Data & Information – Test 2 (1.5 hours)

25 May 2018, 13:45–15:15

Program: Technical Computer Science / Business & IT
Module: Data & Information (201700279)
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: Theatre Tickets

Question 1, 2, and 3a make use of the following case description.

Subsidies for cultural activities have been repeatedly decreased over the last few years. As a consequence, the cultural sector in the region Polderland requires major restructuring. This does not only involve theatre companies and orchestras, the theatres themselves are also having hard times. In order to cut costs, the different theatres in the region have decided to merge their organizations. As a result, there is now a large "Polderland Theatre", of which most of the previously independent theatres have become branches. Their administrations will be merged. There is a need for a new computer system that will support selling tickets for the different theatres.
There are a few smaller theatres that want to remain independent, i.e. not to be merged into the big Polderland Theatre organzation, but they will also use the new system for selling tickets.
Customers will be able to buy tickets for all theatres in Polderland at the box office of each theatre. It is also possible to buy tickets through the web site or to order them in advance by means of a paper order form. If you order tickets in advance, the costs will be debited to your bank account after the start of the cultural season, in the first week of September.

You are asked to help design parts of the new system for selling theatre tickets

Question 1 (Database Schema) (30 points)

Figure 1 shows a class diagram for part of the system, dealing with on clients and sponsors.

Most customers are citizens of Polderland or neighbouring regions, who go to see theatre productions.
If one person buys tickets for the family then the person buying the tickets is registered as customer for all the tickets s/he bought. Of each customer, the theatre wants to have at least one telephone number, so that the customer can be texted when something gets cancelled at short notice (more telephone numbers are optional). For example last year it happened that a choir from the UK got caught up in air traffic problems and had to be cancelled only hours before.
Polderland Theatre is sponsored by various local companies. A sponsoring company pays a yearly amount. In return, a sponsor gets free advertisements in the programme booklets and on the web site, and a number of free tickets each year. Sponsors can choose for which productions they want free tickets. For each sponsor, a contact person is known, for each contact person one or more telephone numbers.
Each ticket that has been issued has a ticket number. Some tickets are sold with a reduction (possibly a different percentage for different age groups, a reduction of 100 % for the free tickets of the sponsors). It is also registered whether the ticket has been paid or not.

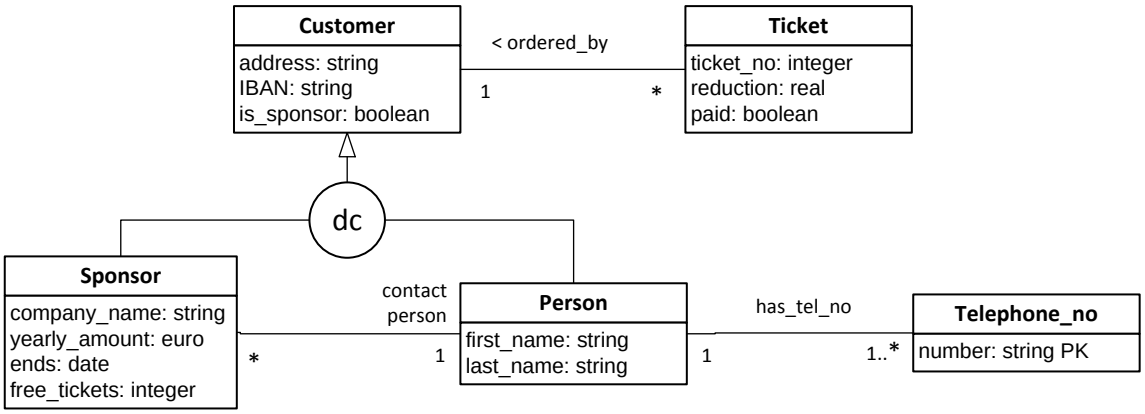


Figure 1: Partial class diagram for the theatre ticket administration

1a) Define a database schema which represents all the information in the class diagram in Figure 1. (See Appendix B for SQL syntax).

1b) If you consider only the generalization (superclass "Customer" with subclasses "Sponsor" and "Person") without the other classes and assocations, 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, given the other elements of the class diagram

Question 2 (Class Diagram) (35 points)

Extend the class diagram of Figure 1 with the following information.
Please Copy the classes "Customer" and "Ticket" into your diagram (no need to include "Sponsor", "Person" and "Telephone_no").

Polderland theatre has different theatre halls. Each hall has a name and a location. Most halls are now owned by the Polderland Theatre organization. The new system will also sells tickets for productions that are performed in other theatres in the region. In that case, the name and e-mail address of the owner are also known.
Seats in a theatre hall are characterized by a row and a seat number. (The row name is actually a string: e.g., row "6" is on the ground floor, row "balcony 6" is on the balcony.)

Polderland theatre shows different productions by different companies. A production can be a theater play, concert, cabaret, etc.*(but you don't have to model different types of productions).*
Of a company, the name, address, e-mail adress and telephone number are known.
A theatre company can run several productions. A production has a name; an entrance fee; a duration (in minutes); a description.
Some productions have a higher entrance fee than others, but there is a fixed entrance fee for each production. Polderland Theatre does not differentiate entrance fees according to seating areas, which many theatres do. *(Some tickets are sold at a reduced fee, see Figure 1, but that is to be interpreted as a reduction on the fixed entrance fee for the production.)*
A production usually is performed multiple times, possibly in different halls across the region. Occasionally a production is performed two times a day, so for the ticket sales we need to know the start time as well as the date.

Hint: in order to cover all seats in all halls for all productions, it helps to introduce the notion of a *sellable seat*. For example: row 6 seat 17 for the concert of the Polderland Symphony Orchestra in the Beethoven Hall on Friday 25 May at 8 PM. This seat for tonight might have been sold or might still be available.

Question 3 (35 points)

3a) (Functional dependencies) (15 points)

For the theatre administration the relation $R(H,R,S,C,P,V,N,T)$ is defined, with the following attributes:

H: Hall	*P*: Production
R: Row	*V*: Visitor (customer)
S: Seat	*N*: telephone Number
C: Company	*T*: Ticket

Furthermore, the following facts are given:

1. A row is in one particular hall
2. A seat is in one particular row
3. All visitors have different telephone numbers
4. A visitor may have multiple telephone numbers
5. A ticket is for one visitor
6. A ticket is for one seat
7. A ticket is for one production
8. A production is run by one company

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) $T \longrightarrow H$

b) $T \longrightarrow N$

c) $N \longrightarrow T$

d) $HT \longrightarrow C$

e) $SP \longrightarrow HC$

f) $V \longrightarrow PC$

g) $P \longrightarrow VC$

h) $C \longrightarrow PV$

i) $T \twoheadrightarrow CPV$

j) $T \twoheadrightarrow HCP$

3b) (Normal forms) (20 points)

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

$$\mathcal{F} = \{ ACE \longrightarrow D,\ C \longrightarrow B,\ D \longrightarrow C \}.$$

1) Compute $\mathcal{F}^+$ and determine the candidate keys.
Which functional dependencies violate the BCNF condition?

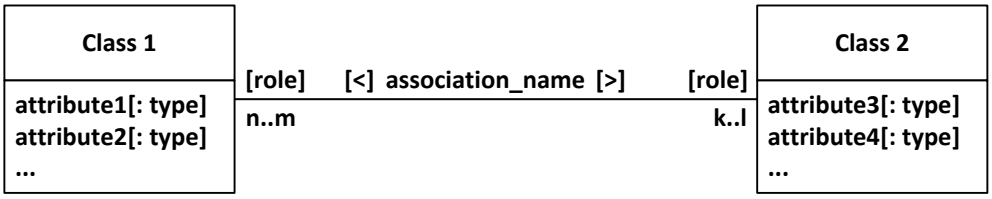
2) Apply the algorithm in appendix C to decompose *R* into a set of relational schemas that are all in BCNF. For each decomposition step, please give the resulting schemas with their sets of functional dependencies and their candidate keys.

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

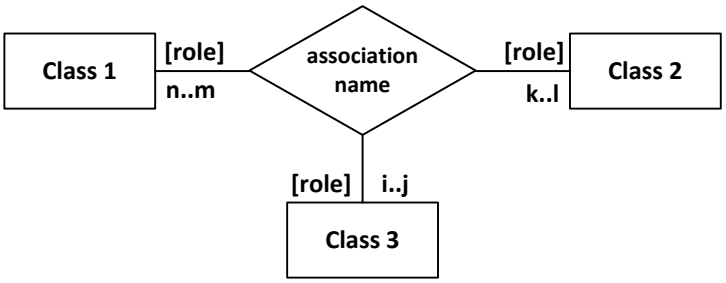
Appendix A: Notations for class diagrams

meta-notation:
[...] Optional element

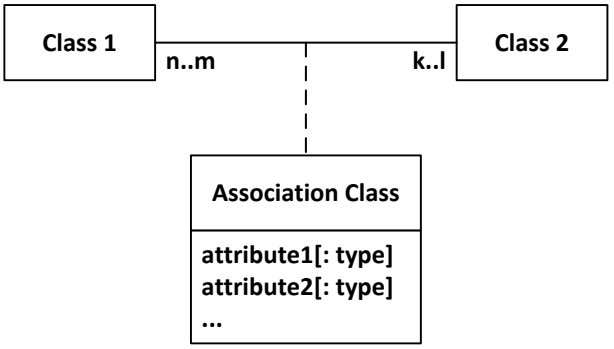
Class and Association



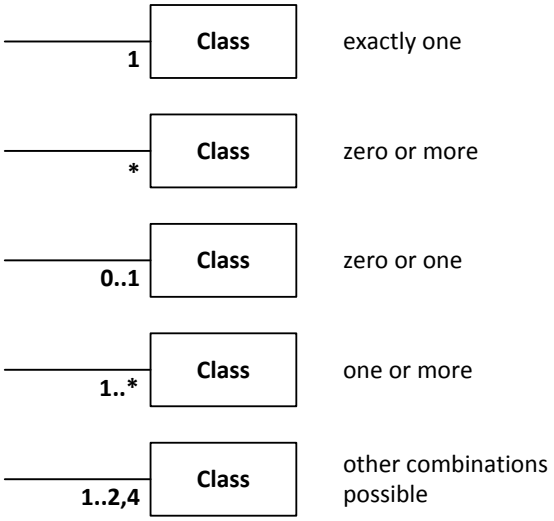
Ternary (or *n*-ary) association



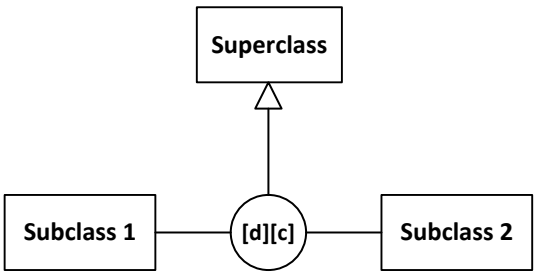
Association class



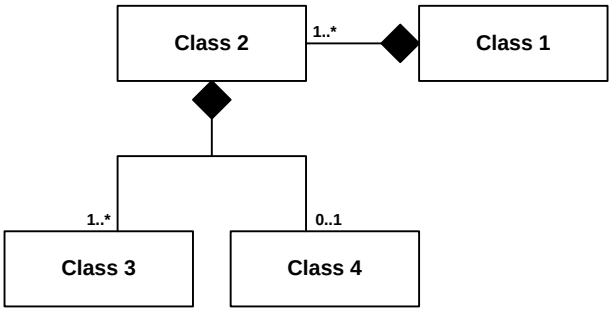
Multiplicity



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 $\mathcal{F}$.
Let $X \rightarrow Y$ be a functional dependency in $\mathcal{F}$ which violates the BCNF constraint.

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