

INF4401 - Database Design

Julien Romero - Télécom SudParis

Previously...

Working With an Existing Database

- We supposed that we already have a relational database and we want to query it.
 - The relational database can be described with a *database schema*
- We saw that we can use relational algebra to represent our queries as a diagram.
- When we use a computer, we need a dedicated programming language SQL
 - SQL implements relational algebra...
 - ... but also add new features like aggregation functions and grouping.

SQL Query

```
SELECT <list of attributes/columns to select>  
FROM <list of tables to consider>  
[JOIN <table>  
  [ON <join condition> ]]*  
[WHERE <condition without aggregation>  
[GROUP BY <list of columns used for grouping>  
[HAVING <condition with aggregation> ]]  
[ORDER BY <list of column + ASC or DESC>  
[LIMIT <number of rows> ];
```

Database Design

Database Design Stages



Identification of Client Needs

- Before we can query a database, we need to create it.
- In general a client arrives with a vague textual project description and we need to translate it into an *organized database*.
 - We need to understand client needs and ask for clarification. Interaction with the client is crucial here.
 - The client also has *functional requirements*, i.e. queries they want to run.

Database Design Stages



Identification of
Client Needs

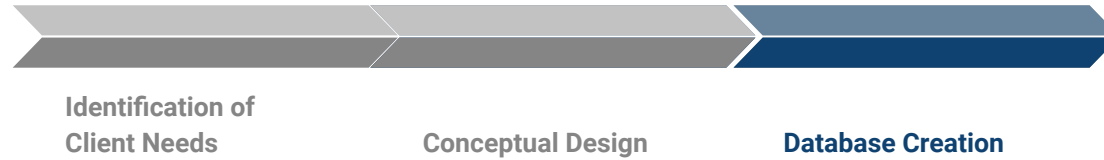
Conceptual Design

Database Creation

What Is a Database Design?

- Once we identify the needs, we need to develop a conceptual design to answer those needs.
- A good design:
 - Satisfies the client needs: Can we represent everything the client wants?
 - Is sound: Does it contain inconsistencies?
 - Does not contain redundancy: Is each information represented only once. If it is not the case, it might be a problem to maintain the database.
 - Movies(movieID, title, year, **directorID**, directorName)
 - Directors(directorID, directorName, birthdata)
 - Meets the functional requirements: Can we query the database to get what the client wants?

Database Design Stages



Database Creation

- Once the design is finished, we have to actually integrate it into a physical system
- Once the system is physically implemented, it is hard to change the conceptual design. Therefore, we want to take a lot of care with the previous stage.
- When the design is implemented, we can start to query the database.

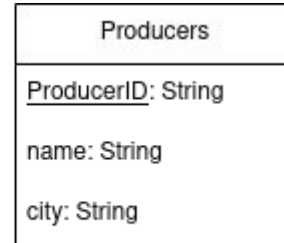
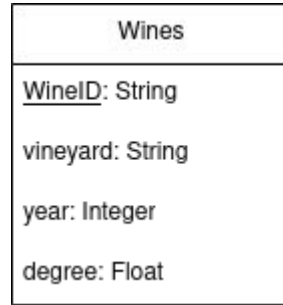
Entity-Relationship Model

Entity-Relationship Model

- The Entity-Relationship Model or E/R model allows to design a database.
 - It is often associated with the E/R diagram, which is a visual representation of the model.
- Main concepts:
 - **Entity:** A physical or conceptual “thing” or “object” that can be uniquely identified.
 - E.g.: a bike, a student, an exam, a grade, a transaction
 - **Property/attributes:** values (numeric, String, dates, ...) that characterize an entity.
 - E.g.: name, birthdate, social security number, passport ID, amount of money in a bank account
 - Properties can also be primary keys! If we do not have a primary key, we have to create one artificially (generally called ID)
 - **Entity set:** A set of entities of the same type that share the same properties.
 - E.g.: All the bikes, all the students, all the exams
 - An entity can be in several sets: A student can be both in the set of all students and of all people.

Representation of an Entity Set

- An entity set is represented by a rectangle containing the name of the set and its attributes.



Example - Identifying the Entities

Our client wants to design an application to manage her employees. When she recruits someone, this person has to fill a form to give his name, address, phone number, and birthdate. Then, they sign at a given date a contract that has a type (CDD/CDI), a start date, and an end date. If the signed contract is of predefined length (CDD), it is possible to renew it by signing a new contract. Each employee has exactly a manager, and a manager manages several employees. Each employee is also assigned to an office with a name. An office can only be occupied by a single employee. An employee can be sent to a mission in a particular city with a specific goal (e.g. develop a new product). A mission can be assigned to several employees. An assignment has a start and an end date.

Example - Identifying the Entities

Our **client** wants to design an application to manage her **employees**. When she recruits someone, this person has to fill a form to give his name, address, phone number, and birthdate. Then, they sign at a given date a **contract** that has a type (CDD/CDI), a start date, and an end date. If the signed contract is of predefined length (CDD), it is possible to renew it by signing a new contract. Each employee has exactly a **manager**, and a manager manages several employees. Each employee is also assigned to an **office** with a name. An office can only be occupied by a single employee. An employee can be sent to a **mission** in a particular city with a specific goal (e.g. develop a new product). A mission can be assigned to several employees. An assignment has a start and an end date.

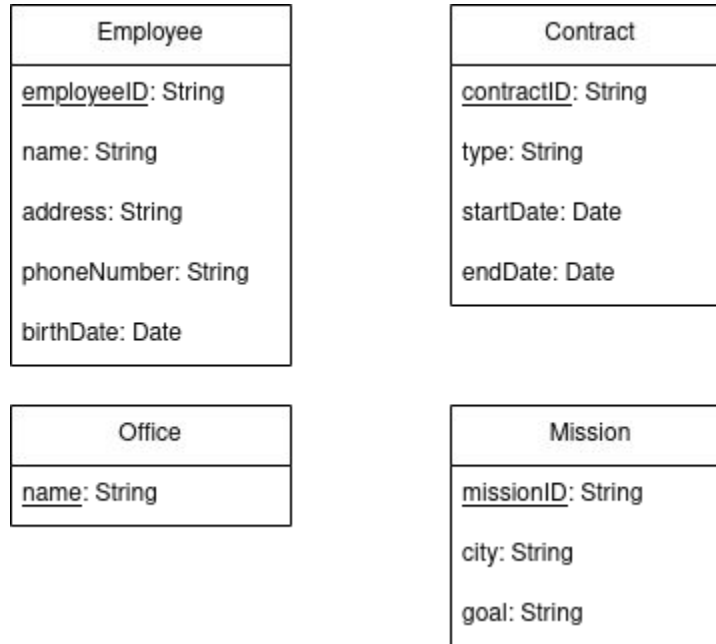
Example - Identifying the Attributes

Our client wants to design an application to manage her employees. When she recruits someone, this person has to fill a form to give his name, address, phone number, and birthdate. Then, they sign at a given date a contract that has a type (CDD/CDI), a start date, and an end date. If the signed contract is of predefined length (CDD), it is possible to renew it by signing a new contract. Each employee has exactly a manager, and a manager manages several employees. Each employee is also assigned to an office with a name. An office can only be occupied by a single employee. An employee can be sent to a mission in a particular city with a specific goal (e.g. develop a new product). A mission can be assigned to several employees. An assignment has a start and an end date.

Example - Identifying the Attributes

Our client wants to design an application to manage her employees. When she recruits someone, this person has to fill a form to give his **name, address, phone number, and birthdate**. Then, they sign at a given date a contract that has a **type (CDD/CDI), a start date, and an end date**. If the signed contract is of predefined length (CDD), it is possible to renew it by signing a new contract. Each employee has exactly a manager, and a manager manages several employees. Each employee is also assigned to an office with a **name**. An office can only be occupied by a single employee. An employee can be sent to a mission **in a particular city with a specific goal** (e.g. develop a new product). A mission can be assigned to several employees. An assignment has a start and an end date.

Example - E/R Diagram For Entities and Attributes

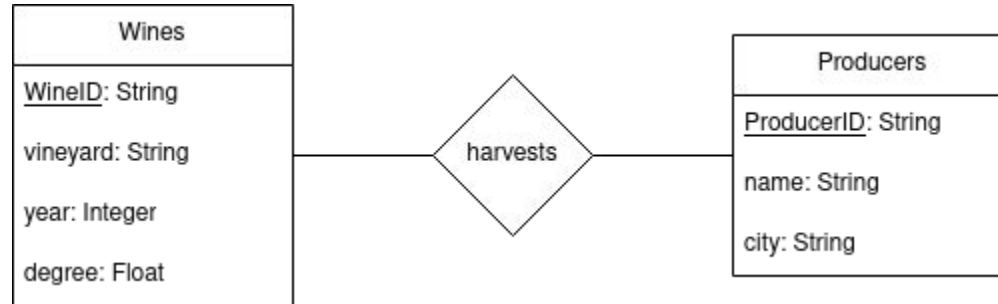


Relationships

- **Relationship:** Association between entities
 - E.g.: Between a teacher and a student: advisor, between a customer and a product: buy, between an employee and his company: work for
- **Relationship set:** Set of association of the same type.
 - E.g.: All the “advisor” relationships, all the “buy” relationships, all the “work for” relationships

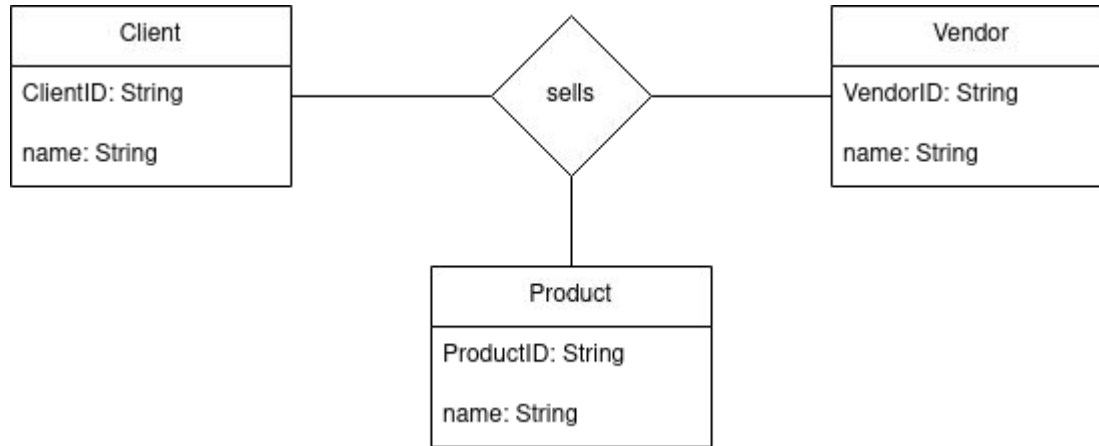
Representation of a Relationship Set

- A relationship set is represented as a diamond between several entity sets.
- Here is a binary relationship harvests between a producer and a wine.



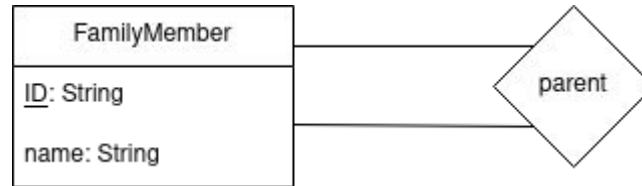
Representation of a Relationship Set

- Relationships can be about more than two entity sets.
- Here, a relationships representing that a vendor sells a product to a client;



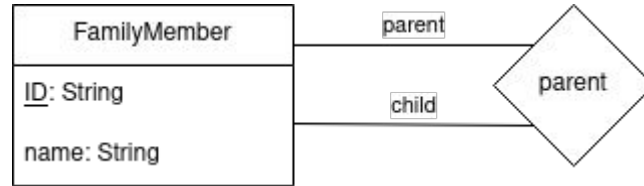
Representation of a Relationship Set

- A relationship can be between only one entity set.
- Here, a relationship between a child and his parents.



Entity's role

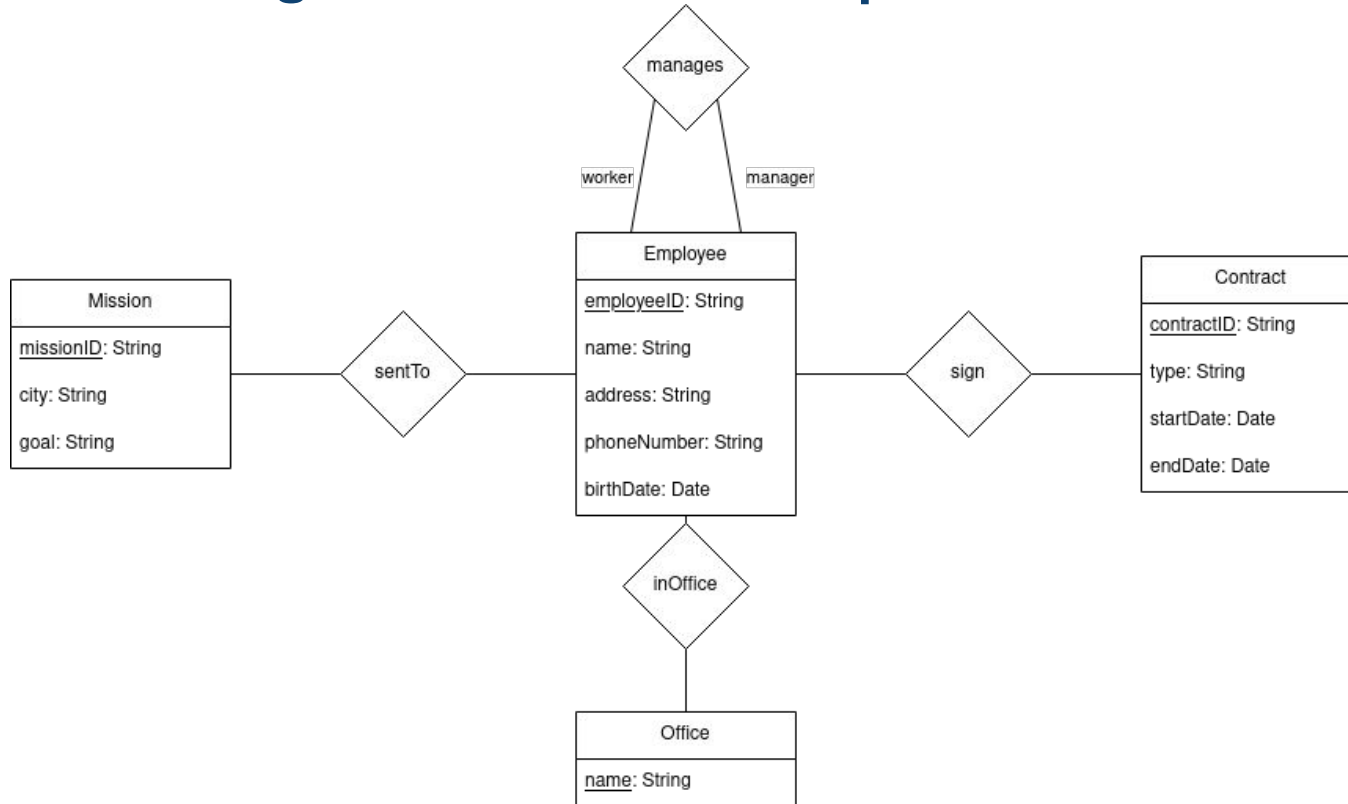
- Sometime, the role of an entity is not clear. In this case, we can add labels on the connections.



Example - Identifying the Relationships

Our client wants to design an application to manage her employees. When she recruits someone, this person has to fill a form to give his name, address, phone number, and birthdate. Then, they **sign** at a given date a contract that has a type (CDD/CDI), a start date, and an end date. If the signed contract is of predefined length (CDD), it is possible to renew it by signing a new contract. Each employee has exactly a manager, and a manager **manages** several employees. Each employee is also **assigned to** an office with a name. An office can only be occupied by a single employee. An employee can be **sent to** a mission in a particular city with a specific goal (e.g. develop a new product). A mission can be assigned to several employees. An assignment has a start and an end date.

Example - E/R Diagram With Relationships

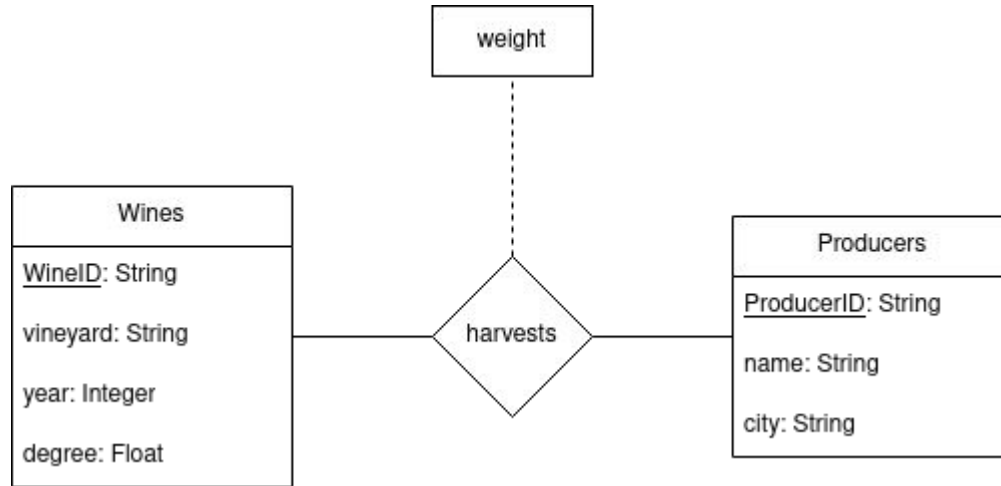


Relationship Attributes

- In some cases, we might need to add attributes to the relationship to better describe the interaction.
 - E.g.: We can give the weight harvested for a producer and wine, or the price of an item for the relation sells.

Representation of Relationship Attributes

- We use a rectangle connected with a dash line to the relationship



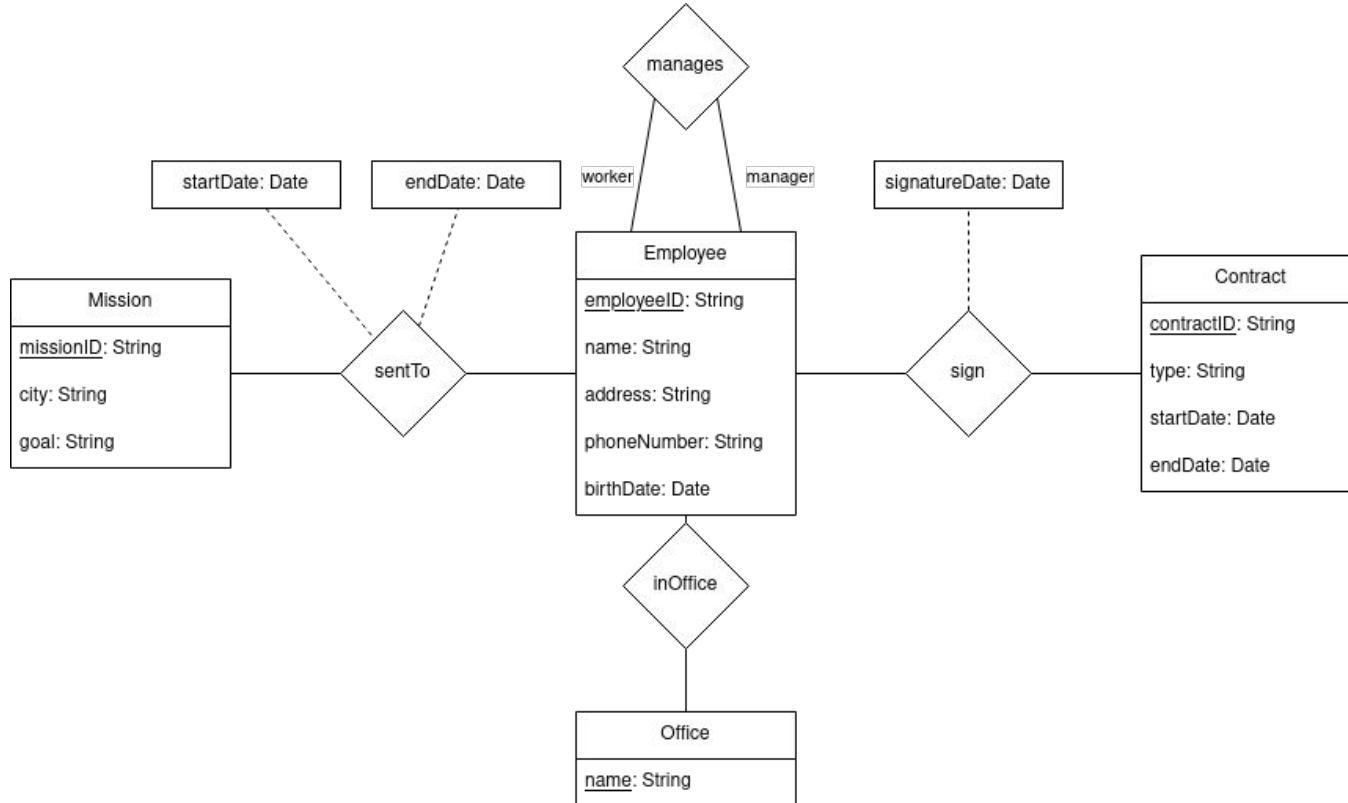
Example - Identifying the Relationships Attributes

Our client wants to design an application to manage her employees. When she recruits someone, this person has to fill a form to give his name, address, phone number, and birthdate. Then, they sign at a given date a contract that has a type (CDD/CDI), a start date, and an end date. If the signed contract is of predefined length (CDD), it is possible to renew it by signing a new contract. Each employee has exactly a manager, and a manager manages several employees. Each employee is also assigned to an office with a name. An office can only be occupied by a single employee. An employee can be sent to a mission in a particular city with a specific goal (e.g. develop a new product). A mission can be assigned to several employees. An assignment has a start and an end date.

Example - Identifying the Relationships Attributes

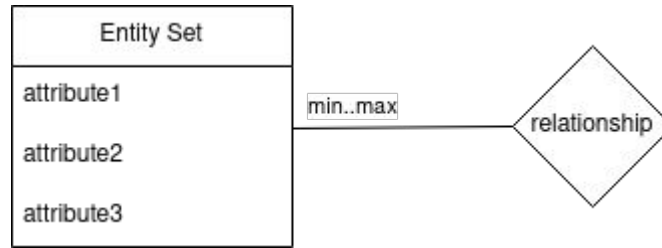
Our client wants to design an application to manage her employees. When she recruits someone, this person has to fill a form to give his name, address, phone number, and birthdate. Then, they sign at **a given date** a contract that has a type (CDD/CDI), a start date, and an end date. If the signed contract is of predefined length (CDD), it is possible to renew it by signing a new contract. Each employee has exactly a manager, and a manager manages several employees. Each employee is also assigned to an office with a name. An office can only be occupied by a single employee. An employee can be sent to a mission in a particular city with a specific goal (e.g. develop a new product). A mission can be assigned to several employees. An assignment has **a start and an end date**.

Example - E/R Diagram With Relationship Attributes



Cardinality of a Relationship

- For a given entity set and relationship, we can indicate in how many relations each entity is enrolled.



- We indicate the cardinality with the notation min..max on the link between the entity set and the relationship.
 - Min is the minimum number of relation for each entity, and max is the maximum number of entity.
 - If max = *, it means there is no limit.

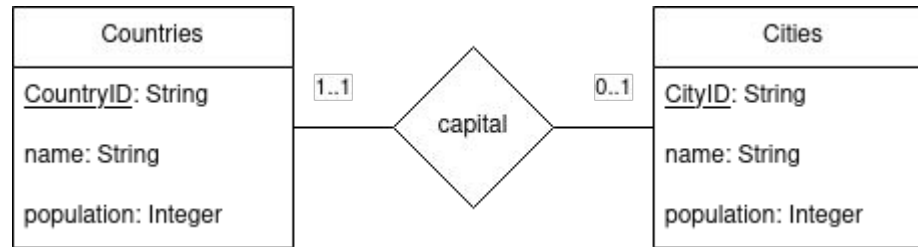
Cardinality of a Relationship - Cardinality types

There are three types of cardinality for a binary relationship:

- **One-to-one:** Both sides of the relationship have a maximum cardinality of one.

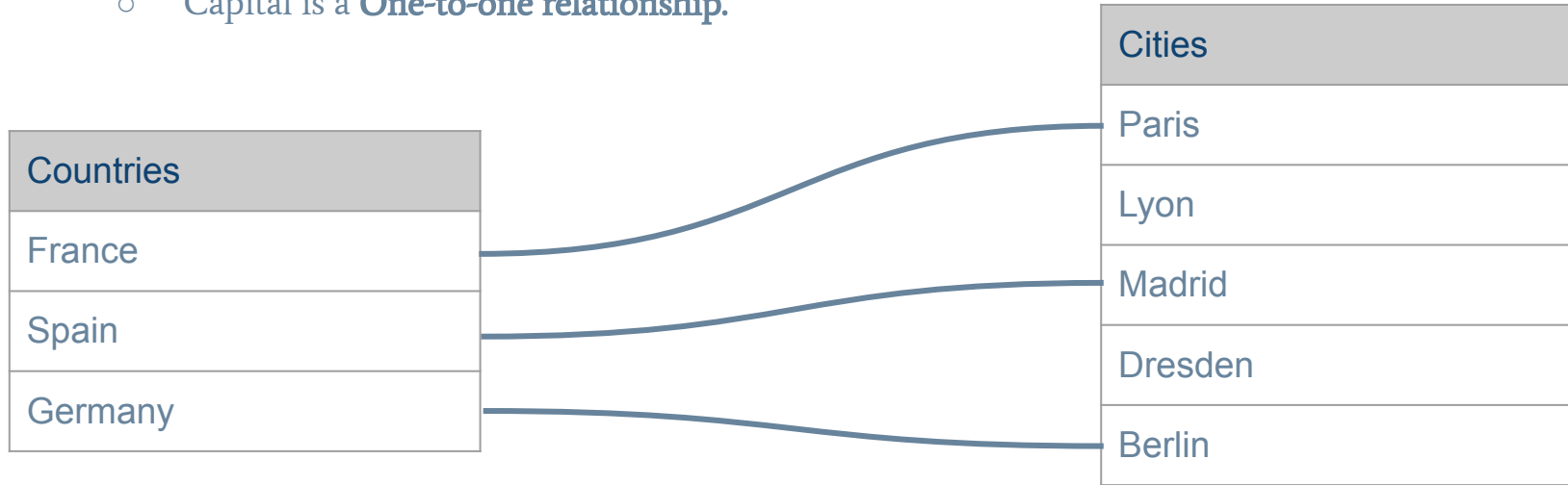
Cardinality of a Relationship - Examples - One-To-One

- We have two cardinalities here:
 - Countries with 1..1: A country has exactly one capital.
 - Cities 0..1: A city is either not a capital (0) or is the capital of a single country (1)
 - Capital is a **One-to-one relationship**.



Cardinality of a Relationship - Examples - One-To-One

- We have two cardinalities here:
 - Countries with 1..1: A country has exactly one capital.
 - Cities 0..1: A city is either not a capital (0) or is the capital of a single country (1)
 - Capital is a **One-to-one relationship**.



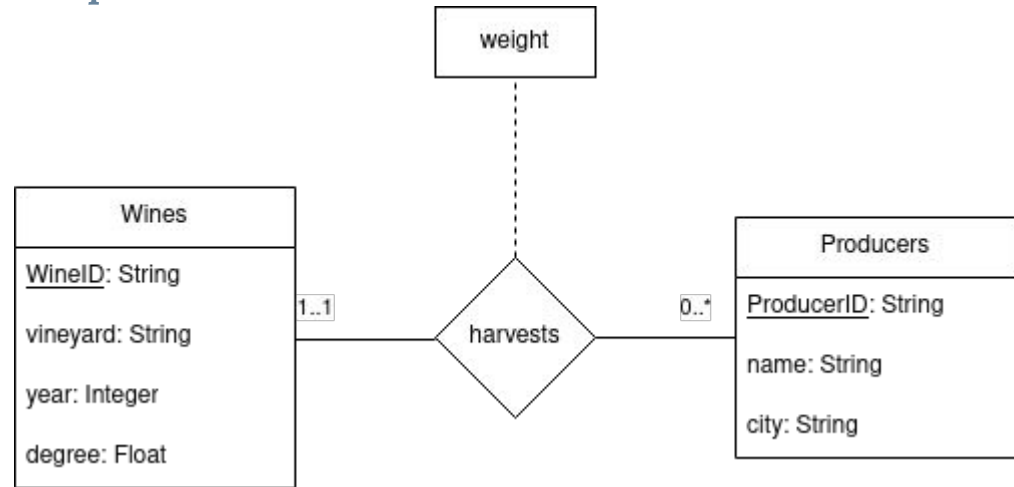
Cardinality of a Relationship - Cardinality types

There are three types of cardinality for a binary relationship:

- **One-to-one:** Both sides of the relationship have a maximum cardinality of one.
- **One-to-many:** One side of the relationship has a maximum cardinality of one, the other has either no limit for the maximum (*) or a maximum strictly more than one.

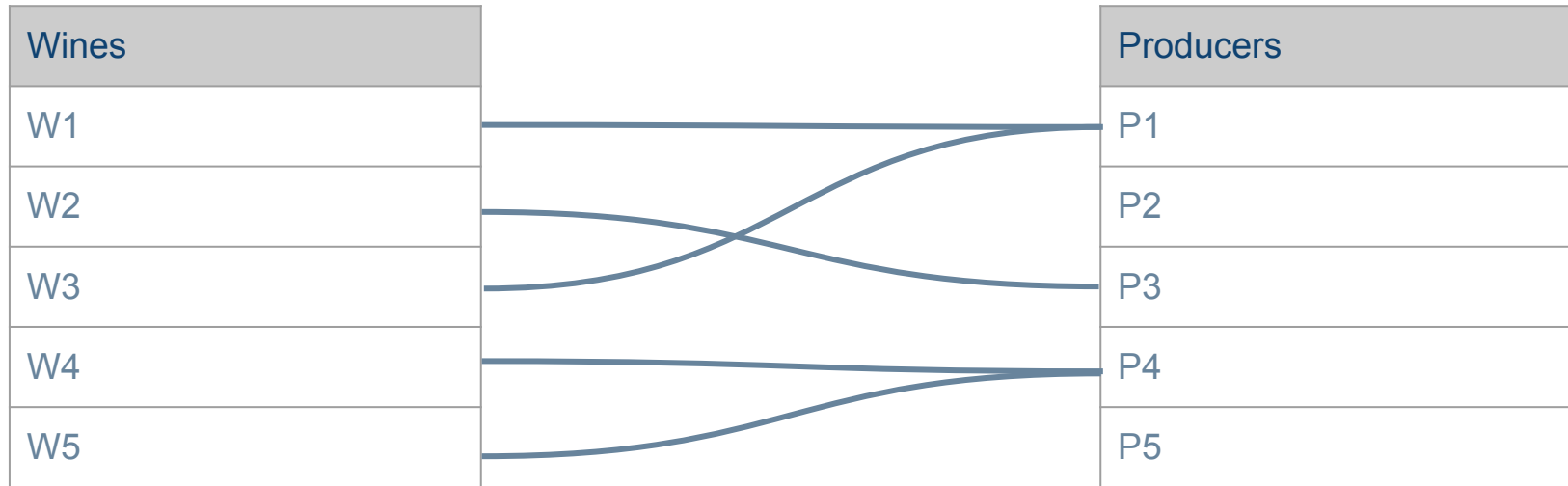
Cardinality of a Relationship - Examples - One-To-Many

- We have two cardinalities here:
 - Wines with 1..1: Each wine is harvested by exactly one producer (min=max=1)
 - Producers with 0..*: Each producer harvests between 0 and any number of wine.
 - Harvests is a **One-to-many relationship**.



Cardinality of a Relationship - Examples - One-To-Many

- We have two cardinalities here:
 - Wines with 1..1: Each wine is harvested by exactly one producer (min=max=1)
 - Producers with 0..*: Each producer harvests between 0 and any number of wine.
 - Harvests is a **One-to-many relationship**.



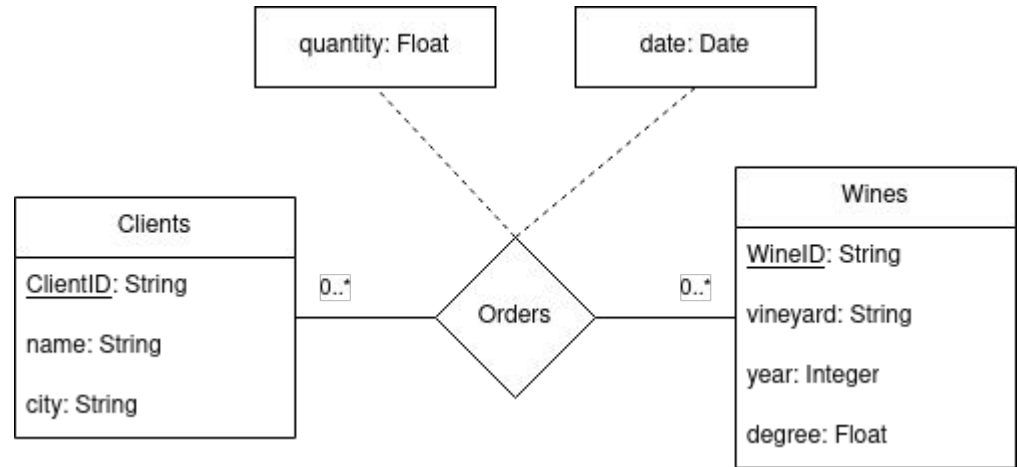
Cardinality of a Relationship - Cardinality types

There are three types of cardinality for a binary relationship:

- **One-to-one:** Both sides of the relationship have a maximum cardinality of one.
- **One-to-many:** One side of the relationship has a maximum cardinality of one, the other has either no limit for the maximum (*) or a maximum strictly more than one.
- **Many-to-many:** Both sides of the relationship have either no limit for the maximum (*) or a maximum strictly more than one.

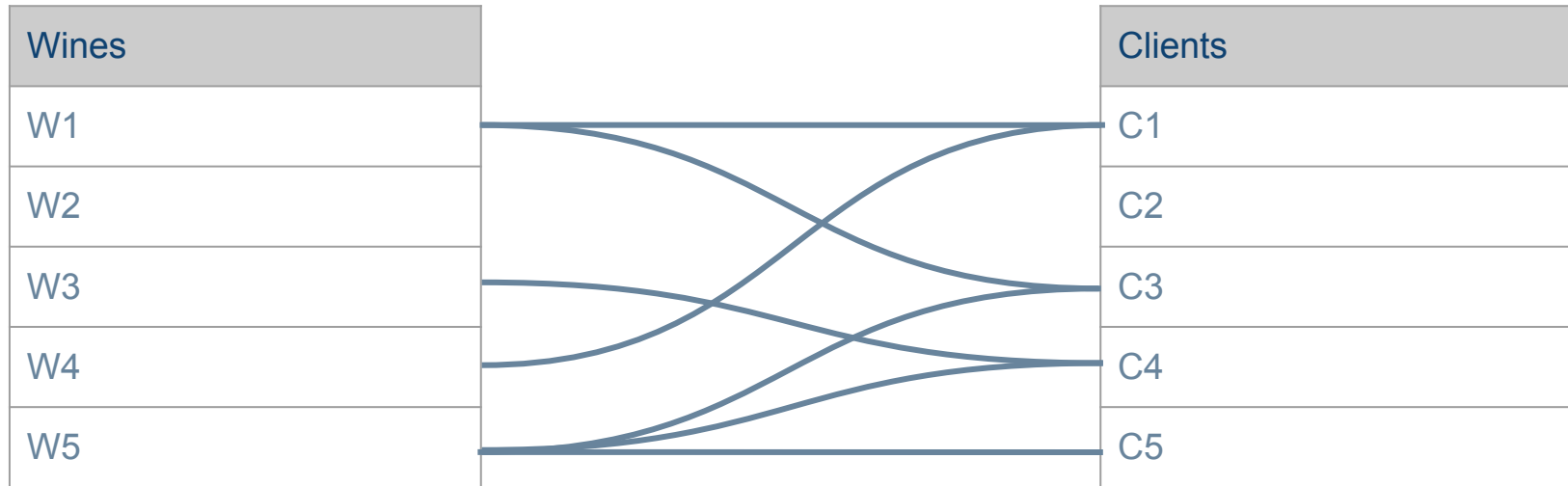
Cardinality of a Relationship - Examples - Many-To-Many

- We have two cardinalities:
 - Clients with 0..*: A client can order any quantity of wines.
 - Wines with 0..*: A wine can be ordered by any number of clients.
 - Orders is a **Many-to-many relationship**.



Cardinality of a Relationship - Examples - Many-To-Many

- We have two cardinalities:
 - Clients with 0..*: A client can order any quantity of wines.
 - Wines with 0..*: A wine can be ordered by any number of clients.
 - Orders is a **Many-to-many relationship**.



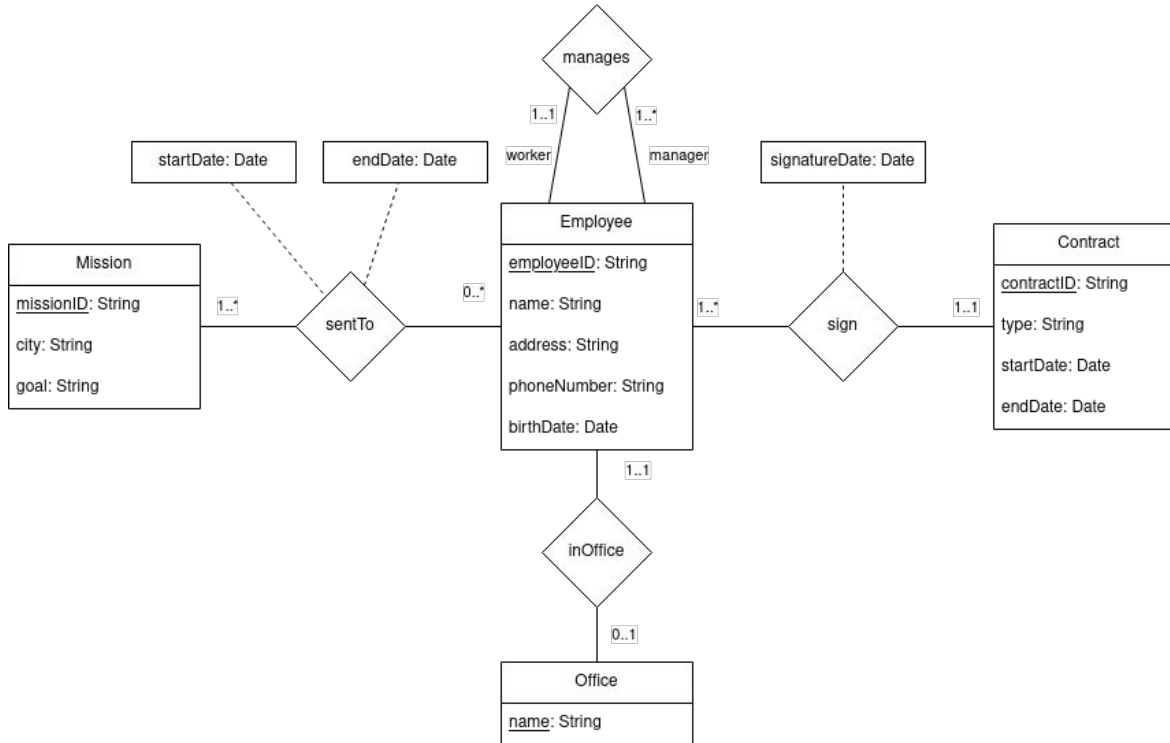
Example - Identifying the Cardinalities

Our client wants to design an application to manage her employees. When she recruits someone, this person has to fill a form to give his name, address, phone number, and birthdate. Then, they sign at a given date a contract that has a type (CDD/CDI), a start date, and an end date. If the signed contract is of predefined length (CDD), it is possible to renew it by signing a new contract. Each employee has exactly a manager, and a manager manages several employees. Each employee is also assigned to an office with a name. An office can only be occupied by a single employee. An employee can be sent to a mission in a particular city with a specific goal (e.g. develop a new product). A mission can be assigned to several employees. An assignment has a start and an end date.

Example - Identifying the Cardinalities

Our client wants to design an application to manage her employees. When she recruits someone, this person has to fill a form to give his name, address, phone number, and birthdate. Then, they sign at a given date a contract that has a type (CDD/CDI), a start date, and an end date. If the signed contract is of predefined length (CDD), it is possible to **renew it by signing a new contract**. Each employee **has exactly a manager**, and **a manager manages several employees**. Each employee is also assigned to an office with a name. **An office can only be occupied by a single employee**. **An employee can be sent to a mission** in a particular city with a specific goal (e.g. develop a new product). **A mission can be assigned to several employees**. An assignment has a start and an end date.

Example - E/R Diagram With Cardinalities



Removing Redundancy In ER Diagram

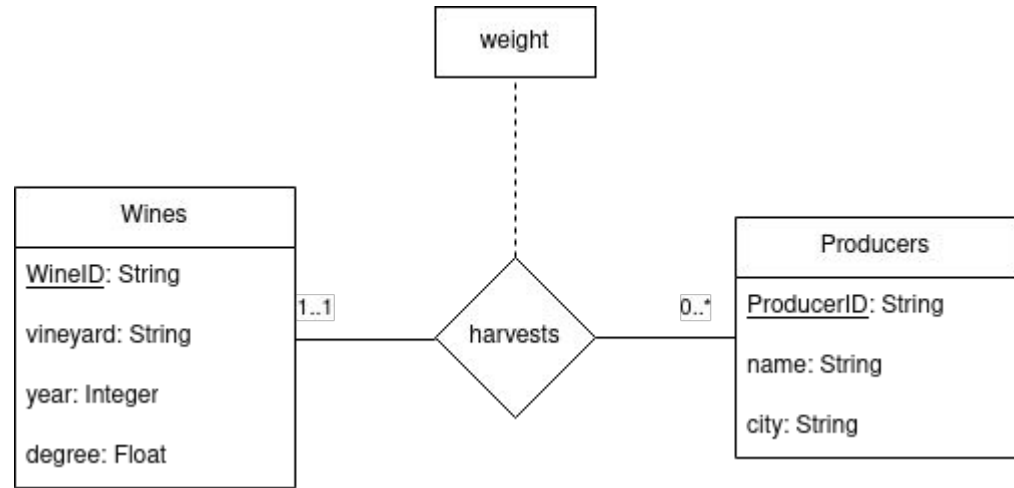
- An attribute is redundant if it appears in two tables, but in fact represents a relationship.
- Here, ProducerID is redundant in Wines, and can be represented by a relationship.

Wines
<u>WineID</u> : String
vineyard: String
year: Integer
degree: Float
ProducerID: String

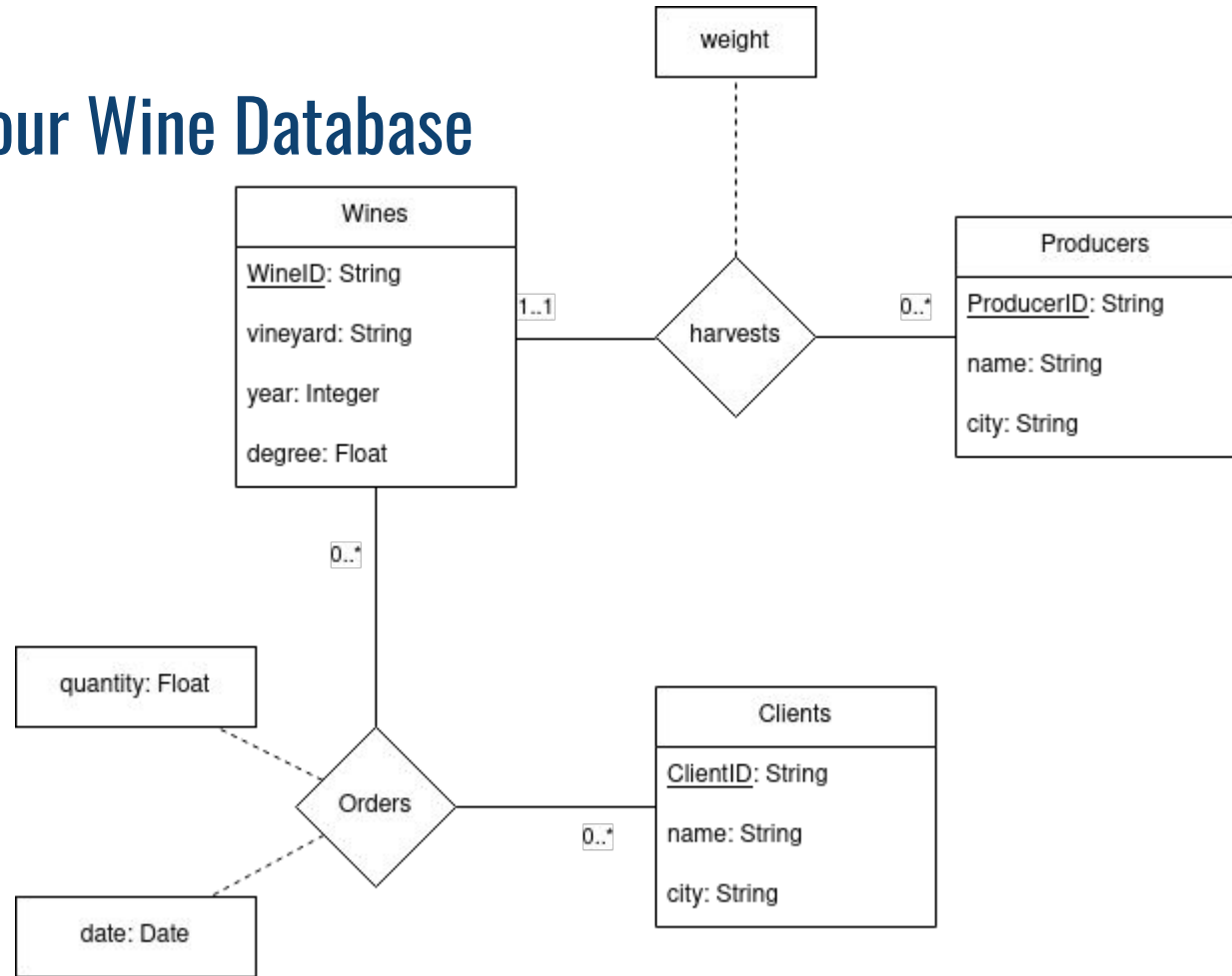
Producers
<u>ProducerID</u> : String
name: String
city: String

Removing Redundancy In ER Diagram

- An attribute is redundant if it appears in two tables, but in fact represents a relationship.
- Here, ProducerID is redundant in Wines, and can be represented by a relationship.



Final ER Diagram for our Wine Database



From E/R to a Database Schema

Translation of Entity Sets

- We start by translating each entity set. For each entity set, we create a new relation schema where:
 - the name is the name of the entity set
 - the attributes are the ones of the entity set
 - the primary key is also the same.

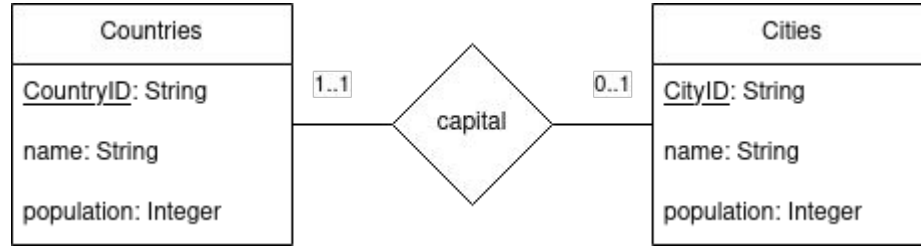
Wines
<u>WineID</u> : String
vineyard: String
year: Integer
degree: Float

becomes Wines(WineID: String, vineyard: String, year: Integer, degree: Float)

Translation of Relationships

- We create a new relation schema for each relationship. For each relationship:
 - The name of relation schema is the name of the relationship.
 - The attributes of the relationship are the ones in the E/R diagram **PLUS** the primary keys of all entity sets involved in the relationship. Note that these new attributes are also foreign keys.
 - We choose the primary key depending on the cardinality type:
 - For a one-to-one relationship: We take as primary key the primary key of one of the entity set in the relationship (not important which one)
 - For a one-to-many relationship: We take as primary key the primary key of the entity set on the side of the “one”.
 - For a many-to-many relationship: We take as the primary key all the primary keys of the entity sets in the relationship.

Translation of Relationships - Example - One-To-One



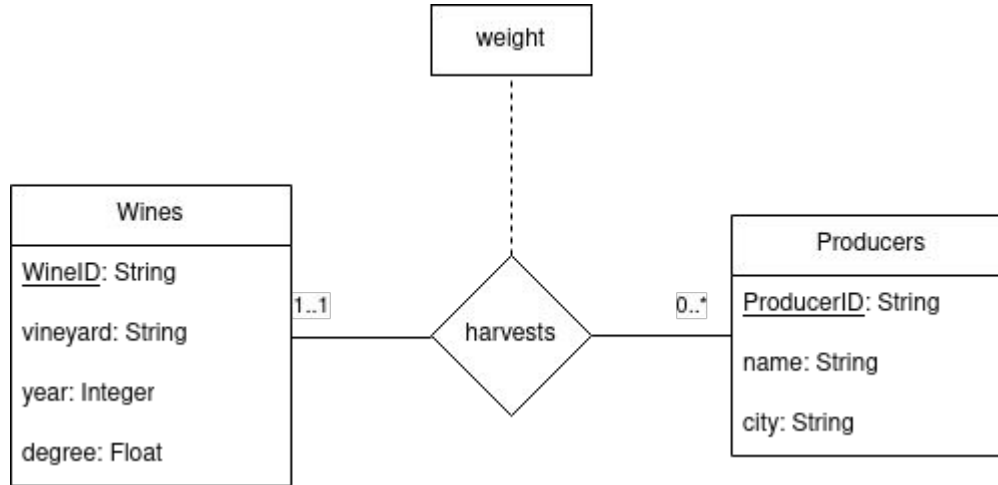
becomes:

capital(CountryID: String, cityID: String)

Or

capital(CityID: String, CountryID: String)

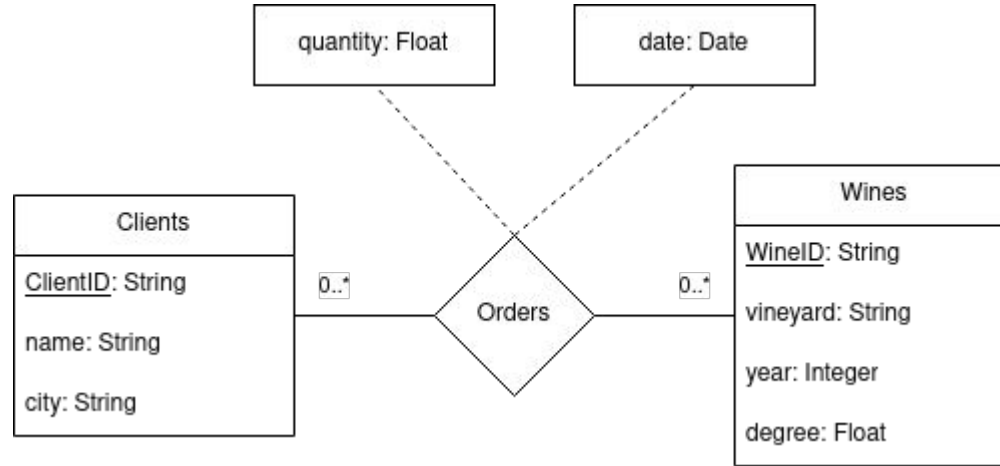
Translation of Relationships - Example - One-To-Many



becomes:

harvests(WineID: String, ProducerID: String, weight: Float)

Translation of Relationships - Example - Many-To-Many



becomes:

Orders(ClientID: String, WineID: String, quantity: Float, date: Date)

Summary

- Before working with a database, we need to design it, i.e. create the database schema from the client needs.
- E/R diagram help representing the design. It is composed of:
 - Entities with attributes and a primary key
 - Relationships with relationship attributes and cardinalities.
- Once we have an E/R diagram, we can translate it to a database schema.