
ORACLE
ACADEMY

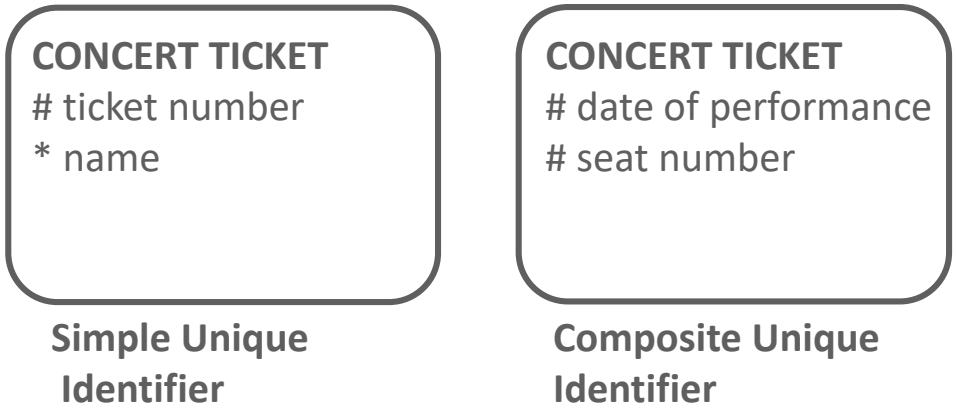
Database Design

6-1
Artificial, Composite, and Secondary UIDs

ORACLE ACADEMY

Simple UIDs vs. Composite UIDs

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

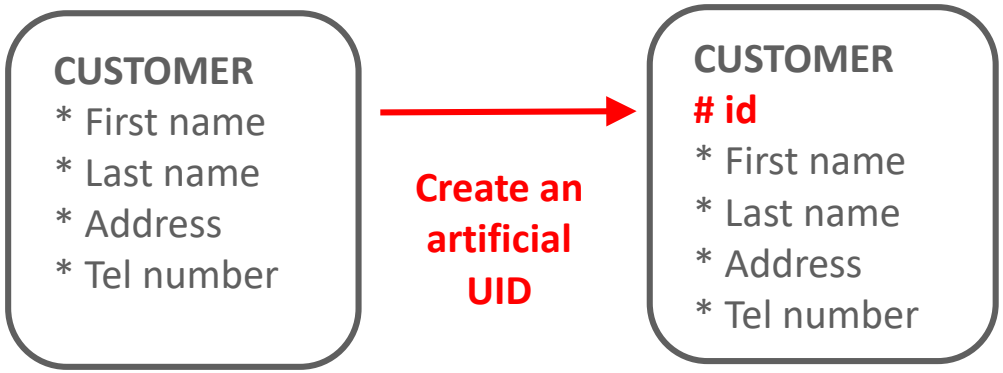


Simple Unique Identifier

Composite Unique Identifier

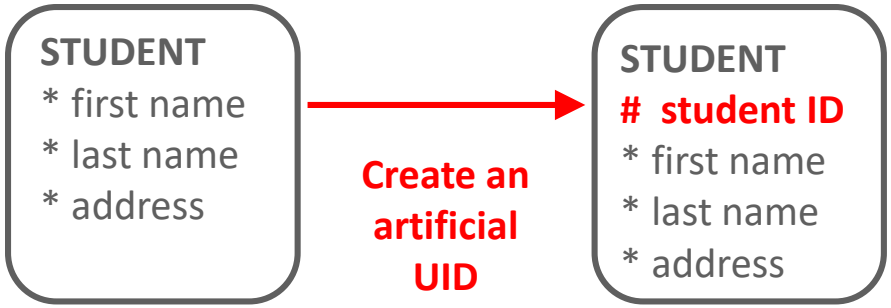
Artificial UIDs

- Artificial UIDs 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.



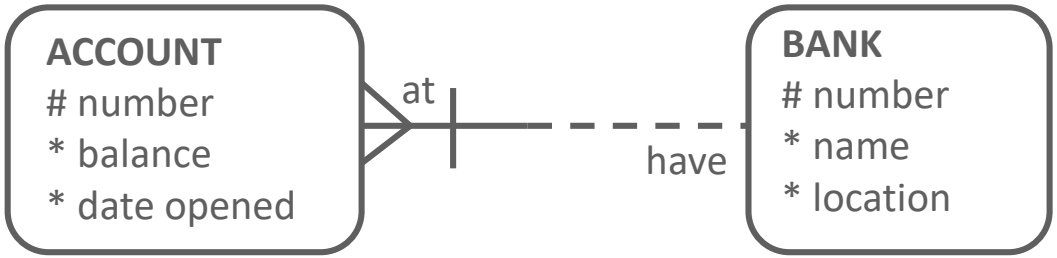
Artificial UID Example

- 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.



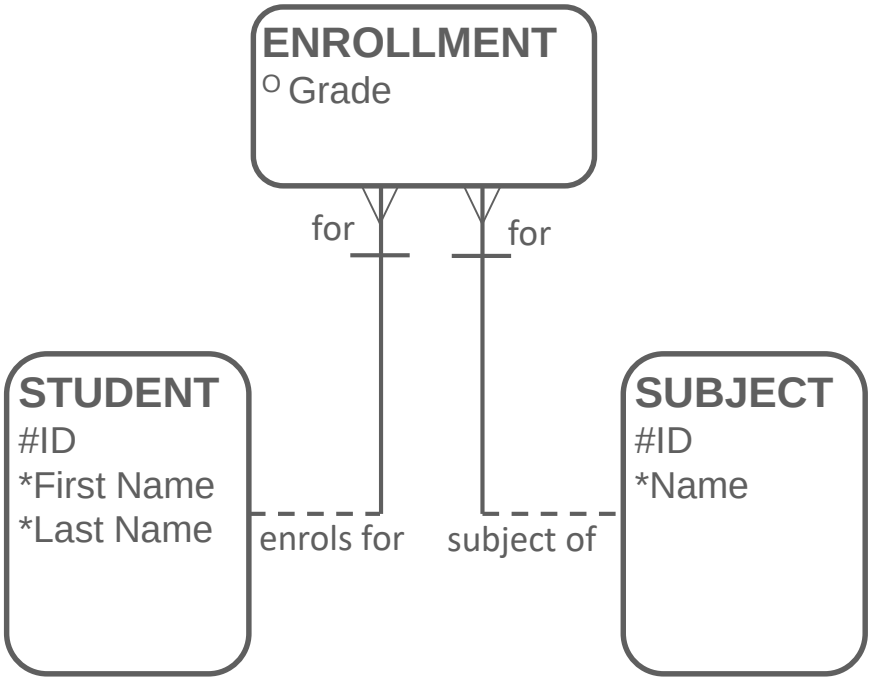
UIDs from Barred Relationships

- Sometimes the UID is a combination of an attribute and a relationship.
- What is the UID of ACCOUNT? Is it artificial? Is it composite?
- Two people could have the same bank account number, but at different banks.
- Bank to bank transfers always need the bank routing number in addition to the bank account number.



UID from Barred Relationship Intersection Entity

- As we've seen before, the resolution of a M:M relationship often results in barred relationships from the intersection entity to the original ones.
- In this example, the UID of ENROLLMENT comes from STUDENT and SUBJECT.
- The bars on the relationships tell you this.

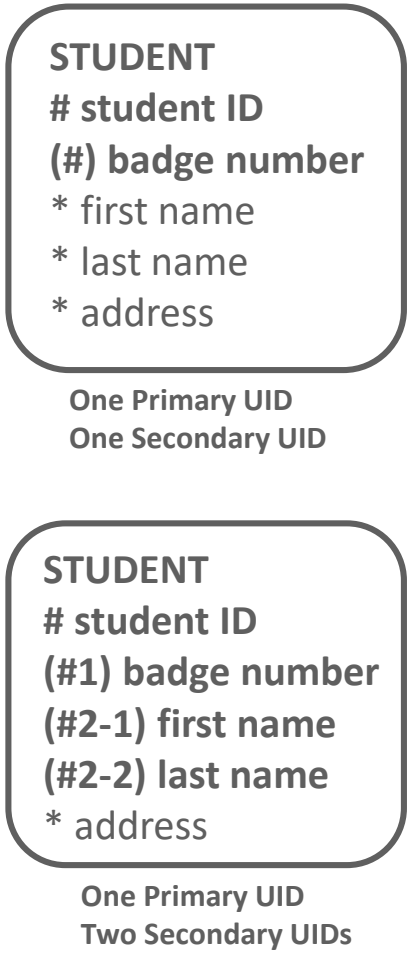


Candidate UIDs

- Sometimes two or more possible UIDs 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 UID. These are both candidate UIDs.
- Only one of the candidate UIDs is chosen as the actual UID. This is called the primary UID.
- The other candidates are called secondary UIDs.

Candidate UIDs

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



Identification: Database vs. Real World

- Unique identifiers make it possible for us to distinguish one instance of an entity from another.
- As you will see later, these become primary keys in the database.
- A primary key allows you to access a specific record in a database.
- In the real world, however, it is sometimes not so easy to distinguish one thing from another.

Database Design

6-2
Normalization and First Normal Form

ORACLE ACADEMY

Objectives

This lesson covers the following objectives:

- Define the purpose of normalization in database models
- Define the rule of First Normal Form in the normalization process
- Determine if an entity conforms to the rule of First Normal Form
- Convert an entity to First Normal Form if needed

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.

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.

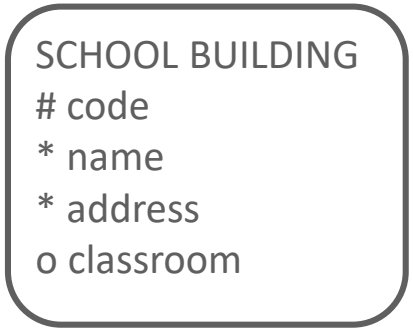
Purpose

- 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.

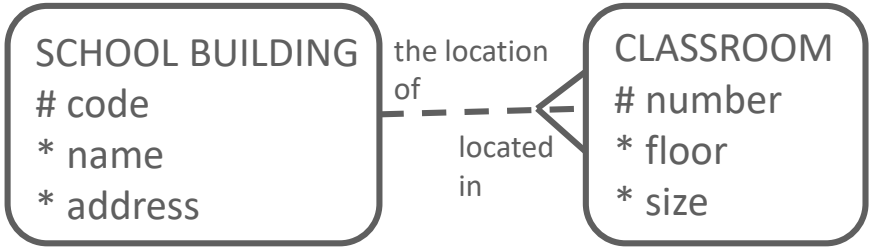
First Normal Form (1NF)

- 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.

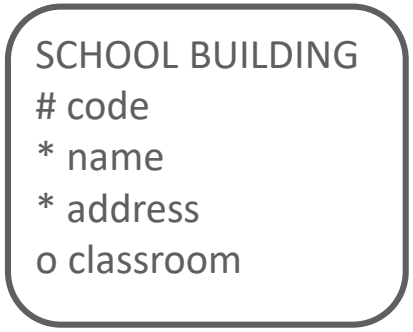


CLASSROOM is now its own entity.
All attributes have only one value per instance.
Both entities are in First Normal Form.

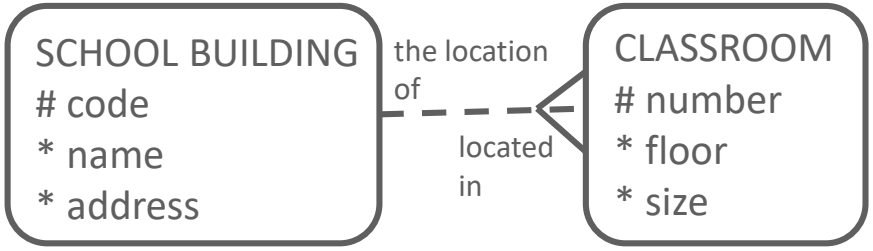
First Normal Form (1NF)

- 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.

SCHOOL BUILDING 1NF



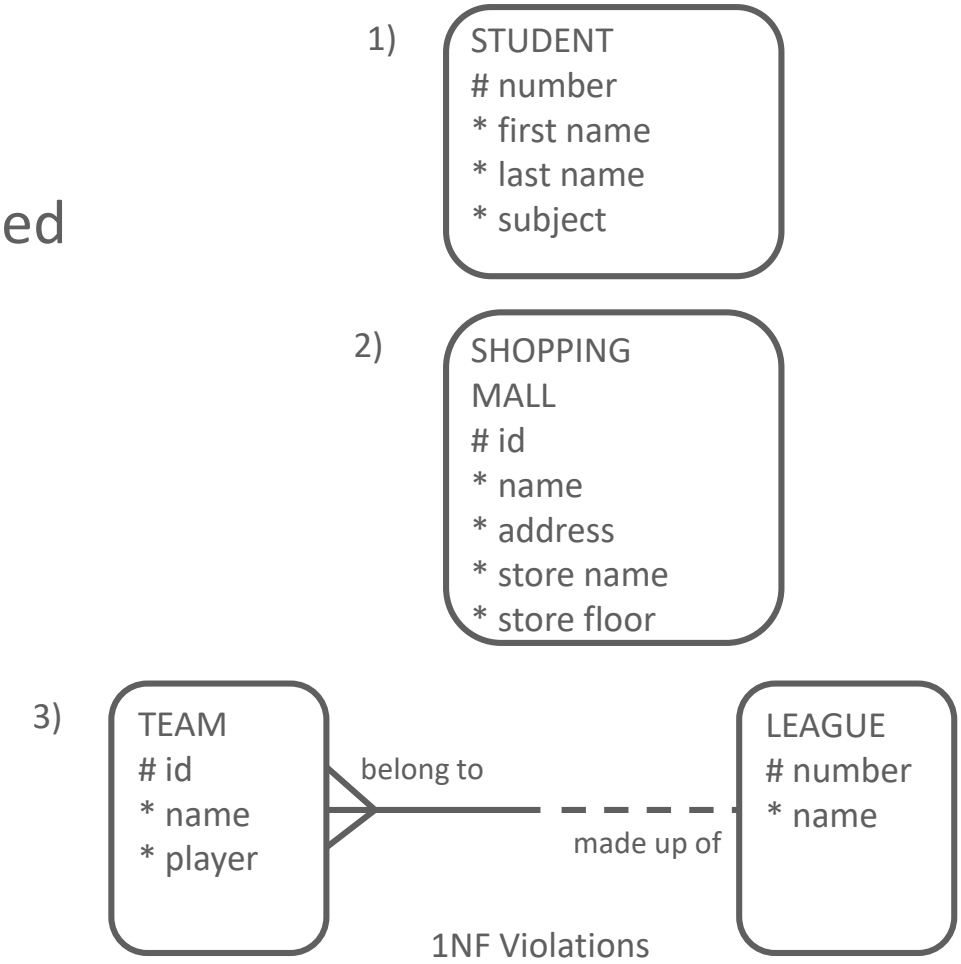
The classroom attribute will have multiple values.
This entity is not in First Normal Form.



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

1NF Violations

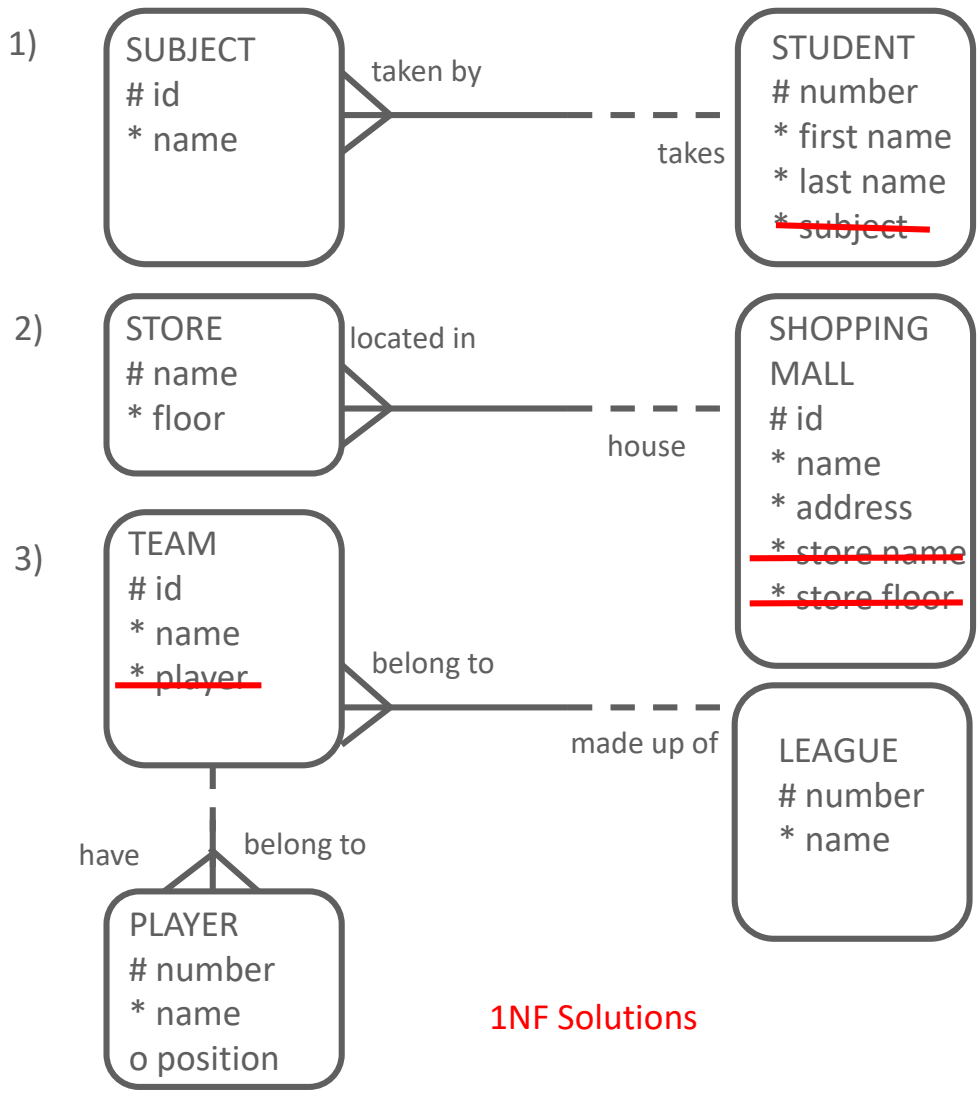
- Examine the entities.
- Are there any multi-valued attributes?



1NF Violations

1NF Solutions

- When all the attributes in an entity are single-valued, that entity is said to be in First Normal Form.



1NF Solutions

Terminology

Key terms used in this lesson included:

- First Normal Form (1NF)
- Normalization
- Redundancy

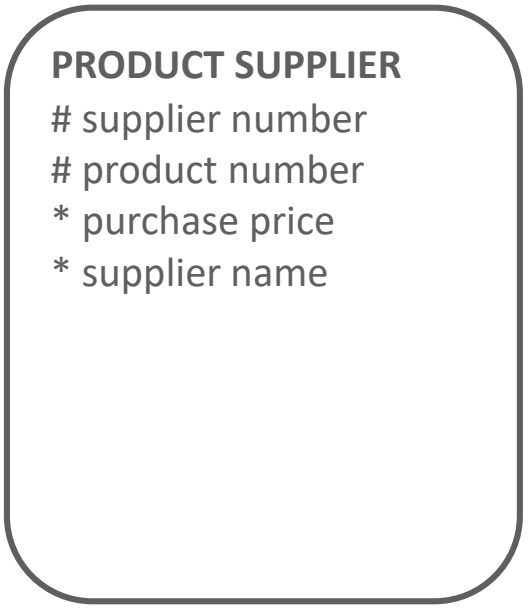
Database Design

6-3
Second Normal Form

ORACLE ACADEMY

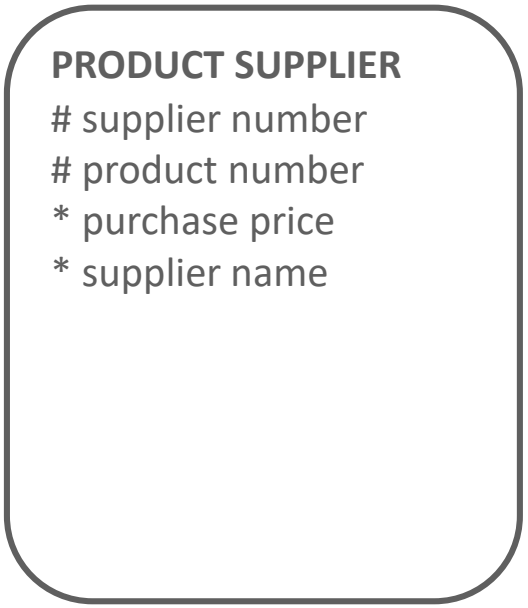
Second Normal Form Example

- Examine the entity PRODUCT SUPPLIER.
- The UID is a composite UID consisting of the supplier number and the product number.
- If one supplier supplies 5 different products, then 5 different instances are created.
- What happens if the supplier name changes?



Second Normal Form Example

- The supplier name would then need to be changed in 5 different instances.
- What if some of them were changed, but not others?
- How would users know which name is the correct name?



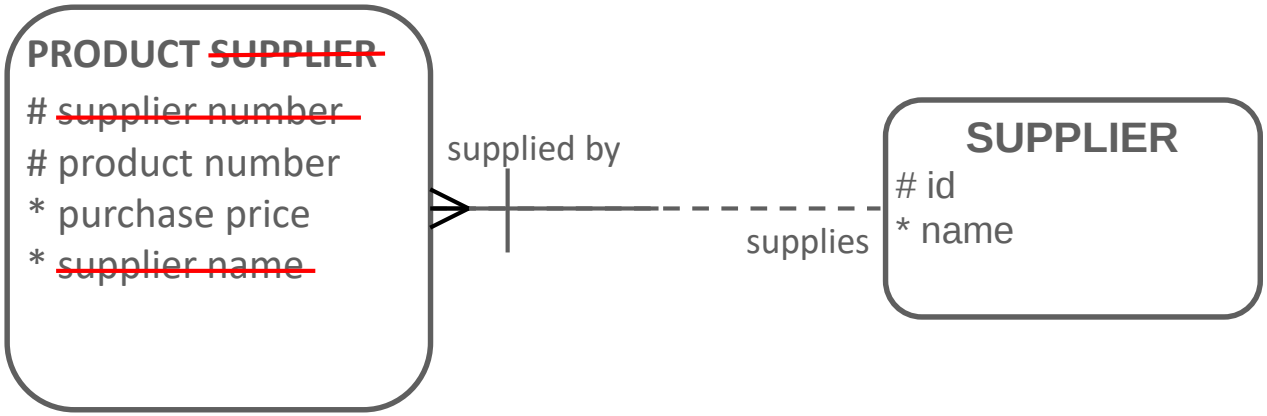
Second Normal Form Description

- Second Normal Form (2NF) requires that any non-UID attribute be dependent on (be a property of, or a characteristic of) the entire UID.
- Is purchase price a property of supplier number, product number, or both?

PRODUCT SUPPLIER
supplier number
product number
* purchase price
* supplier name

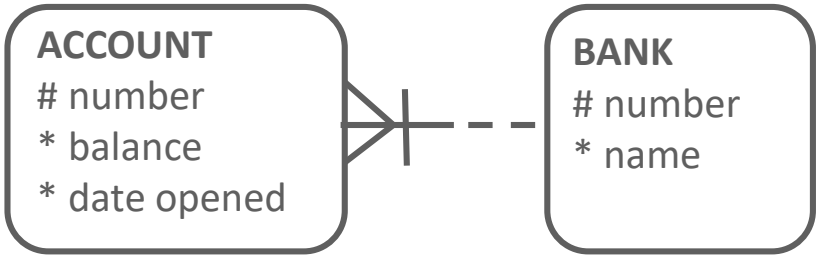
Second Normal Form Description

- Is supplier name a property of supplier number, product number, or both?
- 2NF requires a “both” answer to each question.



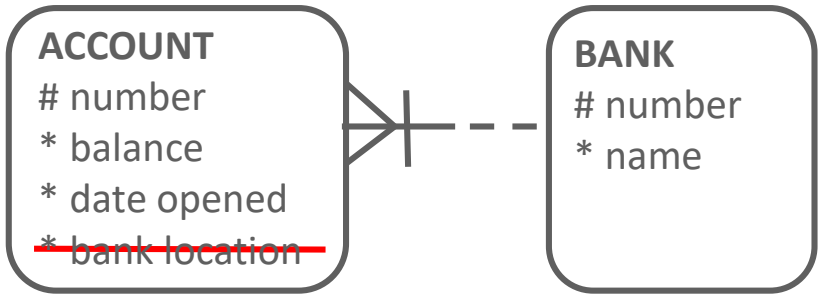
Second Normal Form Bar Relationship

- The UID for ACCOUNT is a composite UID from a barred relationship consisting of ACCOUNT number and BANK number.
- Is balance a property of ACCOUNT number, BANK number, or both?
- Is date opened a property of ACCOUNT number, BANK number, or both?



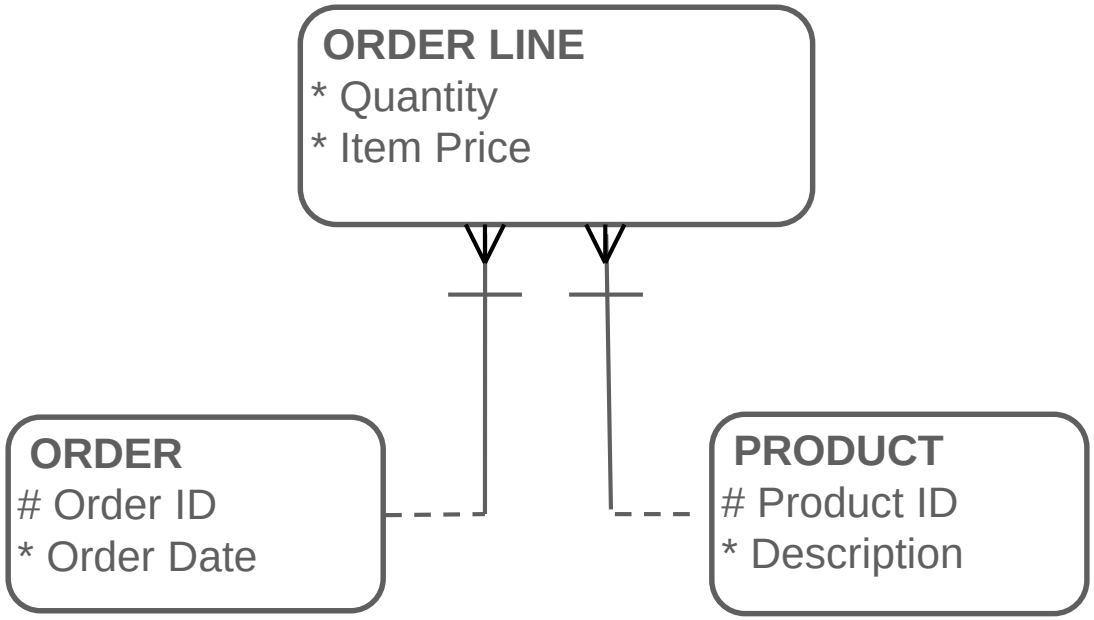
Second Normal Form Violation

- In this ERD, the attribute bank location has been added. Is bank location a property of ACCOUNT number, BANK number, or both?
- It is a property of BANK number only and is thus misplaced. This is a violation of Second Normal Form.
- What would happen if a bank's location changed?
- Every account at that bank would need to be updated.



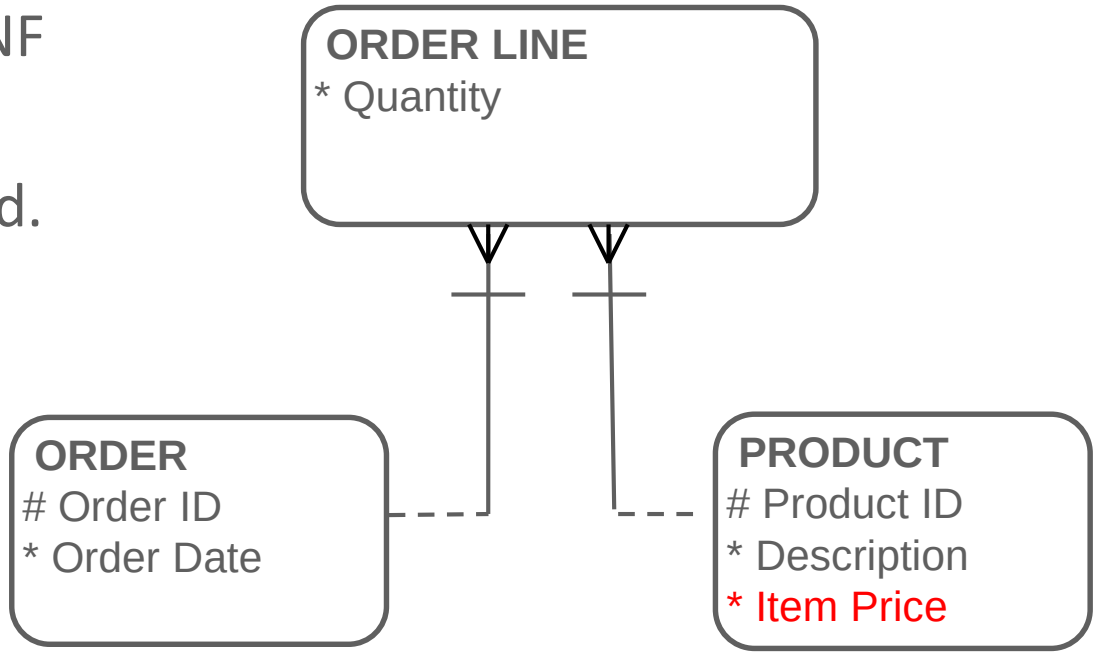
Order ERD

- What is wrong with this diagram?



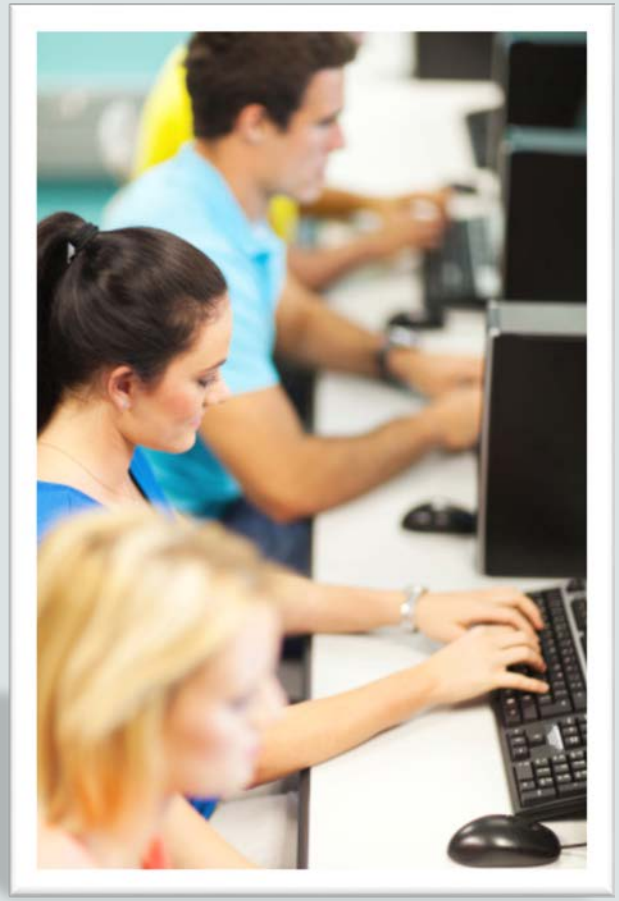
Order ERD

- The ERD is now in 2NF
- Answer: The price attribute is misplaced.
- Item Price depends solely on PRODUCT.
- This is a violation of Second Normal Form.



Database Design

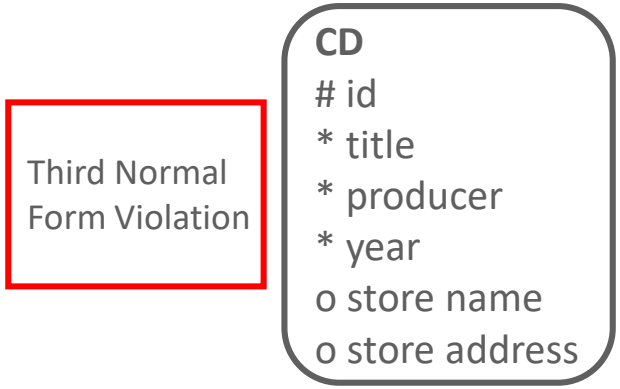
6-4
Third Normal Form



ORACLE ACADEMY

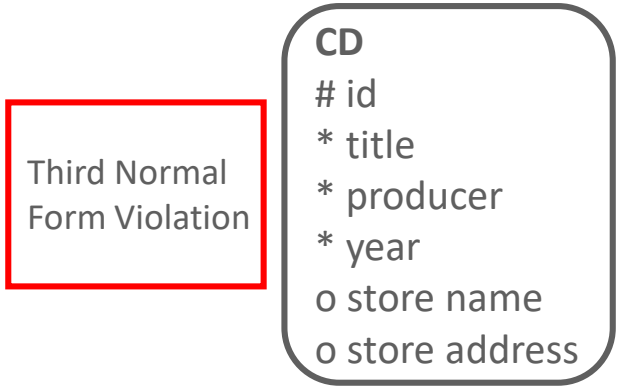
Third Normal Form Rule

- The rule of Third Normal Form (3NF) states that no non-UID attribute can be dependent on another non-UID attribute.
- Third Normal Form prohibits transitive dependencies.
- A transitive dependency exists when any attribute in an entity is dependent on any other non-UID attribute in that entity.



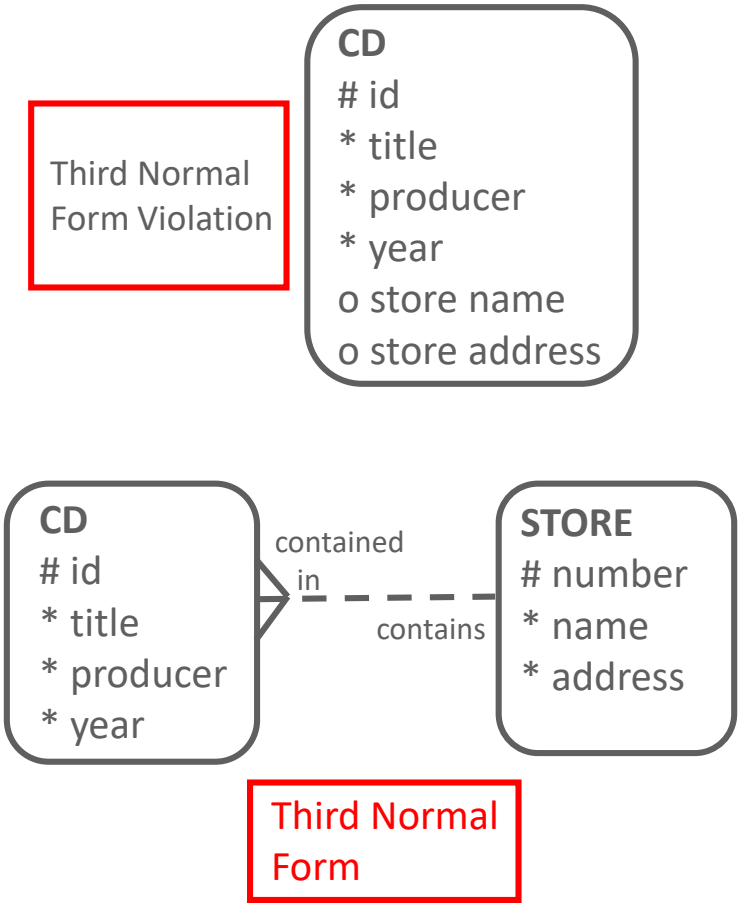
Third Normal Form Rule

- Think of the kind of information you'd like to store about your CD collection.
- Does information about the store where you bought the CD belong in the same entity?
- If the store address changed, you would have to change the information on all the CDs that were bought at that store.



Third Normal Form Transitive Dependency

- The store address is dependent on the CD number, which is the UID of the CD entity. So this entity is in 1NF and 2NF.
- But store address is also dependent on store name, which is a non-UID attribute.
- This is an example of a transitive dependency and a violation of Third Normal Form.



Third Normal Form Transitive Dependency

- The correctly normalized model is shown here: create a second entity STORE, with a relationship to CD.