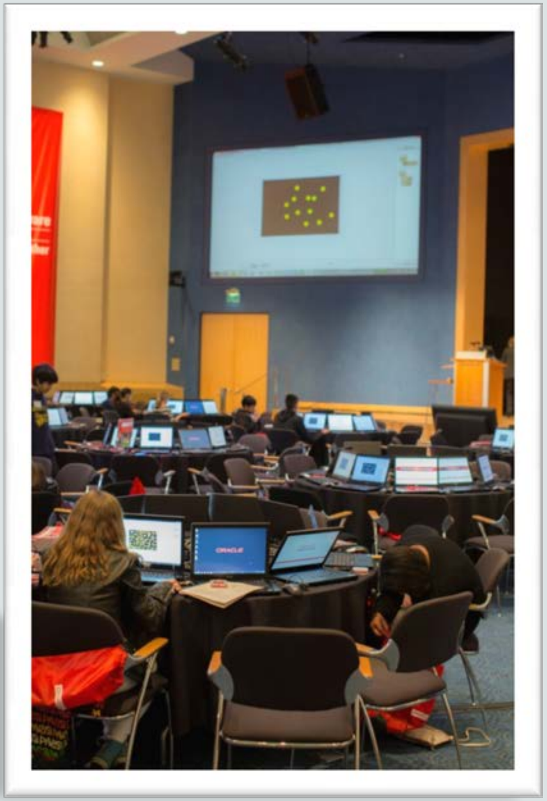
ORACLE® ACADEMY

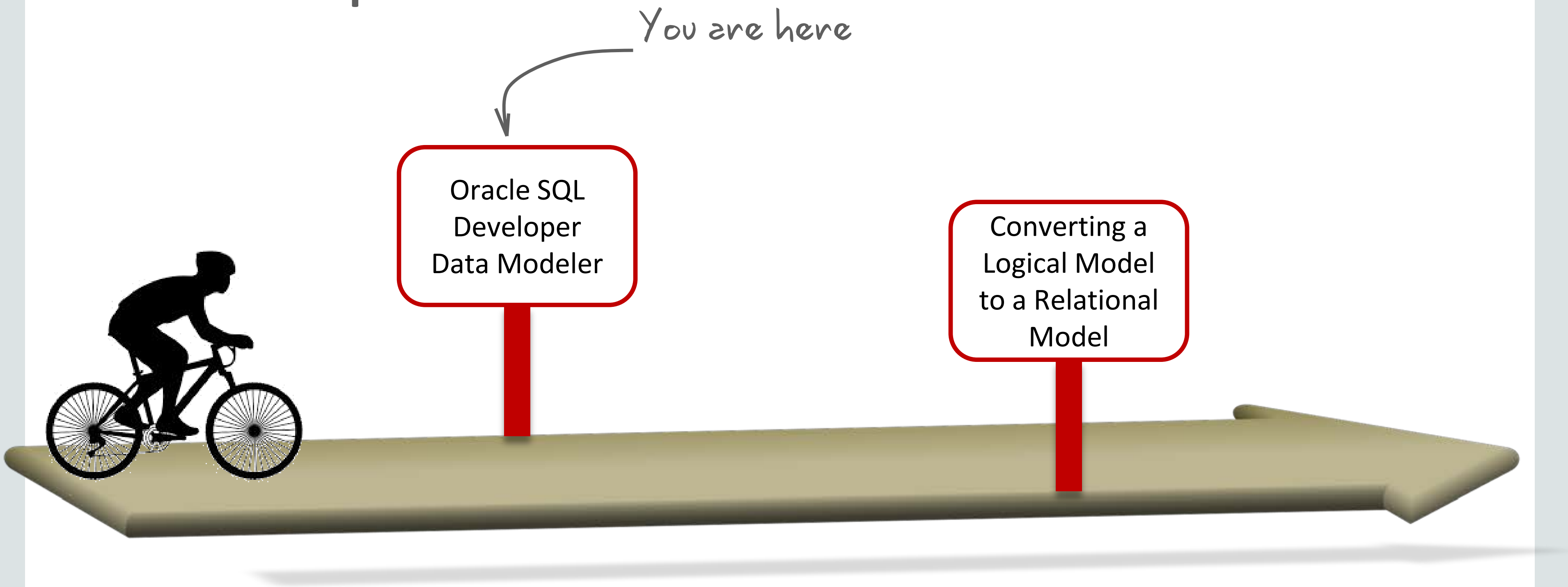
# Database Foundations

## 4-1
## Oracle SQL Developer Data Modeler



ORACLE ACADEMY

# Road Map


You are here
Oracle SQL Developer Data Modeler
Converting a Logical Model to a Relational Model

# Objectives

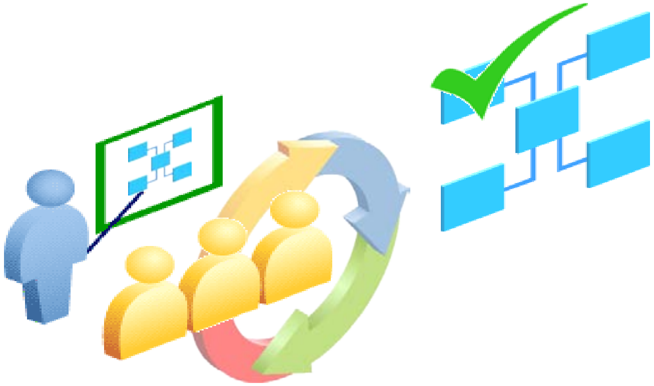
This lesson covers the following objectives:

- Use Oracle SQL Developer Data Modeler to create:
  - Entities, attributes, and UIDs with correct optionality and cardinality
  - Supertype and subtype entities
  - Arc, hierarchical, barred, and recursive relationships

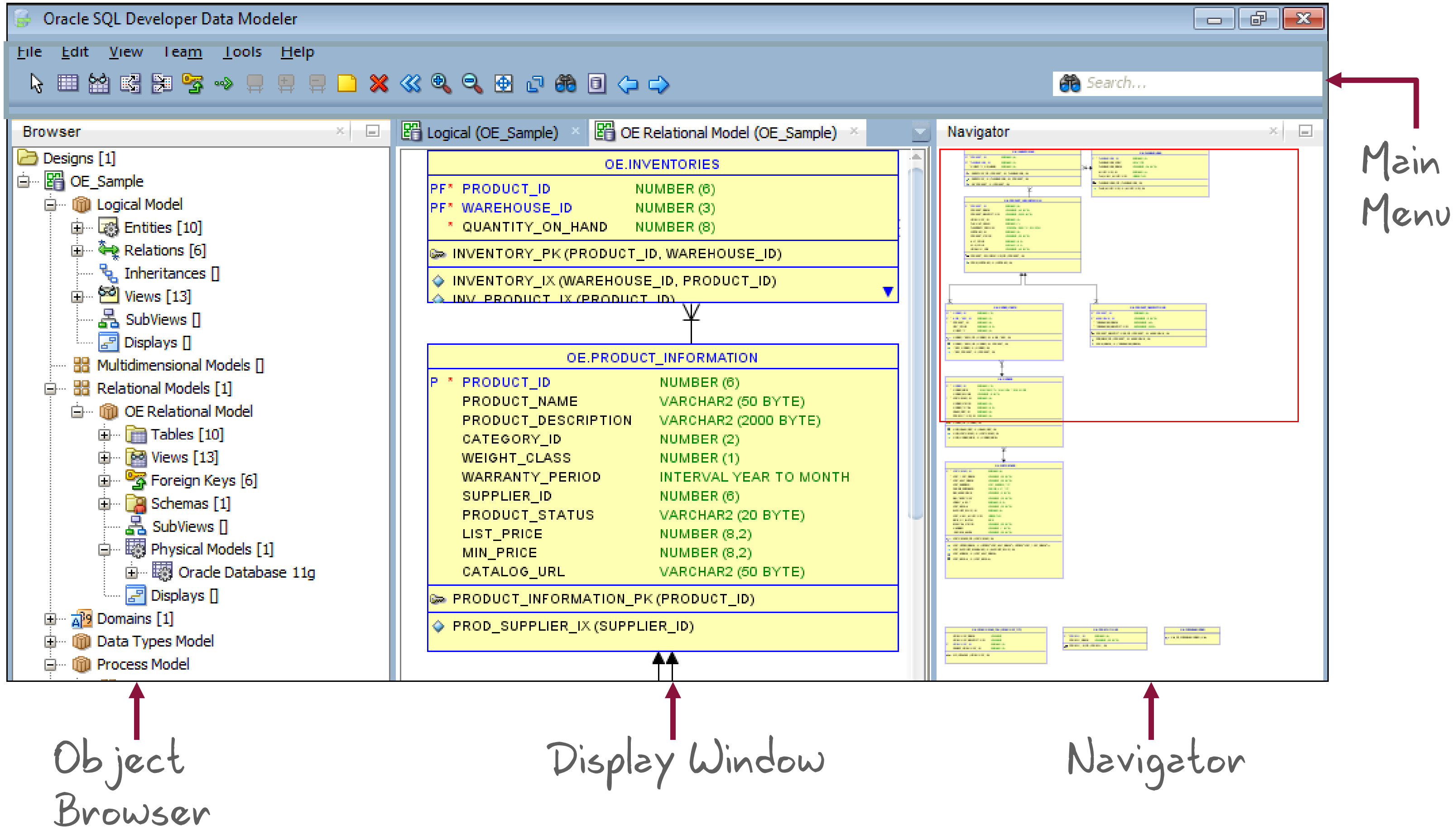
# Introduction to Oracle SQL Developer Data Modeler

Oracle SQL Developer Data Modeler offers a range of data and database modeling capabilities, enabling you to:

- Capture business rules and information
- Create process, logical, relational, and physical models
- Store metadata information in XML files
- Synchronize the relational model with the data dictionary



# Oracle SQL Developer Data Modeler Interface: Example Overview


Oracle SQL Developer Data Modeler
File Edit View Team Tools Help
Search...
Browser
Logical (OE_Sample)
OE Relational Model (OE_Sample)
Navigator
Main Menu
Object Browser
Display Window
Navigator

- Designs [1]
  - OE_Sample
    - Logical Model
      - Entities [10]
      - Relations [6]
      - Inheritances []
      - Views [13]
      - SubViews []
      - Displays []
    - Multidimensional Models []
    - Relational Models [1]
      - OE Relational Model
        - Tables [10]
        - Views [13]
        - Foreign Keys [6]
        - Schemas [1]
        - SubViews []
        - Physical Models [1]
          - Oracle Database 11g
        - Displays []
    - Domains [1]
    - Data Types Model
    - Process Model

OE.INVENTORIES

| | | |
|---|---|---|
| PF* | PRODUCT_ID | NUMBER (6) |
| PF* | WAREHOUSE_ID | NUMBER (3) |
| * | QUANTITY_ON_HAND | NUMBER (8) |

INVENTORY_PK (PRODUCT_ID, WAREHOUSE_ID)

INVENTORY_IX (WAREHOUSE_ID, PRODUCT_ID)
INV_PRODUCT_IX (PRODUCT_ID)

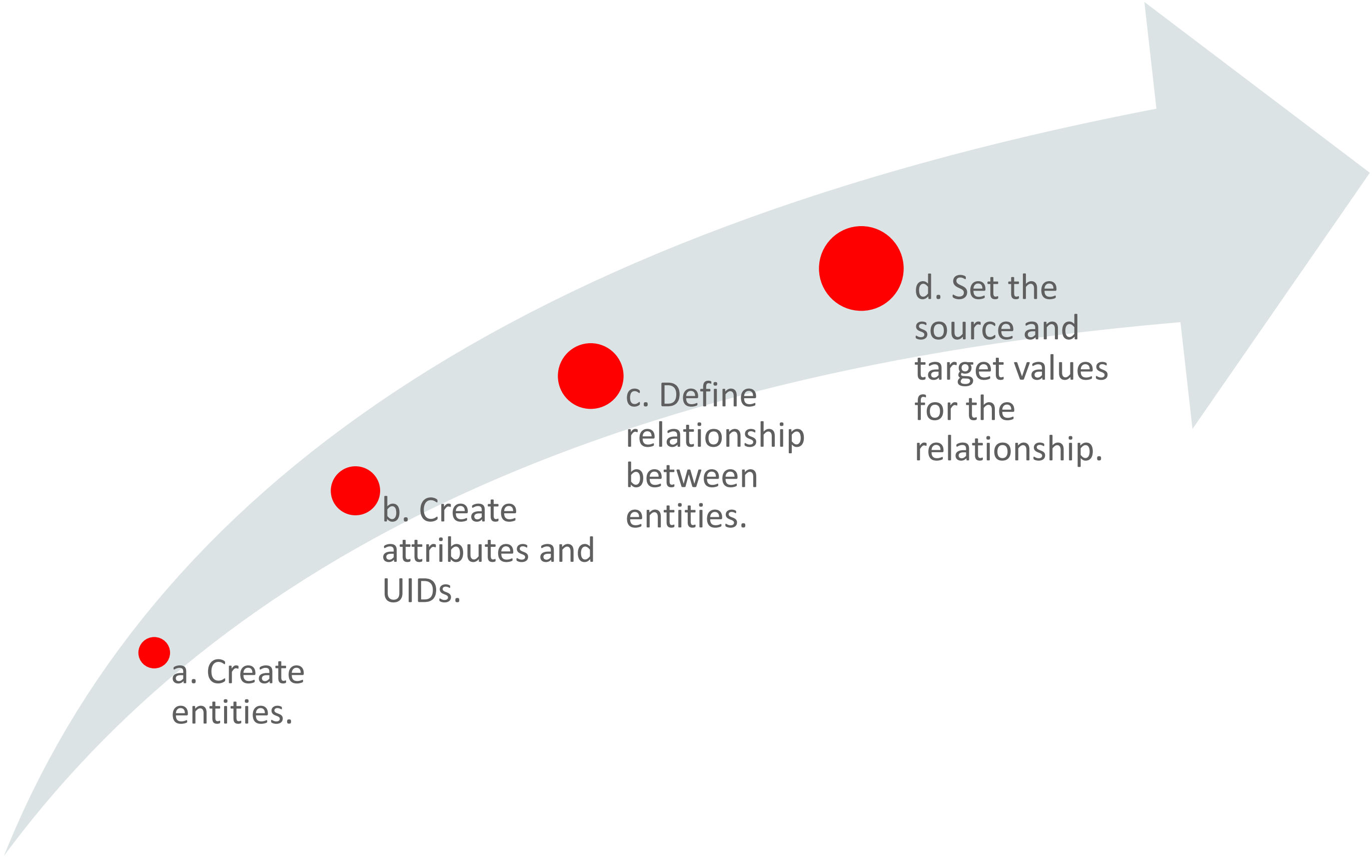
OE.PRODUCT_INFORMATION

| | | |
|---|---|---|
| P * | PRODUCT_ID | NUMBER (6) |
| | PRODUCT_NAME | VARCHAR2 (50 BYTE) |
| | PRODUCT_DESCRIPTION | VARCHAR2 (2000 BYTE) |
| | CATEGORY_ID | NUMBER (2) |
| | WEIGHT_CLASS | NUMBER (1) |
| | WARRANTY_PERIOD | INTERVAL YEAR TO MONTH |
| | SUPPLIER_ID | NUMBER (6) |
| | PRODUCT_STATUS | VARCHAR2 (20 BYTE) |
| | LIST_PRICE | NUMBER (8,2) |
| | MIN_PRICE | NUMBER (8,2) |
| | CATALOG_URL | VARCHAR2 (50 BYTE) |

PRODUCT_INFORMATION_PK (PRODUCT_ID)

PROD_SUPPLIER_IX (SUPPLIER_ID)

# Building an ERD by Using Oracle SQL Developer Data Modeler



a. Create entities.

b. Create attributes and UIDs.

c. Define relationship between entities.

d. Set the source and target values for the relationship.

# Case Scenario: An Introduction

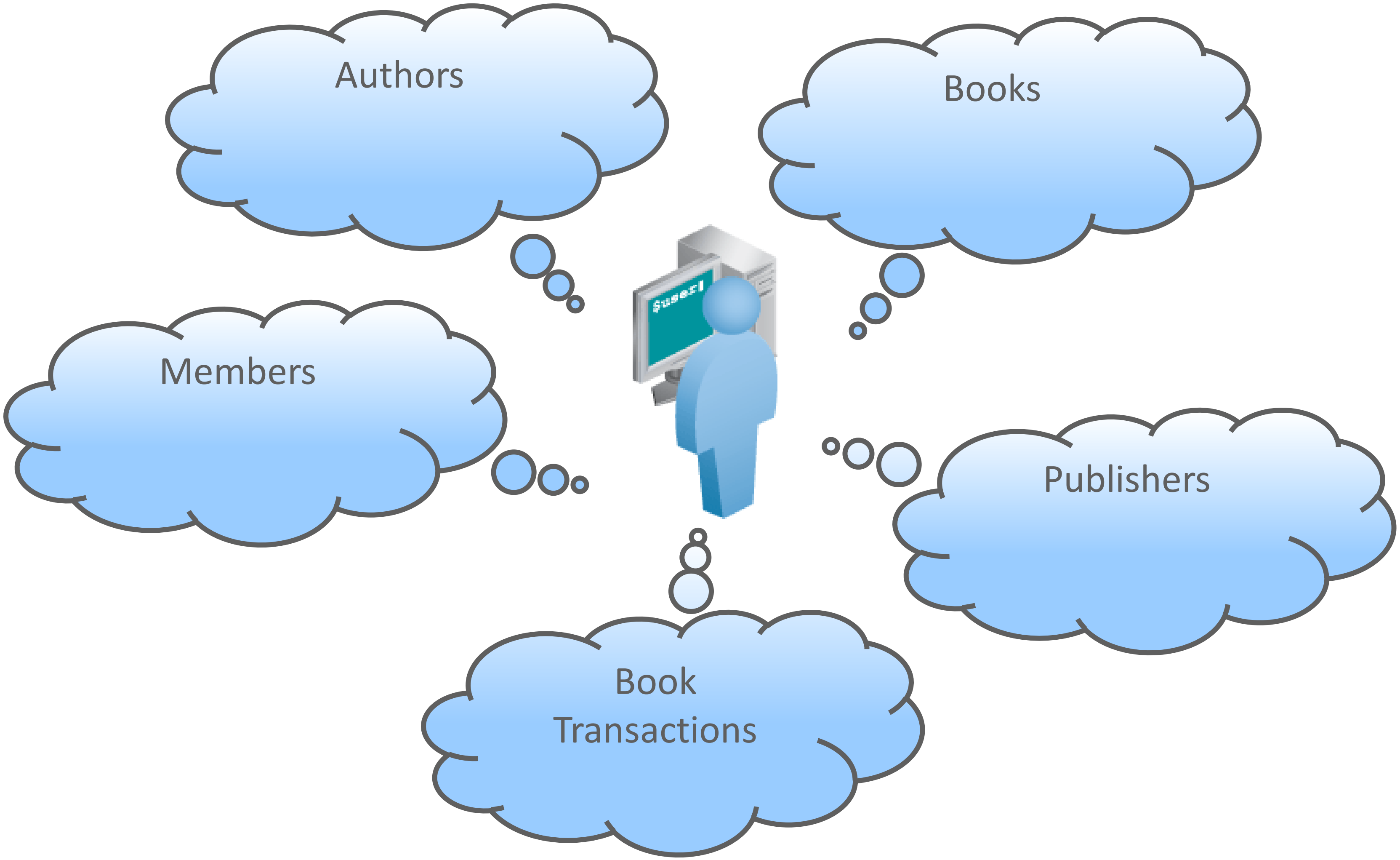
Faculty

Sean, I would like you to create a simplified library database to manage the number of reference books in our department. As a first step, can you build a logical model using Oracle SQL Developer Data Modeler that we have installed in our student machines?

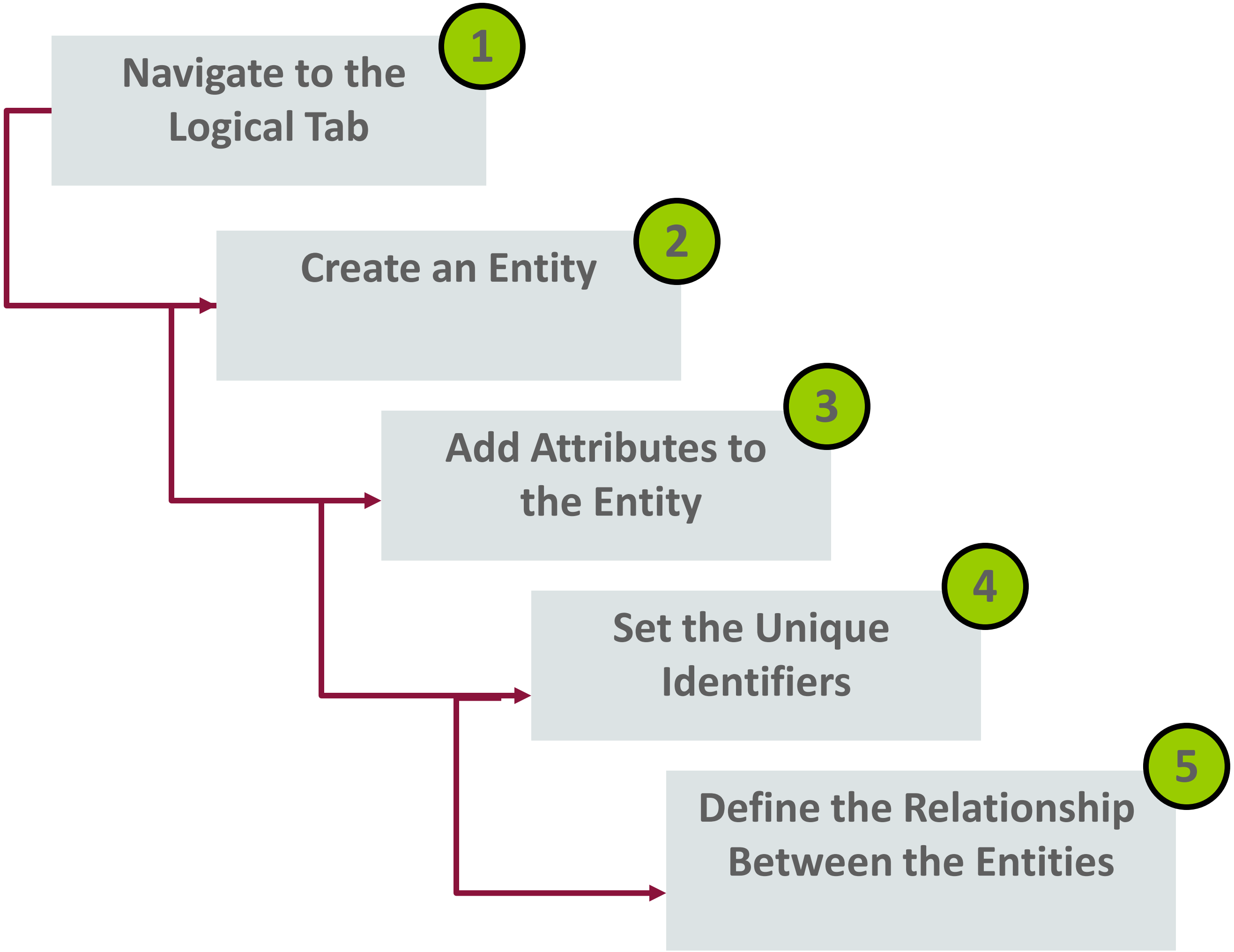
Glad to. I'll start by identifying the entities and their attributes. After that, I can use the Oracle SQL Developer Data Modeler tool to build the logical model.

Student

# Case Scenario: Identifying Entities



# Building an Entity Relationship Diagram



1 Navigate to the Logical Tab

2 Create an Entity

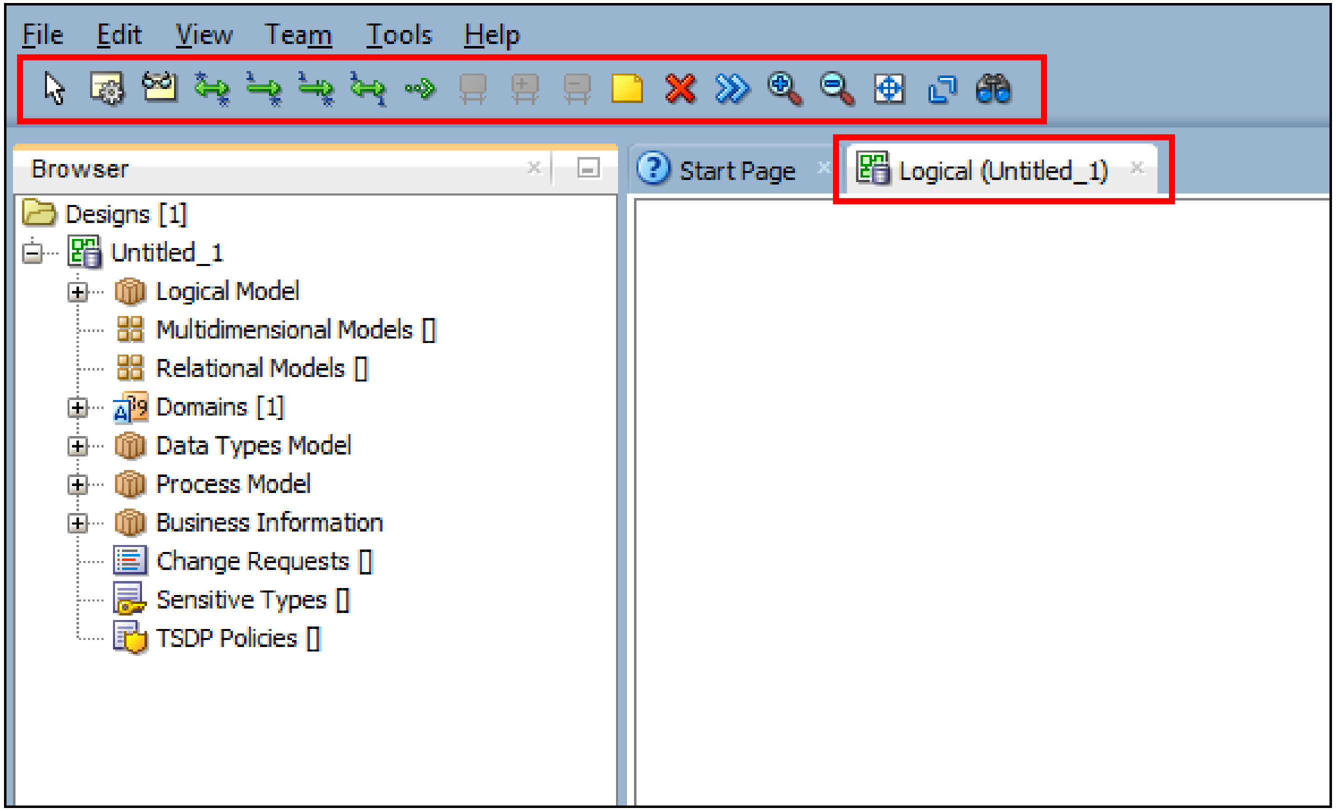
3 Add Attributes to the Entity

4 Set the Unique Identifiers

5 Define the Relationship Between the Entities

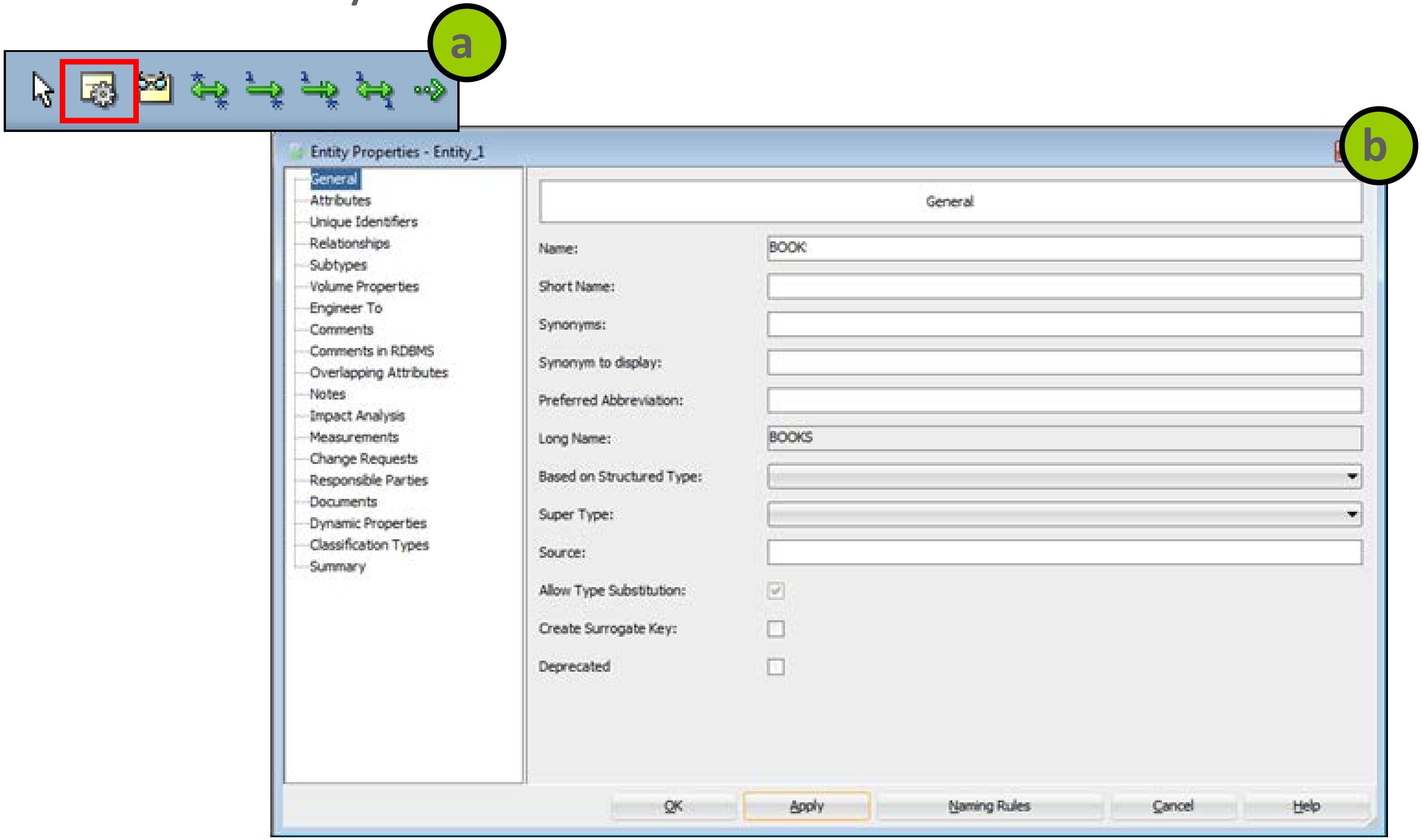
# Building an ERD: Step 1

1. Navigate to the Logical tab.



# Building an ERD: Step 2

2. Create an entity.



Entity Properties - Entity_1

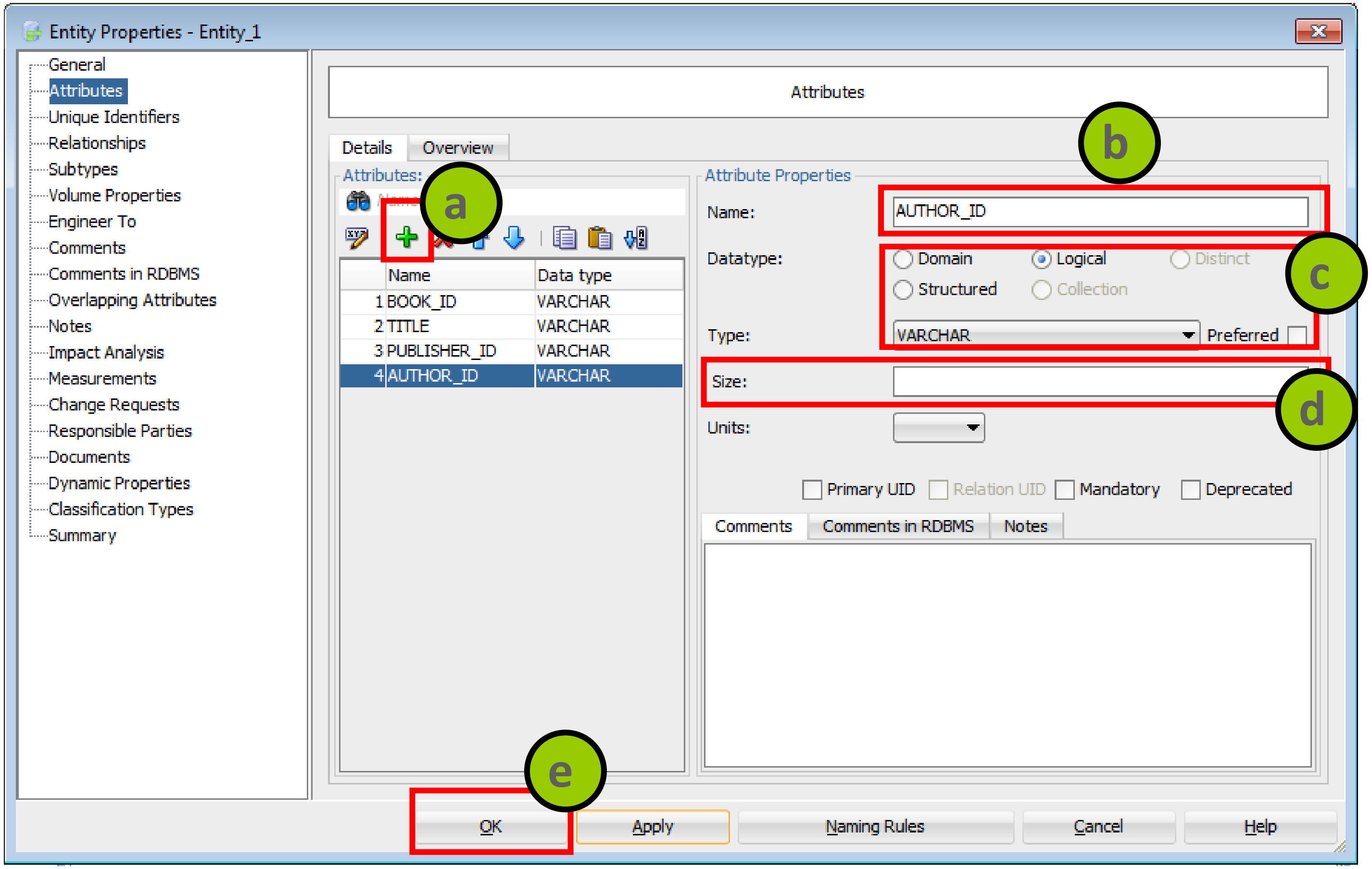
- General
- Attributes
- Unique Identifiers
- Relationships
- Subtypes
- Volume Properties
- Engineer To
- Comments
- Comments in RDBMS
- Overlapping Attributes
- Notes
- Impact Analysis
- Measurements
- Change Requests
- Responsible Parties
- Documents
- Dynamic Properties
- Classification Types
- Summary

General

| | |
|---|---|
| Name: | BOOK |
| Short Name: | |
| Synonyms: | |
| Synonym to display: | |
| Preferred Abbreviation: | |
| Long Name: | BOOKS |
| Based on Structured Type: | |
| Super Type: | |
| Source: | |
| Allow Type Substitution: | ☑ |
| Create Surrogate Key: | □ |
| Deprecated | □ |

# Building an ERD: Step 3

3. Add attributes to the entity.



| | Name | Data type |
|---|---|---|
| 1 | BOOK_ID | VARCHAR |
| 2 | TITLE | VARCHAR |
| 3 | PUBLISHER_ID | VARCHAR |
| 4 | AUTHOR_ID | VARCHAR |

Name: AUTHOR_ID

Datatype: ○ Domain ● Logical ○ Distinct ○ Structured ○ Collection

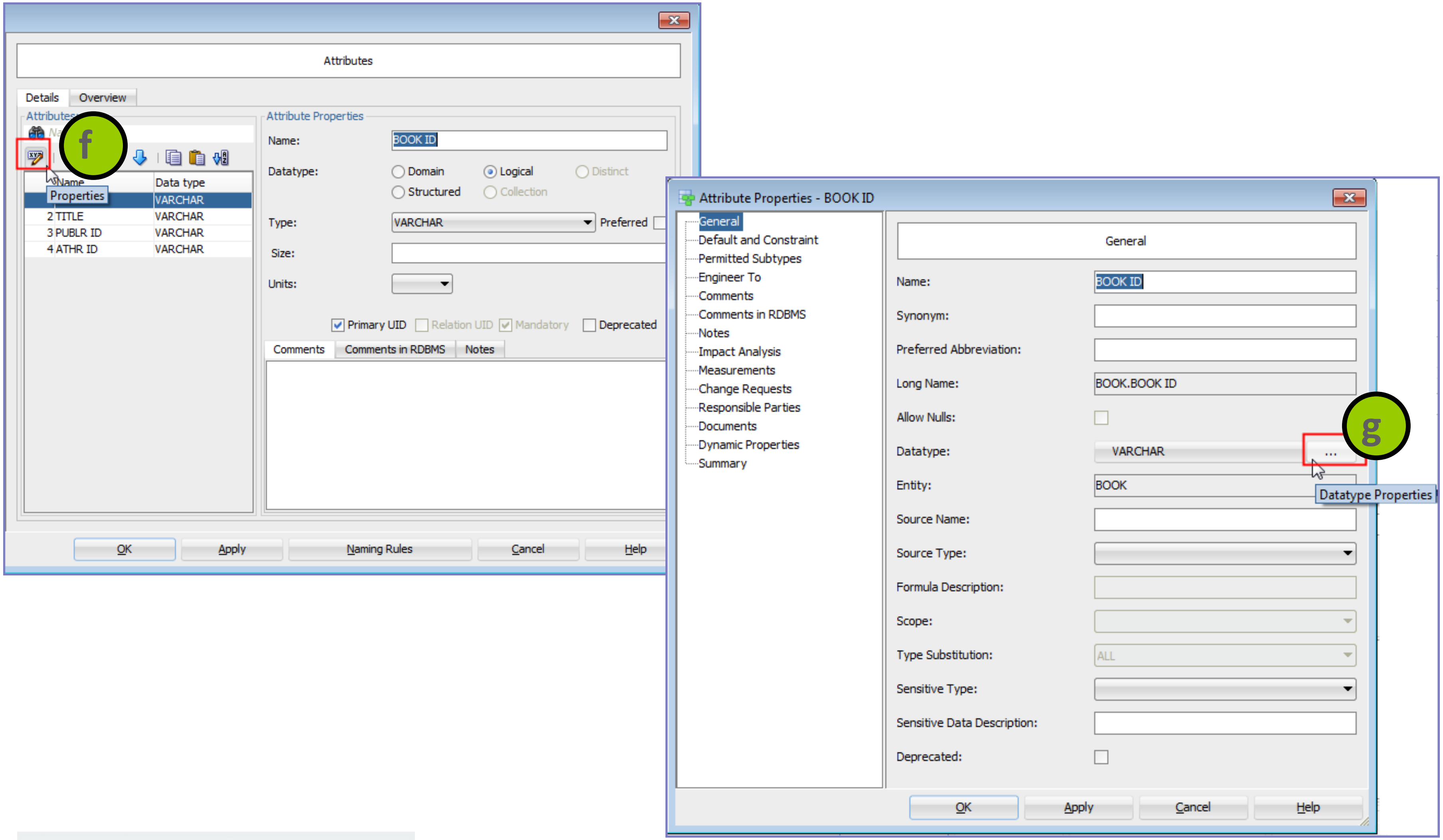
Type: VARCHAR Preferred □

Size:

Units:

□ Primary UID □ Relation UID □ Mandatory □ Deprecated

# Editing the Attribute Properties



Attributes

| Name | Data type |
|---|---|
| | VARCHAR |
| 2 TITLE | VARCHAR |
| 3 PUBLR ID | VARCHAR |
| 4 ATHR ID | VARCHAR |

Attribute Properties

Name: BOOK ID

Datatype: ○ Domain ● Logical ○ Distinct ○ Structured ○ Collection

Type: VARCHAR Preferred □

Size:

Units:

☑ Primary UID □ Relation UID ☑ Mandatory □ Deprecated

Comments | Comments in RDBMS | Notes

Attribute Properties - BOOK ID

- General
- Default and Constraint
- Permitted Subtypes
- Engineer To
- Comments
- Comments in RDBMS
- Notes
- Impact Analysis
- Measurements
- Change Requests
- Responsible Parties
- Documents
- Dynamic Properties
- Summary

General

Name: BOOK ID

Synonym:

Preferred Abbreviation:

Long Name: BOOK.BOOK ID

Allow Nulls: □

Datatype: VARCHAR ...

Entity: BOOK

Source Name:

Source Type:

Formula Description:

Scope:

Type Substitution: ALL

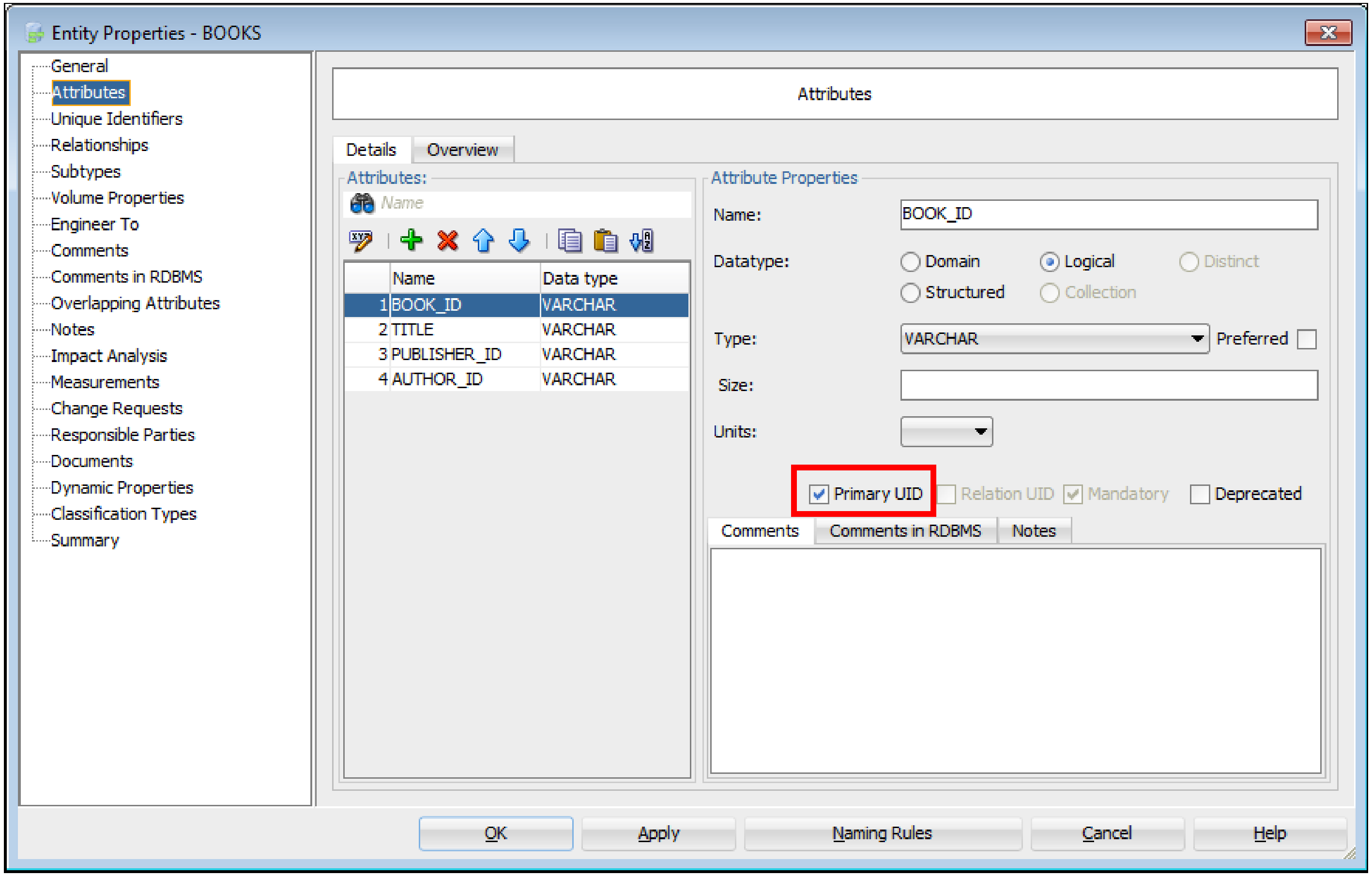
Sensitive Type:

Sensitive Data Description:

Deprecated: □

# Building an ERD: Step 4

4. Set the UID.



|  | Name | Data type |
|---|---|---|
| 1 | BOOK_ID | VARCHAR |
| 2 | TITLE | VARCHAR |
| 3 | PUBLISHER_ID | VARCHAR |
| 4 | AUTHOR_ID | VARCHAR |

Name: BOOK_ID

Datatype: ○ Domain ● Logical ○ Distinct ○ Structured ○ Collection

Type: VARCHAR Preferred □

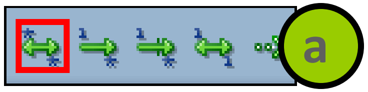
Size:

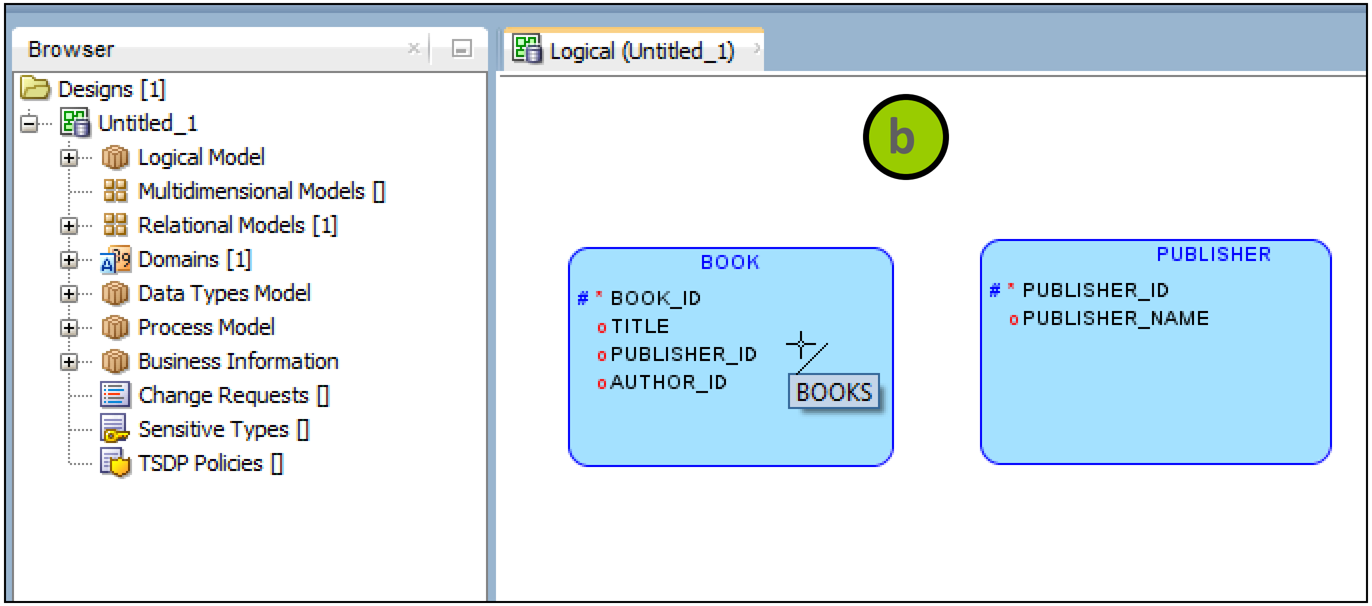
Units:

☑ Primary UID □ Relation UID ☑ Mandatory □ Deprecated

# Building an ERD: Step 5

5. Define the relationships between the entities.





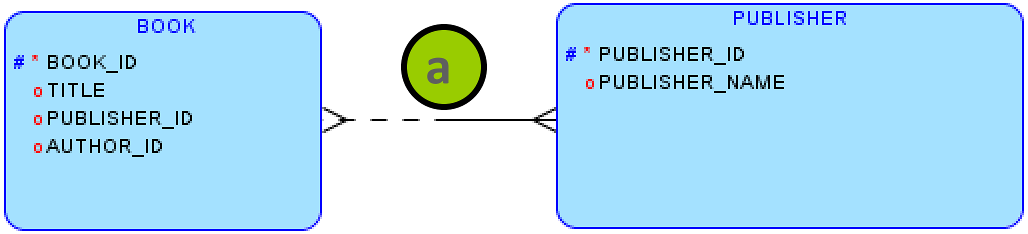
Designs [1]
Untitled_1
Logical Model
Multidimensional Models []
Relational Models [1]
Domains [1]
Data Types Model
Process Model
Business Information
Change Requests []
Sensitive Types []
TSDP Policies []

BOOK
# * BOOK_ID
o TITLE
o PUBLISHER_ID
o AUTHOR_ID

PUBLISHER
# * PUBLISHER_ID
o PUBLISHER_NAME

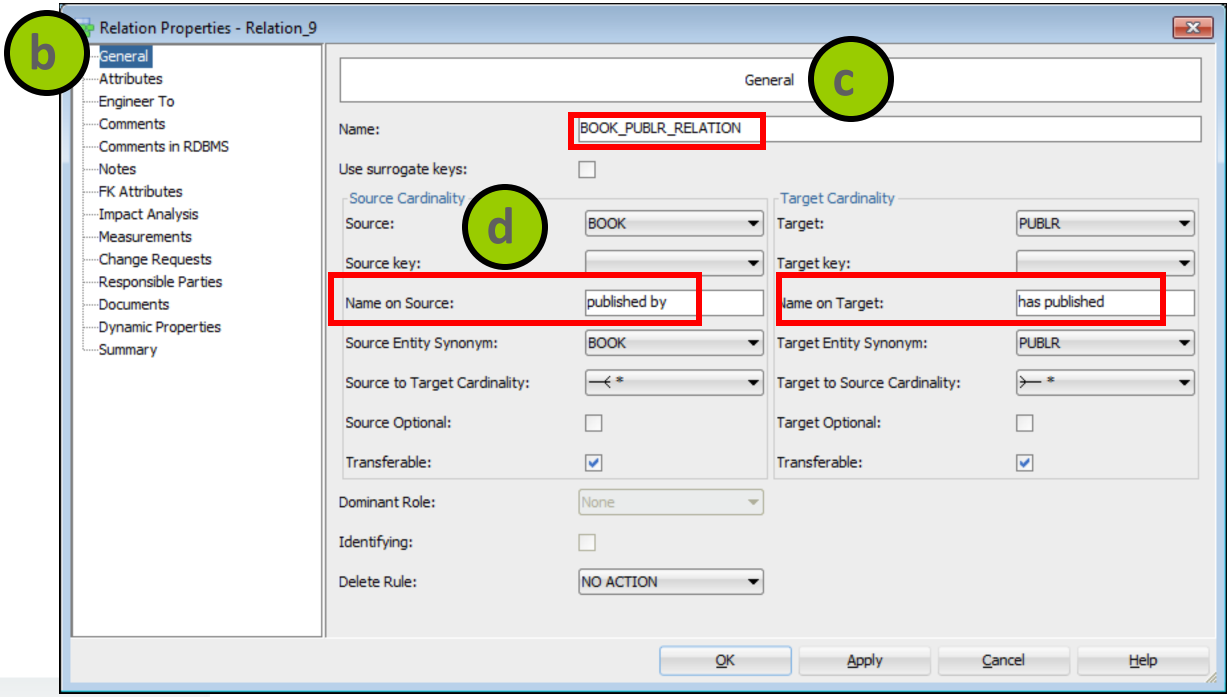
# Building an ERD: Step 6

6. Set the source and target values for the relationship.



BOOK
- # * BOOK_ID
- o TITLE
- o PUBLISHER_ID
- o AUTHOR_ID

PUBLISHER
- # * PUBLISHER_ID
- o PUBLISHER_NAME



Relation Properties - Relation_9

General

Name: BOOK_PUBLR_RELATION

Use surrogate keys: □

| Source Cardinality | | Target Cardinality | |
|---|---|---|---|
| Source: | BOOK | Target: | PUBLR |
| Source key: | | Target key: | |
| Name on Source: | published by | Name on Target: | has published |
| Source Entity Synonym: | BOOK | Target Entity Synonym: | PUBLR |
| Source to Target Cardinality: | ⪪ * | Target to Source Cardinality: | ⪫ * |
| Source Optional: | □ | Target Optional: | □ |
| Transferable: | ☑ | Transferable: | ☑ |

Dominant Role: None

Identifying: □

Delete Rule: NO ACTION

# Case Scenario: Entity Types

Faculty

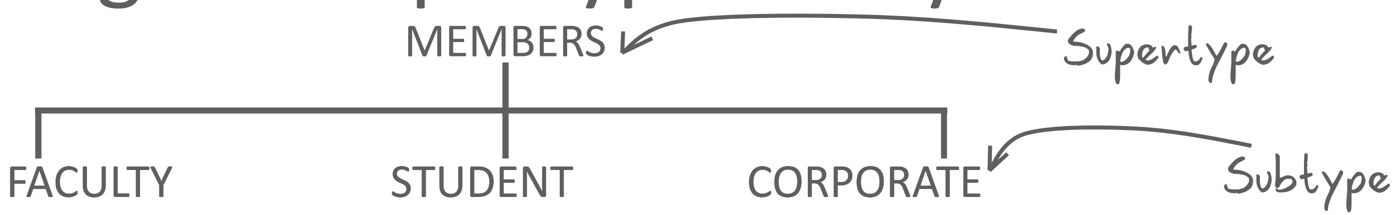
Sean, I was wondering if we could include new types of membership categories such as:

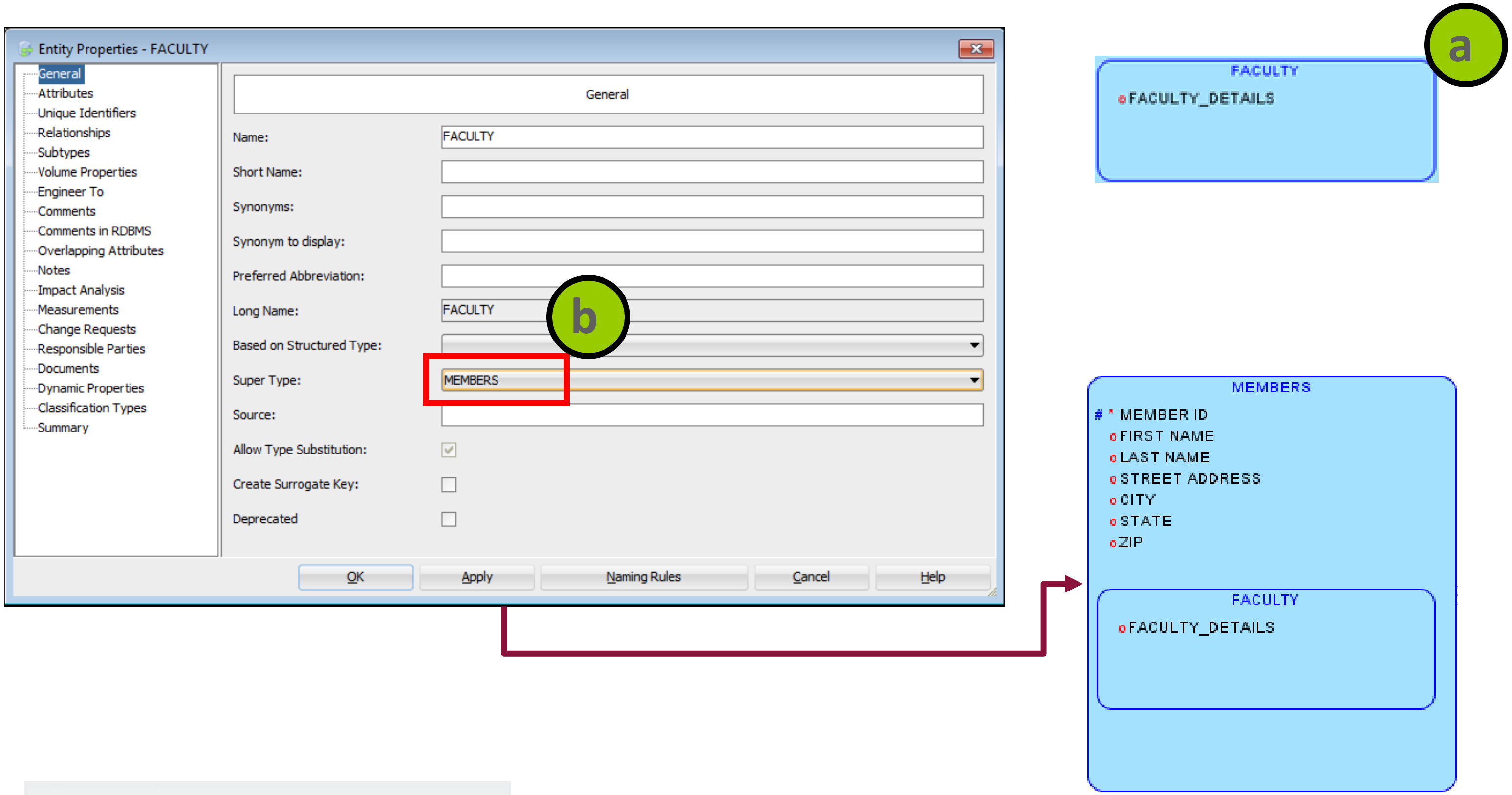
- Student Membership
- Faculty Membership
- Corporate Membership

This can definitely be achieved. I can create a common entity that would hold membership details that are common to all the three membership categories. This would be a **supertype** entity. The specific membership categories would inherit the properties of the supertype entity, in addition to their own specific attributes. Hence, the specific membership category would be a **subtype** entity.

Student

# Creating the Supertype Entity





Entity Properties - FACULTY

General

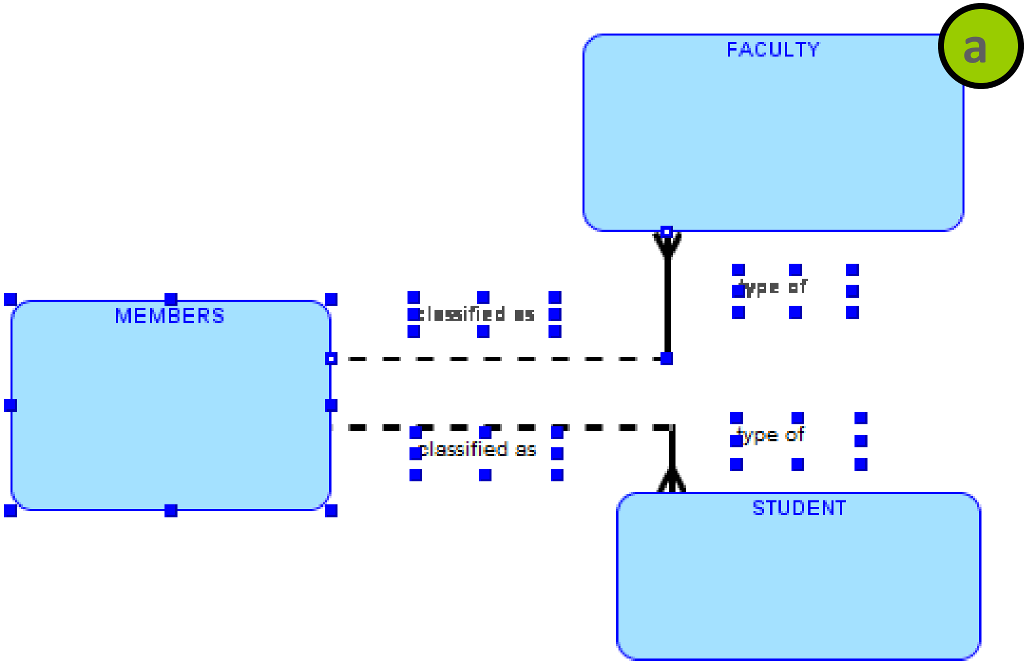
| | |
|---|---|
| Name: | FACULTY |
| Short Name: | |
| Synonyms: | |
| Synonym to display: | |
| Preferred Abbreviation: | |
| Long Name: | FACULTY |
| Based on Structured Type: | |
| Super Type: | MEMBERS |
| Source: | |
| Allow Type Substitution: | ☑ |
| Create Surrogate Key: | □ |
| Deprecated | □ |

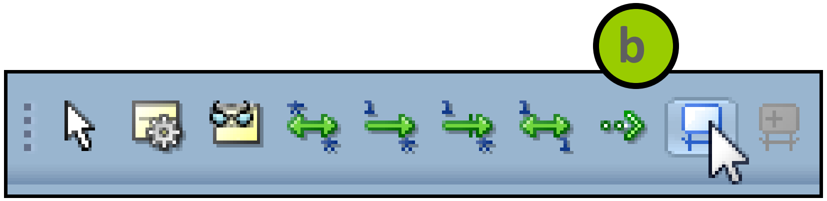
FACULTY
o FACULTY_DETAILS

MEMBERS
# * MEMBER ID
o FIRST NAME
o LAST NAME
o STREET ADDRESS
o CITY
o STATE
o ZIP

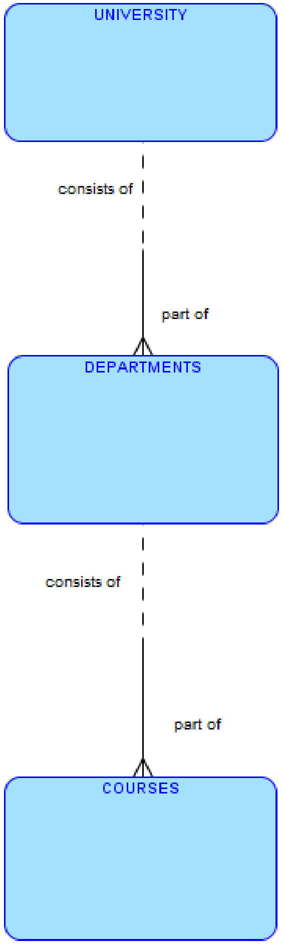
FACULTY
o FACULTY_DETAILS

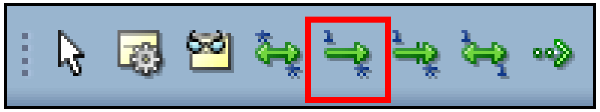
# Creating the Arc Relationship



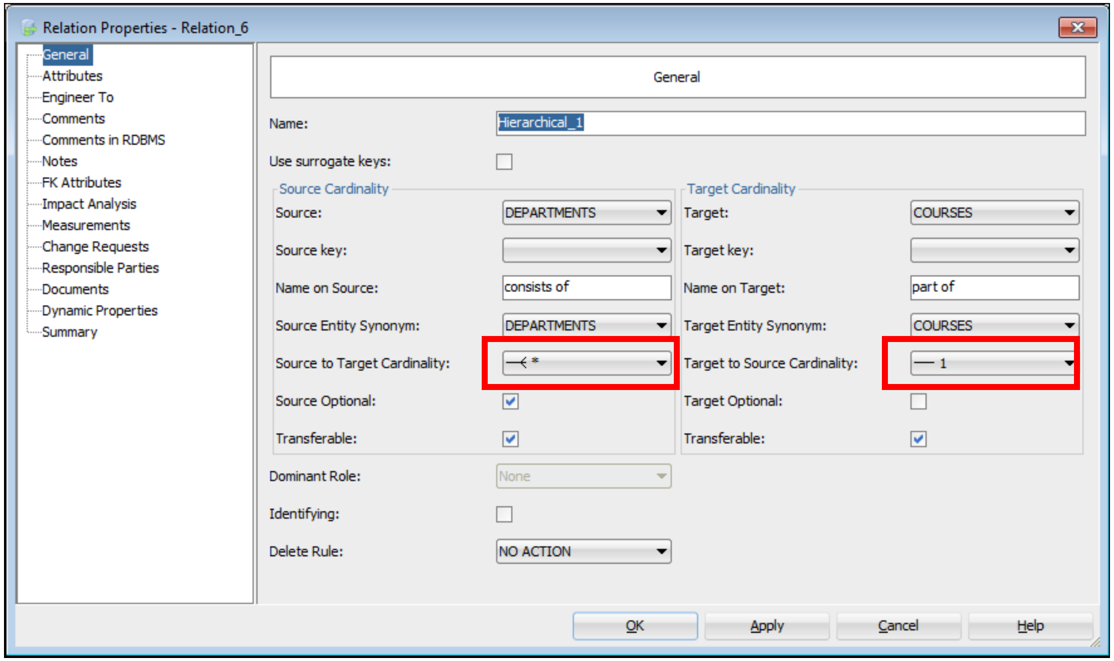


# Creating the Hierarchical Relationship





1:N Relationship



Relation Properties - Relation_6

- General
- Attributes
- Engineer To
- Comments
- Comments in RDBMS
- Notes
- FK Attributes
- Impact Analysis
- Measurements
- Change Requests
- Responsible Parties
- Documents
- Dynamic Properties
- Summary

General

Name: Hierarchical_1

Use surrogate keys: □

Source Cardinality

Source: DEPARTMENTS

Source key:

Name on Source: consists of

Source Entity Synonym: DEPARTMENTS

Source to Target Cardinality: —< *

Source Optional: ☑

Transferable: ☑

Target Cardinality

Target: COURSES

Target key:

Name on Target: part of

Target Entity Synonym: COURSES

Target to Source Cardinality: — 1

Target Optional: □

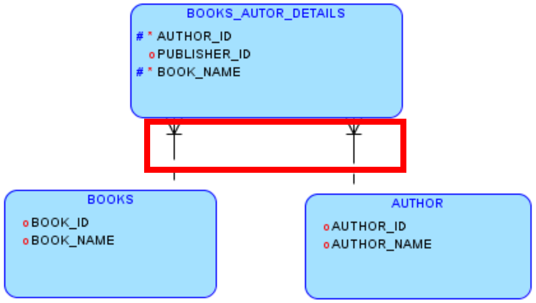
Transferable: ☑

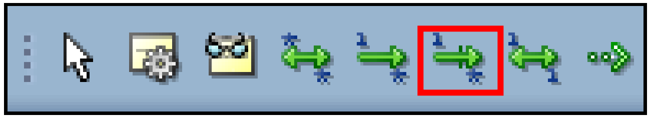
Dominant Role: None

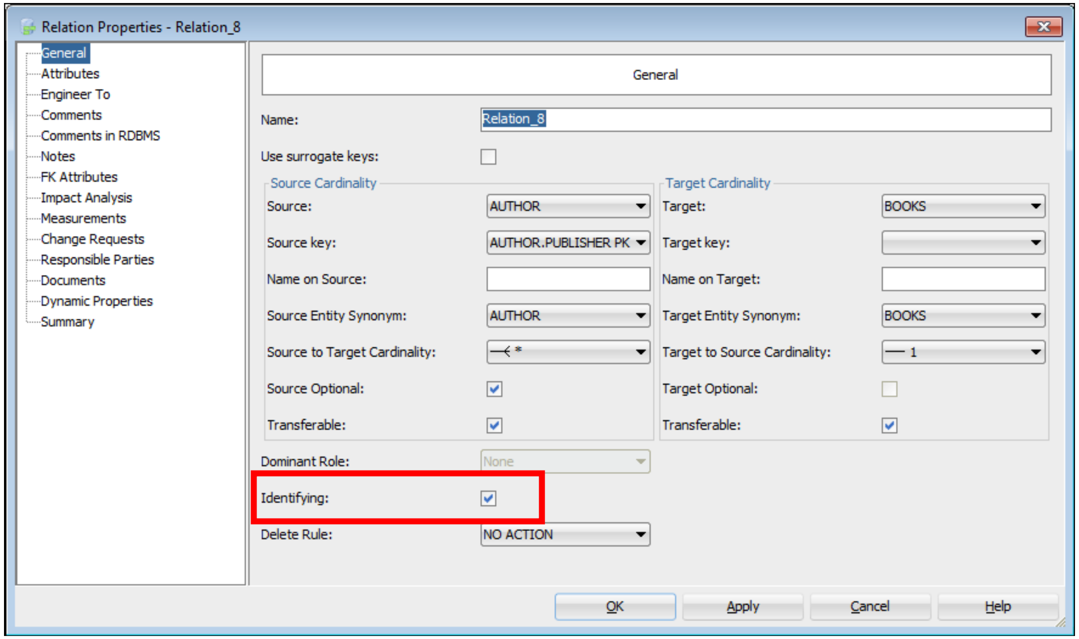
Identifying: □

Delete Rule: NO ACTION

# Creating the Barred Relationship







General

| | |
|---|---|
| Name: | Relation_8 |
| Use surrogate keys: | □ |

Source Cardinality

| | |
|---|---|
| Source: | AUTHOR |
| Source key: | AUTHOR.PUBLISHER PK |
| Name on Source: | |
| Source Entity Synonym: | AUTHOR |
| Source to Target Cardinality: | —< * |
| Source Optional: | ☑ |
| Transferable: | ☑ |

Target Cardinality

| | |
|---|---|
| Target: | BOOKS |
| Target key: | |
| Name on Target: | |
| Target Entity Synonym: | BOOKS |
| Target to Source Cardinality: | — 1 |
| Target Optional: | □ |
| Transferable: | ☑ |

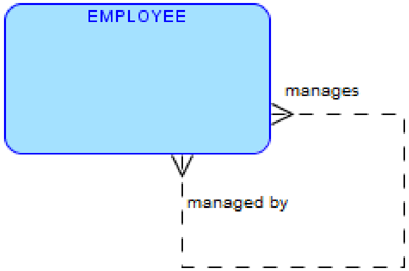
| | |
|---|---|
| Dominant Role: | None |
| Identifying: | ☑ |
| Delete Rule: | NO ACTION |

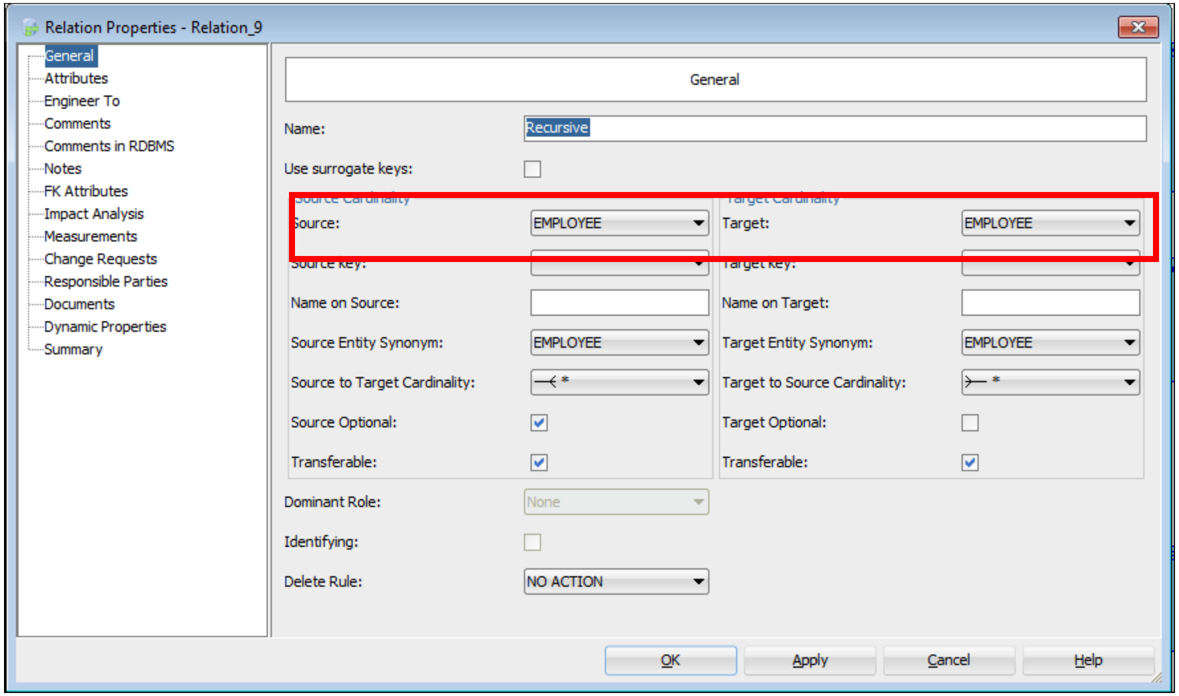
Identifying Relationship

# Creating the Recursive Relationship

In a relationship, if the same entity participates more than once, it is termed a recursive relationship.

Recursive Relationship




Relation Properties - Relation_9

General

Name: Recursive

Use surrogate keys: □

Source Cardinality

Source: EMPLOYEE

Source key:

Name on Source:

Source Entity Synonym: EMPLOYEE

Source to Target Cardinality: —< *

Source Optional: ☑

Transferable: ☑

Target Cardinality

Target: EMPLOYEE

Target key:

Name on Target:

Target Entity Synonym: EMPLOYEE

Target to Source Cardinality: >— *

Target Optional: □

Transferable: ☑

Dominant Role: None

Identifying: □

Delete Rule: NO ACTION

# Summary

In this lesson, you should have learned how to:

- Use Oracle SQL Developer Data Modeler to create :
  - Entities, attributes, and UIDs with correct optionality and cardinality
  - Supertype and subtype entities
  - Arc, hierarchical, barred, and recursive relationships

ORACLE® ACADEMY