CS 207: Applied Database Practicum
Week 8

Varun Dutt

School of Computing and Electrical Engineering
School of Humanities and Social Sciences
Indian Institute of Technology Mandi, India

Scaling the Heights

DBMS - Data Models

Data models define how the logical structure of a database is modeled. Data Models are fundamental entities to introduce abstraction in a DBMS. Data models define how data is connected to each other and how they are processed and stored inside the system.

Entity-Relationship Model

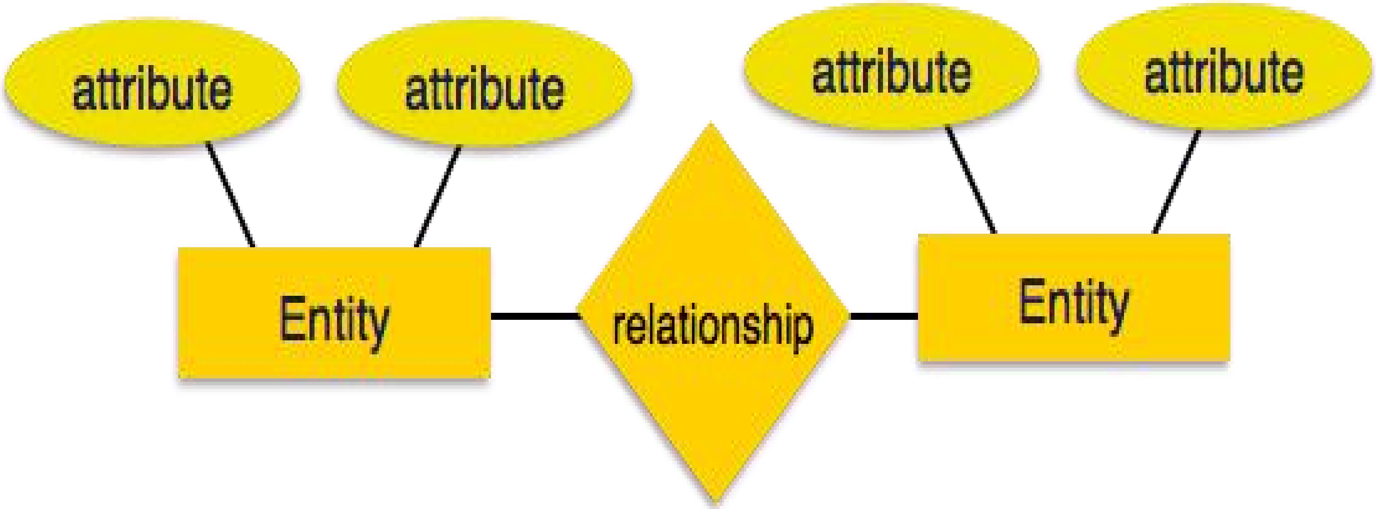
- Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. ER-Diagram is a pictorial representation of data that describes how data is communicated and related to each other.

- ER Model is best used for the conceptual design of a database.

ER Model

ER Model is based on -

- Entities and their attributes.
- Relationships among entities.



Entity

An entity in an ER Model is a real-world entity having properties called **attributes**. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

Attributes

Entities are represented by means of their properties, called **attributes**. All attributes have values. For example, a student entity may have name, class, and age as attributes.

There exists a domain or range of values that can be assigned to attributes. For eg., student’s age can’t be negative.

Types of Attributes

- **Simple -** Simple attributes are atomic values, which cannot be divided further. For example, a employee’s phone number is an atomic value of 10 digits.
- **Composite -** Composite attributes are made of more than one simple attribute. For example, a employee’s complete name may have first_name and last_name.

Types of Attributes

- **Derived -** Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For eg., age can be derived from date of birth.
- **Single value -** Single-value attributes contain single value. For example – Social_Security_Number.

Types of Attributes

- **Multivalue -** Multi-value attributes may contain more than one values. For example, a person can have more than one phone number.

Entity set and keys

An **entity set** is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values. For eg., a Students set may contain all the students of a school.

Key is an attribute or collection of attributes that uniquely identifies an entity among entity set. For eg., the roll no. of a student makes him/her identifiable among students.

Keys

- **Super Key** - A set of attributes (one or more) that collectively identifies an entity in an entity set.
- **Candidate Key -** A minimal super key is called a candidate key. An entity set may have more than one candidate key.
- **Primary Key -** A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

Relationship

The association among entities is called a relationship. For eg., an employee **works_at** a department, here works_at is the name of relationship between entities employee and department.

Relationship set - A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes. These attributes are called **descriptive attributes**.

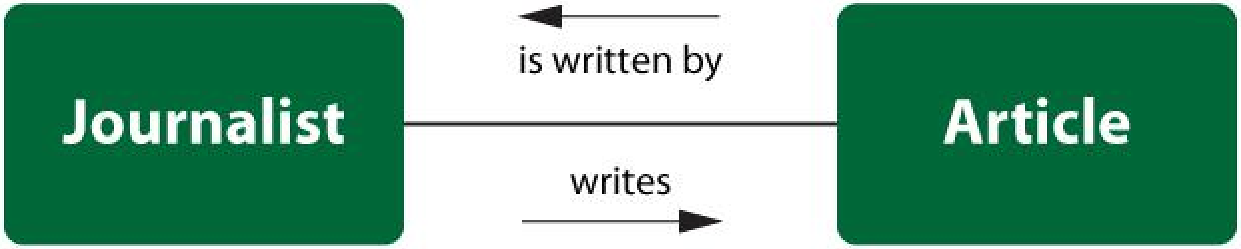
Degree of Relationship

The number of participating entities in a relationship defines the degree of the relationship.

- Binary = degree 2
- Ternary = degree 3
- n-ary = degree n

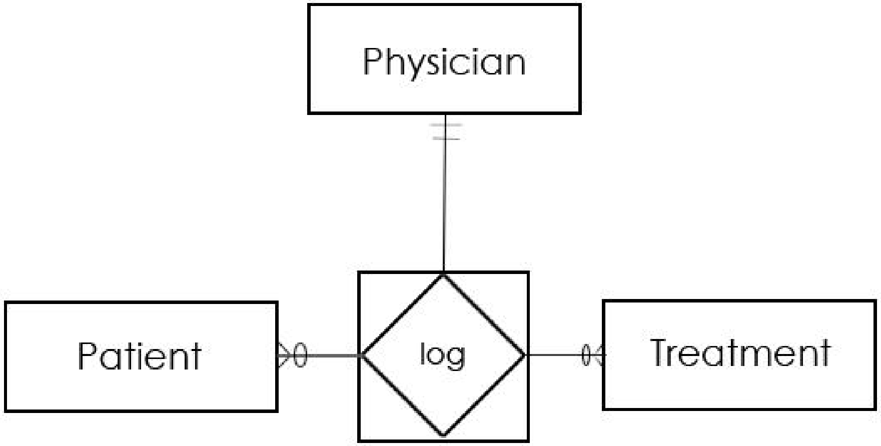
Binary Relationship

- A Journalist writes an article
- A article is written by a Journalist



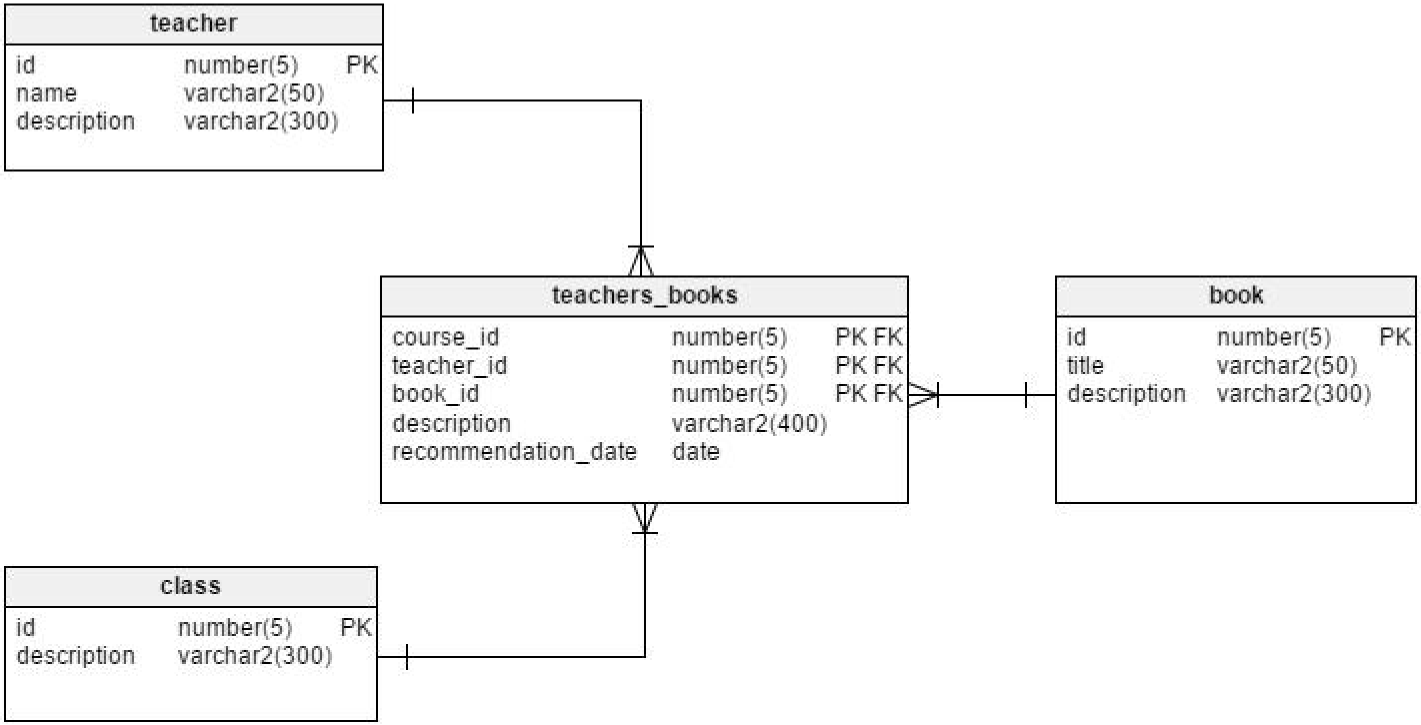
Ternary Relationship

- 1 Physician with 1 specific Patient can log M Treatments
- 1 Physician logs 1 specific Treatment for N Patients
- 1 Patient is logged 1 specific Treatment by 1 Physician



- So the ternary relationship log is an M-N-1 relationship between the participating entities Treatment-Patient-Physician

N-ary Relationship



teacher		
id	number(5)	PK
name	varchar2(50)	
description	varchar2(300)	

teachers_books		
course_id	number(5)	PK FK
teacher_id	number(5)	PK FK
book_id	number(5)	PK FK
description	varchar2(400)	
recommendation_date	date	

book		
id	number(5)	PK
title	varchar2(50)	
description	varchar2(300)	

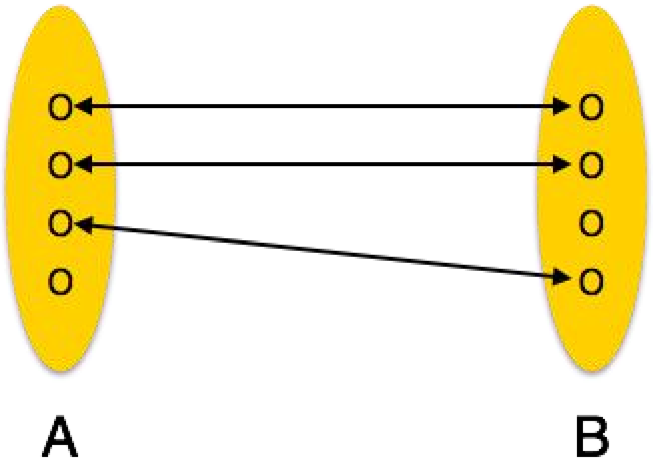
class		
id	number(5)	PK
description	varchar2(300)	

Mapping Cardinalities

Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

Types of Mapping cardinalities

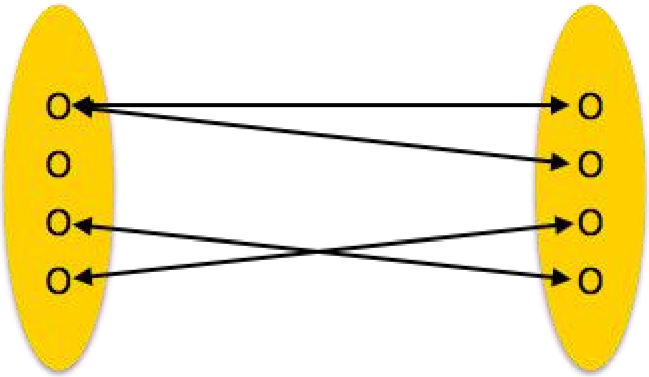
- **One to one -** One entity from entity set A can be associated with at most one entity of entity set B and vice versa.



e.g. One person can have only one Aadhar number

Types of cardinalities

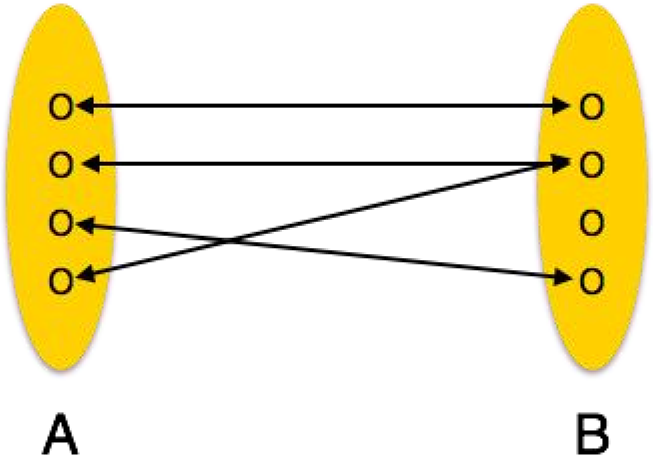
One to many - One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.



e.g. One person can have multiple Phone numbers

Types of cardinalities

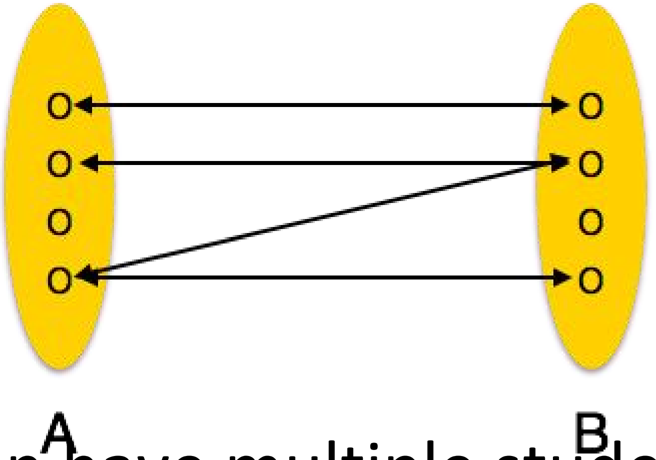
- **Many to one -** More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.



e.g. Many Courses can have a single Instructor

Types of cardinalities

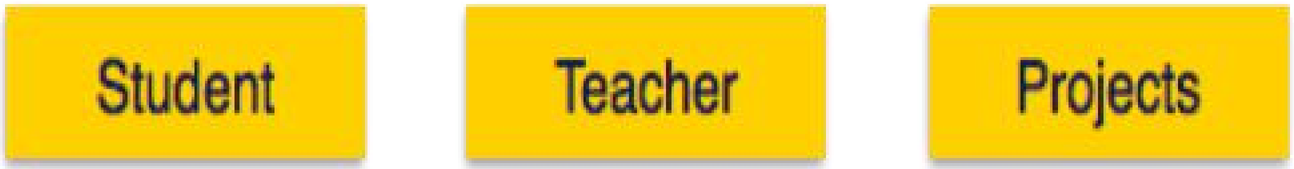
- **Many to many -** One entity from A can be associated with more than one entity from B and vice versa.



e.g. One course can have multiple students enrolled and each student can be enrolled multiple courses

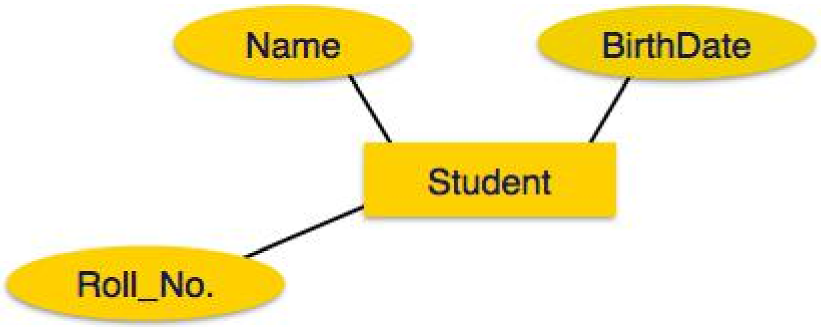
ER Diagram Representation

Entity - Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.



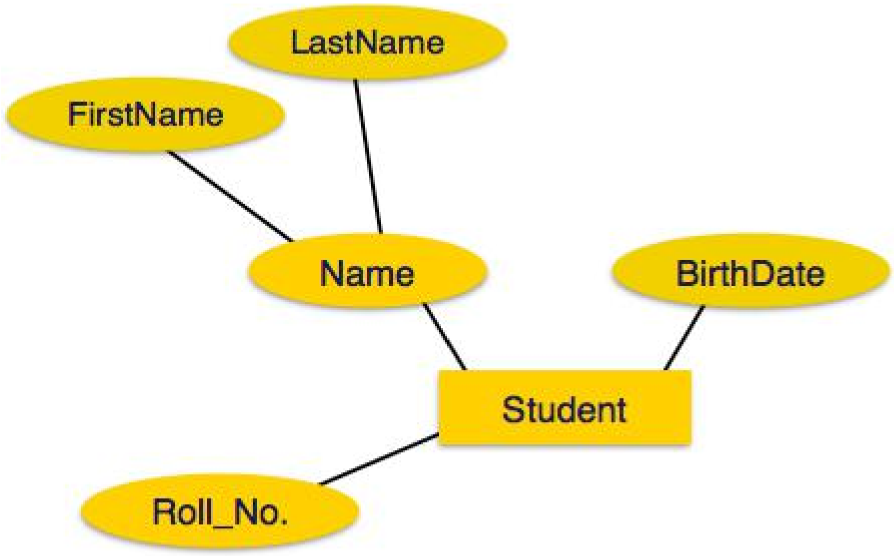
Attributes Representation

Attributes - Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).



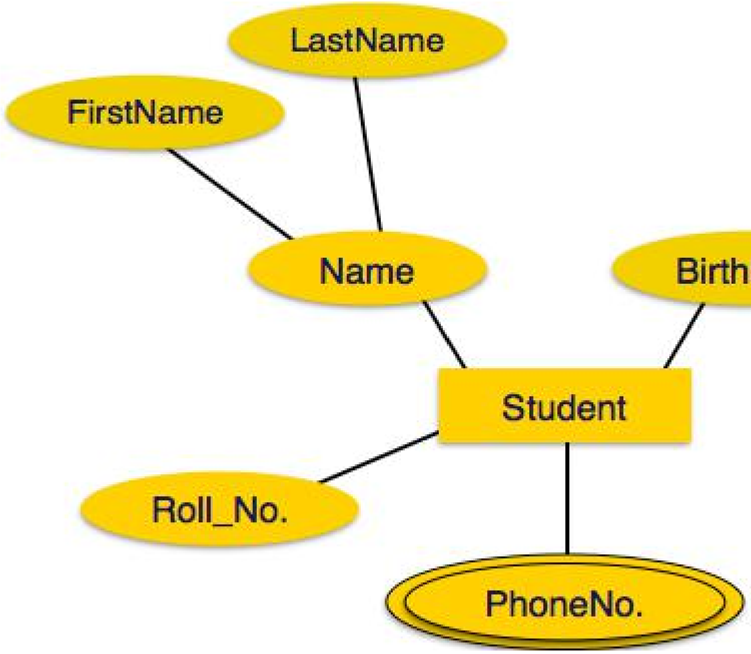
Attributes Representation

Composite Attributes - These are further divided in a tree like structure. Every node is then connected to its attribute.



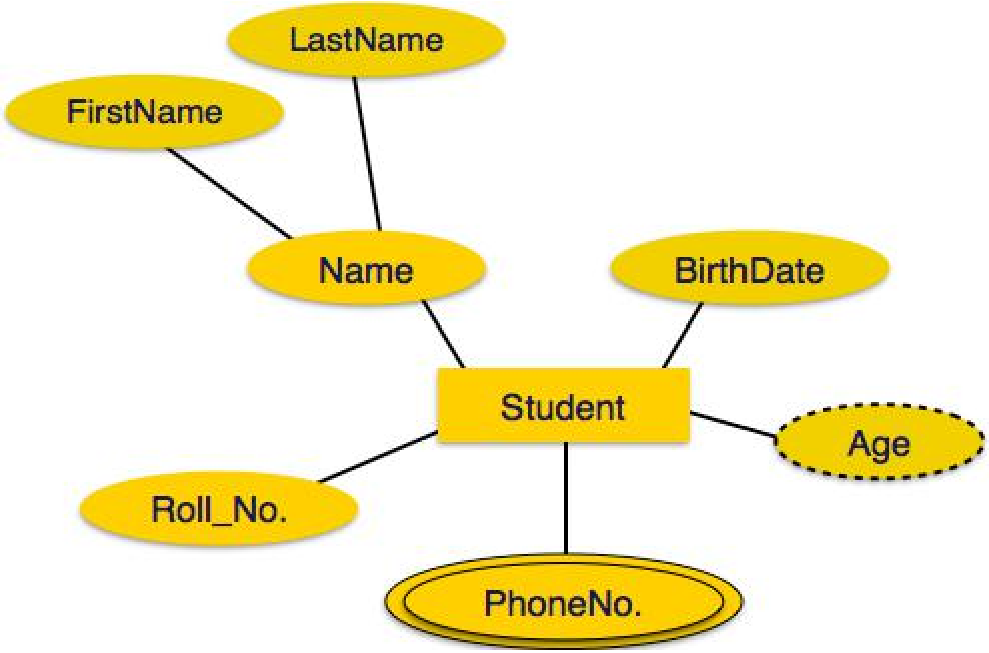
Attributes Representation

Multivalued Attributes are depicted by double ellipse.



Attributes Representation

Derived attributes are depicted by dashed ellipse.

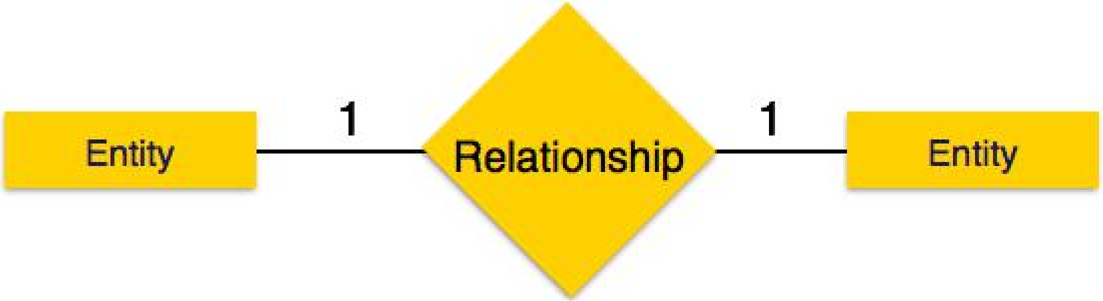


Relationship Representation

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.

Cardinality Representation

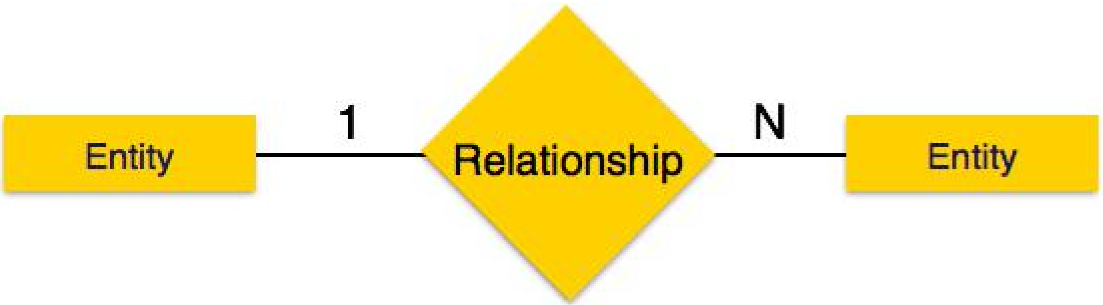
- **One to one -** When only one instance of an entity is associated with the relationship, it is marked as '1:1'.



It depicts one-to-one relationship.

Cardinality Representation

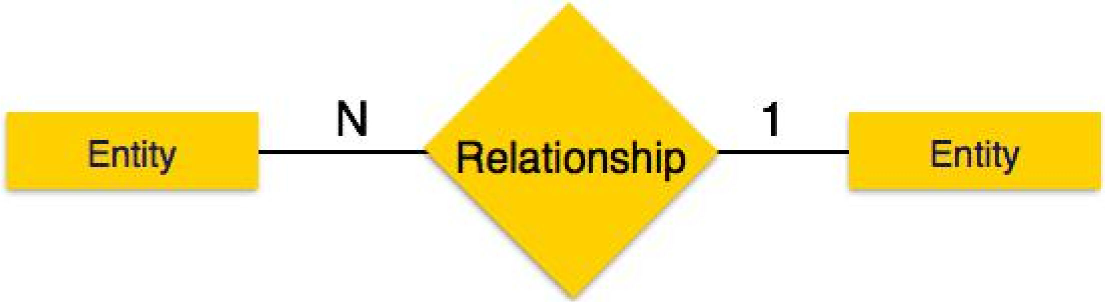
- **One to many -** When more than one instance of an entity is associated with a relationship, it is marked as '1:N'.



It depicts one-to-many relationship.

Cardinality Representation

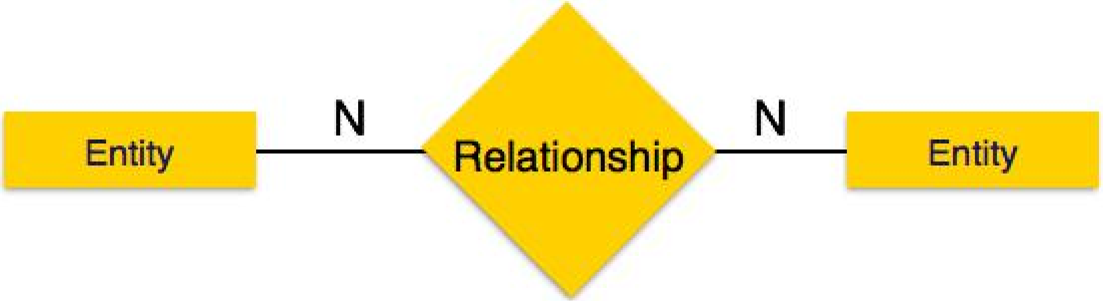
- **Many to one -** When more than one instance of entity is associated with the relationship, it is marked as 'N:1'.



It depicts many-to-one relationship.

Cardinality Representation

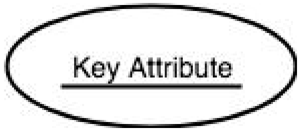
- **Many to many -** When more than one instance of entity is associated with the relationship, it is marked as 'N:N'.



It depicts many-to-many relationship.

Key representation

To represent a Key attribute, the attribute name inside the Ellipse is underlined.



Umbrello Introduction

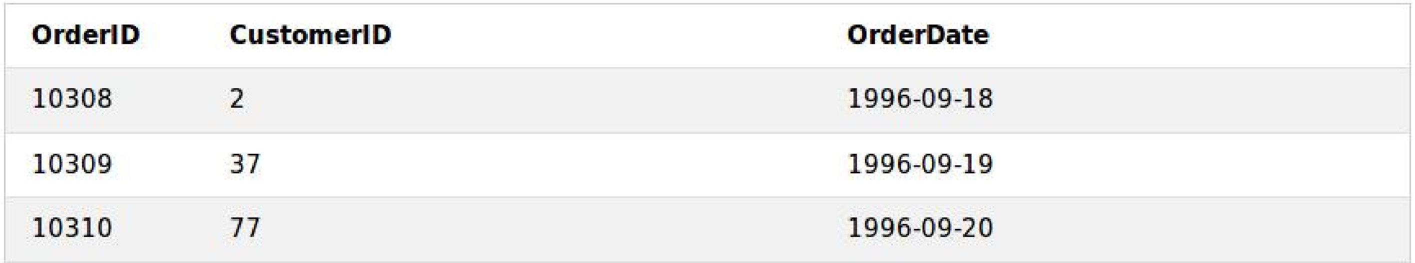
- Umbrello UML Modeller is a UML diagram tool that can support you in the software development process. Especially during the analysis and design phases of this process, Umbrello UML Modeller will help you to get a high quality product.
- Having a good model of your software is the best way to communicate with other developers working on the project and with your customers.

Umbrello Introduction

- UML is the diagramming language used to describing such models. You can represent your ideas in UML using different types of diagrams. Umbrello UML Modeller 2.11 supports the following types:
    - Class Diagram
    - Sequence Diagram
    - Collaboration Diagram
    - Use Case Diagram
    - State Diagram
    - Activity Diagram
    - Component Diagram
    - Deployment Diagram
    - Entity Relationship Diagram

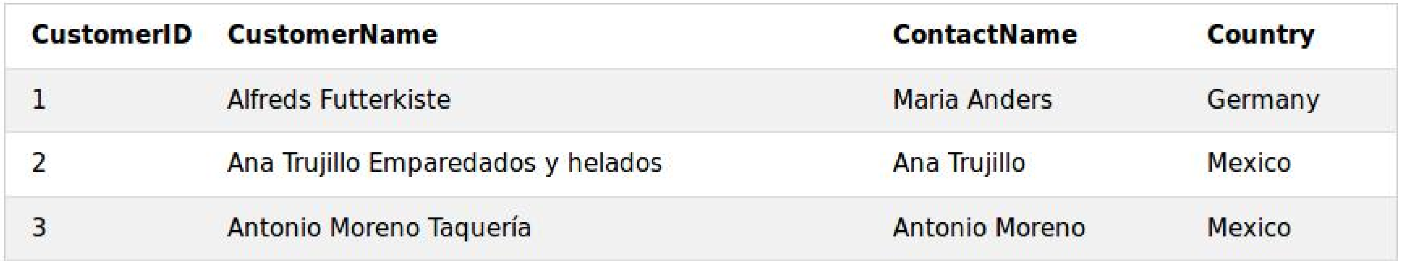
Drawing ER Diagrams using Umbrello

- We will draw the ER-Diagram for the examplar tables that we took earlier:
- “Orders”:



OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

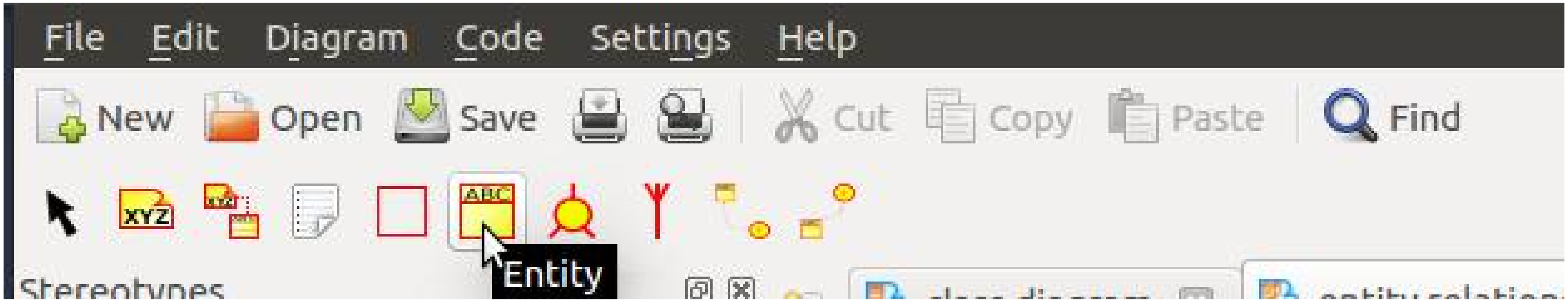
- “Customers”:



CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

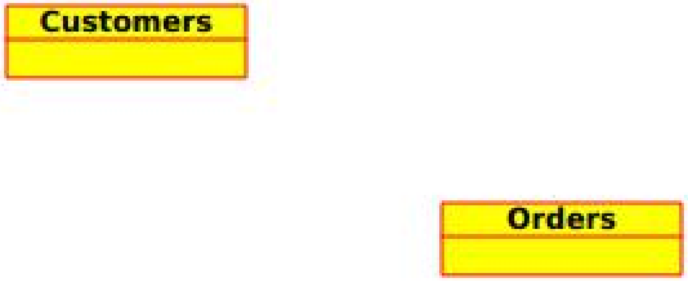
Drawing ER Diagrams using Umbrello

- First, select the “Diagram” drop down from the Umbrello Menu Bar.
- Click on “New” and select “Entity Relationship Diagram”.
- To create an entity, click on the following icon in the toolbar:

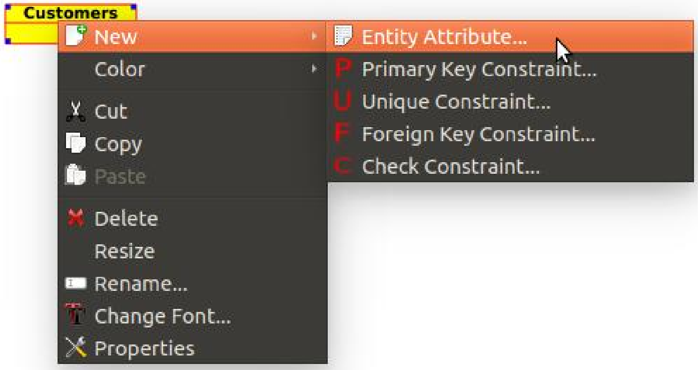


Drawing ER Diagrams using Umbrello

- Make the two entities “Orders” and “Customers”:

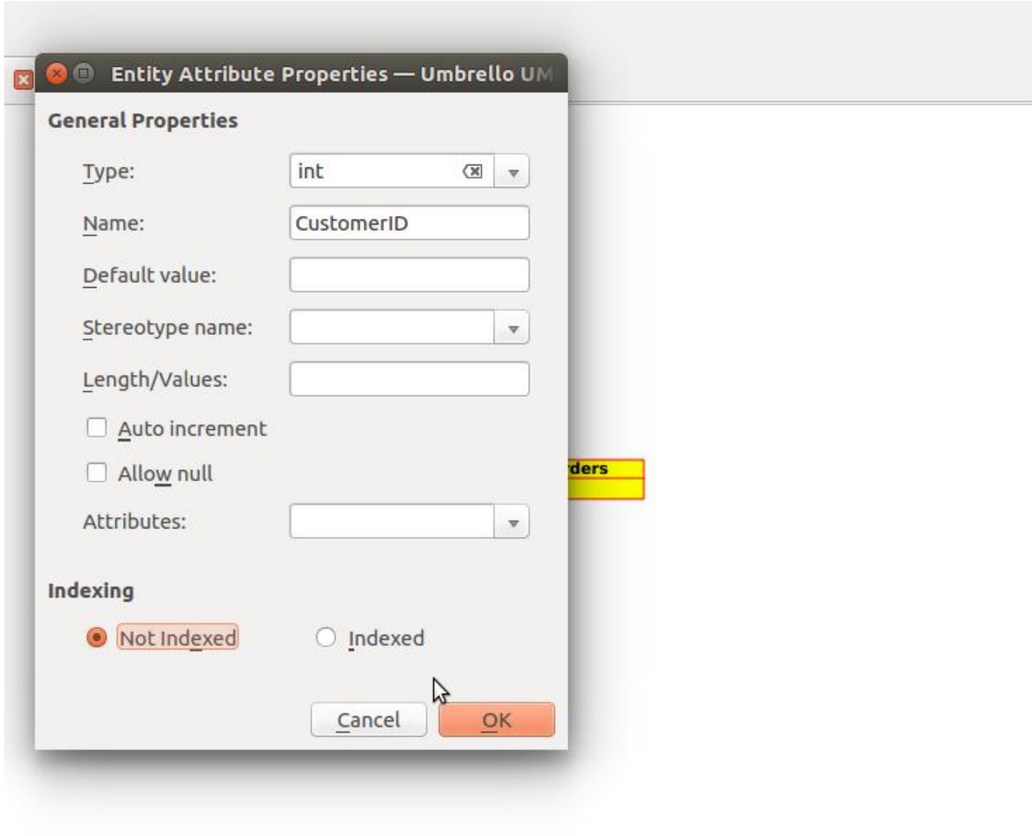


- To create attributes, right click on the entity and select “New”>”Entity Attribute”



Drawing ER Diagrams using Umbrello

- Enter the Data Type (Type) of the attribute and it’s name.



Entity Attribute Properties — Umbrello UM

General Properties

Type: int

Name: CustomerID

Default value:

Stereotype name:

Length/Values:

□ Auto increment

□ Allow null

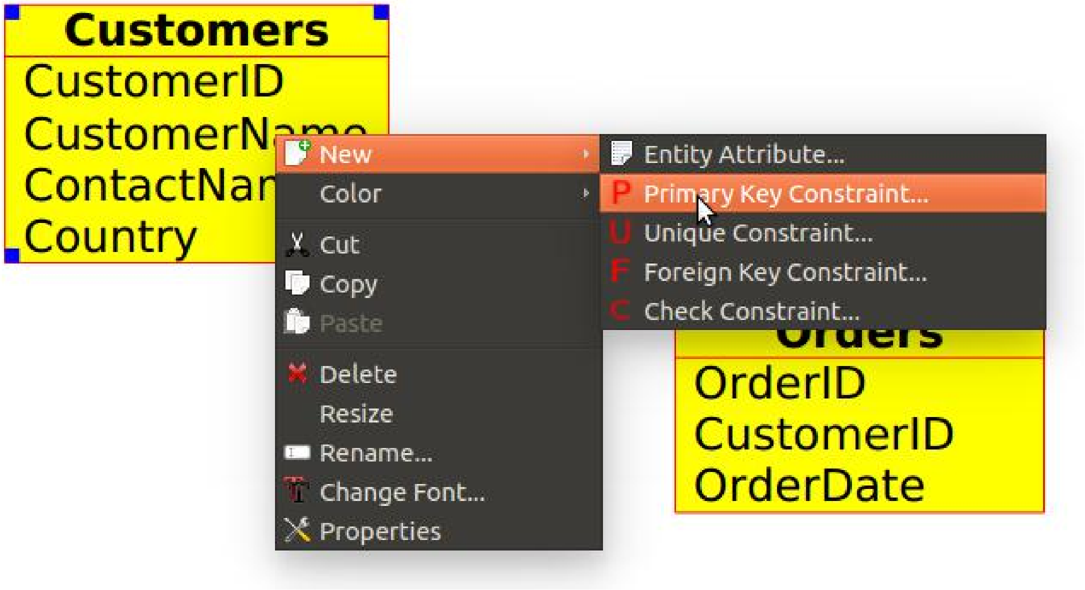
Attributes:

Indexing

● Not Indexed ○ Indexed

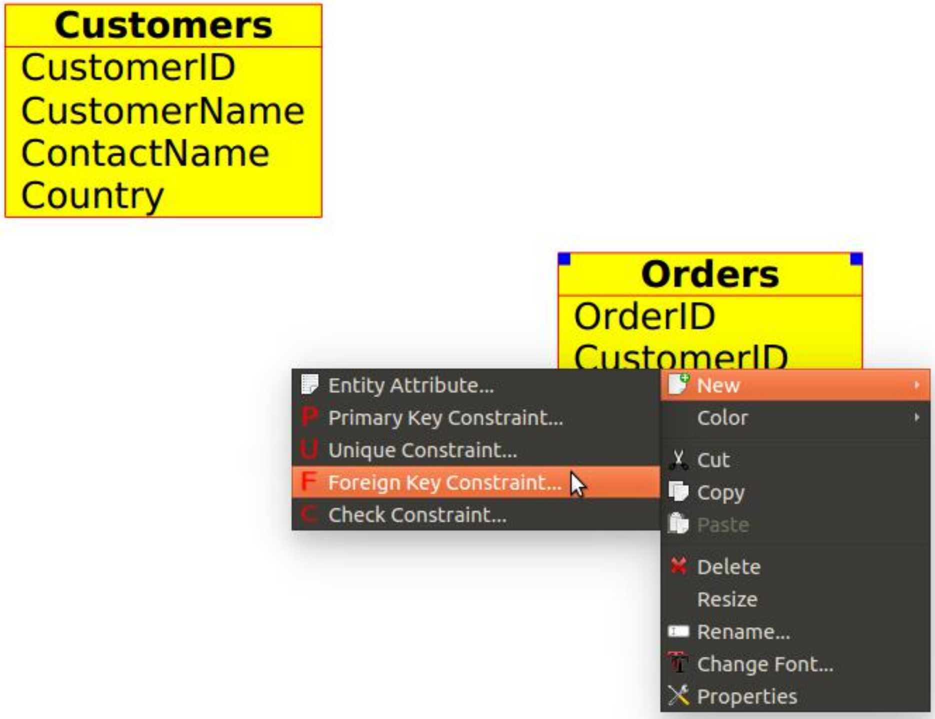
Drawing ER Diagrams using Umbrello

- After creating all the attributes for both the entities, we need to insert the primary key constraint. To do this, right click on the entity, select “New”>”Primary Key Constraint”.



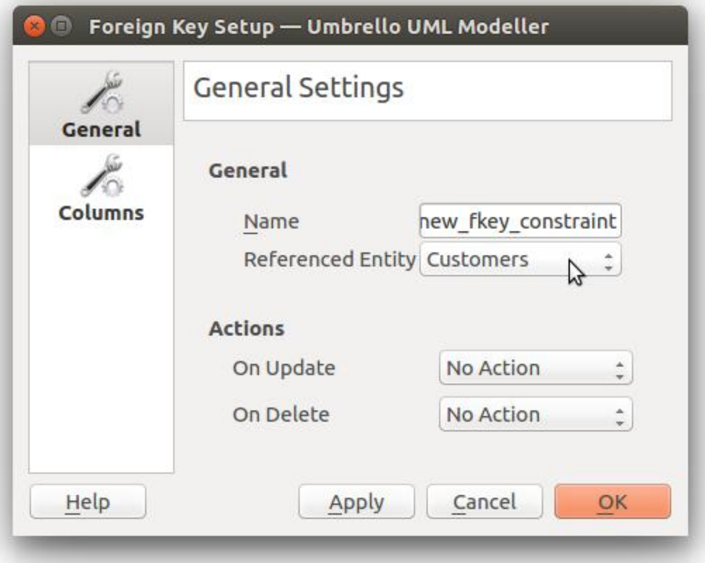
Drawing ER Diagrams using Umbrello

- Now, to create the Foreign Key constraint, right click on the entity, select “New”>”Foreign Key Constraint”.



Drawing ER Diagrams using Umbrello

- In the resulting dialogue box, in the General Settings tab set the Referenced Entity as the Entity whose column you wish to use as reference (e.g. Customers for CustomerID here).



Foreign Key Setup — Umbrello UML Modeller

General Settings

General

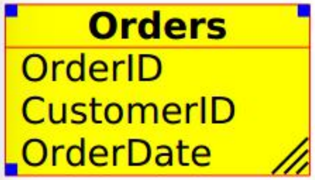
Name: new_fkey_constraint

Referenced Entity: Customers

Actions

On Update: No Action

On Delete: No Action



Orders

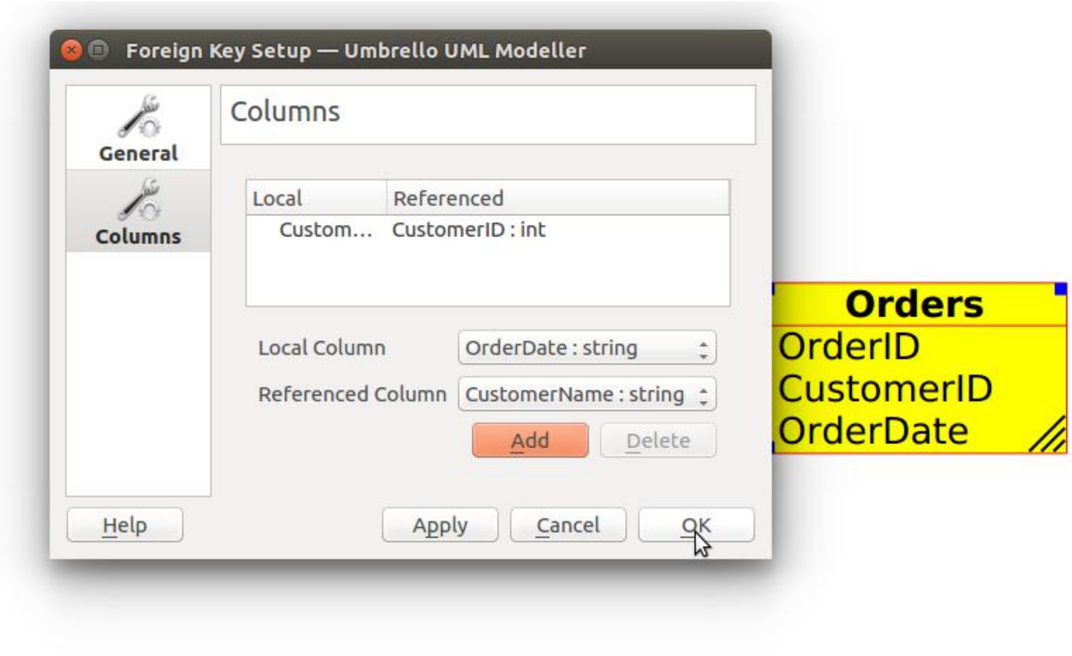
OrderID

CustomerID

OrderDate

Drawing ER Diagrams using Umbrello

- In the “Columns” tab, select the columns which will work as the foreign key constraints.



Foreign Key Setup — Umbrello UML Modeller

Columns

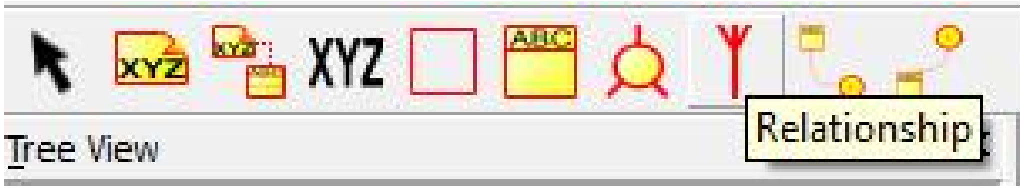
Local	Referenced
Custom…	CustomerID : int

Local Column: OrderDate : string

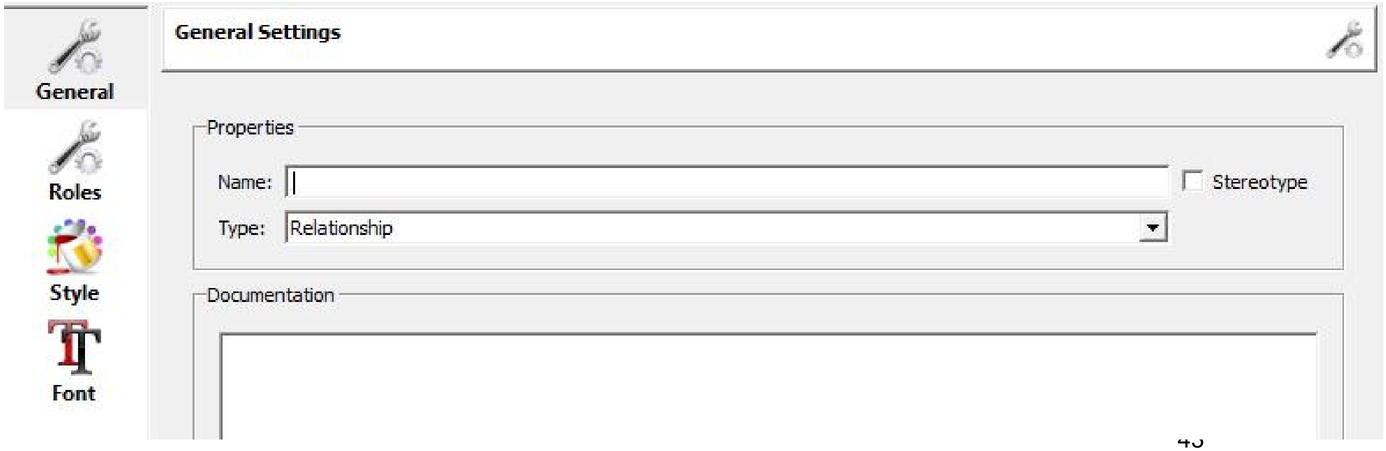
Referenced Column: CustomerName : string

Drawing ER Diagrams using Umbrello

- To create an relationship, click on the following icon in the toolbar



- To define the relationship between Entities double-click on the relationship



General Settings

Properties

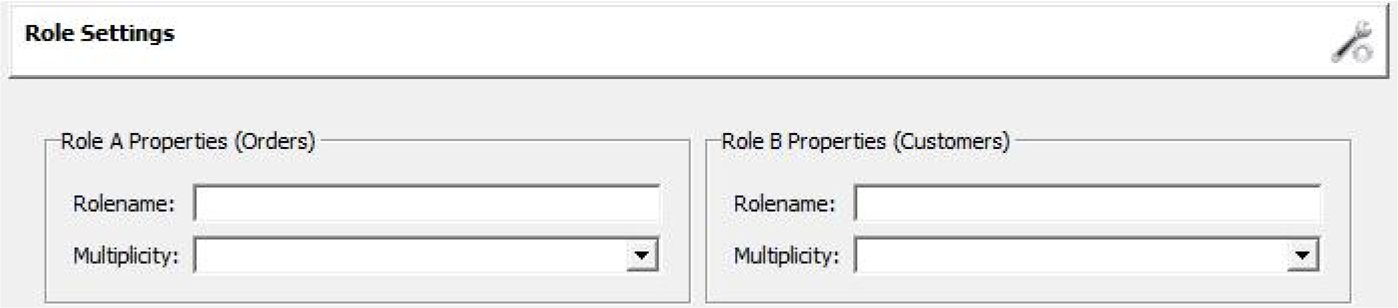
Name: | □ Stereotype

Type: Relationship ▾

Documentation

Drawing ER Diagrams using Umbrello

- To define the role of each entity and the multiplicity of each entity.



Role Settings

Role A Properties (Orders)

Rolename:

Multiplicity:

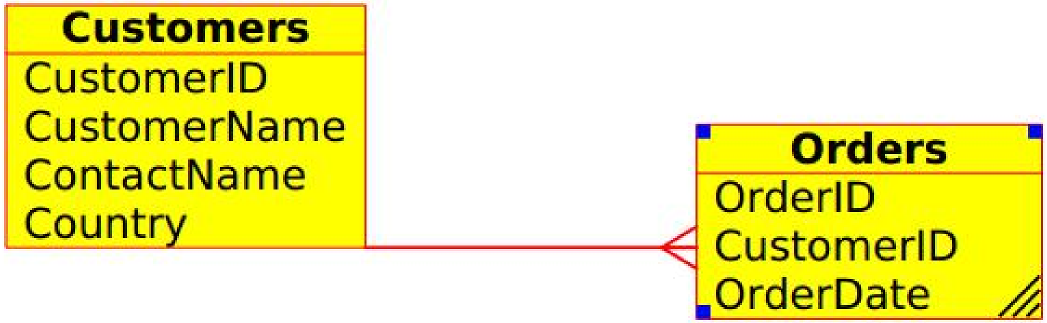
Role B Properties (Customers)

Rolename:

Multiplicity:

Drawing ER Diagrams using Umbrello

- And hence, the complete exemplar ER Diagram will look like:



Customers

CustomerID
CustomerName
ContactName
Country

Orders

OrderID
CustomerID
OrderDate

References

https://w3schools.in

https://tutorialspoint.com

https://docs.kde.org/trunk4/en/kdesdk/umbrello/index.html