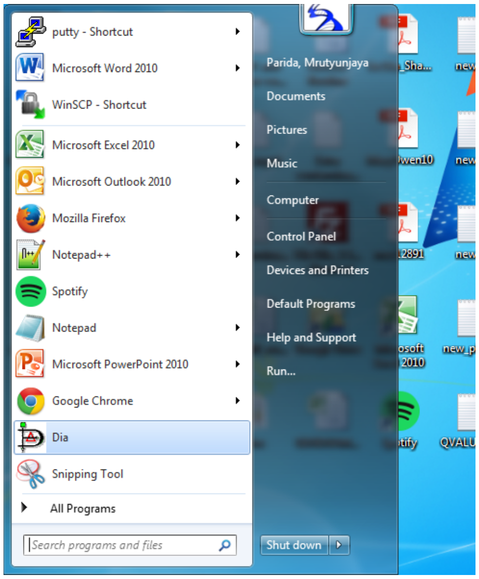
Database modelling tutorial 2

Mrutyunjaya “Rocky” Parida

In this tutorial we will be practicing how to solve a variety of database modelling problems using Dia diagram editor.

Please open Dia on your computer as follows:



Please click on the Drop-down menu below and then click on “Other sheets” option and select Database.

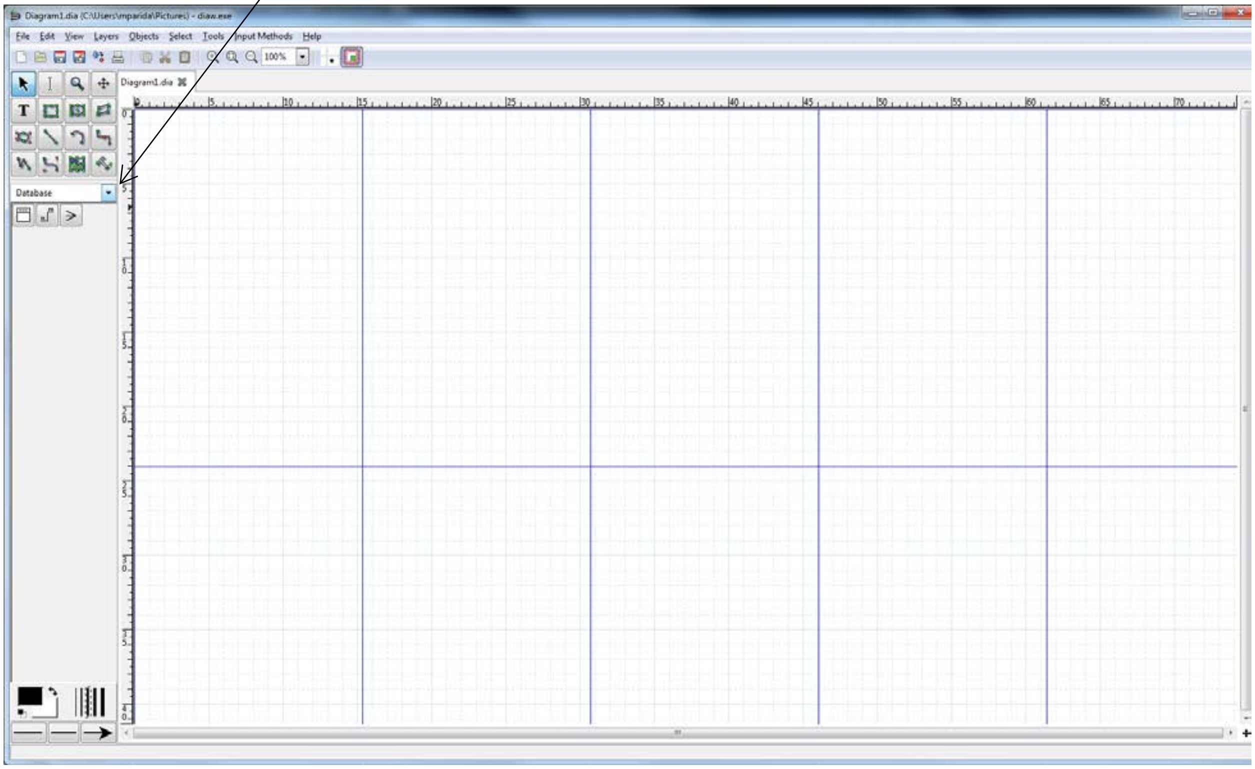
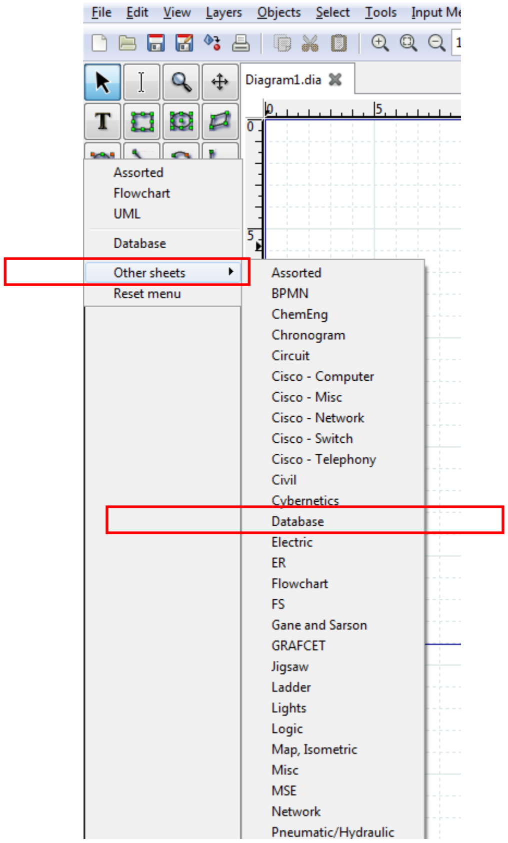
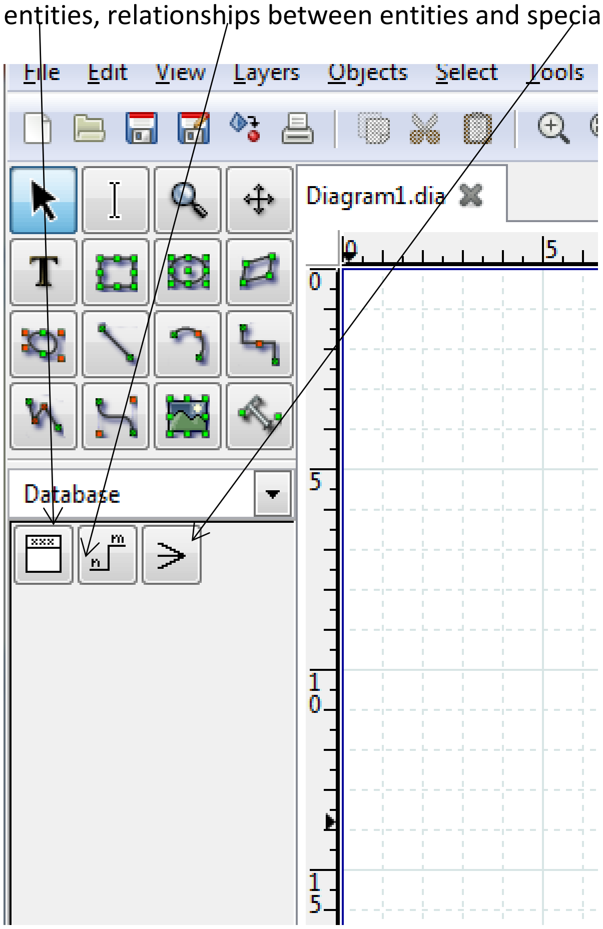


Diagram1.dia (C:\Users\mparida\Pictures) - diaw.exe
File Edit View Layers Objects Select Tools Input Methods Help
100%
Diagram1.dia
Database

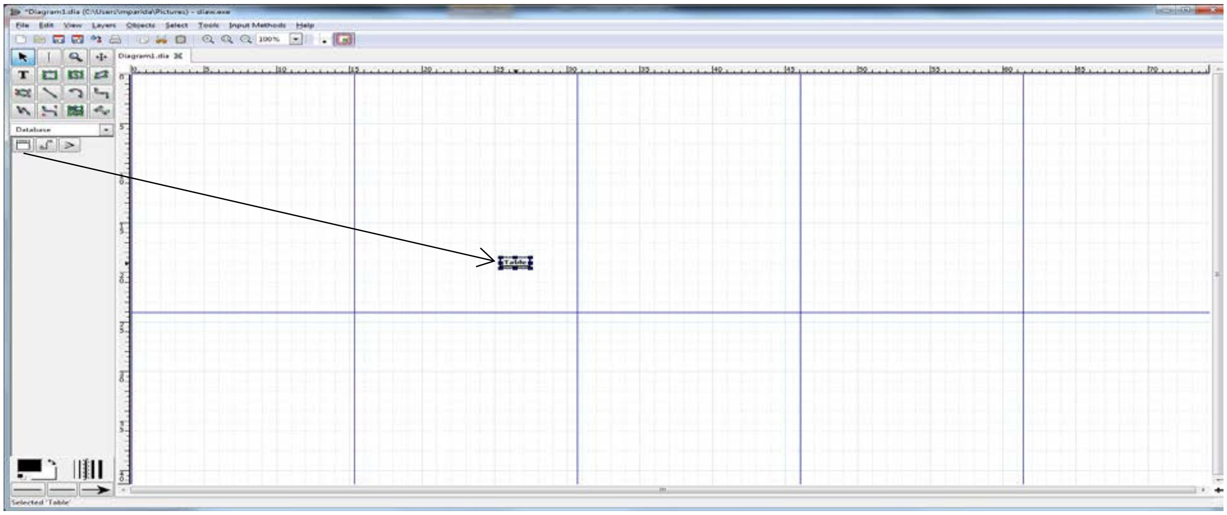


Following is the stencil you may use for database modelling purposes such as drawing entities, relationships between entities and specializations.



Problem 1: Unary relationship

An employee supervises other employees and an employee may be supervised by one supervisor only. To draw this please select entity/Table icon as shown before and drag it to your worksheet as follows. Next, fill in the details for this entity such as Employee id (identifier), Employee name and Supervisor ID.



Please follow Lab2 tutorial on how to create an entity in Dia. Next, please select the relationship/reference icon to add a relationship between employee and supervisors.

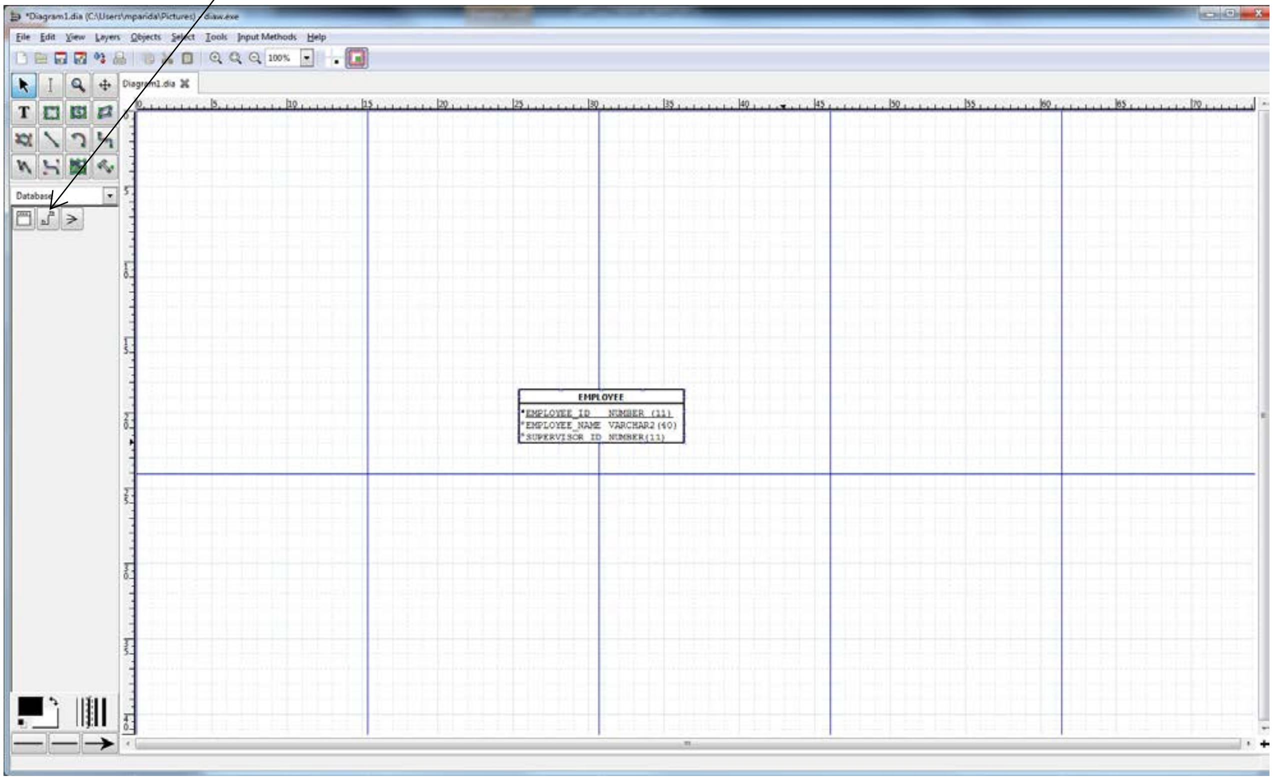
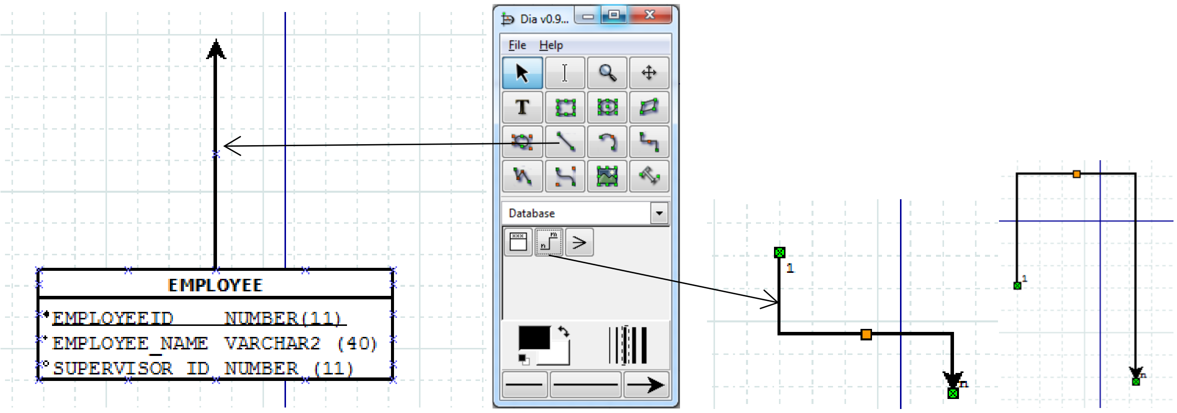


*Diagram1.dia (C:\Users\mparida\Pictures) - diaw.exe
File Edit View Layers Objects Select Tools Input Methods Help
100%
Diagram1.dia
Database

EMPLOYEE

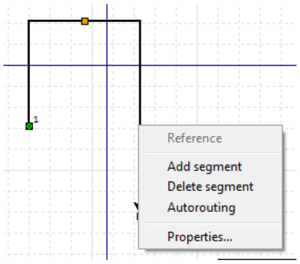
*EMPLOYEE_ID	NUMBER (11)
*EMPLOYEE_NAME	VARCHAR2 (40)
*SUPERVISOR_ID	NUMBER(11)

Please add a line to the EMPLOYEE entity and next play with the reference line to make it look as follows:



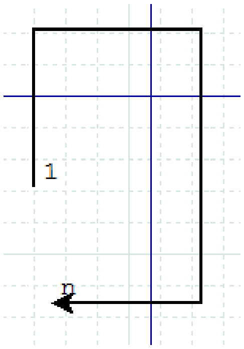
EMPLOYEE

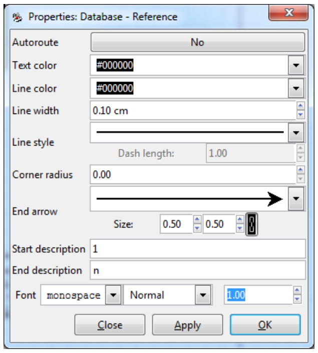
*EMPLOYEEID	NUMBER(11)
*EMPLOYEE_NAME	VARCHAR2 (40)
°SUPERVISOR_ID	NUMBER (11)



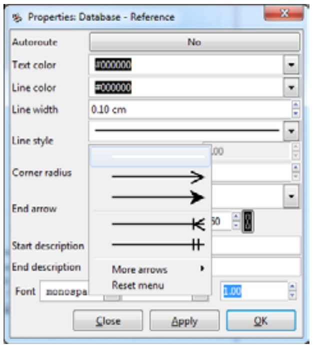
Please do a right click on the reference line to add a segment to the line.

Finally, you can get a shape as follows:

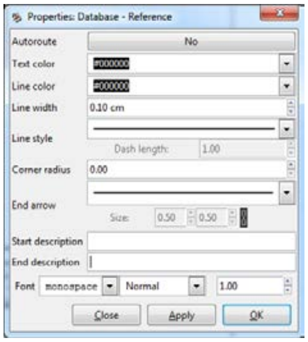



Properties: Database - Reference

Autoroute: No
Text color: #000000
Line color: #000000
Line width: 0.10 cm
Line style
Dash length: 1.00
Corner radius: 0.00
End arrow
Size: 0.50 0.50
Start description: 1
End description: n
Font: monospace Normal 1.00

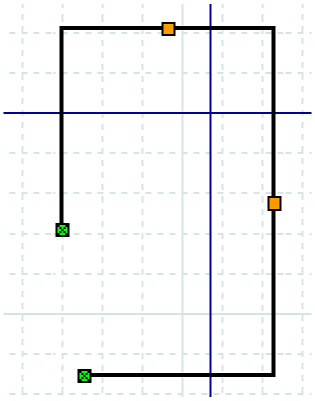

Properties: Database - Reference

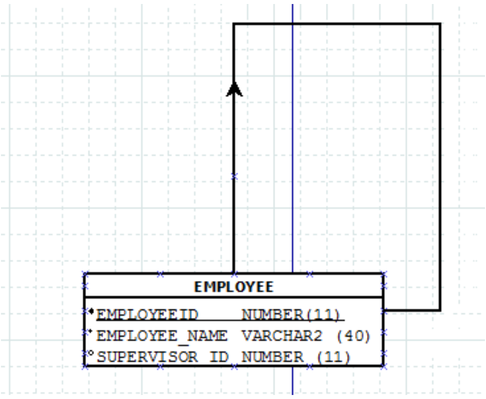
Autoroute: No
Text color: #000000
Line color: #000000
Line width: 0.10 cm
Line style
Corner radius
End arrow
Start description
End description
Font: monospa


Properties: Database - Reference

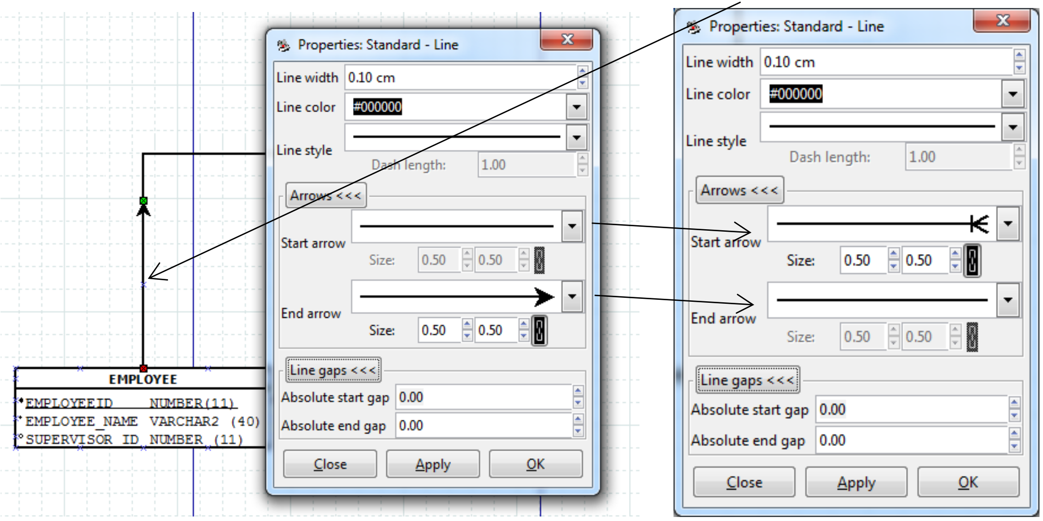
Autoroute: No
Text color: #000000
Line color: #000000
Line width: 0.10 cm
Line style
Dash length: 1.00
Corner radius: 0.00
End arrow
Size: 0.50 0.50
Start description
End description
Font: monospace Normal 1.00

Please access the properties of this reference line by double clicking on it. Please select the line without arrowhead under “End arrow”, clear the Start and End descriptions and then click Apply.





Please join this reference line to the EMPLOYEE entity figure by draggin them closer and adjusting the fit by playing with the lines. Please change the properties of the line to one or || and then click Apply.



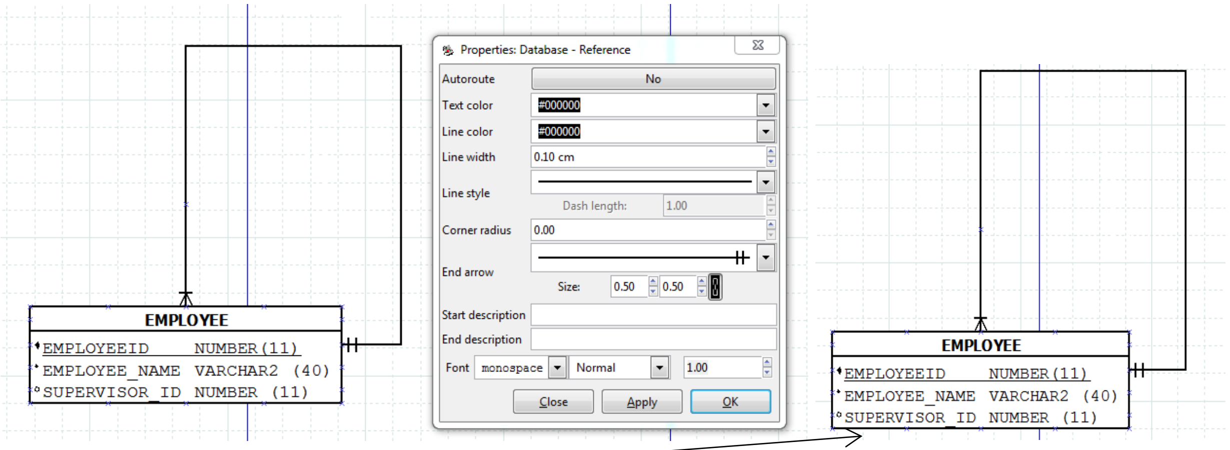
Properties: Standard - Line

Line width: 0.10 cm
Line color: #000000
Line style
Dash length: 1.00
Arrows <<<
Start arrow
Size: 0.50 0.50
End arrow
Size: 0.50 0.50
Line gaps <<<
Absolute start gap: 0.00
Absolute end gap: 0.00

Properties: Standard - Line

Line width: 0.10 cm
Line color: #000000
Line style
Dash length: 1.00
Arrows <<<
Start arrow
Size: 0.50 0.50
End arrow
Size: 0.50 0.50
Line gaps <<<
Absolute start gap: 0.00
Absolute end gap: 0.00

Next, please double click on the reference line to change the End Arrow as follows:

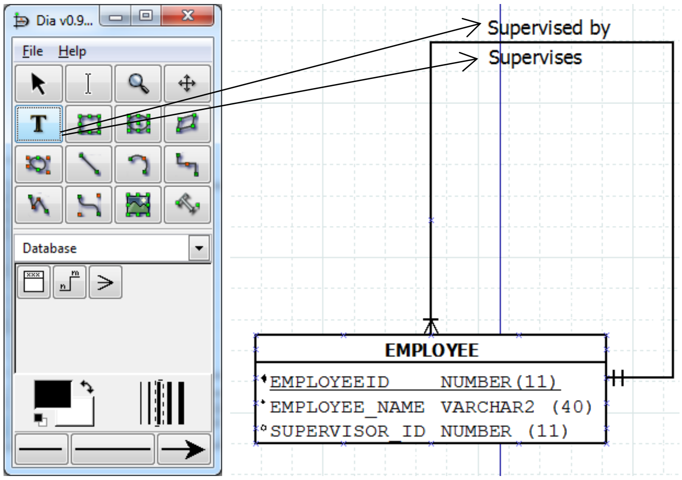

EMPLOYEE
EMPLOYEEID NUMBER(11)
EMPLOYEE_NAME VARCHAR2 (40)
SUPERVISOR_ID NUMBER (11)
EMPLOYEE
EMPLOYEEID NUMBER(11)
EMPLOYEE_NAME VARCHAR2 (40)
SUPERVISOR_ID NUMBER (11)

Properties: Database - Reference

Autoroute: No
Text color: #000000
Line color: #000000
Line width: 0.10 cm
Line style
Dash length: 1.00
Corner radius: 0.00
End arrow
Size: 0.50 0.50
Start description
End description
Font: monospace Normal 1.00

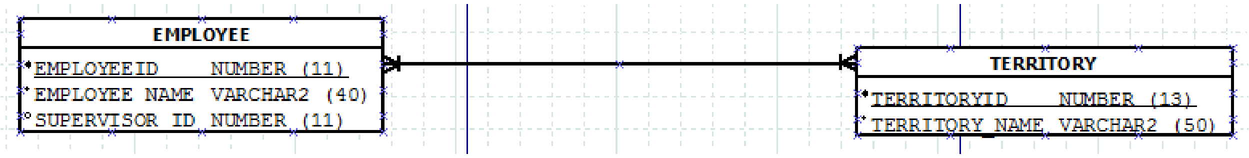
Close Apply OK

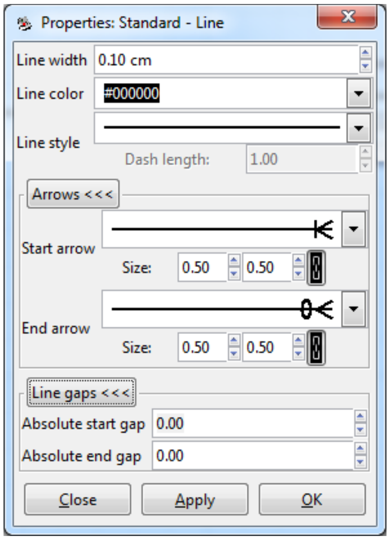
This is the completed unary ER with a many to one relationship between EmployeeID and SupervisorID. Please add relationship names via the Text button for all relationships in an ER diagram.



Problem 2: Binary relationship (many to many)

A Customer does business in one or more territories but a territory should have atleast one customer.


EMPLOYEE
EMPLOYEEID NUMBER (11)
EMPLOYEE NAME VARCHAR2 (40)
SUPERVISOR ID NUMBER (11)
TERRITORY
TERRITORYID NUMBER (13)
TERRITORY_NAME VARCHAR2 (50)


Properties: Standard - Line

Line width: 0.10 cm

Line color: #000000

Line style

Dash length: 1.00

Arrows <<<

Start arrow — Size: 0.50 0.50

End arrow — Size: 0.50 0.50

Line gaps <<<

Absolute start gap: 0.00

Absolute end gap: 0.00

Since this is a many to many relationship it needs an Associative Entity (AE) to keep track of who is doing business in what territory. Please add an AE as follows and change the relationship from the AE to other entities as many to one. An associative entity requires an identifier, in this case, it will be a combination of customer_id and territory_id because a single attribute is not unique enough for this entity/table.

Please check on “Primary key” after adding a new attribute for this entity to create a whole primary key.

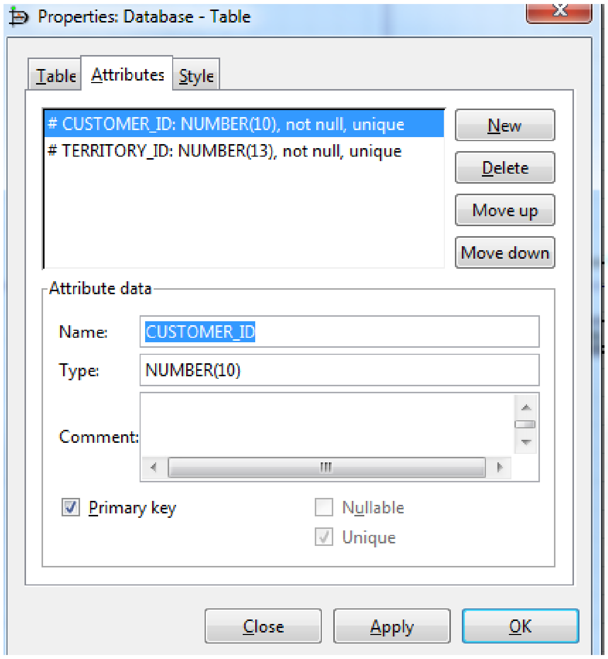

Properties: Database - Table

Table | Attributes | Style

CUSTOMER_ID: NUMBER(10), not null, unique
TERRITORY_ID: NUMBER(13), not null, unique

Attribute data

Name: CUSTOMER_ID

Type: NUMBER(10)

Comment:

☑ Primary key ☐ Nullable ☑ Unique

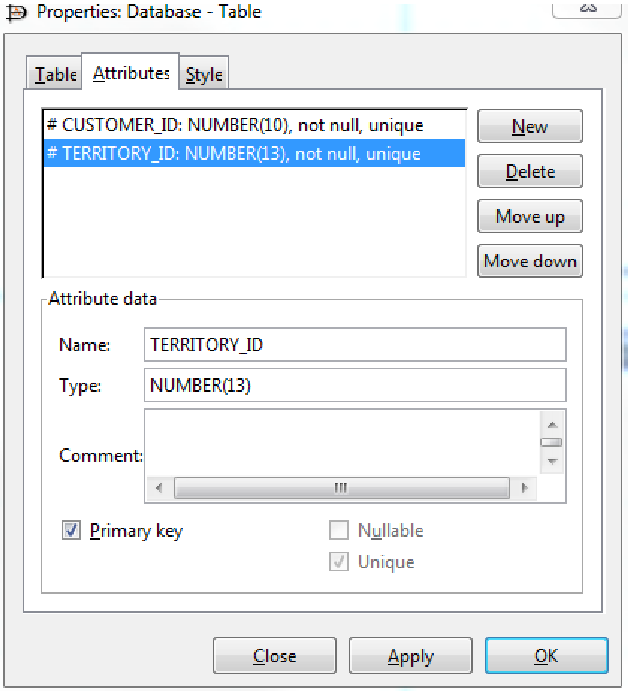

Properties: Database - Table

Table | Attributes | Style

CUSTOMER_ID: NUMBER(10), not null, unique
TERRITORY_ID: NUMBER(13), not null, unique

Attribute data

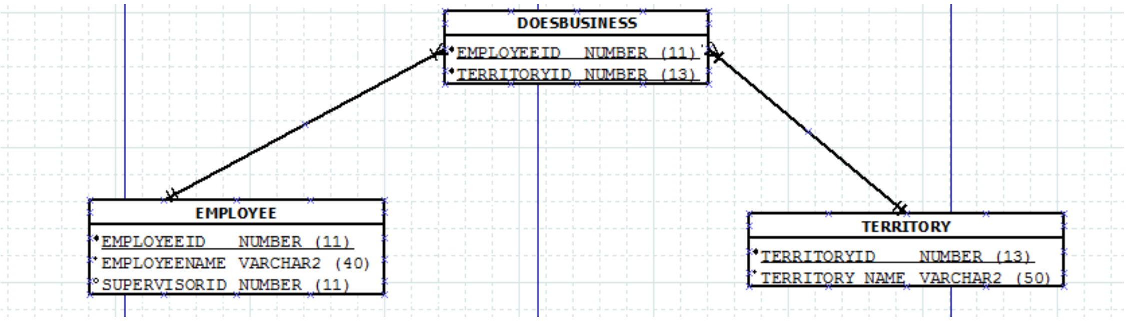
Name: TERRITORY_ID

Type: NUMBER(13)

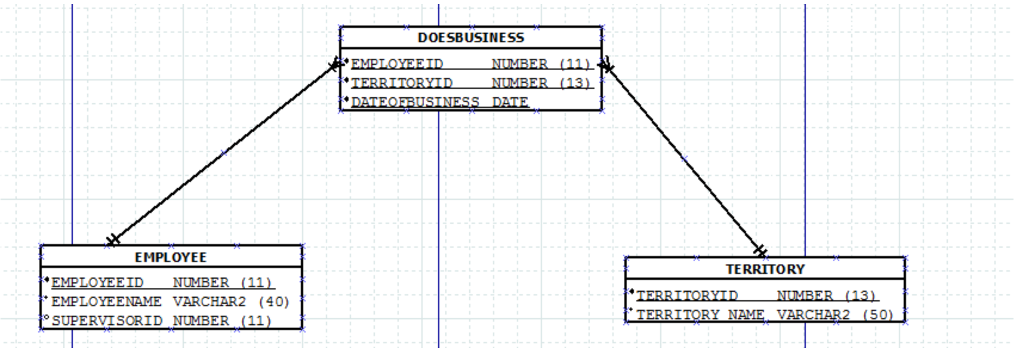
Comment:

☑ Primary key ☐ Nullable ☑ Unique

Please use this example to create whole primary keys for other problems. Please add the appropriate many to one relationships as follows and notice how the cardinalities are placed compared to the many to many relationship before:

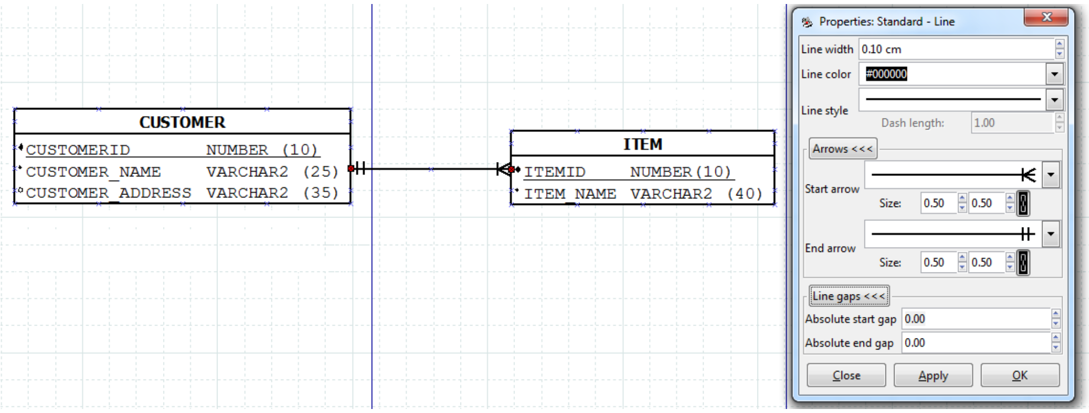


If we want to **keep track of** when the business was done by a customer at a territory, we need to add another attribute to the associated entity called Date of business as follows.



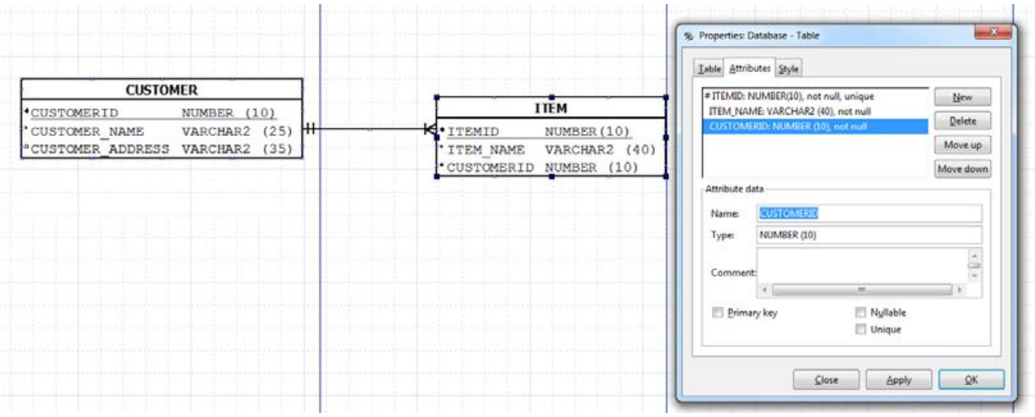
Problem 3: Binary relationship (1 to many)

A customer may order an item or more than one item but each item must be ordered by one and only one customer. Company wants to keep track of who is buying what.



Line width 0.10 cm
Line color #000000
Line style
Dash length: 1.00
Start arrow
Size: 0.50 0.50
End arrow
Size: 0.50 0.50
Absolute start gap 0.00
Absolute end gap 0.00

In a one to many relationship, primary key of the one side is copied for reference to the many side as follows:



ITEMID: NUMBER(10), not null, unique
ITEM_NAME: VARCHAR2 (40), not null
CUSTOMERID: NUMBER (10), not null

Attribute data

Name: CUSTOMERID

Type: NUMBER (10)

Comment:

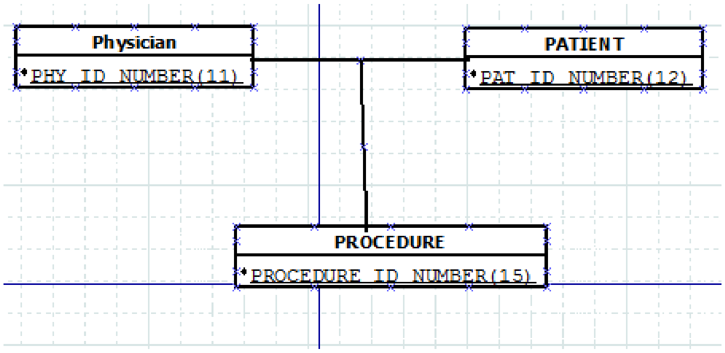
□ Primary key □ Nullable

□ Unique

Now your Binary relationships are complete.

Problem 4: Ternary relationship (many to many)

Ternary relationships are 3 entities participating in one relationship. A patient can be treated by more than one physician, a physician can treat more than one patient, and one procedure can be applied by more than one physician to many patients.

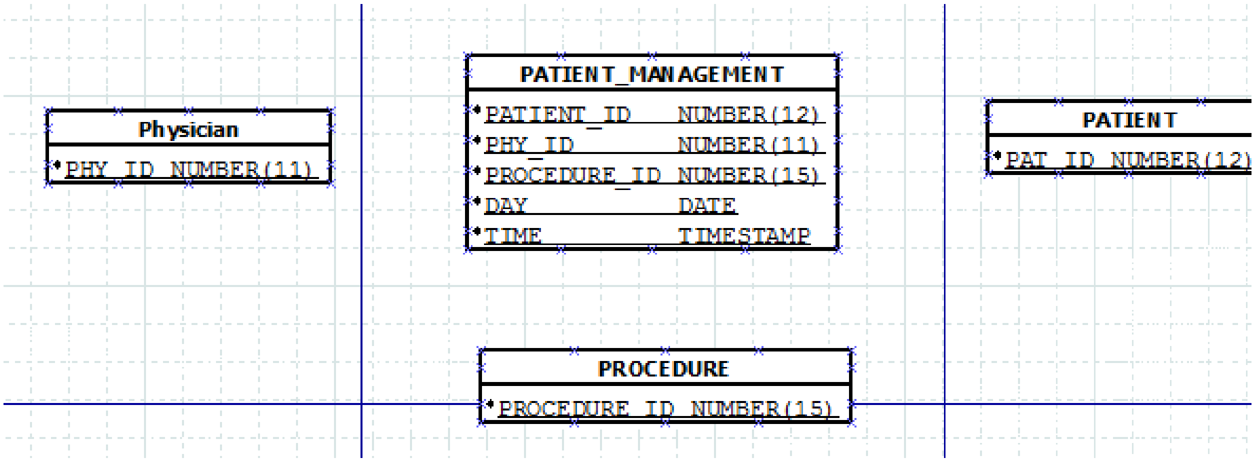


Therefore, to resolve these many to many relationships and keep track of who is treating whom with what procedure we need an AE.Let's call it Patient_management. A whole primary key for this entity that will keep every row unique as follows:

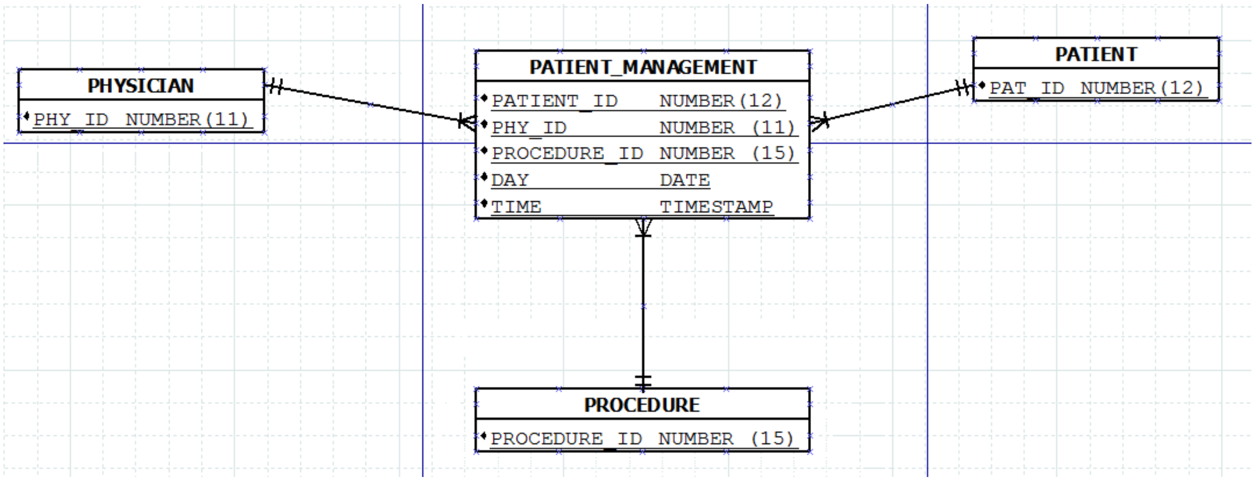
PhysicianID,PatientID,Procedureid, Date, Time

Since, one physician can treat one patient with one procedure on a certain date and at a particular time this whole key is unique for this entity.

The model now looks as follows:



Every relationship from an AE to the other entities are many to one. Finally, the model looks as follows:



Problem 5: Weak entities

A weak entity is an entity that doesnot exist without a strong entity. For example, if an employee doesnot exist then his/her dependents do not exist, but a employee can exist without any dependents. In this case dependents are weak and employee is a strong entity. Please represent weak entities as follows:

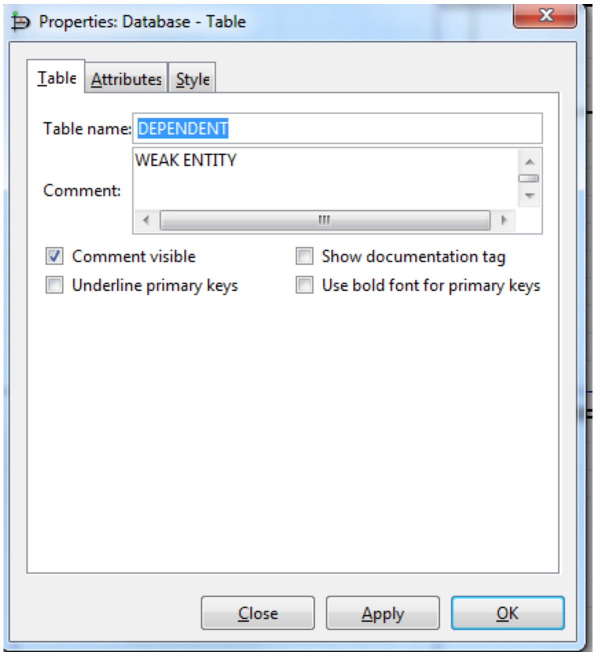
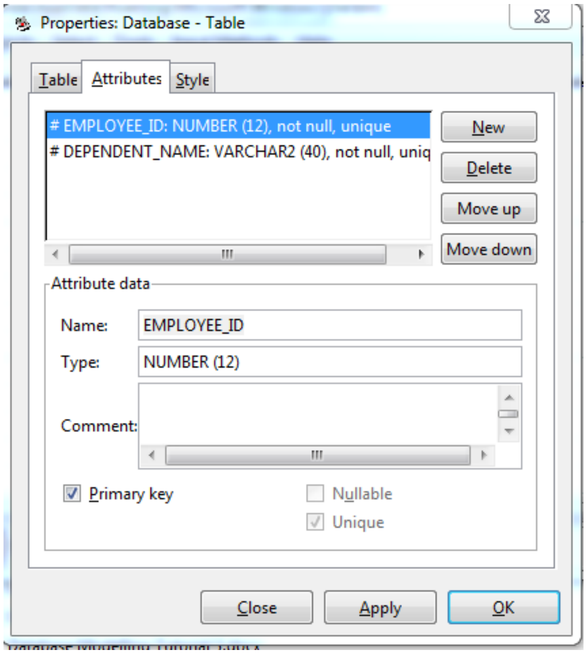


Table name: DEPENDENT

Comment: WEAK ENTITY

☑ Comment visible
□ Show documentation tag
□ Underline primary keys
□ Use bold font for primary keys



EMPLOYEE_ID: NUMBER (12), not null, unique
DEPENDENT_NAME: VARCHAR2 (40), not null, uniq

Attribute data

Name: EMPLOYEE_ID

Type: NUMBER (12)

Comment:

☑ Primary key
□ Nullable
☑ Unique

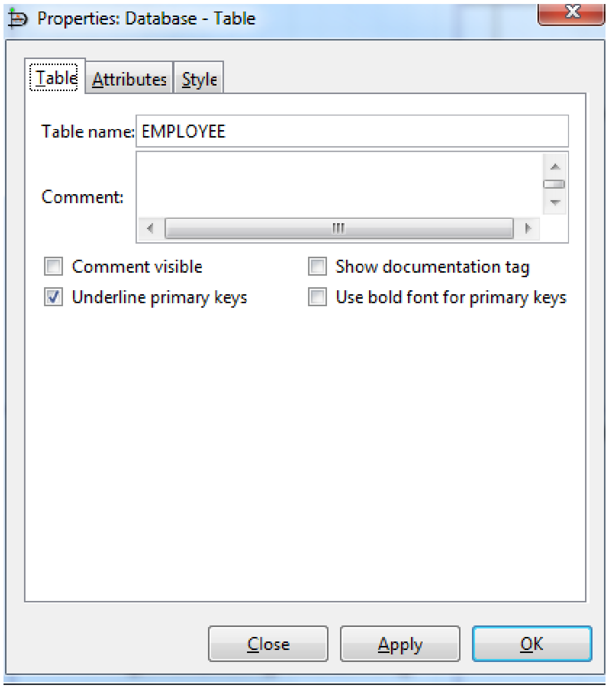
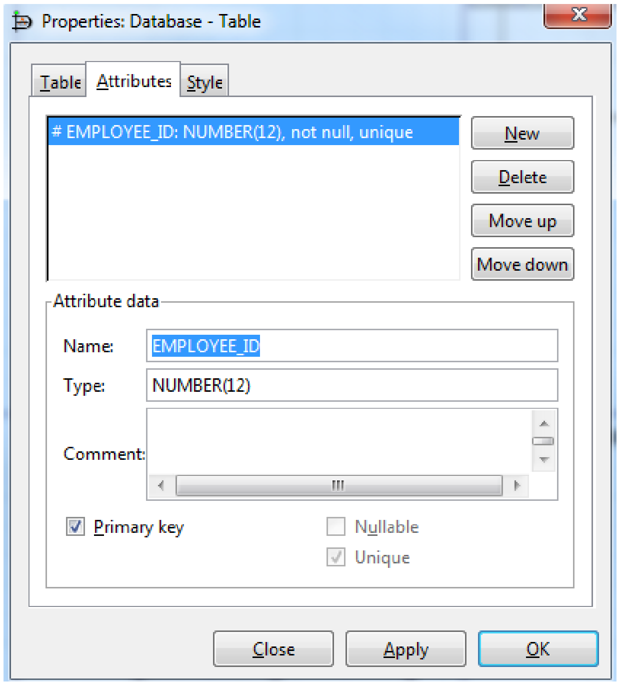


Table name: EMPLOYEE

Comment:

□ Comment visible
□ Show documentation tag
☑ Underline primary keys
□ Use bold font for primary keys



EMPLOYEE_ID: NUMBER(12), not null, unique

Attribute data

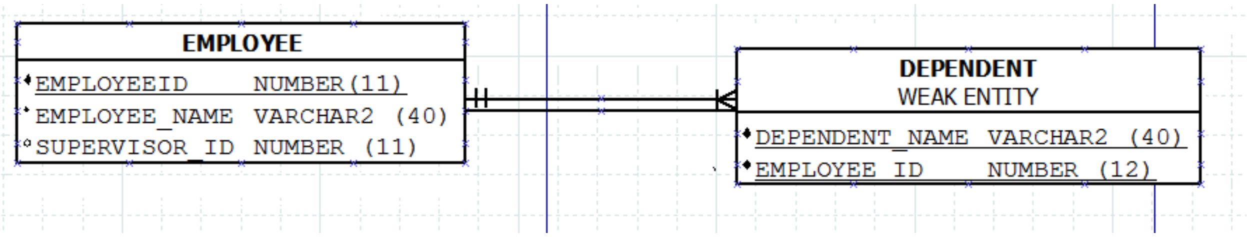
Name: EMPLOYEE_ID

Type: NUMBER(12)

Comment:

☑ Primary key
□ Nullable
☑ Unique

Please create two entities Employees and Dependents. Employees are indentified by Employee_ID and their dependents are identified by a whole key that includes EmployeeID and Dependent_Name both to keep every row unique. Please uncheck underlining primary keys for a weak entity and please check comment visibleas show in these pictures. In the comment section of weak entities please write “WEAK ENTITY”. Next, please add a reference line and a solid line from Dependents to Employees to represent weak entities as follows:



EMPLOYEE

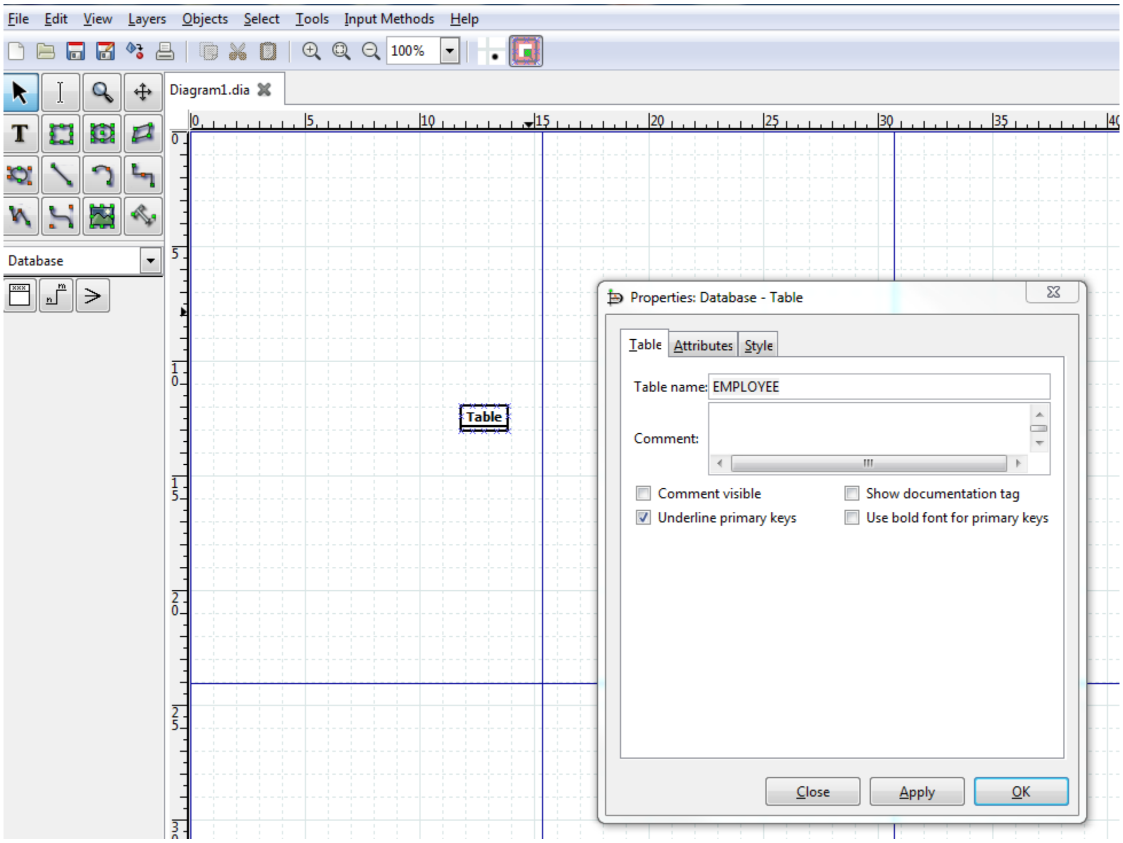
EMPLOYEEID	NUMBER(11)
EMPLOYEE_NAME	VARCHAR2 (40)
SUPERVISOR_ID	NUMBER (11)

DEPENDENT
WEAK ENTITY

DEPENDENT_NAME	VARCHAR2 (40)
EMPLOYEE_ID	NUMBER (12)

Employee_ID is a derived attribute in the dependents entity, therefore, please keep the datatype and size of this attribute same as its original entity. This rule also applies in case you add an AE as shown above in the Patient_management example. Please notice how the derived attributes are maintained the same as their original entities.

Problem 6: Composite attribute.



Properties: Database - Table

Table | Attributes | Style

Table name: EMPLOYEE

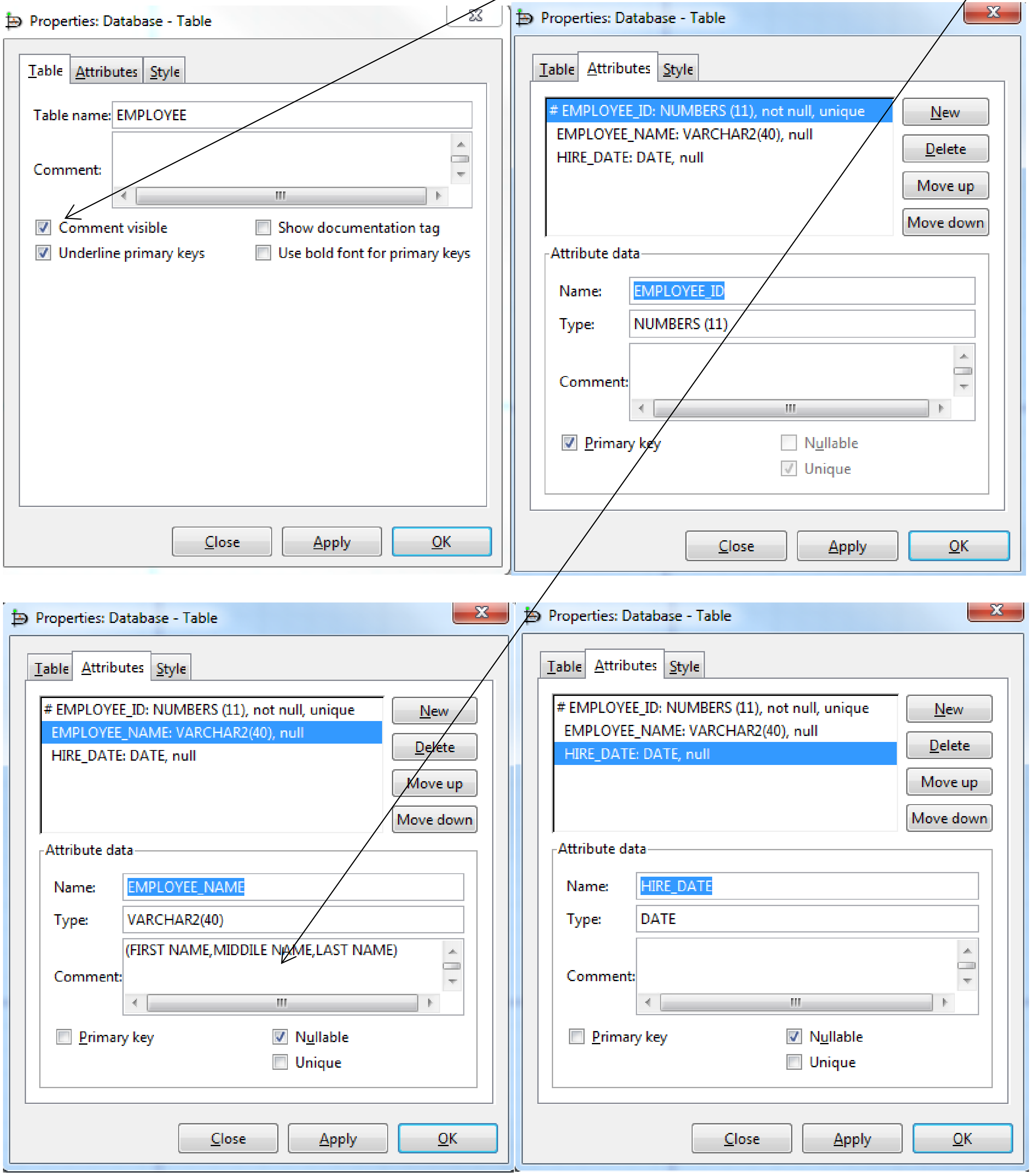
Comment:

☐ Comment visible ☐ Show documentation tag
☑ Underline primary keys ☐ Use bold font for primary keys

Close | Apply | OK

In this case, I am using EMPLOYEE as my entity. Attributes for this entity are Employee_ID, Employee_Name and Hire_Date. Employee_Name is a composite attribute. It can be further broken down into First Name, Middle Name and Last Name attributes.

To represent a composite attribute, please select Comment visible in the Table Properties above. Next, please enter attributes as follows. Please notice how the composite attribute is represented below.



Properties: Database - Table

Table name: EMPLOYEE

Comment:

☑ Comment visible ☐ Show documentation tag

☑ Underline primary keys ☐ Use bold font for primary keys

Properties: Database - Table

# EMPLOYEE_ID: NUMBERS (11), not null, unique
EMPLOYEE_NAME: VARCHAR2(40), null
HIRE_DATE: DATE, null

Attribute data

Name: EMPLOYEE_ID

Type: NUMBERS (11)

Comment:

☑ Primary key ☐ Nullable

☑ Unique

Properties: Database - Table

# EMPLOYEE_ID: NUMBERS (11), not null, unique
EMPLOYEE_NAME: VARCHAR2(40), null
HIRE_DATE: DATE, null

Attribute data

Name: EMPLOYEE_NAME

Type: VARCHAR2(40)

Comment: (FIRST NAME,MIDDILE NAME,LAST NAME)

☐ Primary key ☑ Nullable

☐ Unique

Properties: Database - Table

# EMPLOYEE_ID: NUMBERS (11), not null, unique
EMPLOYEE_NAME: VARCHAR2(40), null
HIRE_DATE: DATE, null

Attribute data

Name: HIRE_DATE

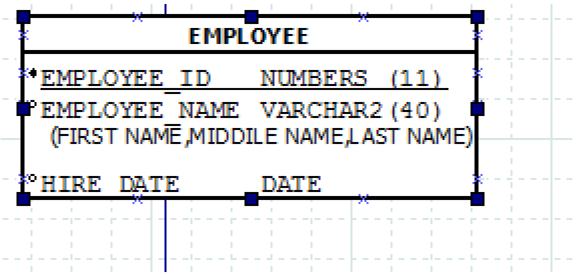
Type: DATE

Comment:

☐ Primary key ☑ Nullable

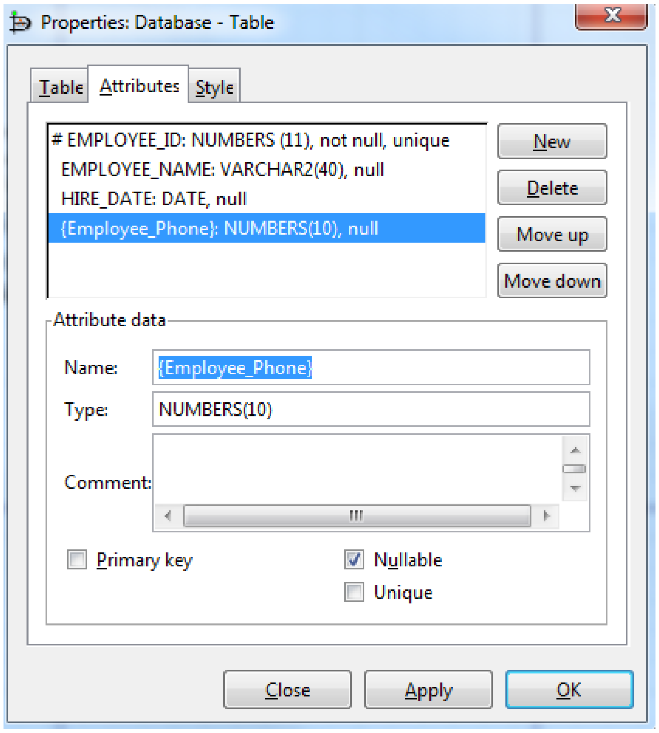
☐ Unique

Finally, click apply and your entity with a composite attribute is ready.


EMPLOYEE
*EMPLOYEE_ID NUMBERS (11)
°EMPLOYEE_NAME VARCHAR2 (40)
(FIRST NAME,MIDDILE NAME,LAST NAME)
°HIRE_DATE DATE

Problem 7: Multivalued attribute

Employee phone numbers is a multivalued attribute and please notice how it is represented below with curly brackets.



EMPLOYEE_ID: NUMBERS (11), not null, unique
EMPLOYEE_NAME: VARCHAR2(40), null
HIRE_DATE: DATE, null
{Employee_Phone}: NUMBERS(10), null

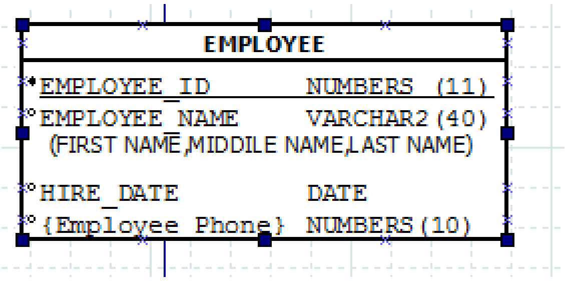
Attribute data

Name: {Employee_Phone}

Type: NUMBERS(10)

Comment:

☐ Primary key ☑ Nullable
☐ Unique


EMPLOYEE
*EMPLOYEE_ID NUMBERS (11)
°EMPLOYEE_NAME VARCHAR2 (40)
(FIRST NAME,MIDDILE NAME,LAST NAME)
°HIRE_DATE DATE
°{Employee_Phone} NUMBERS (10)