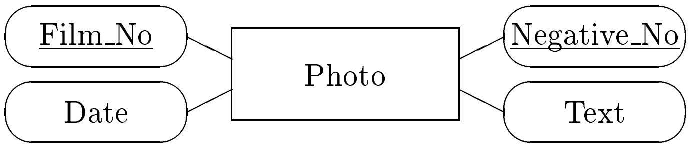
Dr. Stefan Brass
School of Information Sciences
University of Pittsburgh

July 19, 2001

INFSCI 2711 "Database Analysis and Design" — Example III for Midterm Exam: Solution —

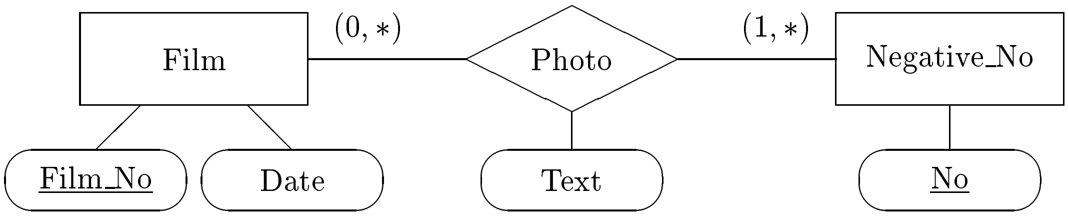
Exercise 1 (ER Model) 12 Points

a) Collegue A would put all information into one entity:



☐ Correct
☒ Wrong, Explanation: Violates 2NF, development date depends only on Film_No.

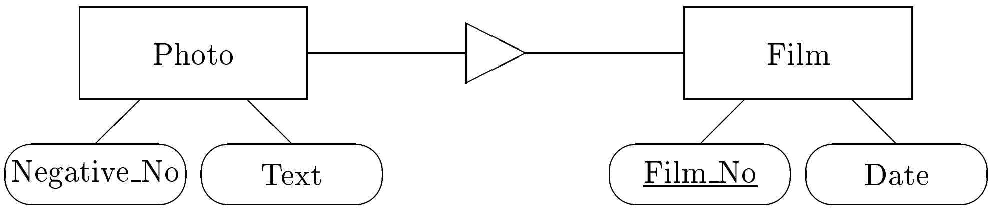
b) Collegue B views the photo as relationship between the film and the number of the negative:



☒ Correct
☐ Wrong, Explanation: ______________________________

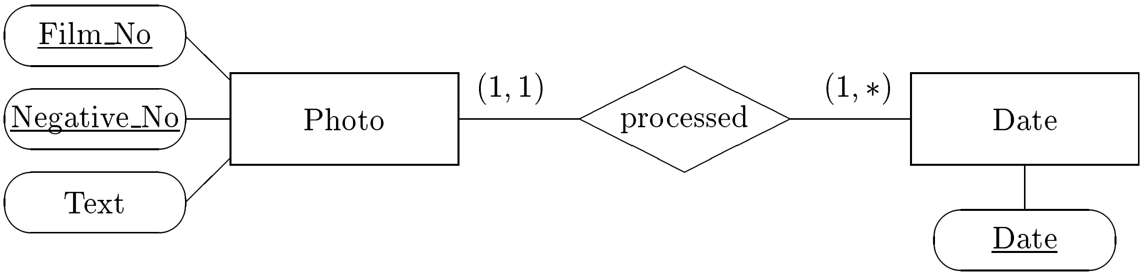
Style remark: "Negative_No" is here a lookup table if one wants to constrain the possible negative numbers. If really only numbers from 1 to 36 are allowed, this is not a good entity type, it is more data. However, if numbers like "00A", "24A" are allowed, a lookup table might be the right thing to do. So this solution is not as strange as it seems first, although "Photo" probably should be an entity (a weak entity). Modelling a thing that you can touch as a relationship seems not very natural. But formally it is equivalent to the given schema.

c) Collegue C thinks that a specialization is needed in this application:



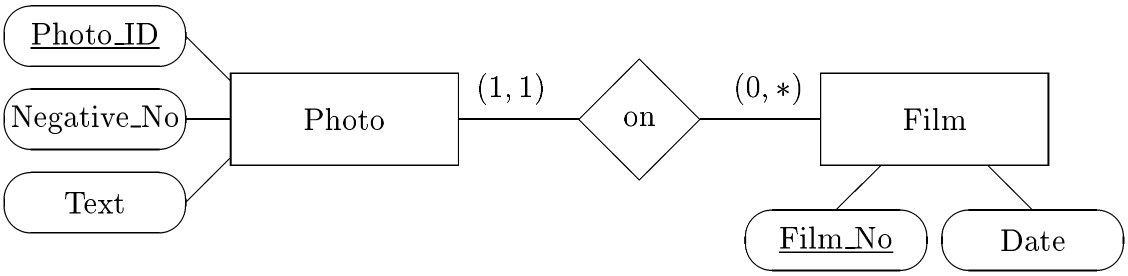
☐ Correct
☒ Wrong, Explanation: In this schema, one can store only one photo per film.

d) Collegue D has a similar solution as collegue A, but he says that in order to get third normal form, it is necessary to put the date into an entity of its own:



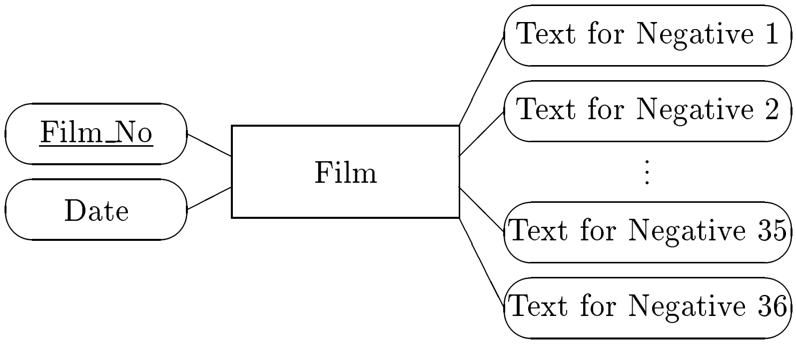
☐ Correct
☒ Wrong, Explanation: Still violates 2NF: The table “Photos” will have foreign key “Date” that depends only on “Film_No”.

e) Collegue E does not like composed keys and suggest to introduced a sequential number for photos:



☐ Correct
☒ Wrong, Explanation: One can store several photos on the same film with the same negative number. Needs an alternative key.

f) Finally, collegue F would create only a single entity for each film, with an attribute for the text of each possible negative number.

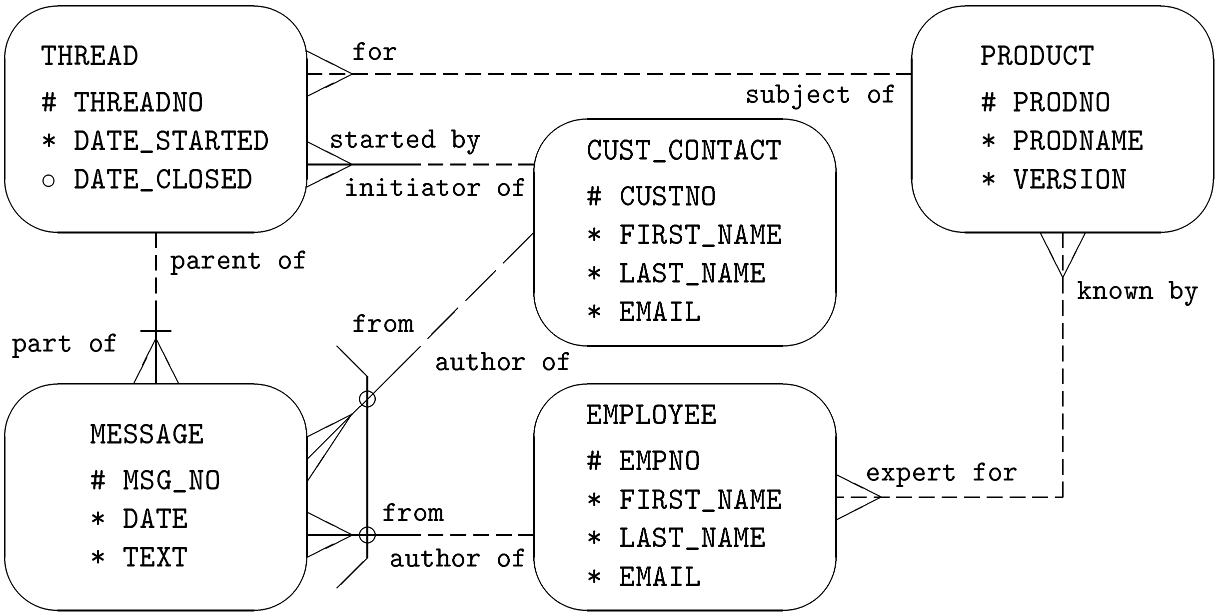


☐ Correct

☒ Wrong, Explanation: At least, the text columns must be optional.
It is not required to store a text about every photo on a film.

Style remark: This is a bad solution. Consider how difficult it would be to write an SQL query that searches for a text saying “Lisa’s first Christmas”. One would need to write 36 conditions combined by “OR”.

Exercise 2 (Reverse Engineering, Oracle Designer) 12 Points

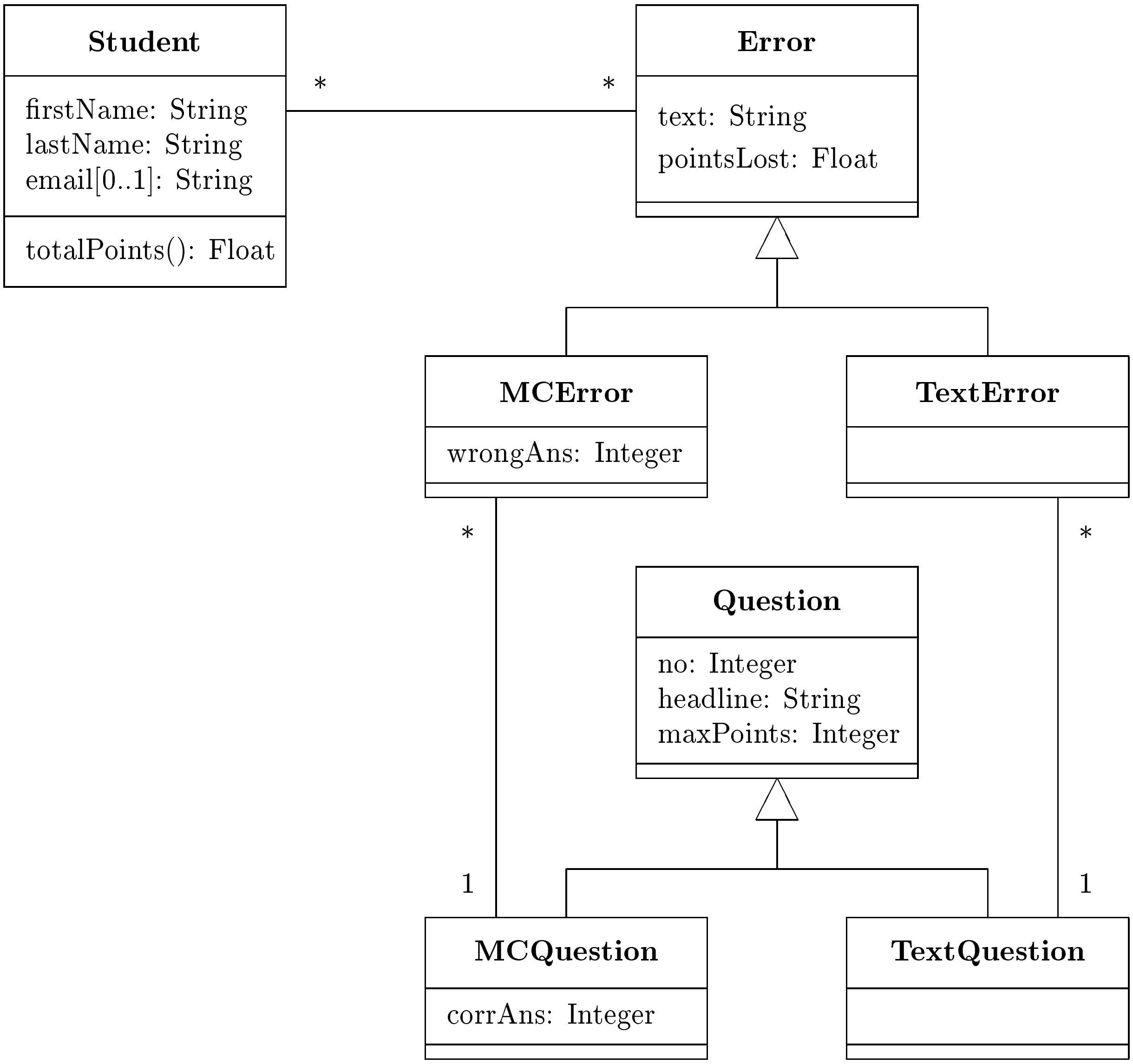


It also makes sense to use a supertype for `CUST_CONTACT` and `EMPLOYEE` (`SUER` or `PARTICIPANT`). However, normally in such cases the key attribute of both relations would have the same name. But there is always some freedom in reverse engineering.

By the way, one probably should change `MSG_NO` to `MSGNO`, since `EMPNO`, `CUSTNO`, and `PRODNO` have no underscore.

The relationship role names are not present in the relational model. They have to be invented in the reverse engineering step, although foreign key names sometimes might help.

Exercise 3 (UML Class Diagram) 12 Points



Exercise 4 (BCNF) 4 Points

The data dictionary of a DBMS contains the following relation:

`KEYS(ID, TAB_NAME, COL_NO, COL_NAME)`

`ID` identifies the key, `TAB_NAME` is the table for which the key was declared. `KEYS` contains one row for each attribute `COL_NAME` of a key (a key can consist of several attributes). `COL_NO` is the number (position) of the attribute within the key. The following functional dependencies hold for this table:

- `ID` → `TAB_NAME`
- `ID, COL_NO` → `COL_NAME`
- `ID, COL_NAME` → `COL_NO`

a) What are all (minimal) keys of this relation? Please check all minimal keys. Exactly two answers are correct.

- ☐ `ID`
- ☐ `ID, TAB_NAME`
- ☒ `ID, COL_NO`
- ☒ `ID, COL_NAME`
- ☐ `ID, COL_NO, COL_NAME`
- ☐ `TAB_NAME, COL_NAME`

b) Is the relation in BCNF? (If several FDs should violate BCNF, it suffices to the check one correct answer.)

- ☐ Yes.
- ☒ No. The FD "`ID` → `TAB_NAME`" violates BCNF.
- ☐ No. The FD "`ID, COL_NO` → `COL_NAME`" violates BCNF.
- ☐ No. The FD "`ID, COL_NAME` → `COL_NO`" verletzt BCNF.

c) Let us assume that some data dictionary relation violates BCNF (not necessarily this one). However, it is a view, and not a stored relation. Would the BCNF violation then be bad? Should the DBMS vendor modify the view definition such that BCNF will not be violated?

- ☐ Yes. Normal forms are important also for views.
- ☒ No. This is no problem.