

# Feedback on cw1

March 2010

Natasha Alechina

# Drug trial scenario

- A research laboratory is running several drug trials on healthy volunteers to check whether drugs have side effects.
- Each drug has a unique name. Each trial involves exactly one drug and several volunteers (who take the drug and report if they had any side effects).

# Drug trial scenario

- For each volunteer in each trial it needs to be recorded whether the volunteer had any side effects, and if yes, what those side effects were (there could be several side effects experienced by the same person, for example headache, dry mouth, and fever).

# Drug trial scenario

- It is important that side effects are described using some standard terminology, so that the laboratory can report what proportion of volunteers had the same side effect. For example, the researchers may tell you that headache should always be recorded as `headache' and not sometimes as `pain in the head' and sometimes as `sore head'. There is however no fixed pre-defined set of possible side effects, because new effects can always be discovered (for example the drug may turn people a bright green colour).

# Drug trial scenario

- For simplicity, assume that each volunteer takes part in at most one drug trial.
- Data stored about volunteers is their National Insurance number, name, age, gender, address and telephone number.

# Entities

- Volunteer
- Drug
- Drug Trial
- Side Effect
- Some students included Research Laboratory which is not a mistake but it isn't doing a lot, same for Researchers.

# Why Side Effect is an Entity

- Some students modelled side effect as an attribute of a volunteer or of trial or of drug
- I don't think it is a good design decision. Side Effects are obviously important and difficult to allocate as a property to any single entity

# Why Side Effect is an Entity

- The only way to make side effects a property of people is to turn them into a list-valued attribute or to fix the number of possible side effects (like SideEffect1, ..., SideEffect4).
- Both are bad choices:
- the first is hard to query (how many people had headache? – need to look inside lists – but in relational model all values are atomic)
- the second is too restrictive (what if some poor person has 4?) and it breeds NULLs.



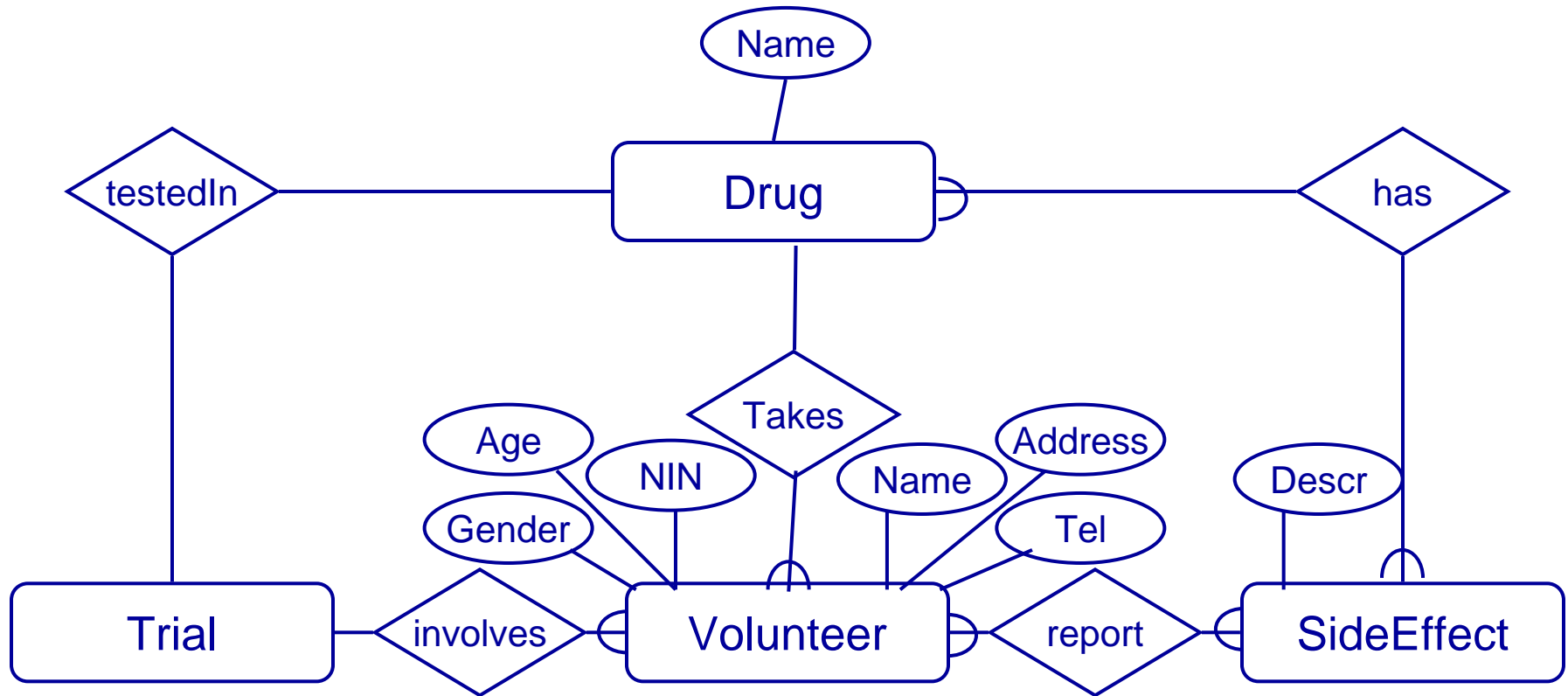
# Attributes

- Volunteer has National Insurance number, name, age, gender, address and telephone number
- Drug has a unique name
- Drug Trial has relationships to Drug and Volunteers but no obvious attributes
- Side Effect has a unique standard description. Common mistake: to make Headache, Fever etc. attributes – they are values!

# Relationships

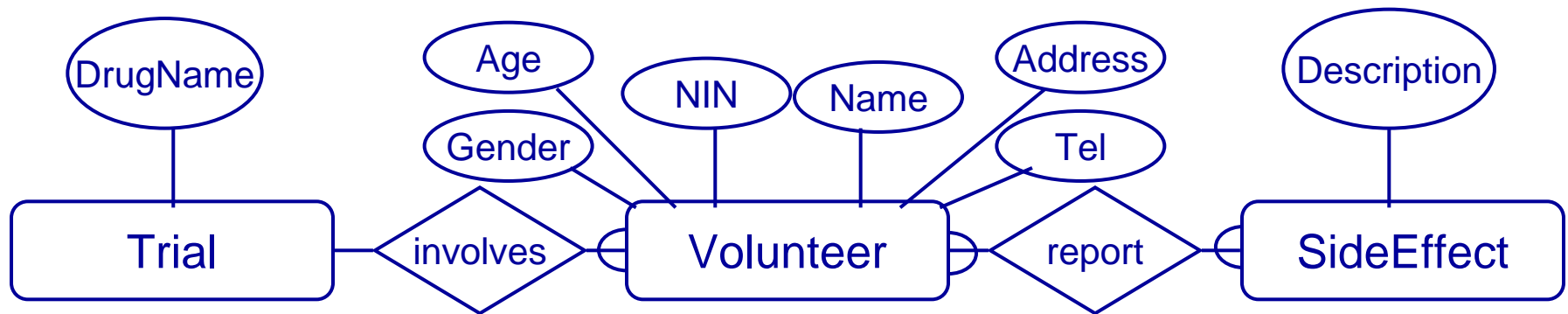
- Drug is being tested in a Trial (1:1)
- Volunteer is involved in a Trial (M:1)
- Volunteer reports Side Effect (M:M)
- We can also introduce relationships between Drug and Side Effect (this is not a mistake) but it is a bit redundant and indirect: in order to discover which side effects a drug has we need to check what people who are testing it reported. If this is recorded, no need to record the same (computable) information again because it may lead to data duplication and mistakes.

# Absolutely full ERD



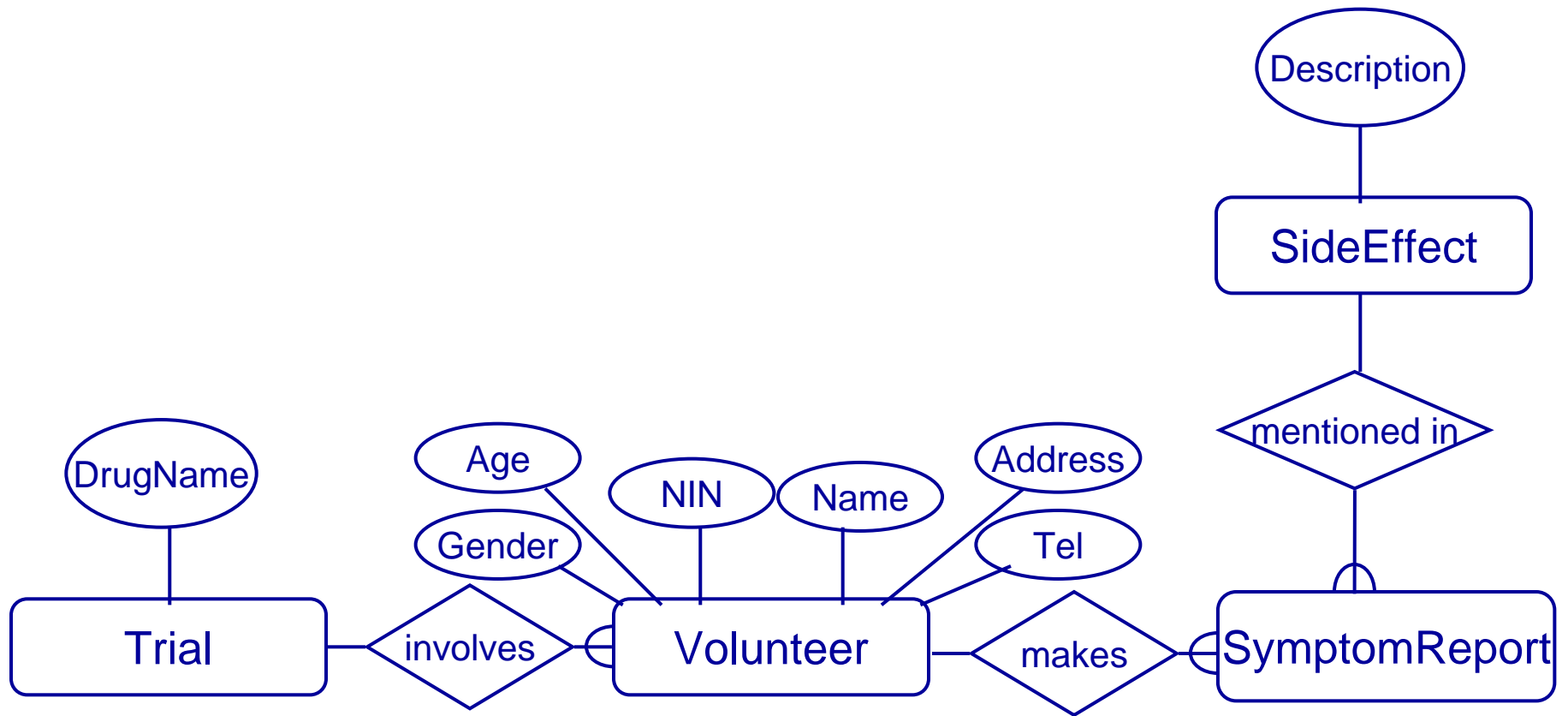
Feedback on cw1

# Simplified ERD



Feedback on cw1

# ERD without M:M



Feedback on cw1

# Common mistake: cardinalities

- The middle entity (Report) is representing reports about side effects experienced by people
- For each person, there may be many such reports, but there is always only one person mentioned in the report. So Volunteer:Report is 1:M
- Same for Side Effect: each report mentions one symptom, but the same symptom may be mentioned in many reports. So SideEffect:Report is 1:M.

# Creating tables

- When ERD has no M:M relationships, the procedure is purely mechanical
- Each entity becomes a table
- Each relationship is represented by a foreign key
- For EntityA:EntityB 1:M, the foreign key is in table B referencing table A

# Why foreign keys are like that

- For example Trial:Volunteer 1:M
- We put foreign key (DrugName) into Volunteer. Then the size of the volunteer table = number of volunteers, each tuple has an additional field for the drug
- If we put foreign key (NI number) into Trial, the size of the trial table = number of drugs times number of volunteers taking that drug.



# Why are foreign keys like this

- Option 1: where the M side of M:1 gets FK

Volunteer

NIN	Name...	Drug
111	John	aspirin
222	Tom	aspirin
333	Mary	aspirin
444	Mark	par
555	Jane	par
666	Sue	aspirin
777	James	par

Drug
aspirin
paracetamol

# Why are foreign keys like this

- Option 2: where the 1 side of M:1 gets FK Volunteer

NIN	Name...	
111	John	
222	Tom	
333	Mary	
444	Mark	
555	Jane	
666	Sue	
777	James	

Drug	NIN
aspirin	111
aspirin	222
aspirin	333
aspirin	666
paracetamol	444
paracetamol	555
paracetamol	666