Databases with *MYSQL*

4. E-R to Relational Model

Dante – Digital Area for Networking Teachers and Educators

Funded by the Erasmus+ Programme of the European Union

Learning Outcomes

After this lesson, the learner will be able to convert the Entitity-Relationship Diagram into the Relational Model. The learner should be able to follow the following rules:

- Rule #1 (1 • :1 •)
- Rule #2 (1:1•)
- Rule #3 (1:1)
- Rule #4 (1:N•)
- Rule #5 (1:N)
- Rule #6 (M:N)
- Rule #7 (M:N:P)
- Rule #8 (Sub-Entities)

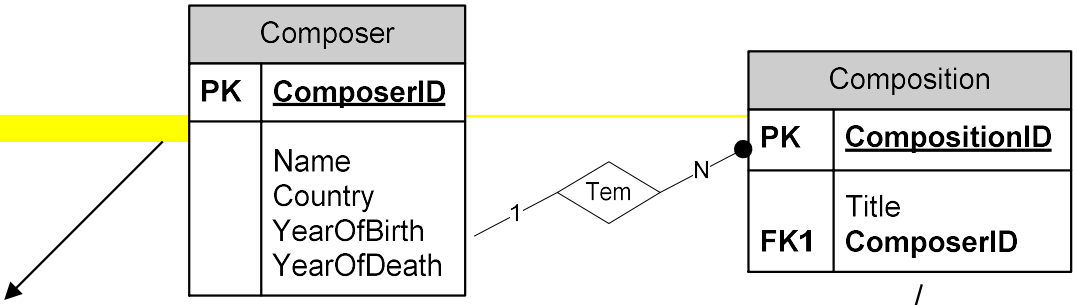
Funded by the Erasmus+ Programme of the European Union

Relational Model

The Relational Model is a logical model based on records. In this model, data and the relationships between data are represented by tables, and all tables in the database and their respective attributes and primary keys are described in it.

Funded by the Erasmus+ Programme of the European Union

Relational Model



ComposerID	Name	Country	YearOfBirth	YearOfDeath
B33	Johann Sebastian Bach	Germany	1685	1750
B76	Ludwig van Beethoven	Germany	1770	1827
B90	Irving Berlin	America	1888	1989
B113	Johannes Brahms	Germany	1833	1897
G18	George Gershwin	America	1898	1937
G80	Orlando Gibbons	England	1583	1625

Primary keys are obvious – they uniquely identify a row.

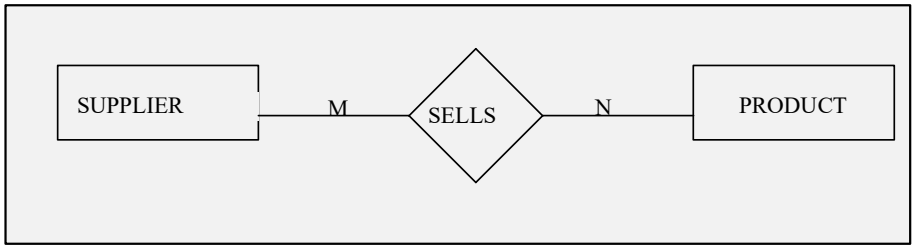
Foreign keys are obvious – they reference a field (column) in another table.

CompositionID	Title	ComposerID
111	Rhapsody in Blue	G18
222	Second Rhapsody	G18
333	Academic Festival Overture	B113
444	Tragic Overture	B113
555	Ein deutsches Requiem	B113

Funded by the Erasmus+ Programme of the European Union

Relational Model

- The entities in the Entity Relationship Diagram (DER) turn into tables
- Entity attributes translate into table fields
- A Table is made up of columns (fields) and rows (records)



Chen Model

Funded by the Erasmus+ Programme of the European Union

Exemple M:N Relationship

Nome de Tabela: Fornecedor
Chave Primária: cod_fornecedor
Chave Forasteira: nenhuma

cod_fornecedor	nome_fornecedor
1	Copiconta Lda
2	Xerox Lda
4	Recicla Lda
0	

Nome de Tabela: Artigo_Fornecedor
Chave Primária: cod_artigo, cod_fornecedor
Chaves Forasteiras: cod_artigo, cod_fornecedor

cod_artigo	cod_fornecedor
1	2
1	4
2	2
2	4
4	4
0	0

Nome de Tabela: Artigo
Chave Primária: cod_artigo
Chave Forasteira: nenhuma

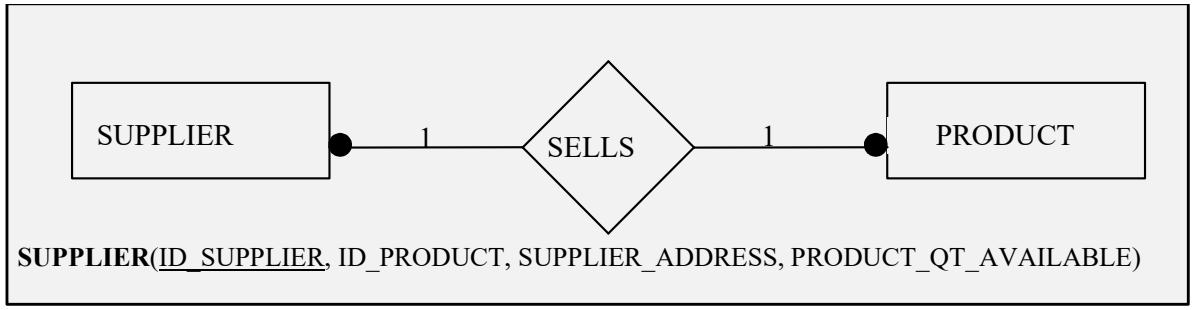
cod_artigo	desc_artigo
1	Papel
2	computador
3	Lapis
4	cartolina
0	

Funded by the Erasmus+ Programme of the European Union

Conversion DER to tables Rule #1 (*1 • :1 •*)

If there is a binary relationship between two entities with mandatory participation in both and their cardinality is of degree ***1:1***, then it is necessary to **define only one table**.

The primary key of this relationship can be the identifier of any of the entities.



In the previous definition, the supplier code was considered as the primary key, but the article code can also be considered as in the following example.

PRODUCT(ID_PRODUCT, ID_SUPPLIER, SUPPLIER_ADDRESS, PRODUCT_QT_AVAILABLE)

Funded by the Erasmus+ Programme of the European Union

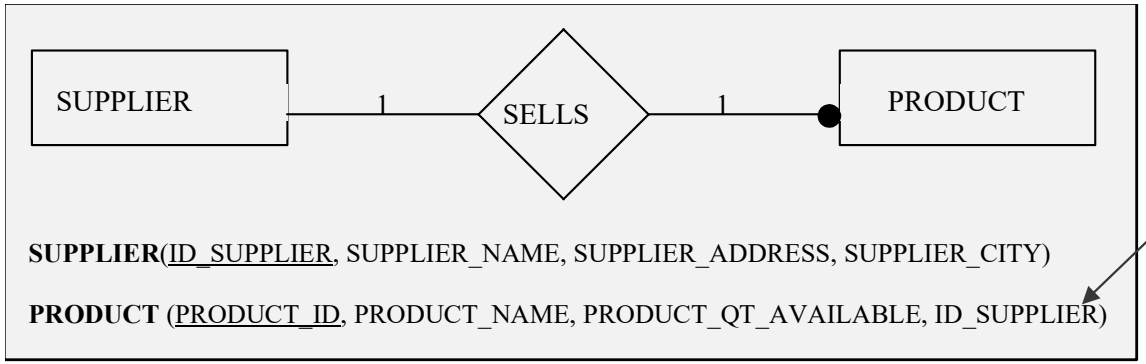
Conversion DER to tables Rule #2 (*1:1●*)

If a binary relationship exists between two entities with mandatory participation in one of the entities and its cardinality is of degree ***1:1***, then it is necessary to **define two tables**.

The identifier of each entity must be the primary key in the corresponding table.

The primary key of the entity with optional participation must be used as an attribute in the table corresponding to the entity whose participation is mandatory.

This attribute is a foreign key, as it is a primary key in the other table and is intended to relate the two tables.



ID_SUPPLIER
is a foreign key in the PRODUCT table

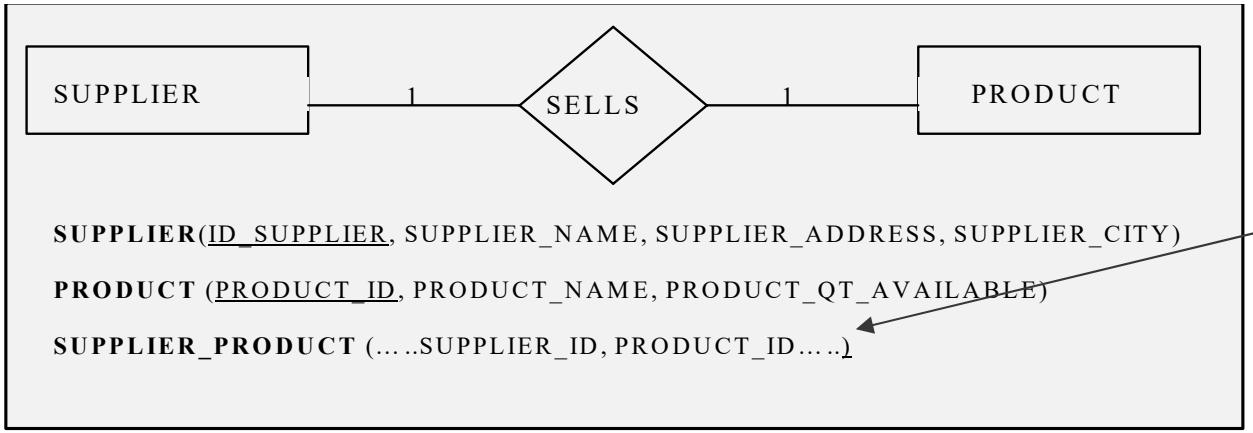
Funded by the Erasmus+ Programme of the European Union

Conversion DER to tables Rule #3 (*1:1*)

If a binary relationship exists between two entities with optional membership in both entities and its cardinality is of degree ***1:1***, then it is necessary to **define three tables**.

The identifier of each entity serves as the primary key in the corresponding table. The table that corresponds to the relationship between the two entities has among its attributes the identifiers of both entities.

The attributes that make up the relationship are foreign keys, as these are primary keys of the other two tables and are intended to relate the two tables.



SUPPLIER_ID, PRODUCT_ID are foreign keys in the SUPPLIER_PRODUCT table

Funded by the Erasmus+ Programme of the European Union

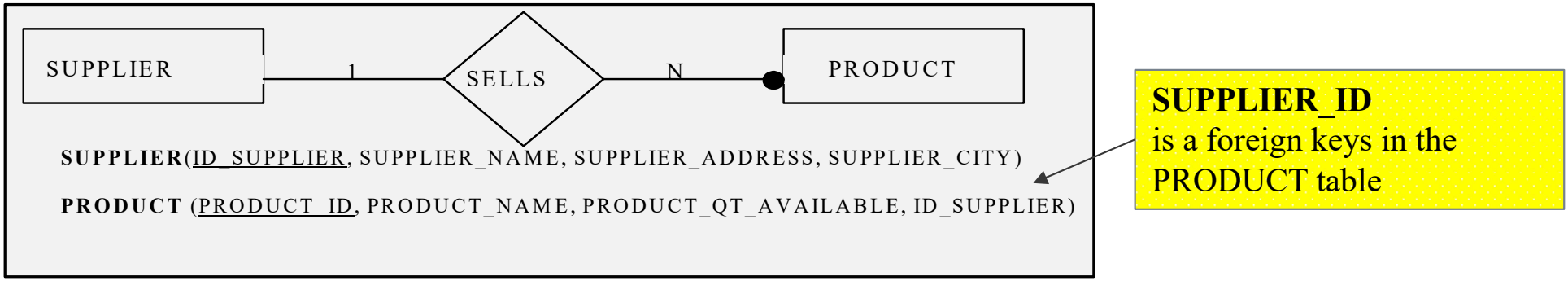
Conversion DER to tables Rule #4 (*1:N●*)

If a binary relationship exists between two entities of degree ***1:N*** with mandatory participation on the N side, then it is necessary to define **two tables**.

It should be noted that the type of participation on side 1 is irrelevant for the definition of the number of tables.

The identifier of each entity must be the primary key in the corresponding table.

The primary key of the entity on side *1* is used as a field in the table corresponding to the entity on side *N*, this field being a foreign key.



Funded by the Erasmus+ Programme of the European Union

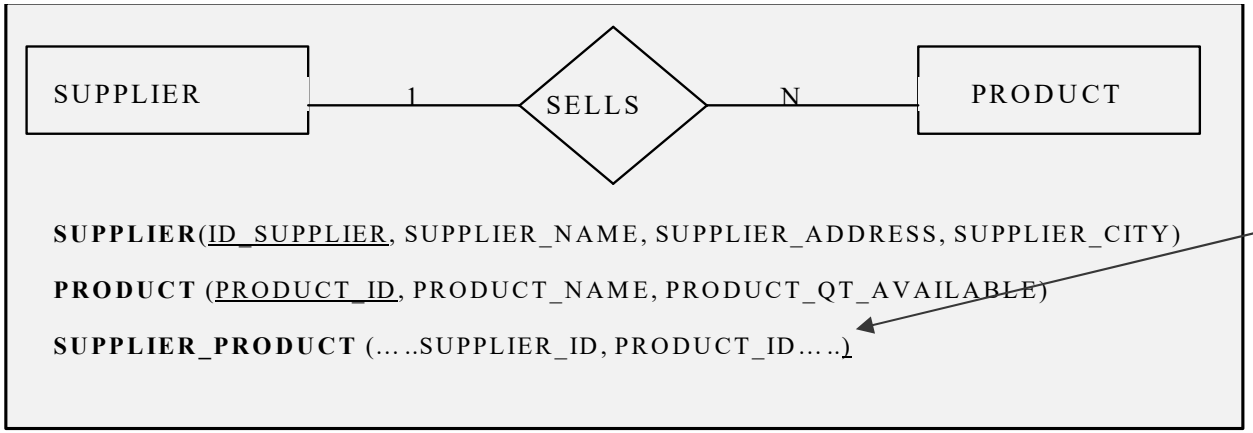
Conversion DER to tables Rule #5 (*1:N*)

If a binary relationship exists between two entities of degree ***1:N*** with optional participation on the N side, then it is necessary to **define three tables**.

It should be noted that the type of participation on side 1 is irrelevant for the definition of the number of tables.

The identifier of each entity must be the primary key in the corresponding table.

The table that corresponds to the relationship between the two entities has among its fields the identifiers of both entities.



SUPPLIER_ID, PRODUCT_ID are foreign keys in the SUPPLIER_PRODUCT table

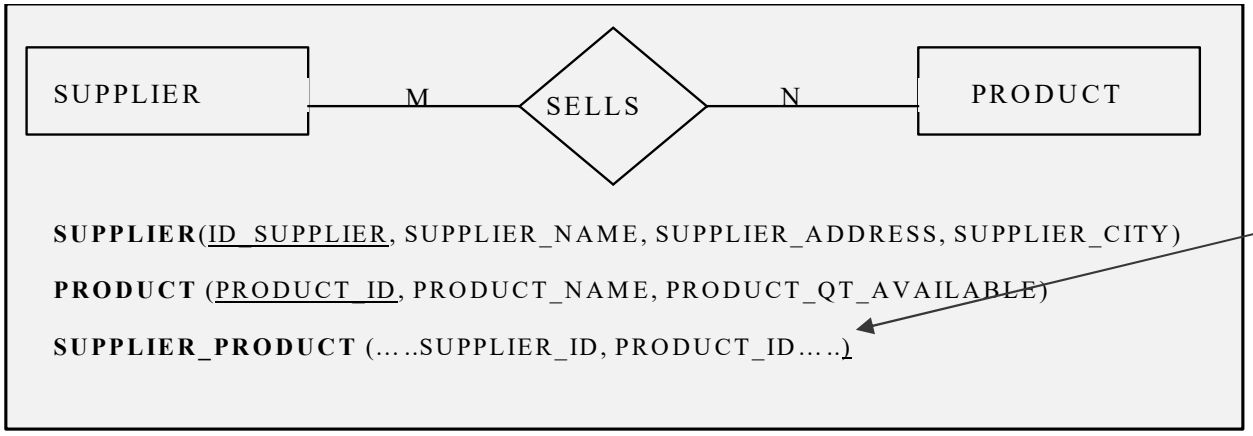
Funded by the Erasmus+ Programme of the European Union

Conversion DER to tables Rule #6 (*M:N*)

If there is a binary relationship between two ***M:N*** entities, it is always necessary to define **three tables whatever the type of participation**.

The identifier of each entity must be the primary key in the corresponding table.

The table that corresponds to the relationship between the two entities has among its fields the identifiers of both entities.



SUPPLIER_ID, PRODUCT_ID are foreign keys in the SUPPLIER_PRODUCT table

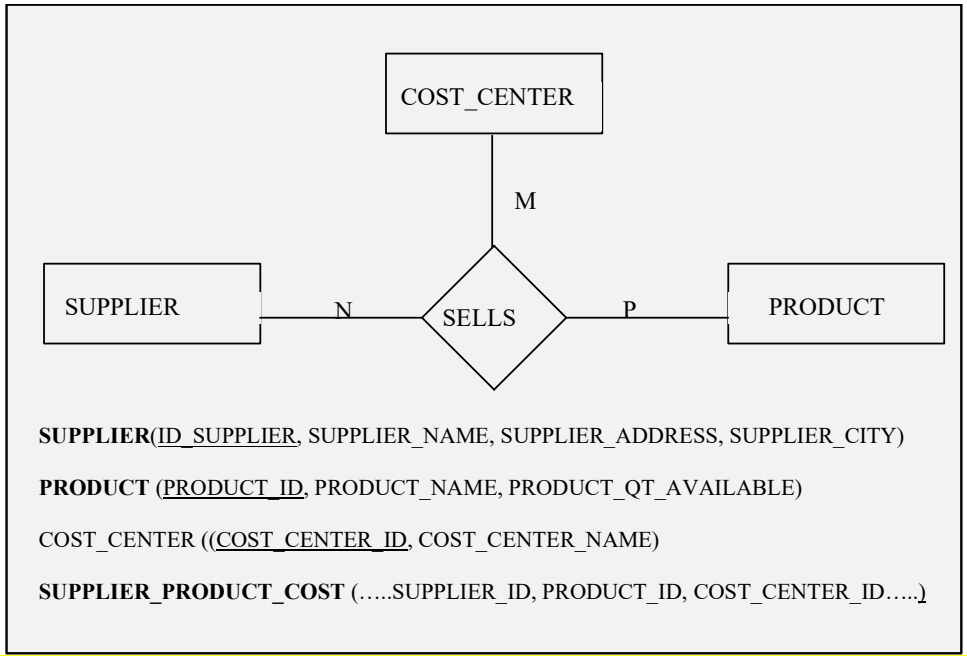
Funded by the Erasmus+ Programme of the European Union

Conversion DER to tables Rule #7 (*M:N:P*)

If a trinary (or higher) relationship exists between three entities of degree ***M:N:P*** regardless of the membership types, **it is always necessary to define one table for each entity and another table for the relationship between the entities**.

The identifier of each entity serves as the primary key in the corresponding table.

The table that corresponds to the relationship has among its fields the dentifiers of all the entities.



SUPPLIER(ID_SUPPLIER, SUPPLIER_NAME, SUPPLIER_ADDRESS, SUPPLIER_CITY)

PRODUCT (PRODUCT_ID, PRODUCT_NAME, PRODUCT_QT_AVAILABLE)

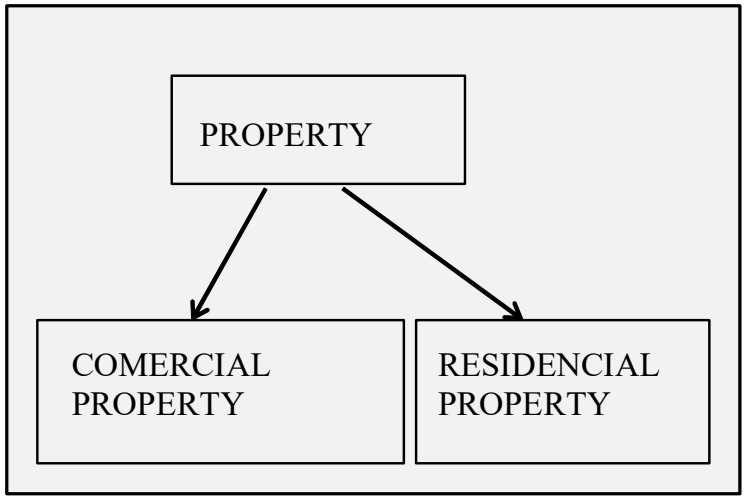
COST_CENTER ((COST_CENTER_ID, COST_CENTER_NAME)

SUPPLIER_PRODUCT_COST (…..SUPPLIER_ID, PRODUCT_ID, COST_CENTER_ID…..)

Funded by the Erasmus+ Programme of the European Union

Conversion DER to tables Rule #8 (Sub-Entities)

f there are **subsets of entities**, it is always necessary to **define a table for the source entity and a table for each sub-entity**. **Sub-entities have the same primary key as the source entity.**



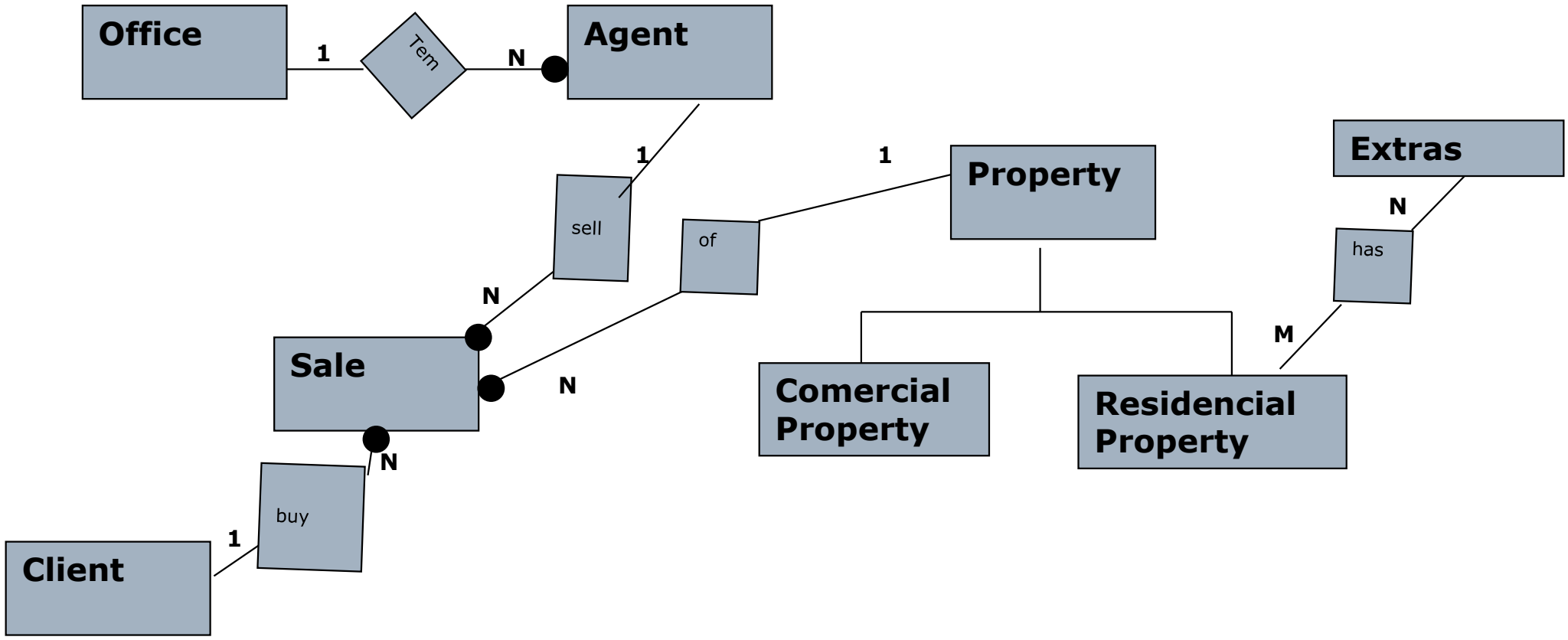
PROPERTY (PROPERTY_ID, ADDRESS_PROPERTY, SQUARE_METERS, TAXES, CLOSE_DATE)

COMERCIAL_PROPERTY (PROPERTY_ID, NUMBER_OFFICES, NUMBERE_DOCKING_BAYS)

RESIDENCIAL_PROPERTY (PROPERTY_ID, NUMBER_BEDROOMS, NUMBER_BATHS)

Funded by the Erasmus+ Programme of the European Union

Conversion ERD into tables - Example



Funded by the Erasmus+ Programme of the European Union

Conversion ERD into tables - Example

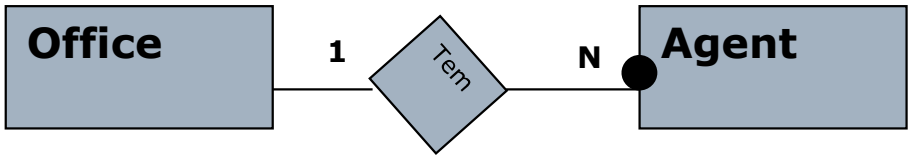
1. One table for each entity:

Office (office_id, office_name, morada, telefone)
Agent (agent_id, agent_name, agent_phone, agent_email)
Client (client_id, cliente_name, cliente_address, cliente_phone)
Sale (valor)
Property (Property_id, Address_property, Square_meters, Taxes, Close_date)
Comercial_Property (Property_id, Number_offices, Numbere_docking_bays)
Residencial_Property (Property_id, Number_bedrooms, Number_baths)
Extra (extra_id, extra_name)

2. Analize Relationships

Funded by the Erasmus+ Programme of the European Union

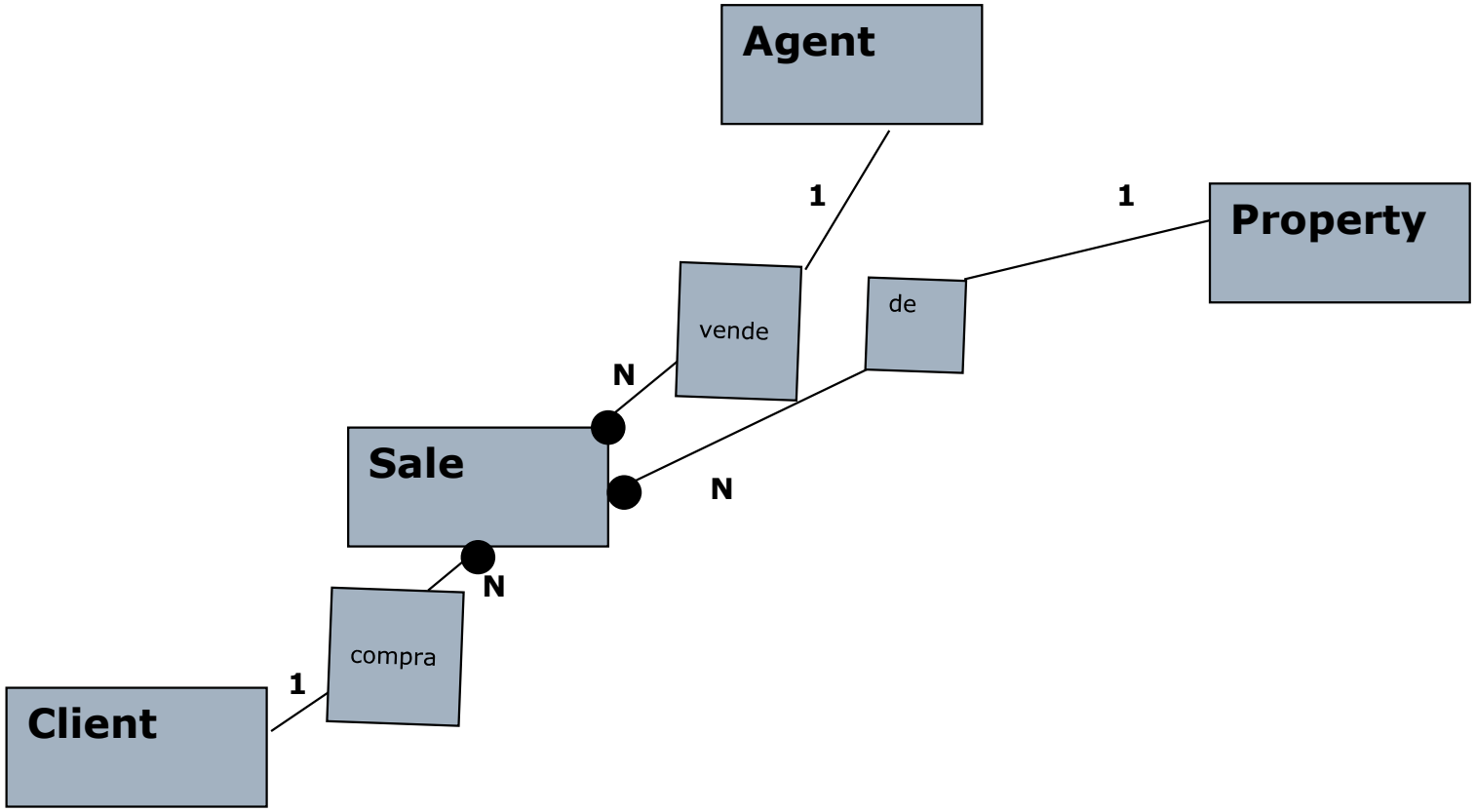
Conversão para tabelas – Exemplo
Analisar Relacionamentos



Office (<u>office_id</u>, office_name, office_address, office_phone)
Agent (<u>agent_id</u>, agent_name, agent_phone, agent_email, **office_id FK**)

Funded by the Erasmus+ Programme of the European Union

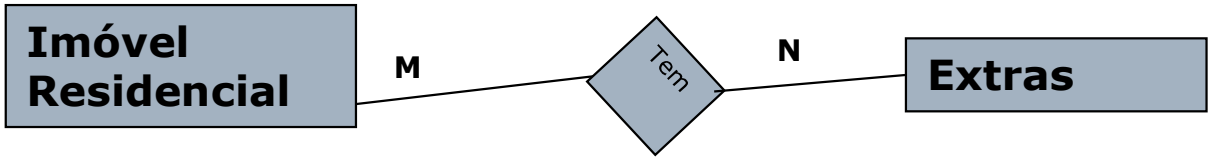
Conversão para tabelas – Exemplo
Analisar Relacionamentos



Sale (sale_id, **property_id FK, agente_id FK, cliente_id FK, price**)

Funded by the Erasmus+ Programme of the European Union

Conversão para tabelas – Exemplo
Analisar Relacionamentos



Residencial_Property (property_id, number_bedrooms, number_baths)
Extra (extra_id, descrição)
Extra_Property (property_id, extra_id)

Funded by the Erasmus+ Programme of the European Union

Conversão para tabelas - Exemplo

Office (office_id, office_name, office_address, office_phone)

Agent (agente_id, agente_name, agente_phone, agente_email, **office_id FK**)

Cliente (idcliente, nome, morada, telefone)

Sale (sale_id, **property_id FK, agente_id FK, cliente_id FK, price**)

Property (Property_id, Address_property, Square_meters, Taxes, Close_date)

Comercial_Property (Property_id, Number_offices, Numbere_docking_bays)

Residencial_Property (property_id, number_bedrooms, number_baths)

Extra (extra_id, extra_name)

Extra_Property (property_id, extra_id)