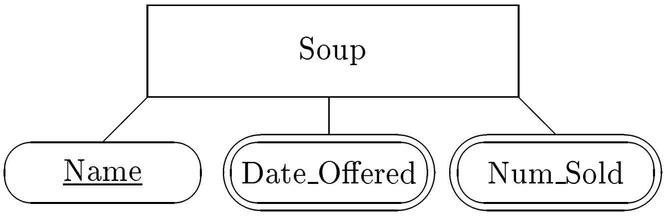
Dr. Stefan Brass
School of Information Sciences
University of Pittsburgh

July 12, 2001

INFSCI 2711 "Database Analysis and Design" — Solution to Midterm Example I —

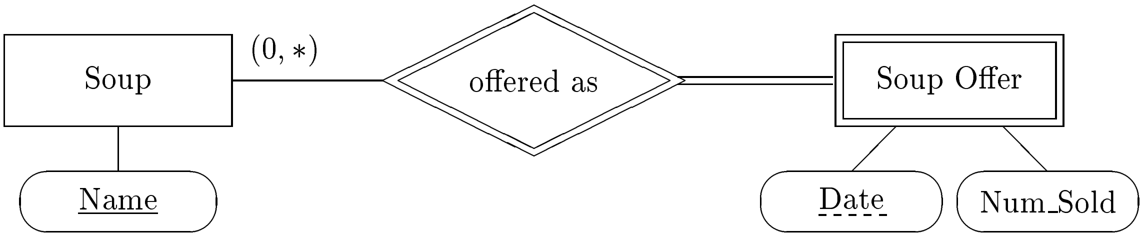
Exercise 1 (ER Model) 14 Points

a) This solution has an entity for every kind of soup and uses multivalued attributes to store when a soup was offered and how often it was sold.



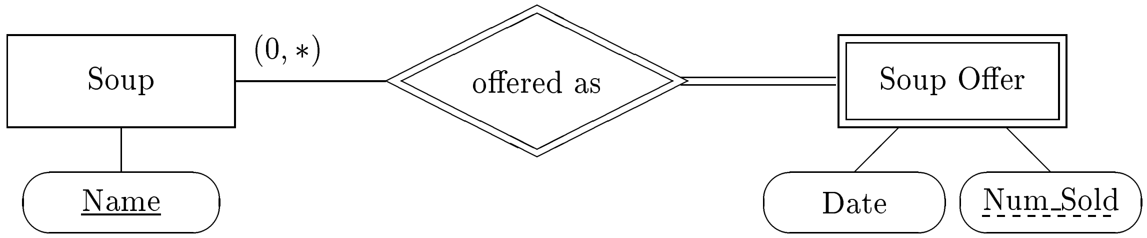
☒ Wrong, Reason: The connection between the date and the number of soups sold is missing. You cannot answer the question how often the soup was sold on a particular date. You only have two unrelated sets of dates when the soup was offered and sales numbers.

b) This solution has an entity for every kind of soup uses a weak entity to model the sales:



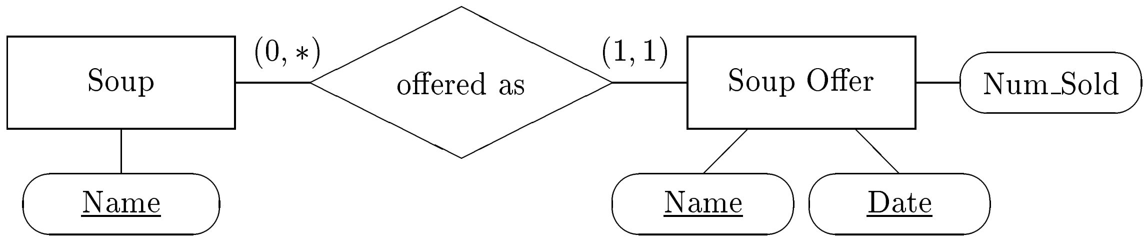
☒ Wrong, Reason: This is nearly correct (it allows to store the required information), but unfortunately it allows more than one soup to be offered on the same day (which is excluded by the requirements). So the schema is too general.

c) This is very similar to b), but the key for the weak entity is different:



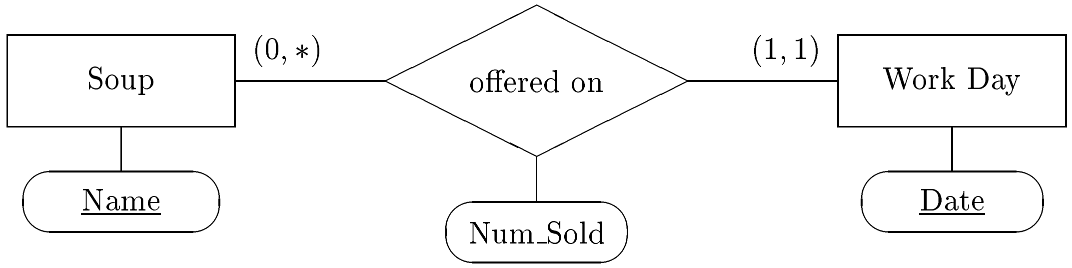
☒ Wrong, Reason: This would not allow to store the situation where the same soup is sold the same number of times on different days.

d) This is again similar, but it does not use a weak entity:



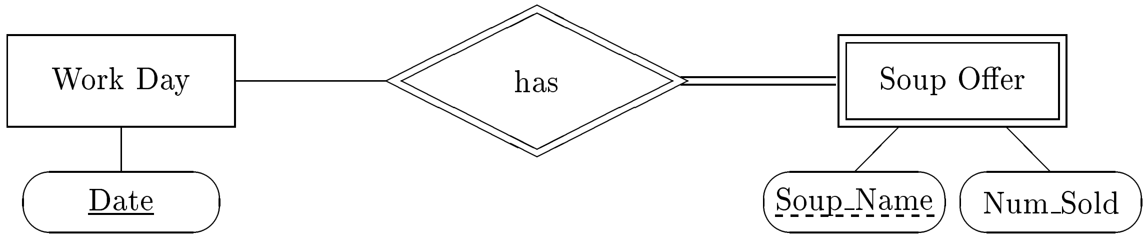
☒ Wrong, Reason: This allows to store all real world states and to answer the given queries, however it has two flaws: First, it also allows to store states which are meaningless. What happens if a "Soup Offer" entity is connected to a "Soup" entity with a different name? A constraint is needed here (but this constraint could be much better expressed with a weak entity). Second, it allows to store more than one soup per day (as above).

e) This solution uses an entity for work days (i.e. when the canteen/cafeteria was open) and a relationship attribute for the number of copies sold:



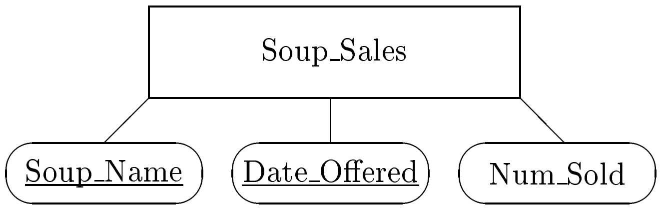
☒ Correct

f) This again uses a weak entity, but this time the date is the master entity:



☒ Wrong, Reason: Again, more than one soup per day is possible. Otherwise it is legal. It allows also days without soup offers, but this was not explicitly excluded.

g) Finally, here is a solution with only one entity and without multivalued attributes:



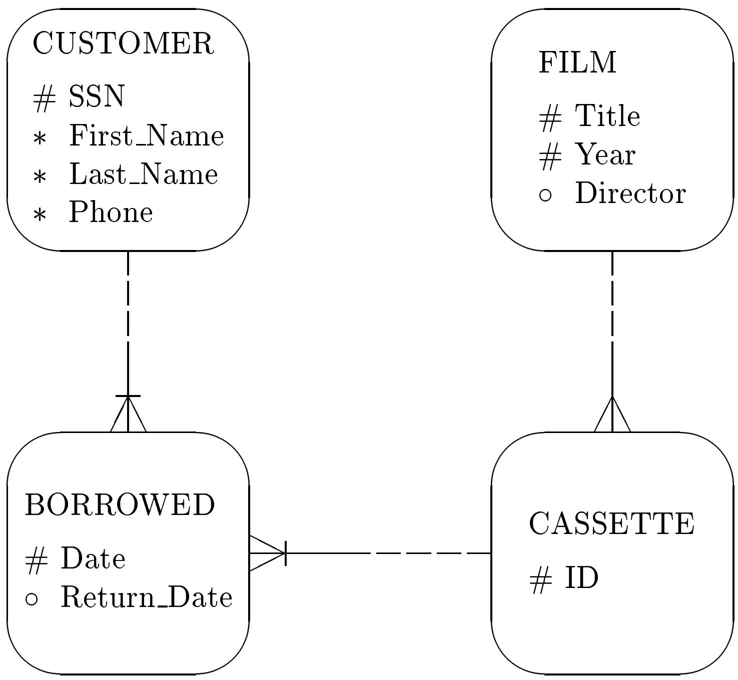
☒ Wrong, Reason: Again, more than one soup per day is possible. In addition, you cannot represent soups which were never offered.

You may have noticed that many of these solutions result more or less in the same table when translated into the relational model. This is also a possibility if you cannot directly decide the equivalence of two ER-schemas.

Exercise 2 (ER Design, Oracle Designer Notation) 12 Points

Please develop an Entity-Relationship diagram in Oracle Designer notation for a video rental service:

- About customers, we store first name, last name, phone number and social security number. All attributes are required.
- Some films which are often requested exist in multiple copies. I.e. there can be several video cassettes which contain the same film. Every video cassette has a unique identifying number.
- We want to store which customer borrowed which cassette on which day, and also when he/she brought the cassette back. The return date is an optional attribute: It is null as long as the customer still has the cassette. We want to store historical information, i.e. also previous lending events.
- About films, we want to store the title and the year, which together uniquely identify the film. In addition, we want to store the director (optional attribute).



The specification leaves open whether customers must have rented a cassette and whether only films could be entered which also exist as a video cassette. I personally believe that this is too strong, but it would not be wrong to use mandatory participation here. Non-transferability of the relationships also would make sense.

Exercise 3 (Oracle Designer) 8 Points

a) Oracle Designer supports weak entities.

- ☒ True.
- ☐ False.

b) There can be several diagrams which show only a subset of the entities, but there must be one diagram which shows the global schema (all entities and relationships).

- ☐ True.
- ☒ False.

c) All cardinalities which Oracle Designer supports are automatically translated into constraints for the relational schema.

- ☐ True.
- ☒ False.

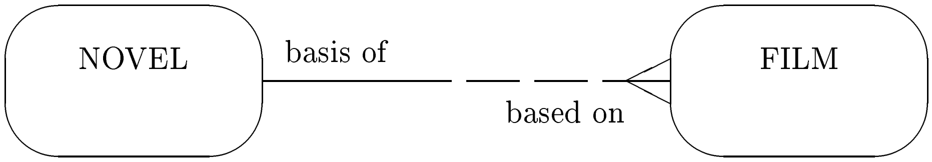
d) Oracle Designer can translate ER-schemas automatically into relational schemas, however you can (and often should) change the relational schema manually afterwards.

- ☒ True
- ☐ False.

e) Oracle Designer stores the design data in a relational database (which must be an Oracle database).

- ☒ True.
- ☐ False.

f) Consider this relationship:



Which of the two entities has mandatory participation in the relationship?

- ☒ NOVEL
- ☐ FILM
- ☐ None
- ☐ Both

g) [2 Points] There are patterns for automatically generated English sentences which describe the cardinalities. Which two sentences would be generated for the relationship in f)?

1. Each NOVEL must be basis of one or more FILMS.

2. Each FILM may be based on one and only one NOVEL.

Exercise 4 (FDs, BCNF) 4+1+2+4+2+0=13 Points

Consider the relation:

`FAMILY_DATA(CHILD, CHILD_BORN, MOTHER, MOTHER_BORN, FATHER, FATHER_BORN, MARRIAGE_YEAR)`

- We assume that every person is uniquely identified by his/her name which is entered into the columns `CHILD`, `MOTHER`, `FATHER`.
- `CHILD_BORN`, `MOTHER_BORN`, `FATHER_BORN` are the year of birth of the corresponding person.
- We assume that each two persons can marry only once. (So we don't consider the case that they divorce and later marry again. This can be seen as a continuation of the first marriage.)
- However, the same person can marry again, e.g. if the spouse died. But let us assume that two marriages in one year are impossible.
- Of course, the same husband and wife can have more than one child, also more than one per year (e.g. twins).

a) [4 Points] Please list all functional dependencies which hold for this relation (except trivial and implied ones). There is space for 8 FDs, but this does not mean that there have to be that many. Listing trivial FDs counts as an error. Also, listing too many implied dependencies might count as an error, but I will be generous with this (you need one implied dependency for b).

1. `CHILD` → `CHILD_BORN, MOTHER, FATHER`
2. `MOTHER` → `MOTHER_BORN`
3. `FATHER` → `FATHER_BORN`
4. `MOTHER, FATHER` → `MARRIAGE_YEAR`
5. `MOTHER, MARRIAGE_YEAR` → `FATHER`
6. `FATHER, MARRIAGE_YEAR` → `MOTHER`

`MOTHER, CHILD_BORN` → `FATHER` and `FATHER, CHILD_BORN` → `MOTHER` are violated only in very sad circumstances, but we will not consider them further.

b) [1 Point] Name one FD which is implied by the ones you have listed:

`CHILD` → `MOTHER_BORN, FATHER_BORN`

c) [2 Points] List all minimal keys of the relation. There is space for three, but there might be less (i.e. only one or two):

1. `CHILD`

d) [4 Points] Is the relation in BCNF?

- ☐ Yes.
- ☒ No.

If you checked “No” name one FD which violates BCNF (there might be multiple violations, but one example is enough):

`MOTHER` → `MOTHER_BORN`

Also give a decomposition of the relation into relations which are in BCNF. Note that if there were multiple violations, it might be necessary to split more than once:

1. `BIRTH(CHILD, CHILD_BORN, MOTHER, FATHER)`
2. `MOTHER(MOTHER, MOTHER_BORN)`
3. `FATHER(FATHER, FATHER_BORN)`
4. `MARRIAGE(MOTHER, FATHER, MARRIAGE_YEAR)`

e) [2 Points] Independent of whether the relation is in BCNF or not and the result of the decomposition, do you feel that there might be a better set of relations to store the information in the given relation?

1. `PERSON(NAME, BIRTH_YEAR, SEX, MOTHER→PERSON, FATHER→PERSON)`
   `MOTHER` and `FATHER` can be null (or else you could never break off the ancestor list). Note that the original design does not store the sex of the child, which is probably a mistake. If this information is not available, `SEX` could also be null.
2. `MARRIAGE(HUSBAND→PERSON, WIFE→PERSON, MARRIAGE_YEAR)`

f) [This question is optional and gives one extra point.] What knowledge about the relation have you used in e) which was not expressed by the FDs?

`CHILD`, `MOTHER`, and `FATHER` are from the same domain (persons).