricity meters that have the ability to provide Ruhrstrom with the data.

## Question 1 (Database Schema) (30 points)

Figure 1 shows a class diagram for part of the administration of Ruhrstrom. A description is given below. *(N.B.: colours have no meaning in a class diagram, the different colours indicate for which parts you are asked to give a database schema)*

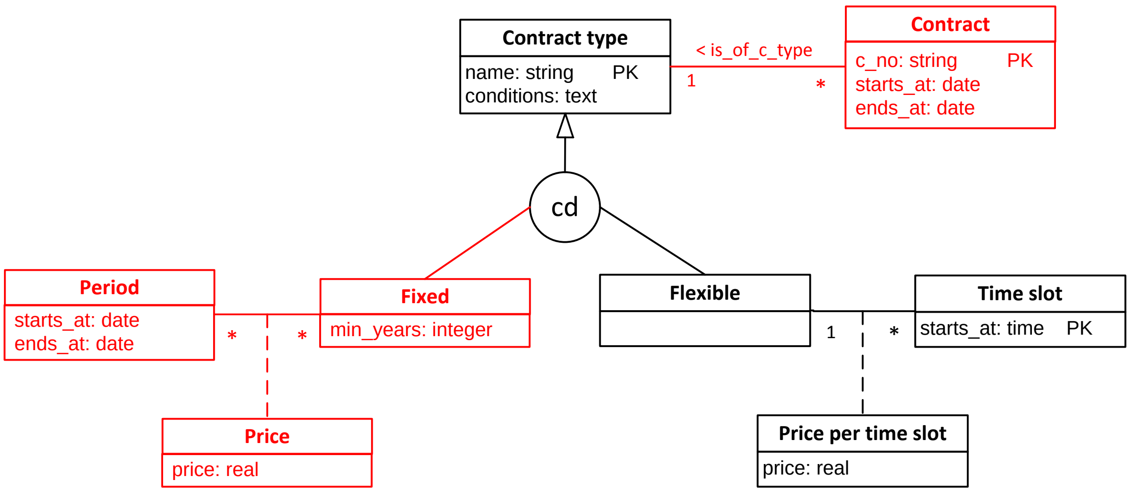



*Figure 1: Partial class diagram for the administration of Ruhrstrom*

A standard contract is for one year, but if you sign for more years, you will get a price reduction. There are specials deals to attract new customers, in which you will get a reduction for the first year.
Also, the price of electricity increases from time to time with a percentage that is determined by the Federal Ministry of Economy and Energy.
A fixed contract (more precisely, a fixed contract type) has a name, various conditions attached (*which we model as a single text attributed for the sake of simplicity*), and a minimum number of years. The price per kWh is constant for a certain period, typically 1 January – 31 December. Different contract types have different prices. Please note that information about previous periods is still available (typically, a yearly electricity bill covers parts of two periods which differ in price).
(*N.B.: Contracts with different prices for day-time and night-time/weekend electricity consumption, which are widespread in the Netherlands, do not exist in Germany.*)
New for this pilot is the flexible contract. Prices change per time slot of 30 minutes. The price is the same for everybody, there is a single flexible contract type.
Individual contracts have a contract number, a start date and end date, and are of a particular type.

**1a)** Define a database schema for the <u>black</u> parts of diagram in Figure 1. (See Appendix B for SQL syntax). Please include a table for "Contract type" as well as a table for "Flexible".
(*You can ignore the red parts in order to limit the amount of writing*).

**1b)** If you consider only the generalization (superclass "Contract type" with subclasses "Fixed" and "Flexible"), would there have been other way(s) to model this in a database schema?
For each possible alternative,

- Describe in half a line what the alternative would be,
- Explain in one or two lines which problem this creates for the database schema and how the problem could be addressed.

## Question 2 (Class Diagram) (35 points)

Complete the class diagram for the administration of Ruhrstrom according to the information given below. (See Appendix A for an overview of class diagram notations.)

*From Figure 1 you need to copy (only) the classes "Contract" and "Time slot" (the other classes will not be involved in further associations/generalizations and are not needed in your class diagram).*

The following information is relevant for the class diagram.

- A contract is for a single property, e.g., a house or an apartment. Buildings that are used for other purposes than living are also properties. (We do not make a distinction between different kinds of properties). Over time, a property will have different contracts. For example, anyone who will have a flexible contract for the pilot study had a fixed contract before.
- For each property the following data are known: registry number of the Land Registry (*kadaster*), street name, house number (which may include additional letters), postal code (in Germany this is a 5-digit number). Furthermore, for each property it is known for each past and present customers (*see next bullet*) for which period this was a customer (start date and end date).
- A customer is the person (or institution) whose responsibility it is that the electricity bill gets paid. This is typically the tenant for a rented property, the owner for a privately owned property. In some cases the customer is not a person, but a company or other institution. The administration make a distinction between private customers and corporate customers.
    - For each customer the following data are known: customer number, name and IBAN (bank account number).
    - For corporate customers, furthermore, the following data are known: HK[1] registration number and address of the customer (not necessarily the property itself).

  *(N.B.: IBAN is "snapshot" information. If a customer gets a new IBAN, the old IBAN need not be remembered. The same holds for the address of a corporate customer.)*
- Each property always has an electricity meter. Occasionally a meter gets replaced, when it is broken, or when and old meter is exchanged for newer type. (Thus, over time, a property can have multiple meters.) A meter that has been installed at one property is never reused at another property.
- In Neustadt different kinds of digital meters have been installed. Some properties have had them already for longer time, other properties only recently were equipped with the newest model. Meters have a brand name (the company who produced them) and a type name (the specific model). Each meter has a serial number (assigned by the producer) and a bar code (assigned by the electricity company who installs the meter).
- Each meter has two counters: a positive counter for consumed electricity and a negative counter for produced electricity[2]. For each meter for each time slot, the readings of the positive and negative counter at the beginning of the time slot are stored. (the time indicated by the attribute "starts_at" of the class "Time slot" in Figure 1).
  How much energy is consumed/produced during the time slot can be computed by combining these data with the counter readings at the start of the next time slot.

---

[1] *Handelskammer*, i.e., Chamber of Commerce

[2] To be very precise: meters register positive/negative differences between consumption and production. When 3 kW is consumed and 1 kW is produced simultaneously, the meter registers a consumption of 2 kW.

## Question 3 (35 points)

### 3a) (Functional dependencies) (15 points)

For the administration of energy demand management the relation *R*(*Co,Cu,D,I,M,Pp,Pt,R,T*) is defined, with the following attributes:

| | |
|---|---|
| *Co*: **Co**ntract | *Pp*: **Pro**p**erty |
| *Cu*: **Cu**stomer | *Pt*: **P**rice of electricity for a **t**ime slot |
| *D*: **D**ate | *R*: **R**eading of the (positive) counter of a meter at the beginning of a time slot |
| *I*: **I**BAN of the customer | *T*: **T**ime slot |
| *M*: Electricity **M**eter | |

Furthermore, the following facts are given:

1. A contract is for one customer and for one property
2. Every customer has a unique IBAN
3. For every property it is known who the customer is at any specific day
4. A meter is associated with one property. However, in the course of time a property can have multiple meters, e.g. when a broken meter is replaced.
5. For each time slot the company sets the price in €/kWh at least 24 hours in advance
6. For each time slot for each meter the reading (at the beginning) of the (positive) counter is known

For each of the following – potential – FDs a)–h) and MVDs i)–j), indicate whether they hold ("yes") or not ("no"). Please motivate your answer, possibly referring to the statements 1–7 above.

a) $M\ T \longrightarrow R$

b) $R\ T \longrightarrow M$

c) $Pp\ D \longrightarrow Cu$

d) $Co\ D \longrightarrow Pp$

e) $Cu\ D \longrightarrow Pp$

f) $M \longrightarrow Co$

g) $Co \longrightarrow Pt$

h) $Co \longrightarrow I$

i) $Co \twoheadrightarrow I$

j) $Co \twoheadrightarrow Cu\ Pp$

### 3b) (Normal forms) (20 points)

Consider the relational schema *R*(*A,B,C,D,E,F,G*) with functional dependencies $\mathcal{F}$, defined by

$$\mathcal{F} = \{ C \rightarrow F,\ D \rightarrow G,\ F \rightarrow ABEG \}.$$

**1)** Which functional dependencies violate the BCNF condition? Why?
(Hint: compute $\mathcal{F}^+$ and the candidate keys first)

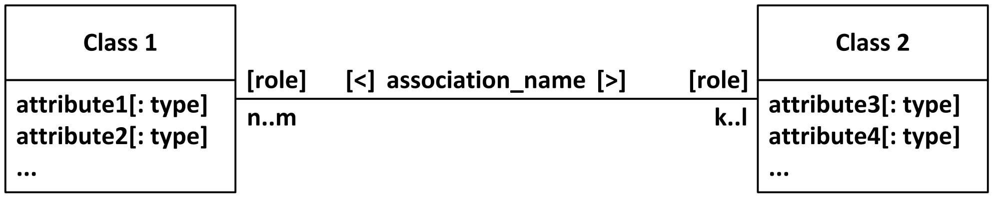
**2)** Apply the algorithm in appendix C to decompose *R* into a set of relational schemas that are all in BCNF. For each decomposition step, please give the resulting schemas with their sets of functional dependencies and their candidate keys.

**3)** Which (if any) of the functional dependencies in $\mathcal{F}$ were lost in the decomposition?

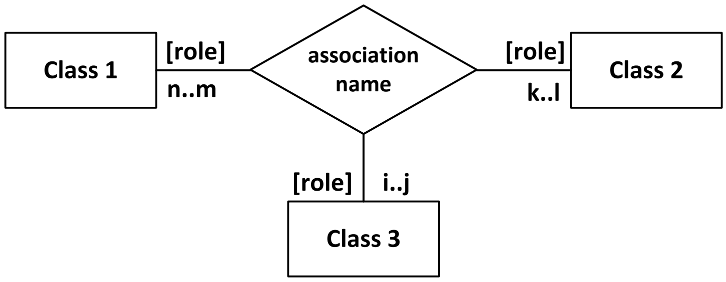
## Appendix A: Notations for class diagrams

*meta-notation:*
[ ... ]   Optional element

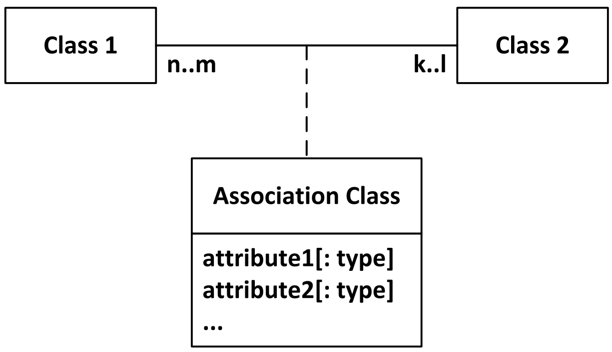
### Class and Association



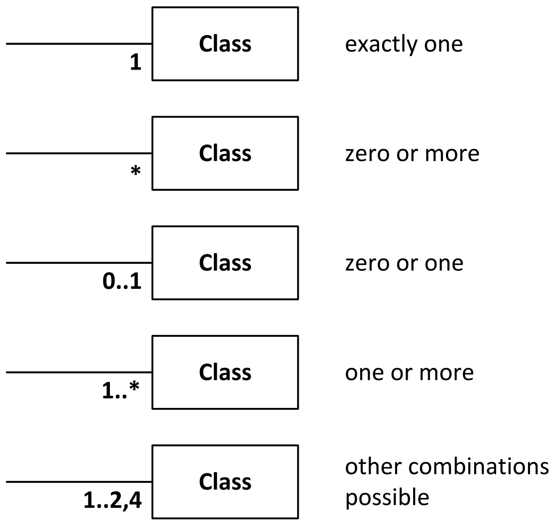

### Ternary (or *n*-ary) association



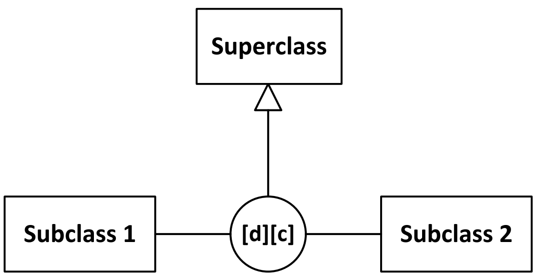

### Association class



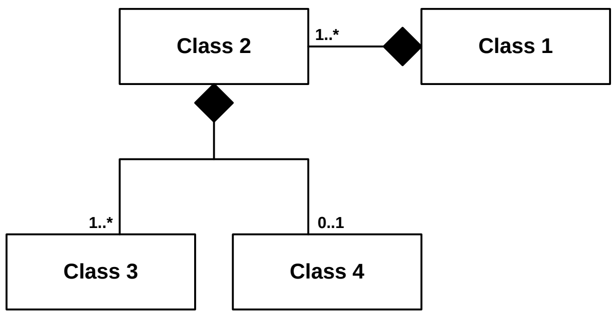

### Multiplicity




### Generalization




### Composition



## Appendix B: Informal syntax for database schema

Informal syntax ('|' for choice and '[ ]' for optional):

```
Table(column [NOT NULL] [UNIQUE] [PK]
     [, column … ]
     [, PK (column, … )]
     [, FK (column, …) REF table[(column, …)]
         [, FK …] ]
     [, CHECK(condition)]
);
```

Examples of condition:

```
  column = value [(OR|AND) [NOT] column <> value ] |
  column IS [NOT] NULL |
  column [NOT] IN (value, ...) |
  …
```

## Appendix C: Losless BCNF decomposition algorithm

Definition of BCNF:
A relational schema is in BCNF if for every nontrivial functional dependency the left-hand side is a superkey.

Decomposition algorithm:
Let $R$ be a relational schema with a set of functional dependencies $\mathfrak{F}$.
Let $X \rightarrow Y$ be a functional dependency in $\mathfrak{F}$ which violates the BCNF constraint.

- Decompose R into
  - $R_1(X^+)$
  - $R_2(Z)$ with $Z = \{X\} \cup$ {attributes of $R$ not in $X^+$}.
- For $i = 1,2$:
  - determine $\mathfrak{F}_i$ for $R_i$ by restricting $\mathfrak{F}^+$ to functional dependencies within $R_i$
  - if $R_i$ is not in BCNF, recursively apply the algorithm