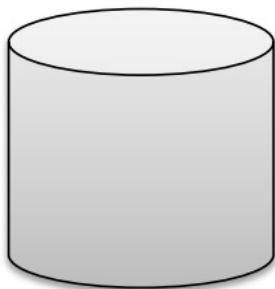


Database Technologies

MS-218

Semester 1

Session 2021-22






Oracle SQL Developer Data Modeler

SQL Developer Data Modeler 21.2

Version 21.2.0.183.1957 - July 14, 2021

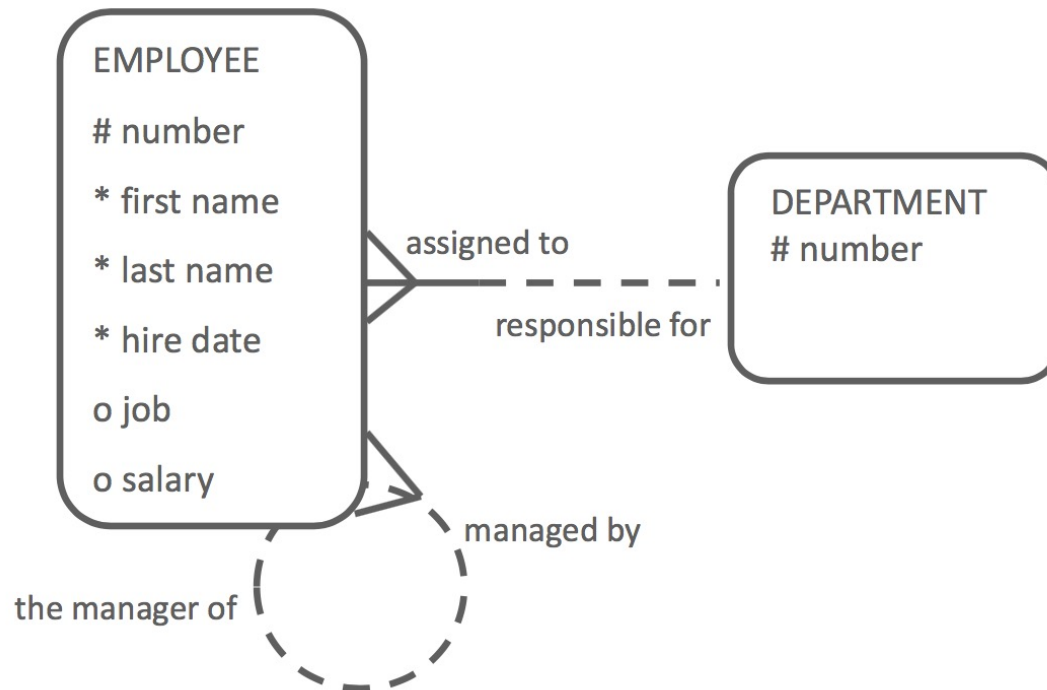
- [Release Notes](#)
- [Bugs Fixed](#)
- [Documentation](#)

Platform	Download	Release notes
Windows 64-bit with JDK included	 Download (317 MB)	<ul style="list-style-type: none">• MD5: 9476880e3fa961e8c4b2b87bf4675149• SHA1: 4d1b23be5ae0c19ef7e448c2beeb5a68dd61d749• Installation Notes, JDK 8 or 11 required
Windows 32-bit/64-bit	 Download (328 MB)	<ul style="list-style-type: none">• MD5: 32350b16329415eb0585cf7cce14203b• SHA1: e2e34e591049338a7f39c4baf00f8f253c2c2ab4• Installation Notes, JDK 8 or 11 required
Mac OSX	 Download (242 MB)	<ul style="list-style-type: none">• MD5: d3b25f06e26ffdf7ebc5e7ece074cc98• SHA1: 6e7e0b55cbe3f5dec12f809fcdc221f93b0390ea

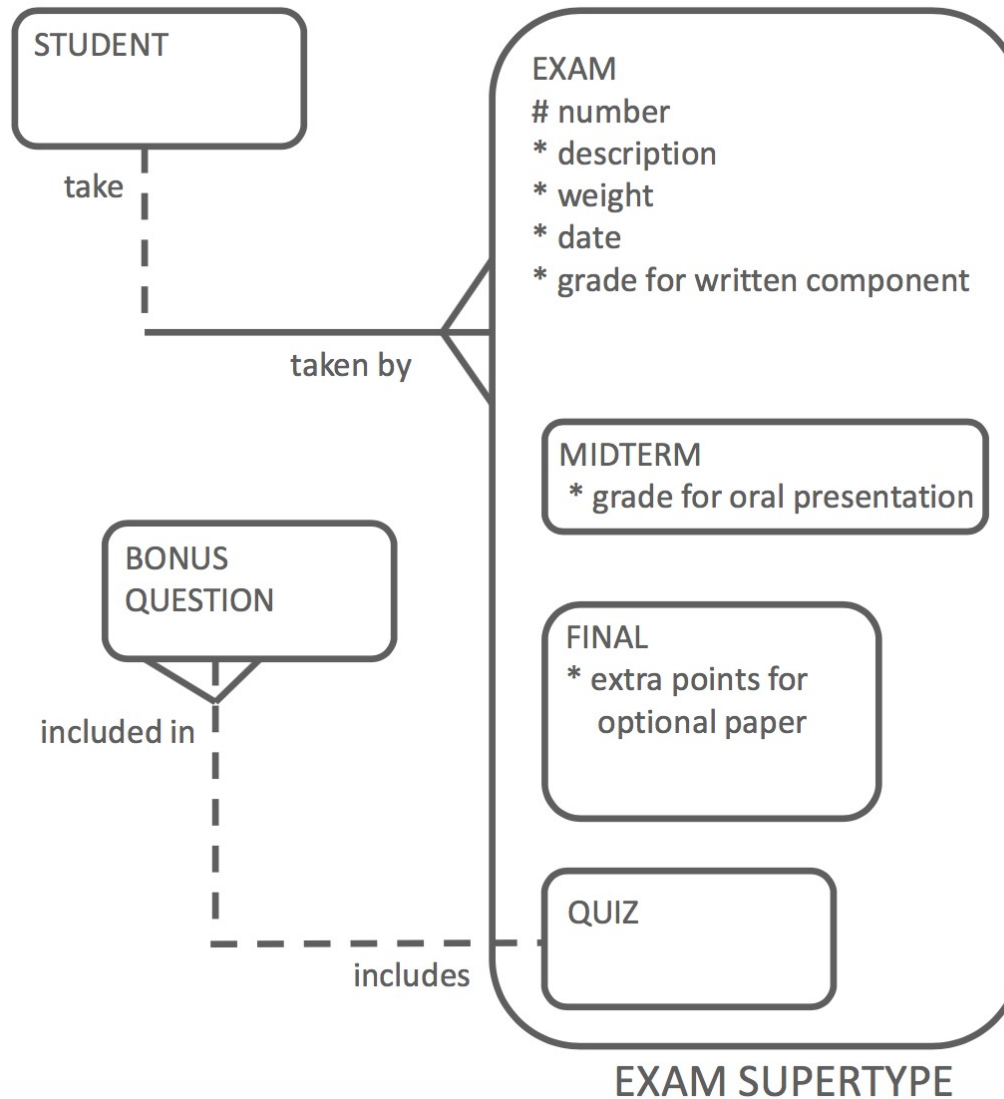
<https://www.oracle.com/tools/downloads/sql-data-modeler-downloads.html>

Good Tutorial

- ❖ <https://mikesmithers.wordpress.com/2018/05/30/fir-st-steps-in-sqldeveloper-data-modeler/>



SuperType Example



Arcs

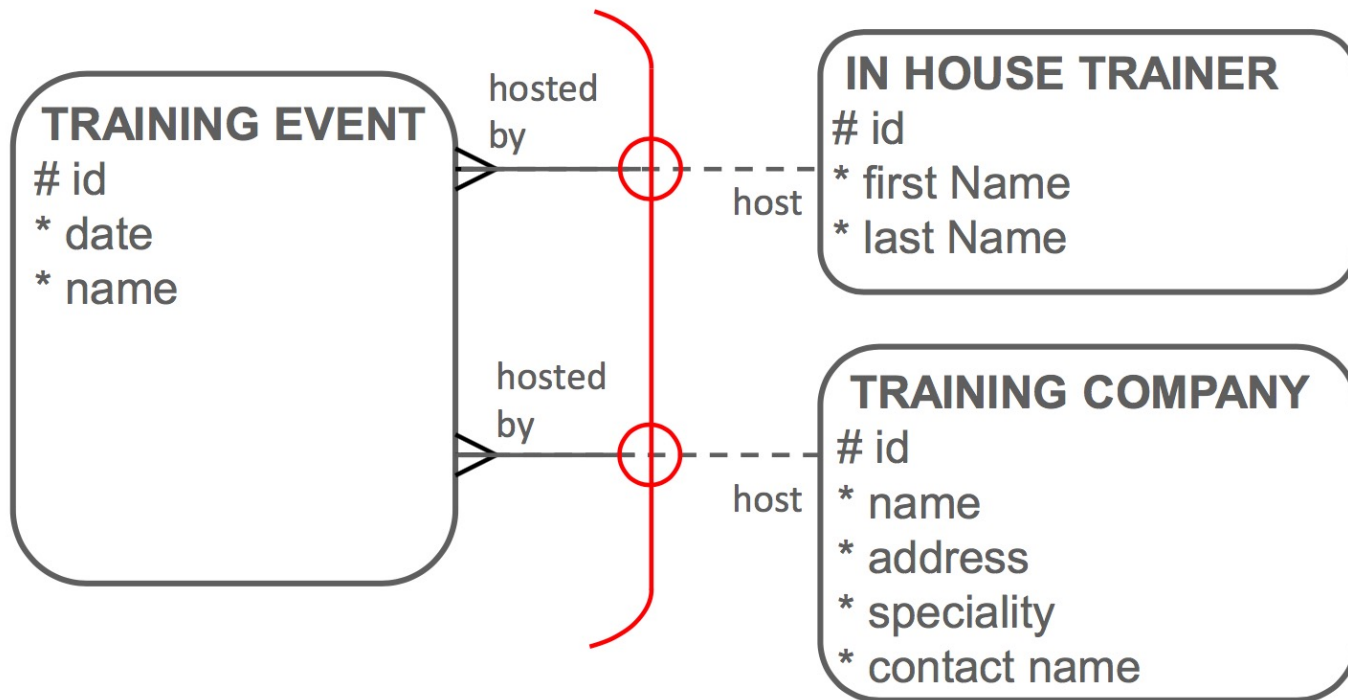
- ❖ Arcs in data modeling help designers clarify an exclusive OR across relationships.
- ❖ The more explicitly you can define the client's requirements, the more accurate your final implementation will be.

Exclusive OR Relationship

- ❖ Mutually exclusive relationships sometimes exist between entities and are also known as Exclusive OR Relationships
- ❖ An Exclusive OR relationship is a relationship between one entity and two (or more) other entities where only one of the relationships can exist at a time
- ❖ In ERDs, we model this type of relationship with an Arc

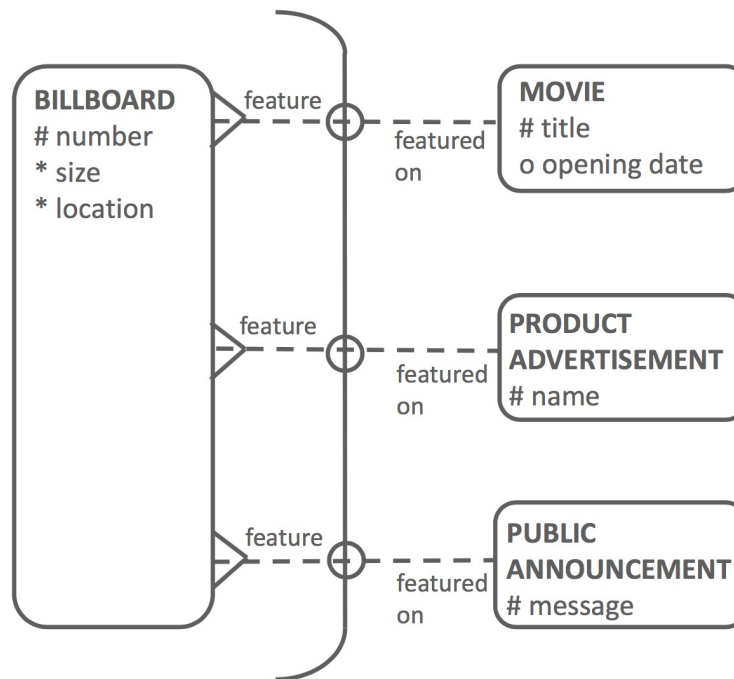
Exclusive OR Relationship

- ❖ For example: a TRAINING EVENT can be hosted by either an IN HOUSE TRAINER or an external TRAINING COMPANY.



Exclusive OR Relationship

- ❖ Another Example: A billboard is an advertising space that can feature a movie, a product, or a public announcement. It may contain advertising about only one of these at a time.

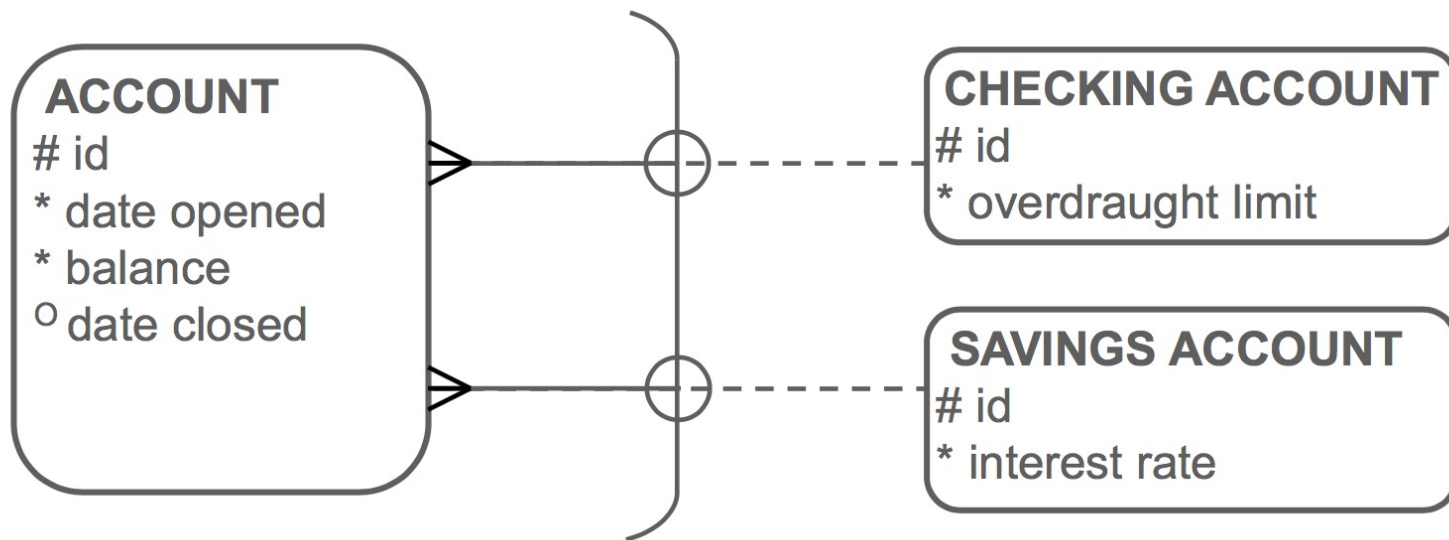


Arcs, Supertypes, and Subtypes

- ❖ Arcs and Super/subtypes both model mutual exclusiveness.
- ❖ Certain situations are best modeled as an arc, and others as supertype and subtypes.

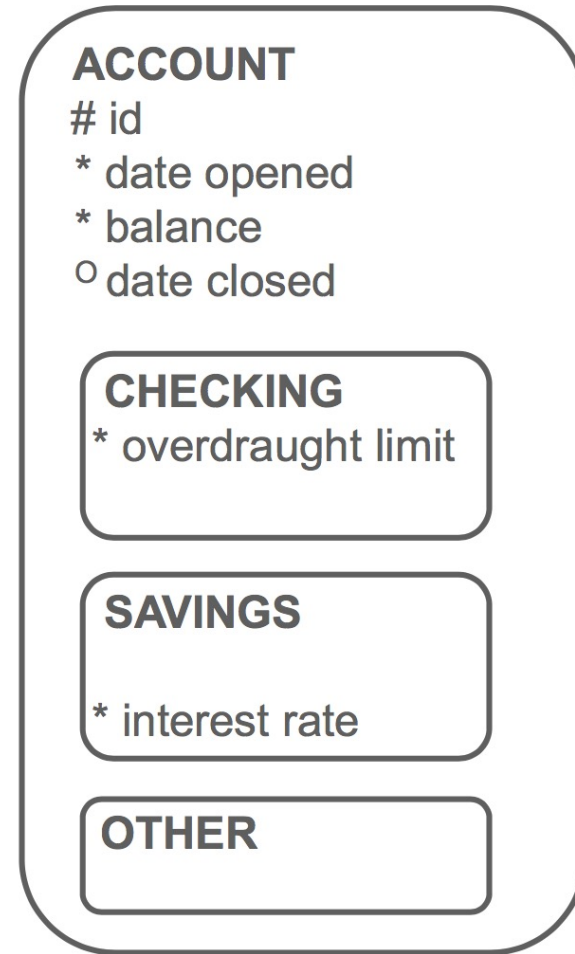
Arcs, Supertypes, and Subtypes

- Example 1: CHECKING ACCOUNT and SAVINGS ACCOUNT are “types” of ACCOUNT.

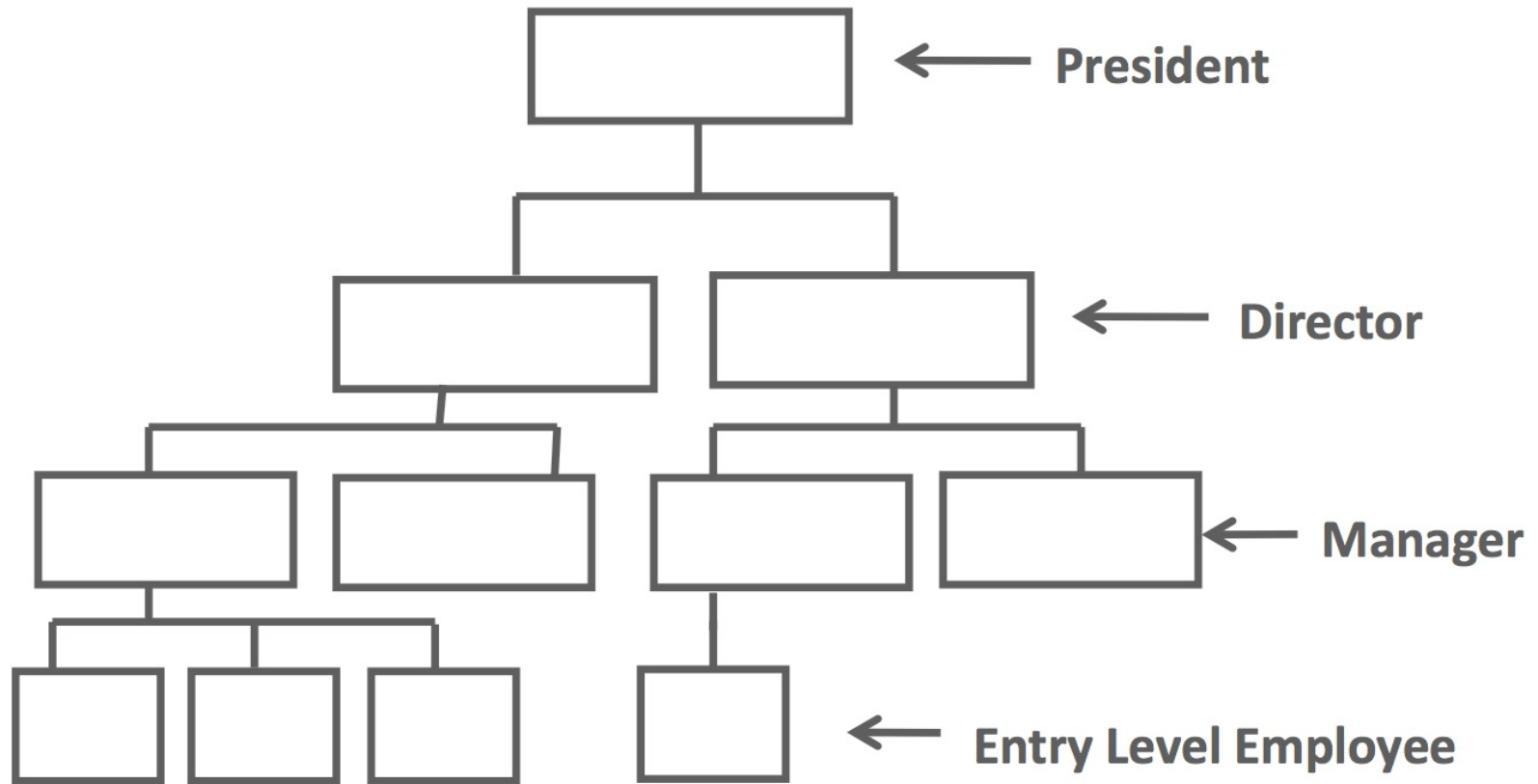


Arcs, Supertypes, and Subtypes

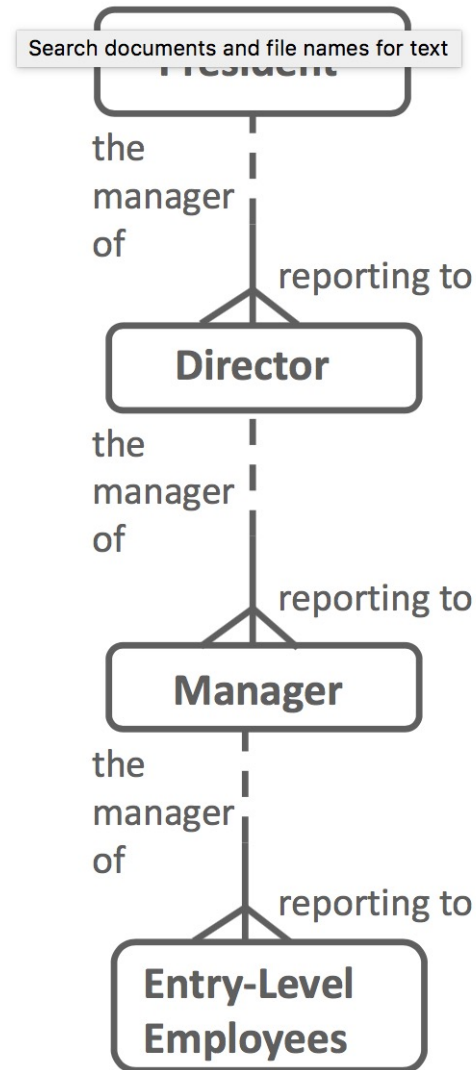
- This should be modeled as supertype and subtypes



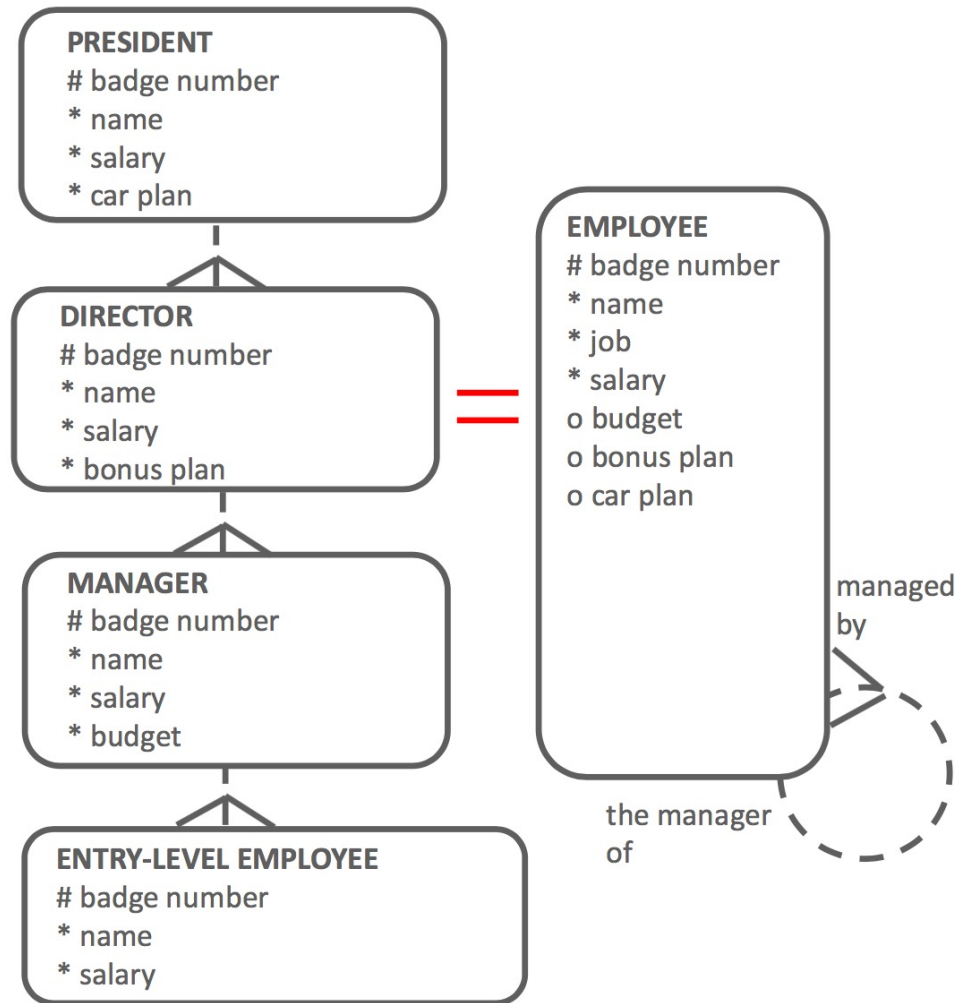
Hierarchy



Relationships in an Organizational Chart



Hierarchy Versus Recursive Relationship



Modeling Historical Data

- ❖ How tall were you at age 5? How tall were you at age 10? How tall are you right now?
- ❖ If your parents wrote this down when you were young, they were keeping track of historical data.

Modeling Historical Data

- ❖ Most businesses need to track some historical data.
- ❖ This helps them find trends and patterns that are the basis for business innovations or process improvements.
- ❖ For example, rental history of a movie is useful to a video store. It tells managers which movies are popular and which should be moved to the back shelf.

Attendance

[Attendance Record](#)

2122-MS218 Database Technologies

**Check-In is currently running .
Students can check in until check in period closes or is ended.**

6 1 7 5

Model Data Over Time

- ❖ When is it necessary to model data over time?
- ❖ Ask your client:
 - Is an audit trail required?
 - Can attribute values change over time?
 - Can relationships change over time?
 - Do you need to produce reports on older data?
 - Do you need to keep previous versions of the data? If so, for how long?

Data Over Time Example

- An organization needs to keep data about employees' salaries.
- All employees are paid weekly.
- Initially, the following EMPLOYEE entity was modeled.
- Additional requirements now specify that the organization needs to keep a historical record of how and when employees' salaries have changed during their employment.

EMPLOYEE

id

* first name

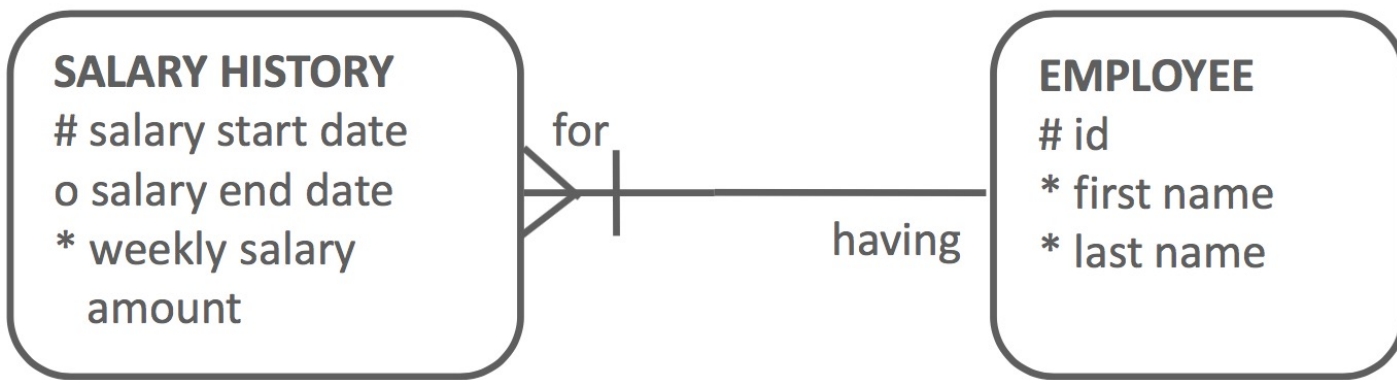
* last name

* weekly salary amount

* salary start date

Model Salary Changes

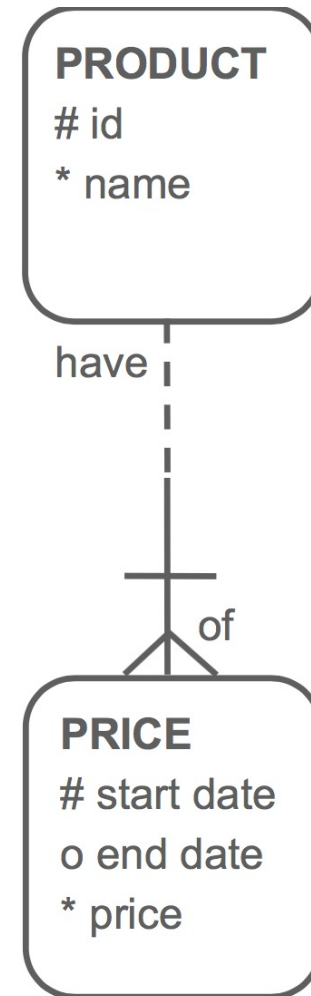
- To model salary changes over time, add a SALARY HISTORY entity.



- The UID of the SALARY HISTORY entity is the related EMPLOYEE id and the salary start date.

Model Historical Price

- It is often useful to have information on past prices.
- The model shown here tracks the historical price of a product.



Project Groups

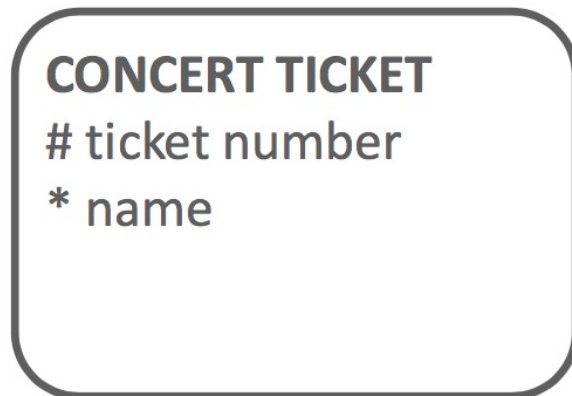
- ❖ 3 students in each group
- ❖ Select a group leader
- ❖ Group leader will email me group info, like names and registration numbers of other students. Please keep other members in CC
- ❖ If you have any difficulty in finding group members, then please email me, jamal.nasir@nuigalway.ie

Unique Identifier

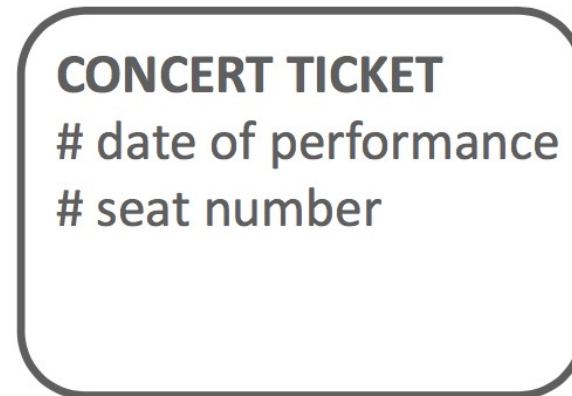
- ❖ The unique identifier (UID) is very important in relational databases.
- ❖ It is the value or combination of values that enables the user to find that one unique item among all the rest.
- ❖ Identifying just the right attribute, or combination of attributes and relationships, is a skill that any database designer must master.
- ❖ The unique identifier enables you to find your record in a file, a particular card in a deck of cards, your package in a warehouse, or a specific piece of data in a database.

Simple UUIDs vs. Composite UUIDs

- ❖ A UUID that is a single attribute is a simple UUID.
- ❖ However, sometimes a single attribute is not enough to uniquely identify an instance of an entity.
- ❖ If the UUID is a combination of attributes, it is called a composite UUID.



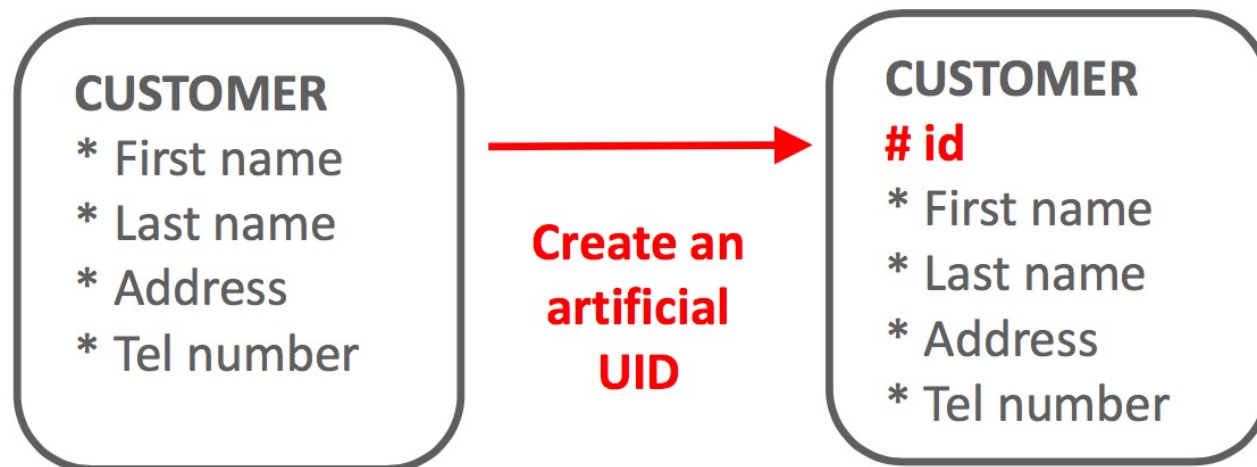
Simple Unique Identifier



Composite Unique Identifier

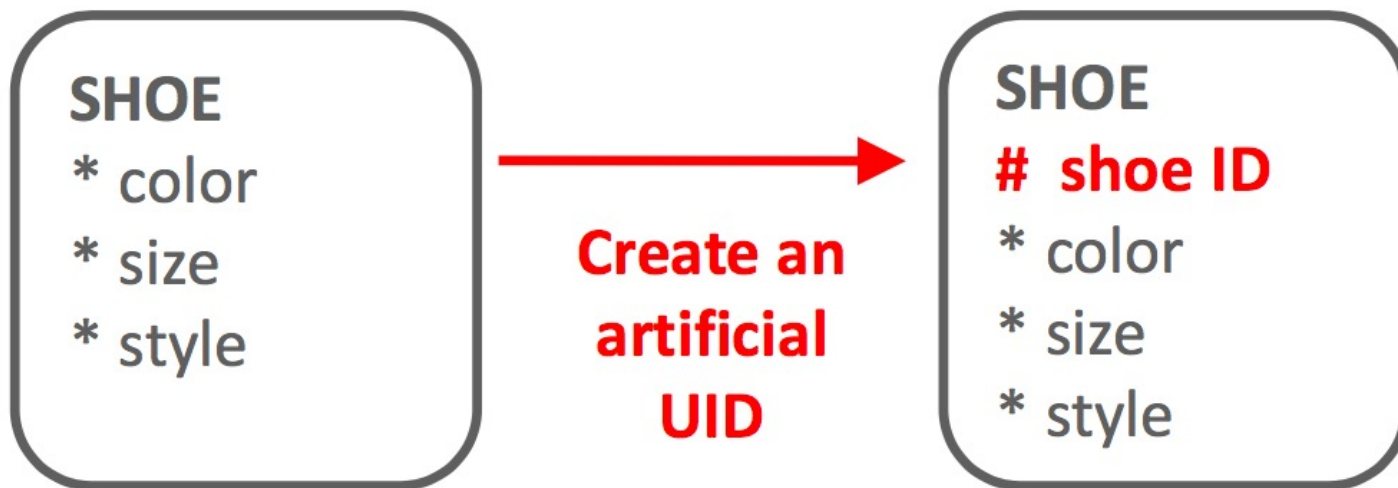
Artificial UUIDs

- ❖ Artificial UUIDs are those that don't occur in the natural world but are created for purposes of identification in a system.
- ❖ People are not born with “numbers,” but a lot of systems assign unique numbers to identify people: student numbers. customer IDs. etc.



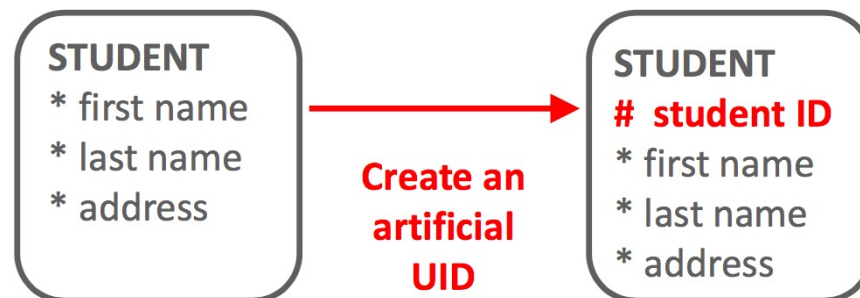
Artificial UUIDs

- ❖ A shoe has a color, a size, a style, but no truly descriptive “number.”
- ❖ However, a shoe store will assign unique numbers to each pair of shoes so they can be uniquely identified.



Artificial UUIDs

- ❖ How can we uniquely identify a STUDENT?
- ❖ Could we use a combination of first name and last name? – Only if we are sure that the combination is unique.
- ❖ Often, it is simpler and more straightforward to create an artificial attribute and make it the unique identifier.
- ❖ A UID can be both artificial and composite.



Candidate UUIDs

- ❖ Sometimes two or more possible UUIDs exist.
- ❖ For example, when you order a product from a commercial website, you will usually be assigned a unique customer code and asked to enter your e-mail address.
- ❖ Each of these uniquely identifies you, and each could be chosen as the UUID. These are both candidate UUIDs.
- ❖ Only one of the candidate UUIDs is chosen as the actual UUID. This is called the primary UUID.
- ❖ The other candidates are called secondary UUIDs.

Candidate UUIDs

- Student ID has been chosen as the primary UUID in both of these STUDENT entities.
- The first entity has one secondary UUID, while the second has two secondary UUIDs (one of which is composite).

STUDENT

student ID

(#) badge number

* first name

* last name

* address

One Primary UUID

One Secondary UUID

STUDENT

student ID

(#1) badge number

(#2-1) first name

(#2-2) last name

* address

One Primary UUID

Two Secondary UUIDs

Normalization(Purpose)

- ❖ Think about storing your friends' phone numbers in three different places: your address book, your cell phone, and a sheet of paper that you have taped to your refrigerator.
- ❖ It's a lot of work if a friend changes his/her phone number.
- ❖ You have to change it in your address book, cell phone, and the sheet of paper taped to your refrigerator.

Normalization(Purpose)

- ❖ What happens if data is stored in more than one place in a database?
- ❖ What if someone changes the information in one place and not the other—how do you know which information is correct?
- ❖ Redundancy like this causes unnecessary problems in a database.

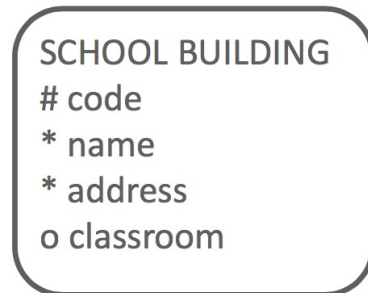
Normalization(Purpose)

- ❖ Normalization is a process that is used to eliminate these kinds of problems.
- ❖ One of your goals as a database designer is to "store information in one place and in the best possible place".
- ❖ If you follow the rules of normalization, you will achieve this goal.

First Normal Form (1NF)

- ❖ First Normal Form requires that no multi-valued attributes exist.
- ❖ To check for 1NF, validate that each attribute has a single value for each instance of the entity.
- ❖ One code, one name, and one address exist for the school building, but not one classroom.

SCHOOL BUILDING 1NF

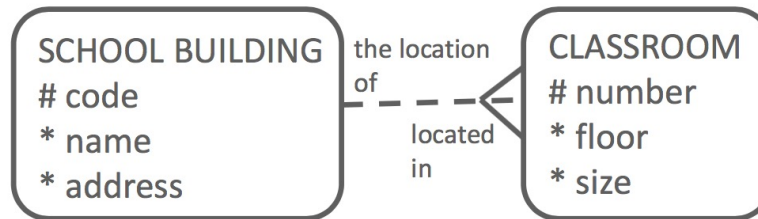
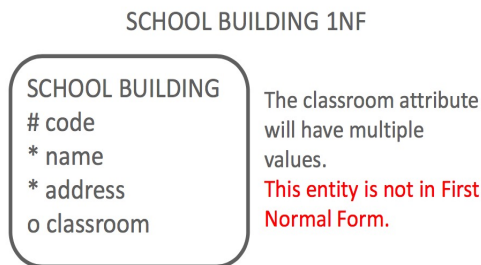


The classroom attribute will have multiple values.

This entity is not in First Normal Form.

First Normal Form (1NF)

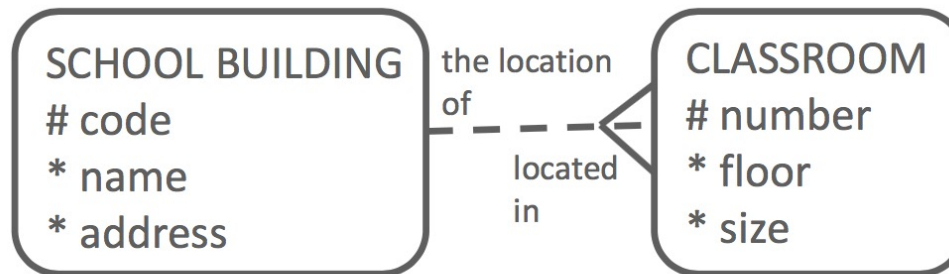
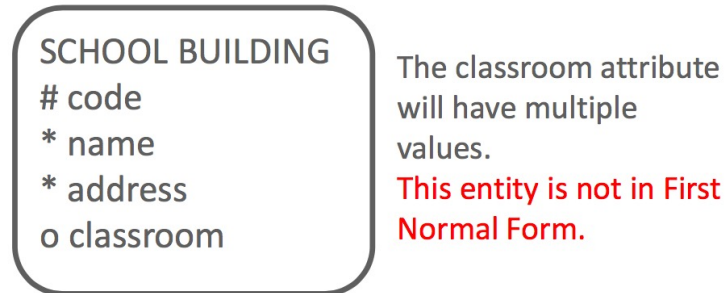
- ❖ Since many classrooms exist in a school building, classroom is multi-valued and violates 1NF.
- ❖ If an attribute is multi-valued, create an additional entity and relate it to the original entity with a 1:M relationship.



CLASSROOM is now its own entity.
All attributes have only one value per instance.
Both entities are in First Normal Form.

First Normal Form (1NF)

SCHOOL BUILDING 1NF



CLASSROOM is now its own entity.
All attributes have only one value per instance.
Both entities are in First Normal Form.

1NF Violations

1)

STUDENT

number

* first name

* last name

* subject

1-NF

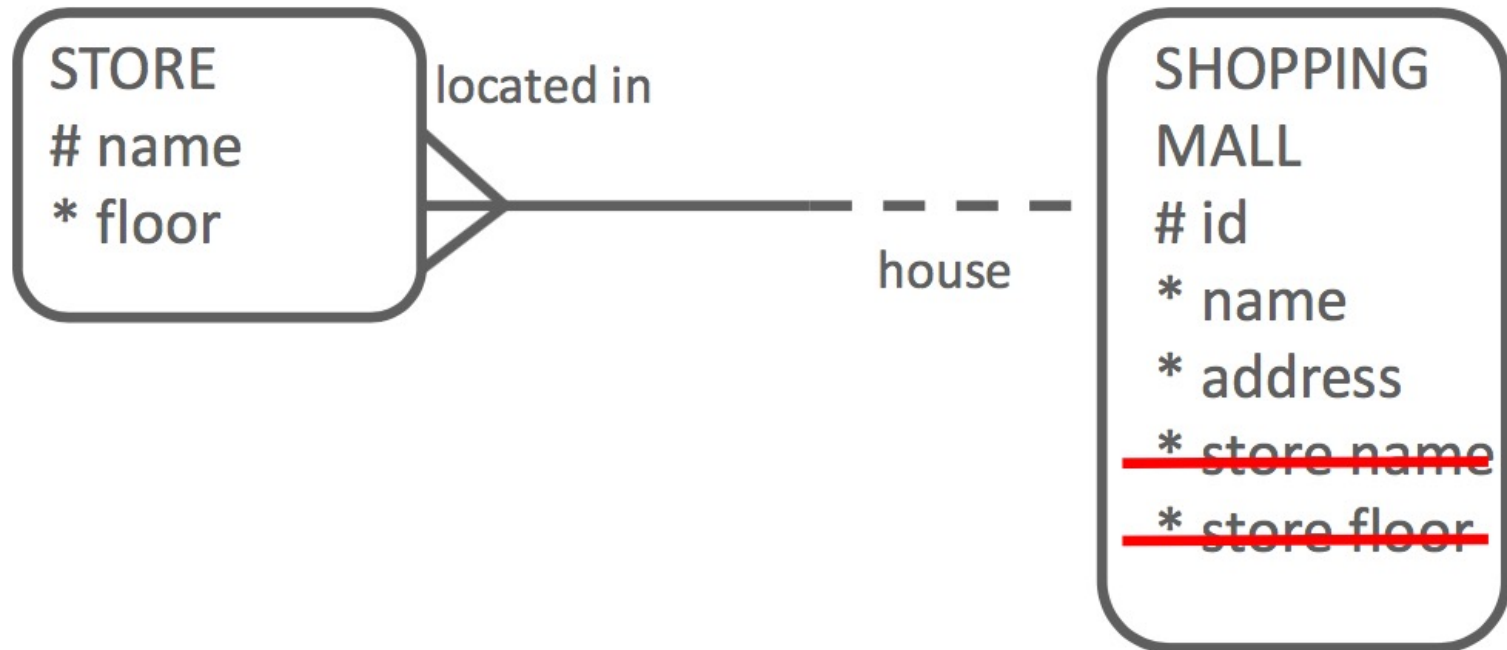


1-NF Violation

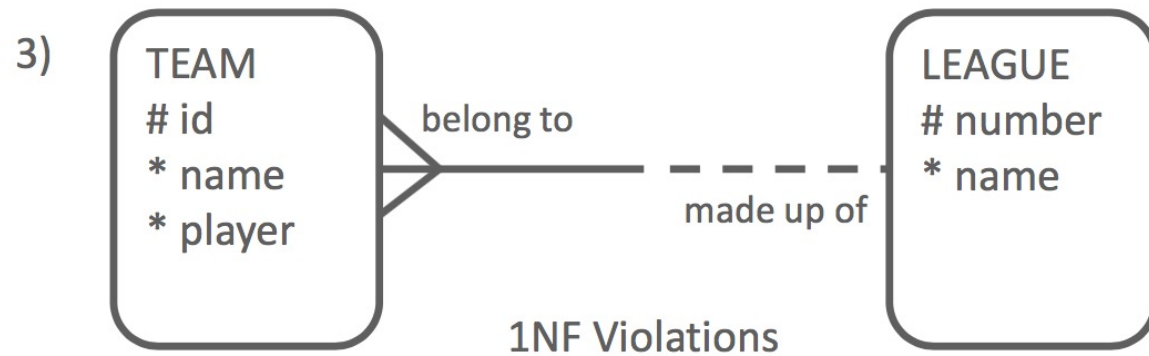
2)

SHOPPING
MALL
id
* name
* address
* store name
* store floor

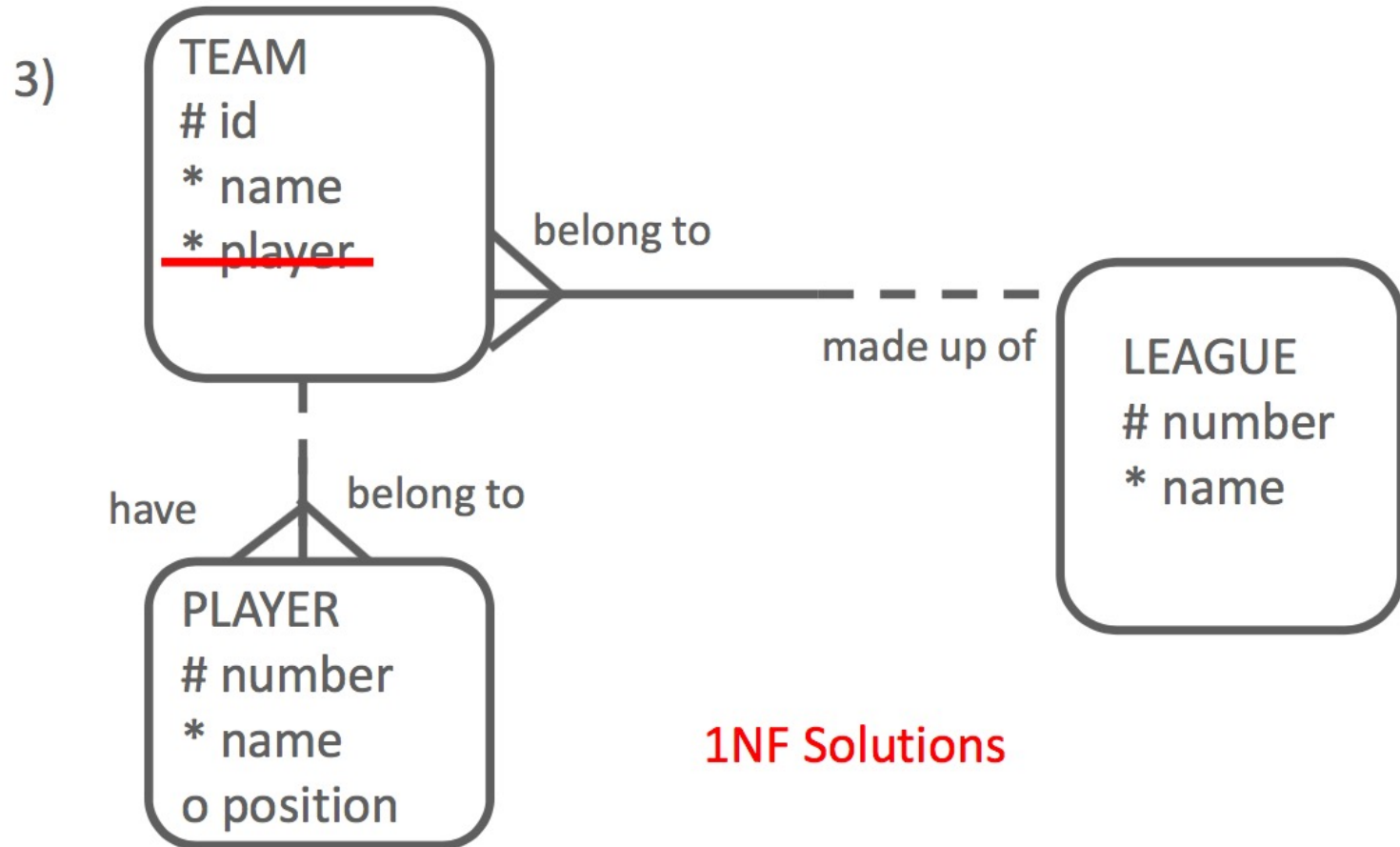
1-NF



1-NF Violation



1-NF



1NF Solutions