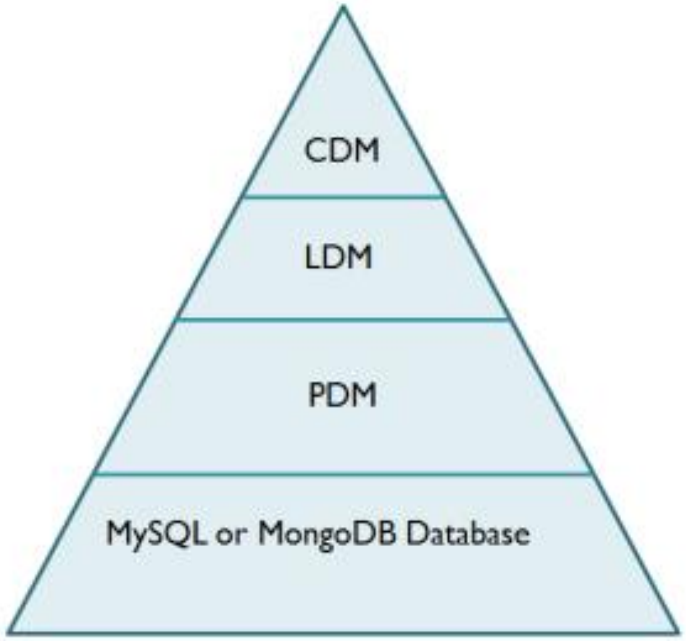
Entity Relationship Diagrams

Contents

- Four Levels of Design



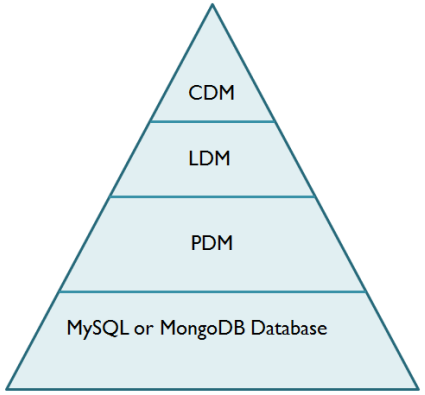
- Entity-Relationship (ER) Modeling
  1. Specify the Entities
  2. Specify Relationships
  3. Specify Connectivity and Optionality
  4. Add the attributes

Conceptual Data Modelling

- Conceptual Data modelling is the process of capturing the satellite view of the business requirements

- What problem does the business need to solve?

- What do these ideas or concepts mean?



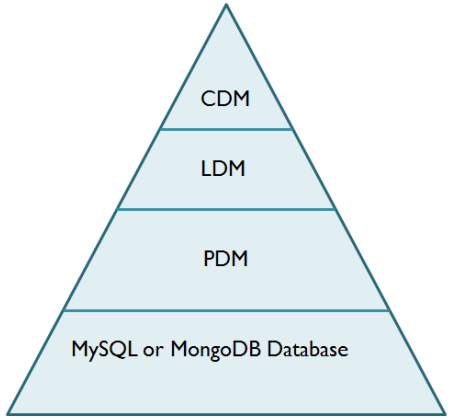
Concept Explanation

- A concept is a key idea that is both basic and critical to your audience.
- "Basic" – it's probably mentioned many times a day with the people who are the audience for the model.
- "Critical" – means the business would be very different or non-existent without this concept.
- Examples: Customer, Employee, Product

Logical Data Modelling

- Logical data modelling is the process of capturing the detailed business solution.

- Note: The logical data model will look the same regardless of whether we are implementing in MySQL or Oracle

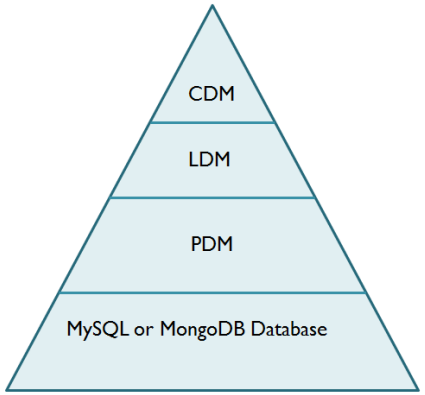


Logical Data Modelling

- Conceptual data modelling will reveal that a Customer may place many Orders.
- Logical data modelling will uncover all of the details behind Customer and Order, such as the customer's name, their address, the order number, and what is being ordered.
- ERDs & Normalisation

Physical Data Modelling

- Physical data modelling is the process of capturing the detailed technical solution.
- After understanding both the high level and detailed business solution, we move on to the technical solution.
- We are focusing on the technology at this stage, e.g. MySQL etc.



Physical Data Modelling

- Objects such as tables and columns are created based on entities and attributes that were defined during logical modelling.
- Constraints are also defined, including primary keys, foreign keys, other unique keys, and check constraints.
- Views can be created from database tables to summarize data or to simply provide the user with another perspective of certain data.
- Physical modelling is when all the pieces come together to complete the process of defining a database for a business.

Physical Data Modelling

- Physical modelling is database software specific, meaning that the objects defined during physical modelling can vary depending on the database software being used.
- For example, most relational database systems have variations with the way data types are represented and the way data is stored, although basic data types are conceptually the same among different implementations.
- Some database systems have objects that are not available in other database systems.

Entity-Relationship (ER) Modeling

ERM and ERD

- ***Entity-Relationship Data Model (ERM)*** is a detailed, logical representation of the data for an organization or for a business area.
  - Expressed in terms of:
    - Entities
    - Attributes
    - Relationships

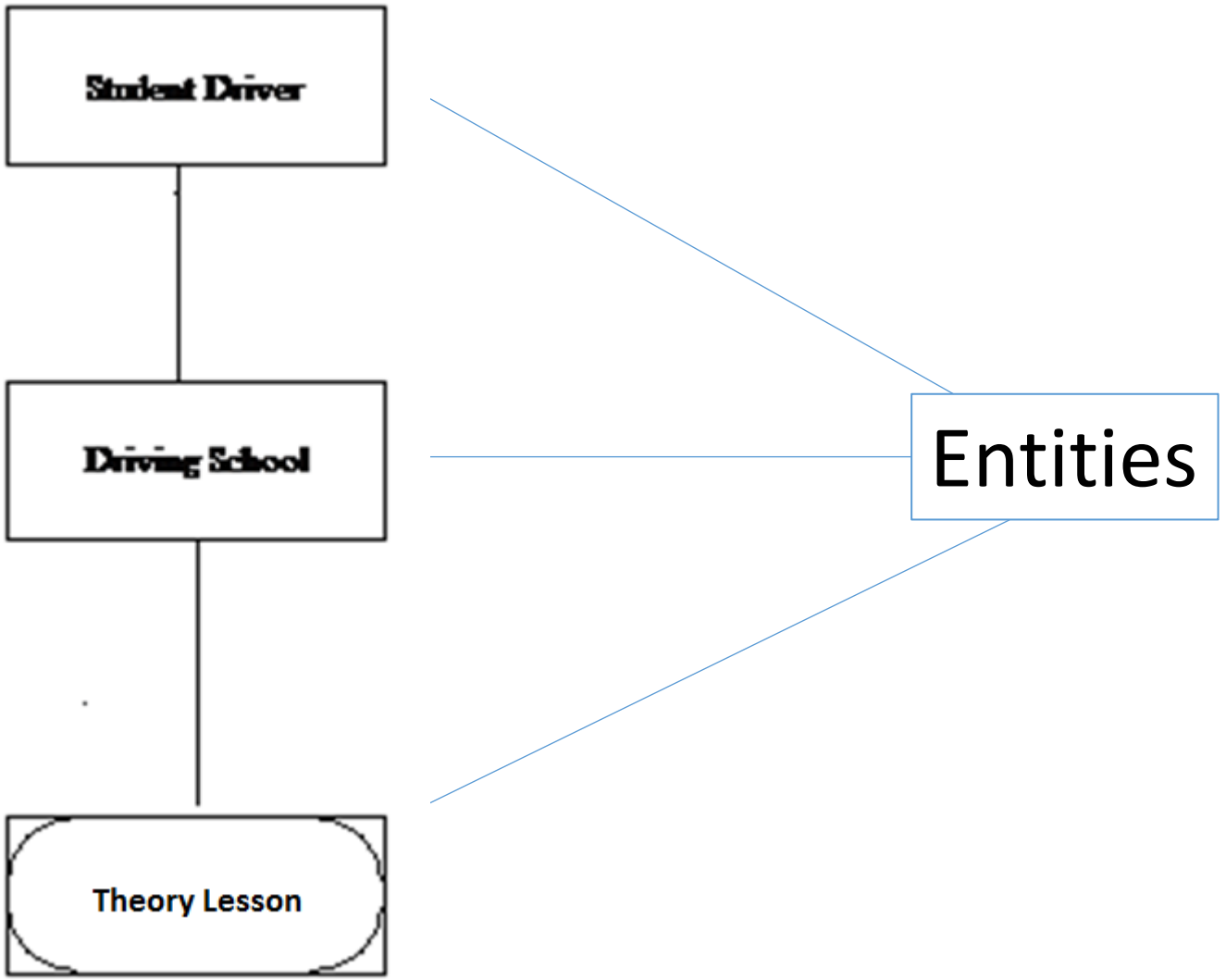
- ***Entity-Relationship Diagram (ERD)*** is a graphical representation of a Entity-Relationship Model.

Entities

- An entity represents a collection of information about something that the business deems important and worthy of capture

- Examples of entities:
  - Person: EMPLOYEE, STUDENT, PATIENT
  - Place: STORE, WAREHOUSE
  - Object: MACHINE, PRODUCT, CAR
  - Event: SALE,REGISTRATION, RENEWAL
  - Concept: ACCOUNT, COURSE

An Example with 3 entities

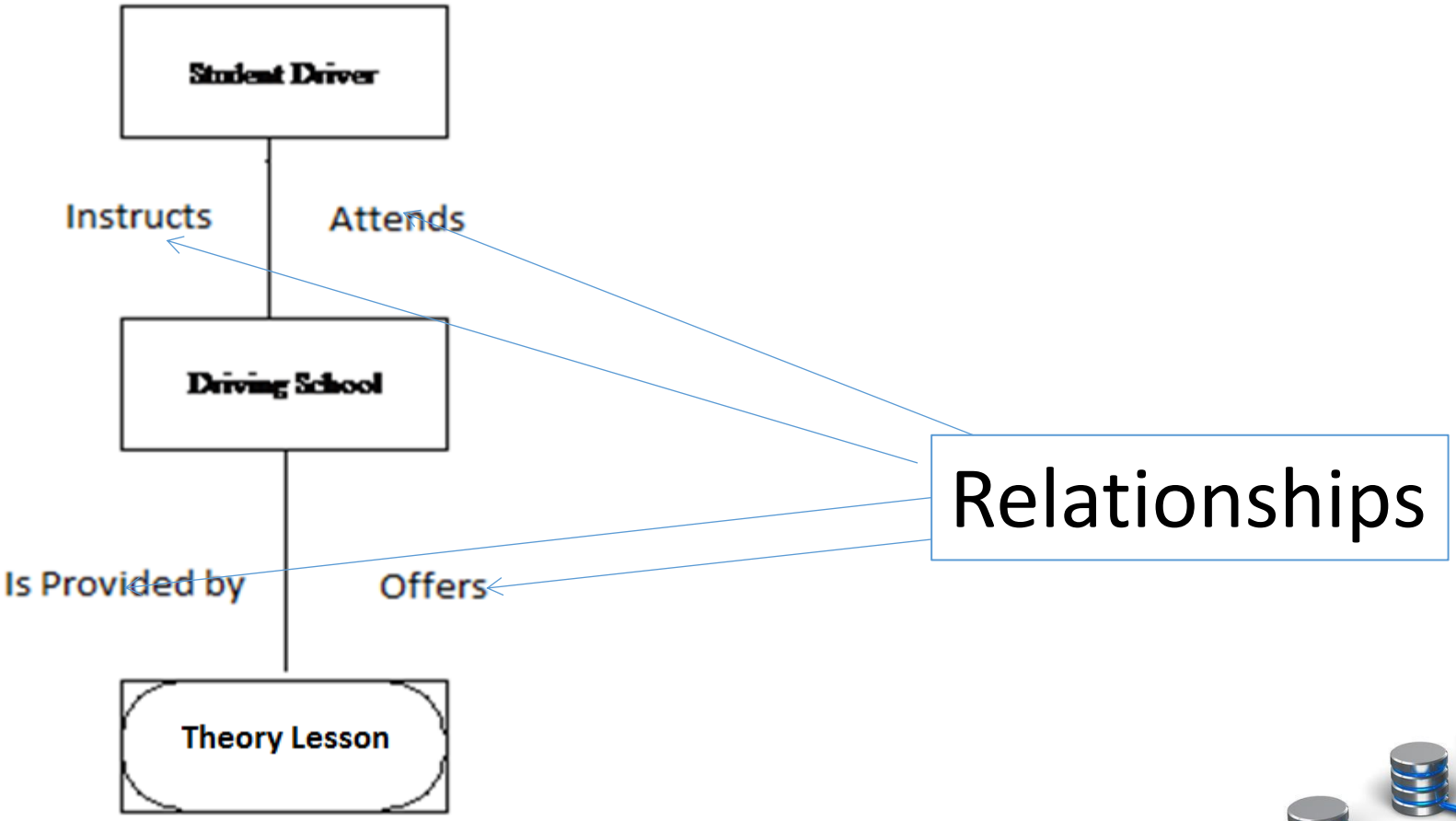


Relationships

- A relational model captures how the business works and contains business rules such as : A **Customer** must have at least one **Account**.

- Rules are captured on our data model through relationships. A relationship is displayed as a line connecting two entities.

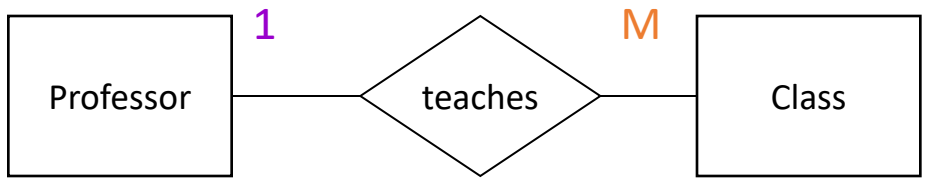
An Example

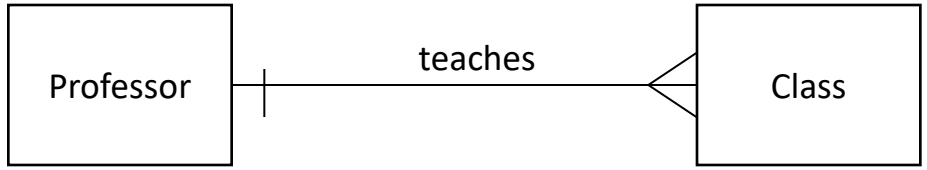

Student Driver
Instructs
Attends
Driving School
Relationships
Is Provided by
Offers
Theory Lesson

Cardinality

- Relationships can be classified as either
  - one – to – one
  - one – to – many
  - many – to –many

Cardinality





Cardinality

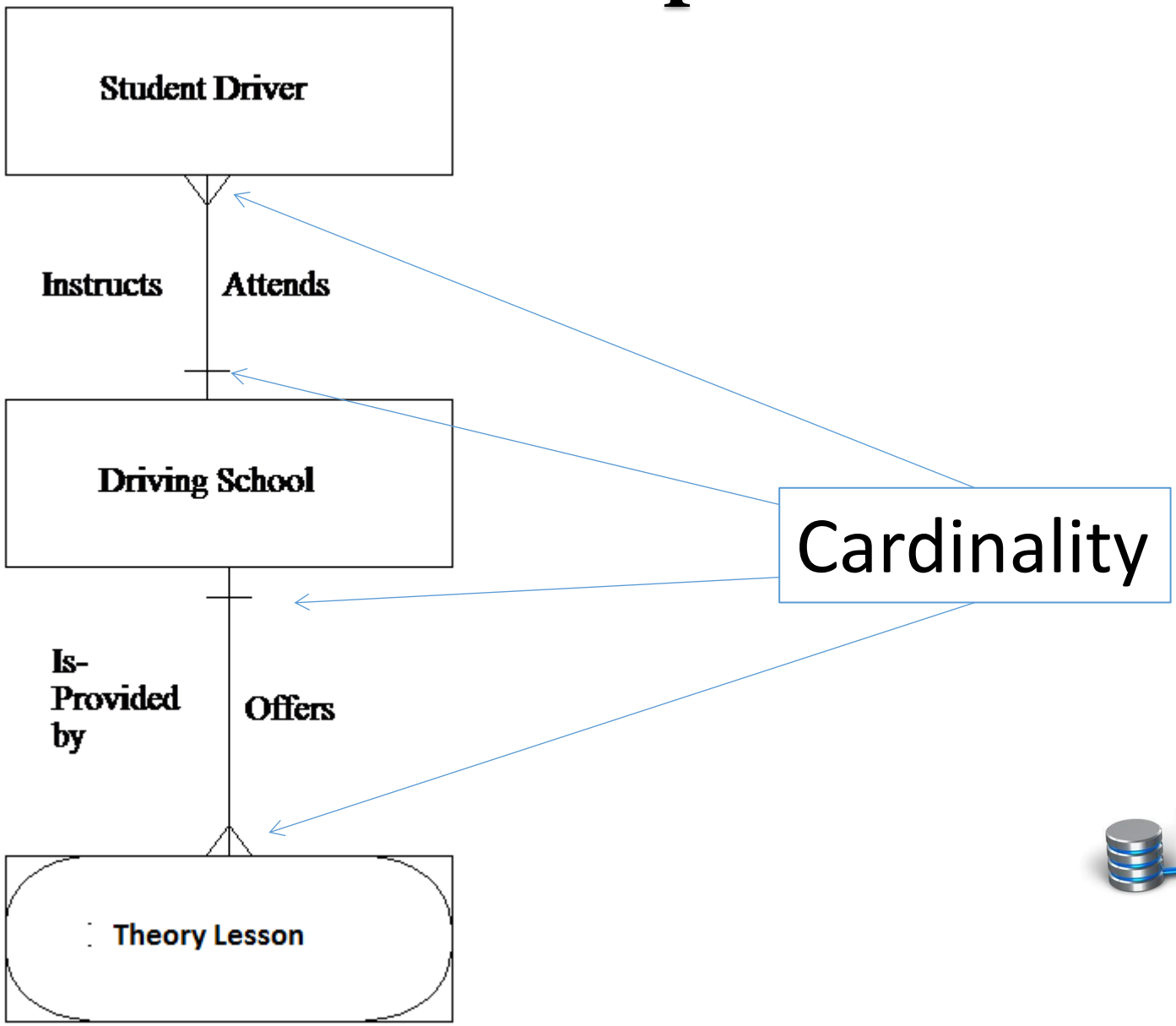
- Chen Model
  - 1 to represent one
  - M to represent many

1

M

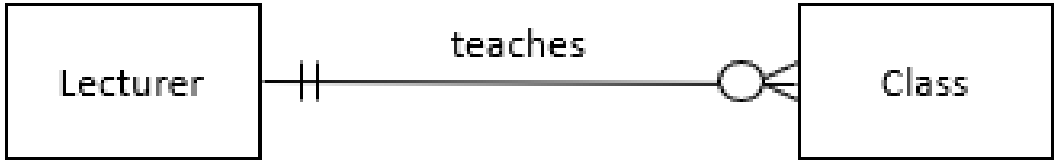
- Crow's Foot
  - One
  - many

An Example


Student Driver
Instructs
Attends
Driving School
Is-
Provided
by
Offers
Theory Lesson
Cardinality

Mandatory and Optional Relationships

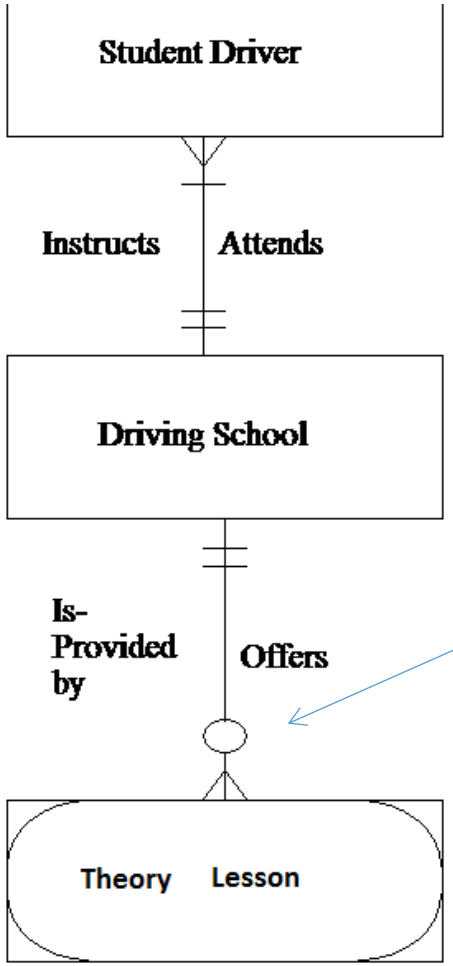
- Specifies whether an instance must exist or can be absent in the relationship



A Lecturer may teach **zero** or many classes.

A class is taught by one and only one Lecturer.

An Example



Optional

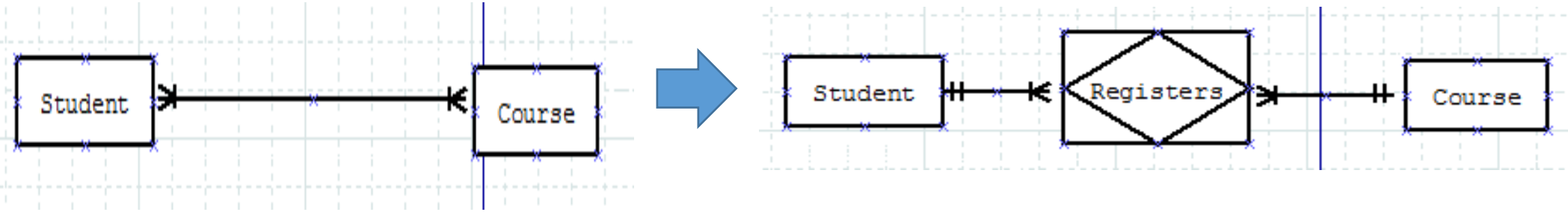
Types of Entity: Strong Entity

Student Driver

Strong entities are the basic elements in your application:

- Student
- Course
- Class
- School etc

Types of Entity: Associative Entity



The relational model does not offer direct support to many-to-many relationships, even though such relationships happen frequently in normal usage. The solution to this problem is the creation of another table to hold the necessary information for this relationship. This new table is called an **associative entity**.

You cant add a foreign key to either table

student_id	student_name	dateOfBirth
1	John	15-04-78
2	Alice	15-06-79
3	Billy	15-04-66

module_id	module_name
101	databases
102	Maths
103	Networks
104	Python

Create an associative entity

module_id	student_id
101	1
102	1
103	2
103	3
104	3

Many-to-Many Relationships

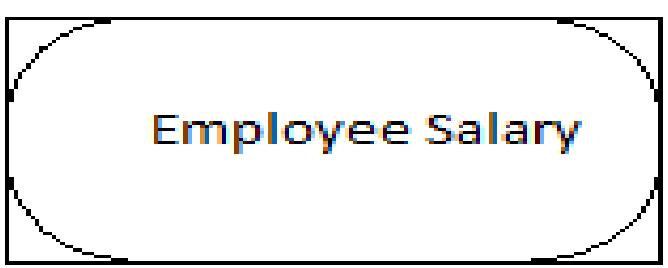
- It is appropriate to identify and illustrate many-to-many relationships at the conceptual level of detail.
- Such relationships must be broken down to one-to-many relationships at the logical level of detail.

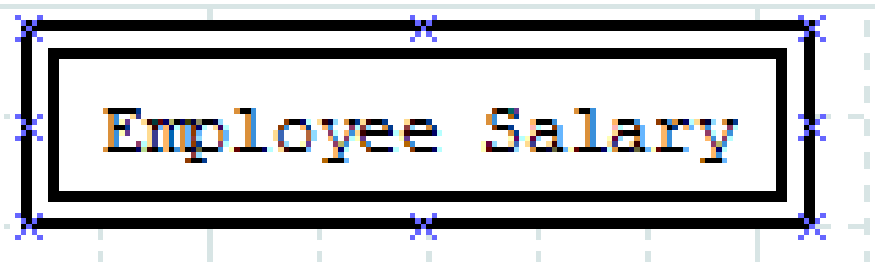
Associative Entity

- You can eliminate a many-to-many relationship by creating an associative entity.
- In the above figure, the many-to-many relationship between **Student** and **Course** is resolved by adding the **Registers** entity.

- Primary Keys:
  student – Student ID
  course – Course ID, (databases, java etc)
  registers – Student ID, Course ID

Types of Entity: Weak Entity





This is used to show data that is wholly dependant upon the existence of a fundamental entity.

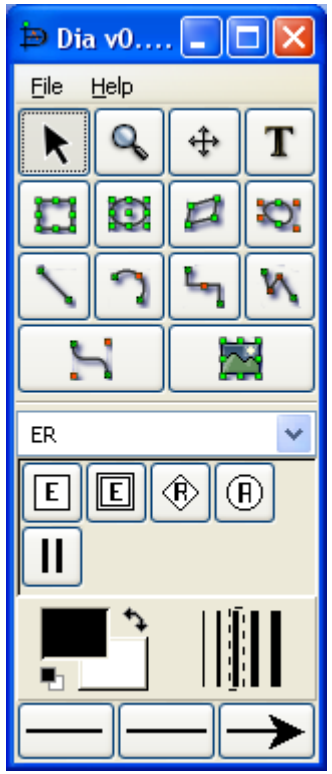
For example Employee and Employee Salary. It depends on another entity for part of its primary key.

Dia

- Dia is an open source diagramming tool with many different diagram templates.
- It can be downloaded free from

  dia-installer.de/

Dia


Dia v0....
File
Help
ER

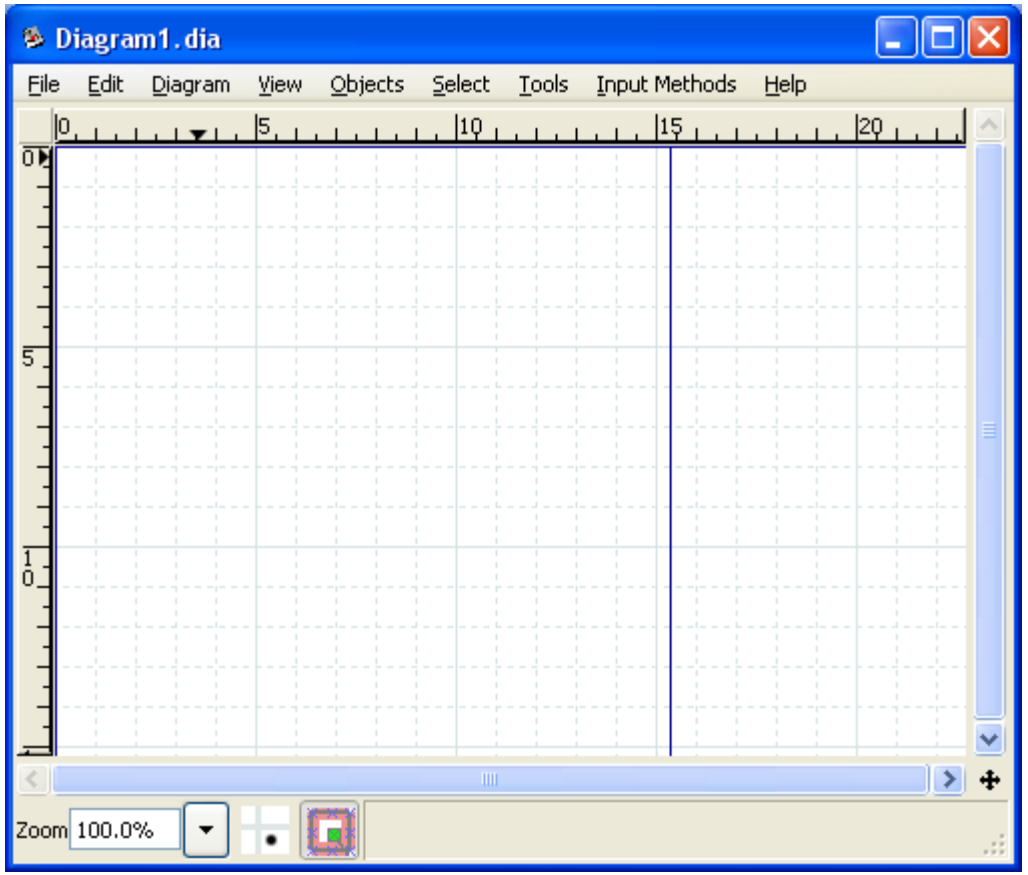
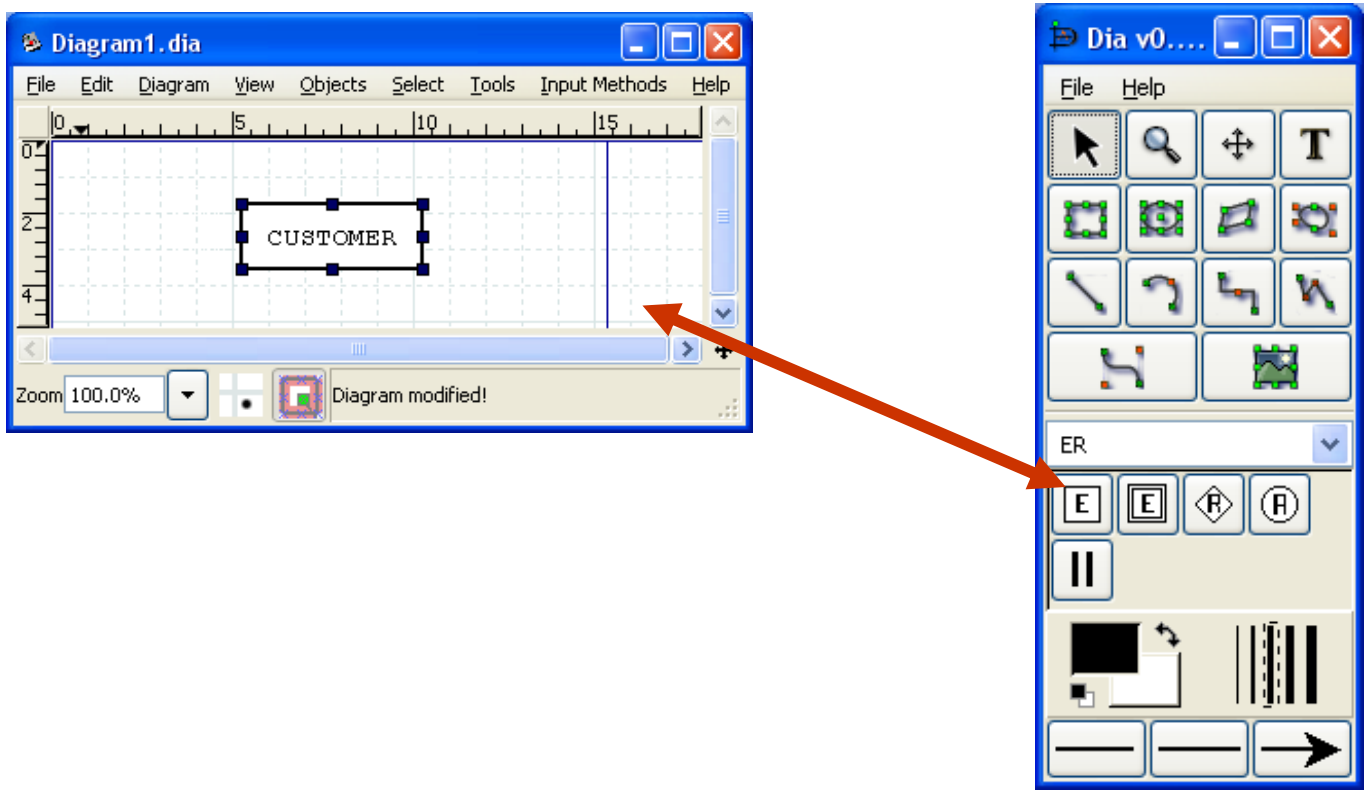


Diagram1.dia
File
Edit
Diagram
View
Objects
Select
Tools
Input Methods
Help
Zoom
100.0%

Entities

- Refers to the *entity set* and not to a single entity occurrence
- Corresponds to a table and not to a row in the relational environment

Just drag the entity Symbol over to the work space, the rectangle will then be automatically displayed. Double click to enter in name



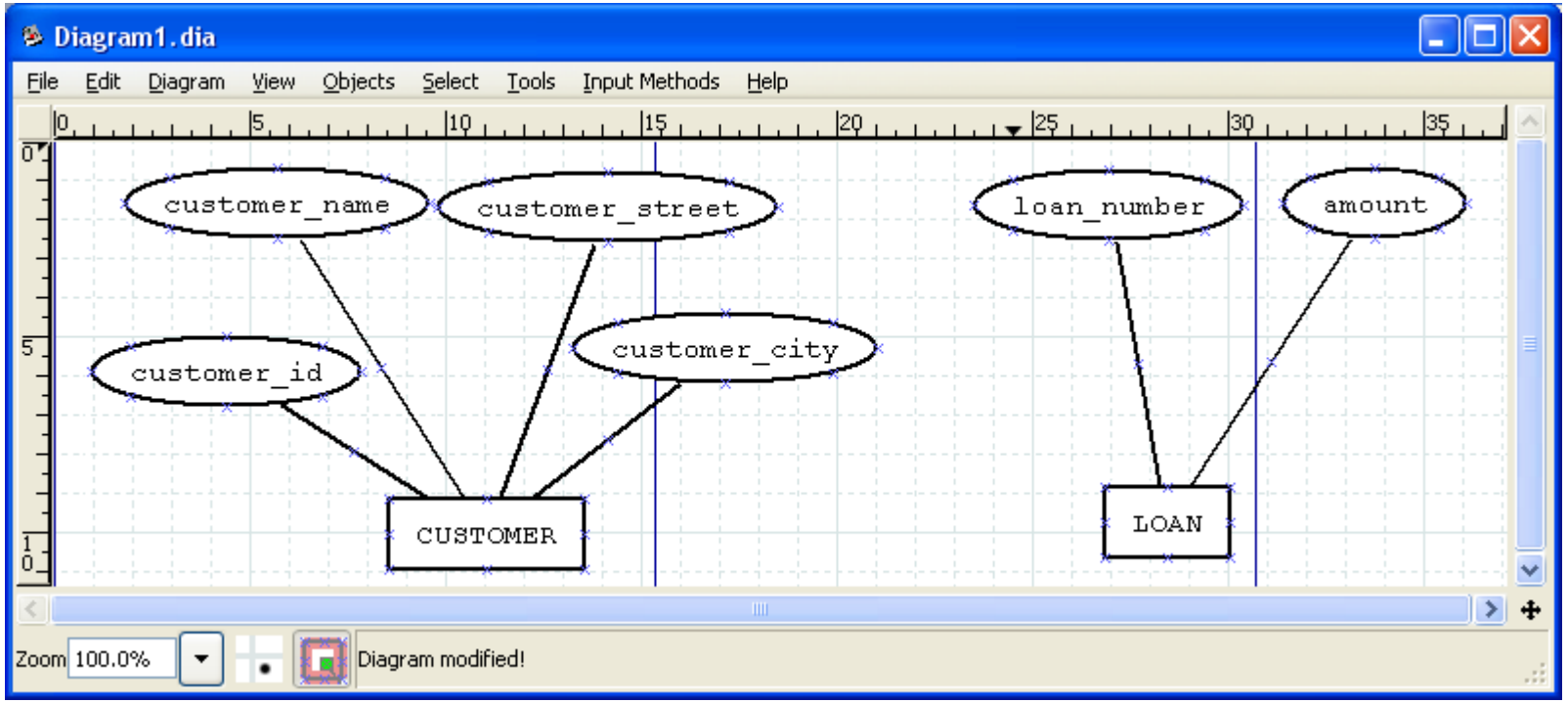
Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

Example:

customer = (*customer_id, customer_name, customer_street, customer_city*)

loan = (*loan_number, amount*)



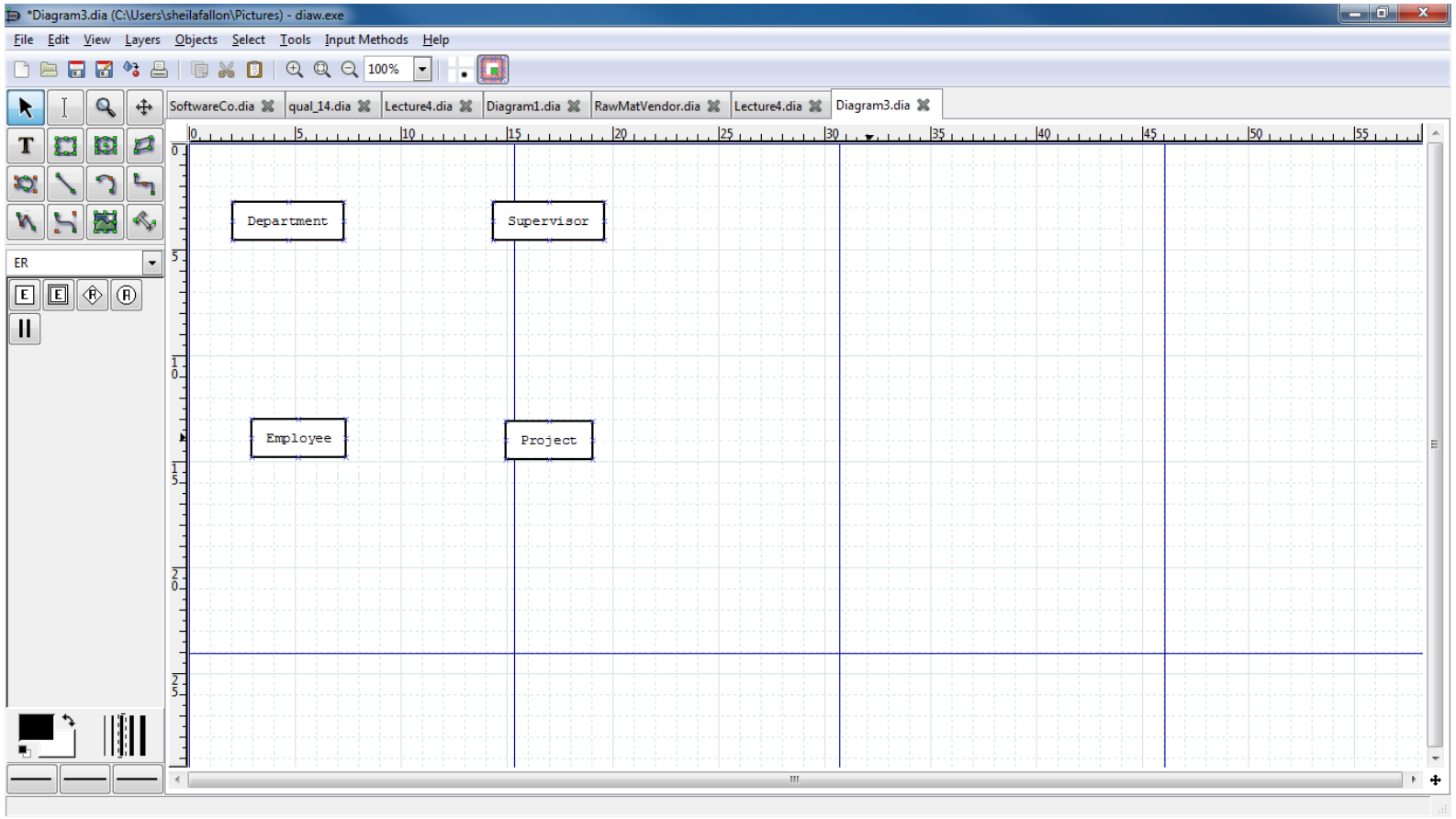
ERD Question

- You have been asked to prepare an Entity-Relationship Diagram (ERD) for a Small Company. The company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned to at least one, but possibly more departments. At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects.
- The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.

Steps

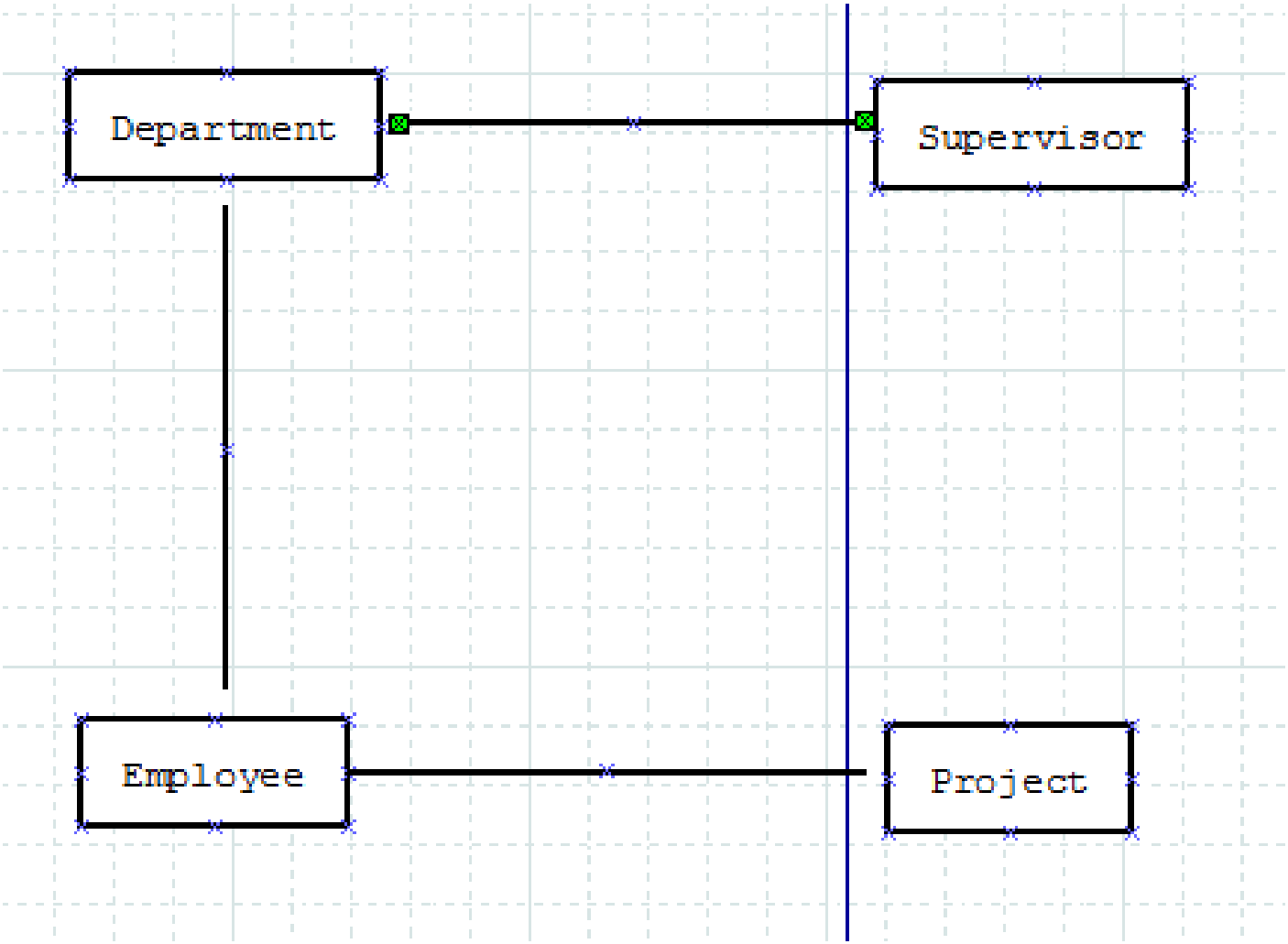
1. Specify the Entities
2. Specify Relationships
3. Specify Connectivity and Optionality
4. Add the attributes

Specify the Entities



1. Specify the Entities
2. Specify Relationships
3. Specify Connectivity and Optionality

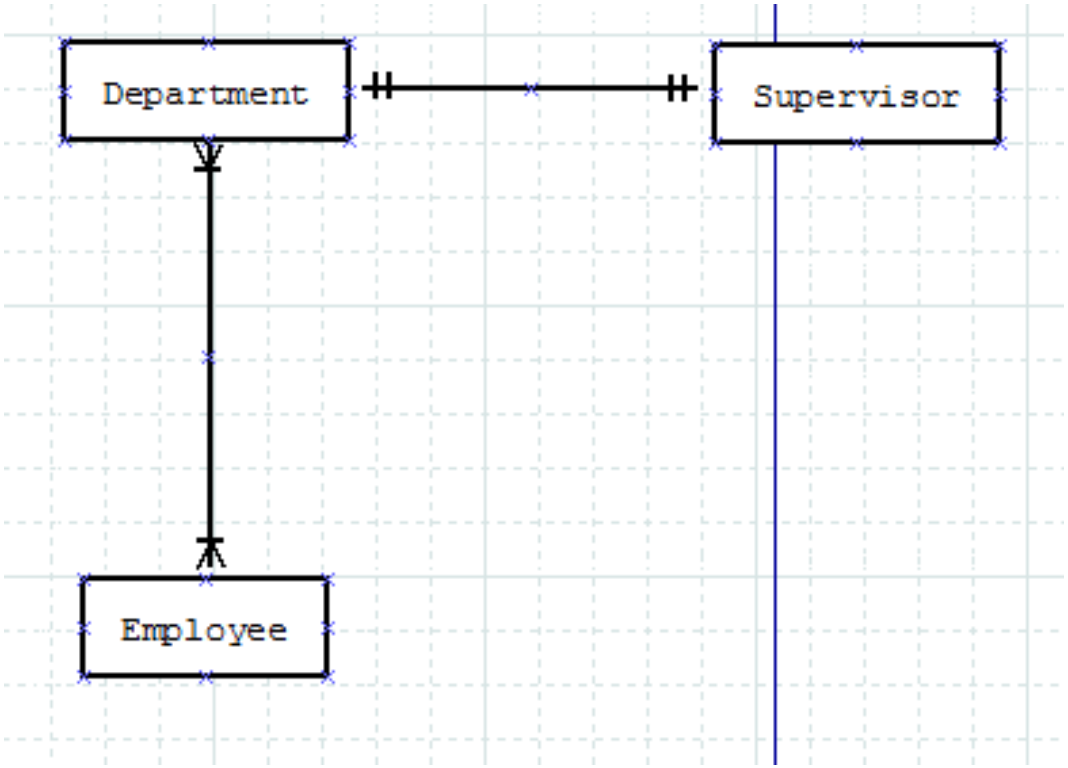
Specify Relationships



1. Specify the Entities
2. Specify Relationships
3. Specify Connectivity and Optionality
4. Add the attributes

Specify the Connectivity and Optionality

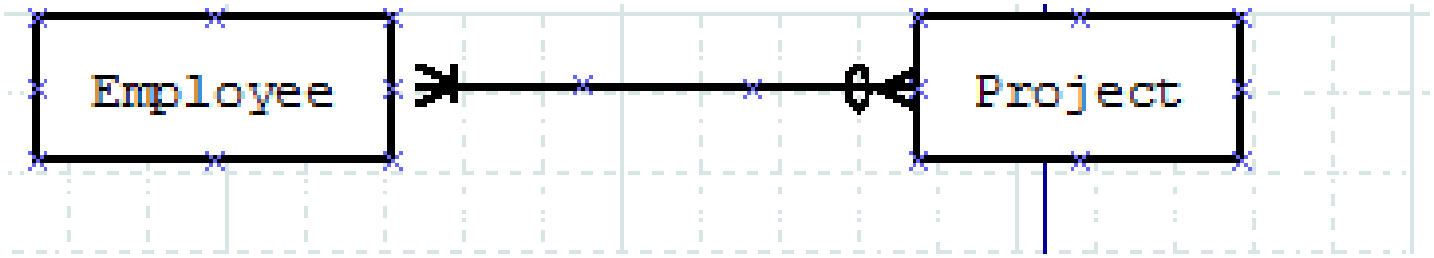
- Each department has a supervisor and at least one employee. Employees must be assigned to at least one, but possibly more departments. A supervisor is assigned to one department.



1. Specify the Entities
2. Specify Relationships
3. Specify Connectivity and Optionality
4. Add the attributes

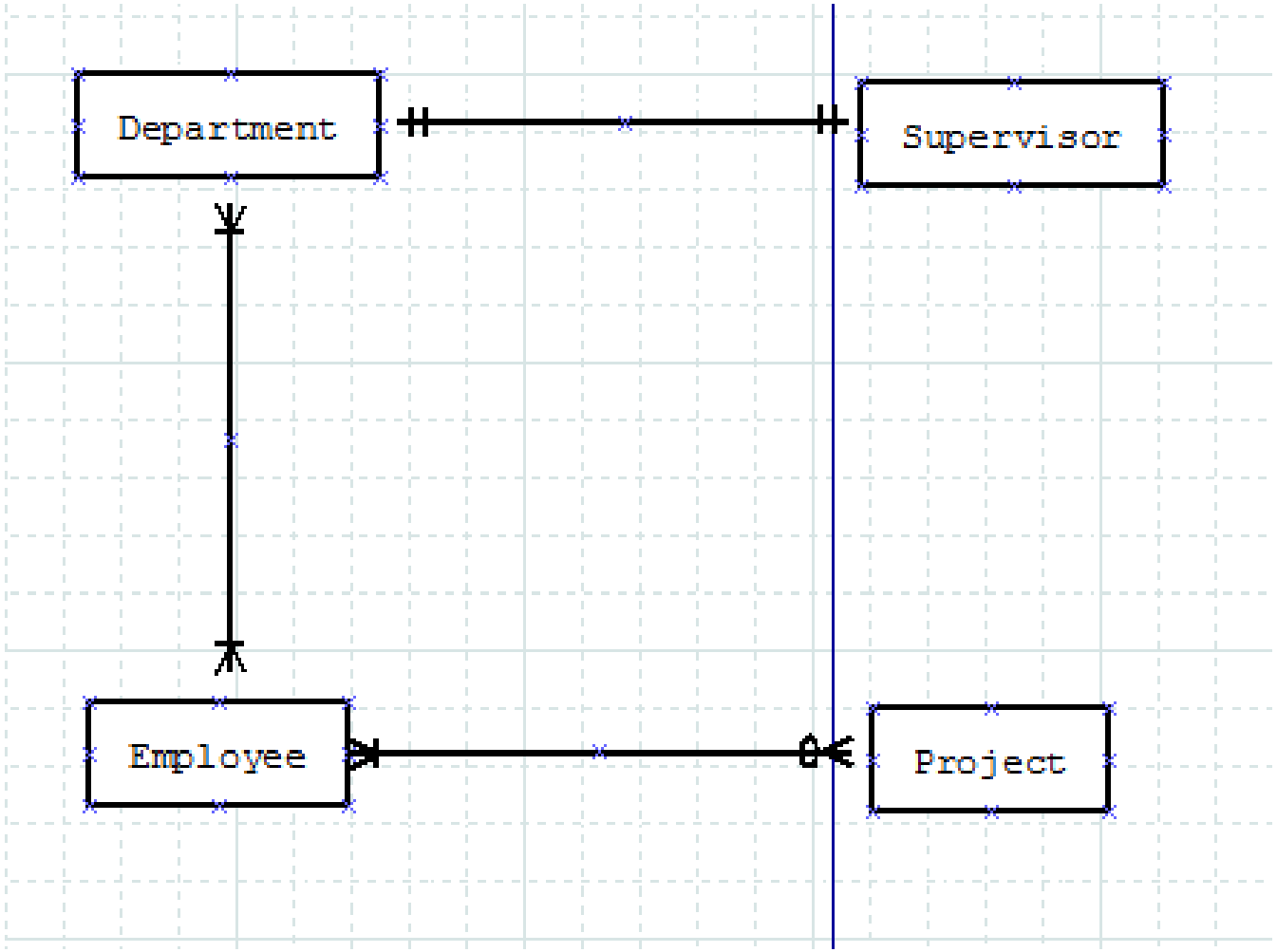
Specify the Connectivity and Optionality

- At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects.



1. Specify the Entities
2. Specify Relationships
3. Specify Connectivity and Optionality
4. Add the attributes

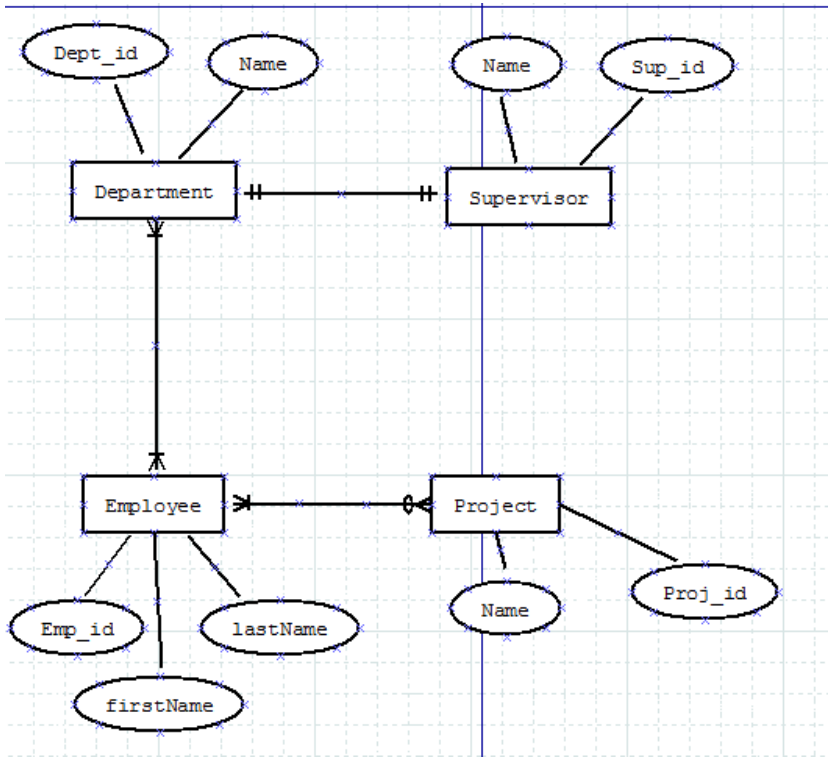
Specify the Connectivity and Optionality



1. Specify the Entities
2. Specify Relationships
3. Specify Connectivity and Optionality

Add the attributes

The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.

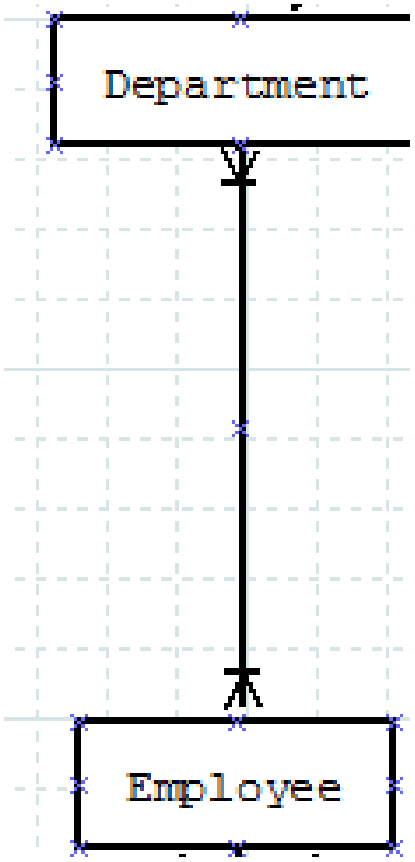


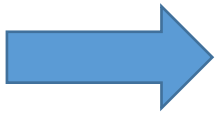
1. Specify the Entities
2. Specify Relationships
3. Specify Connectivity and Optionality
4. Add the attributes

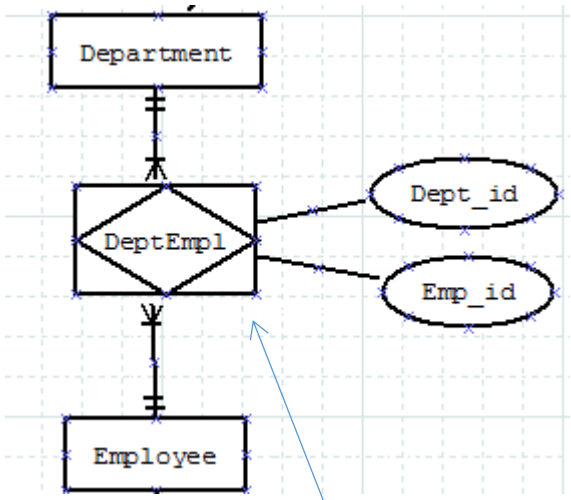
Many to Many Relationships

- It is appropriate to identify and illustrate many-to-many relationships at the conceptual level of detail.
- Such relationships must be broken down to one-to-many relationships at the logical level of detail.

Many to Many Relationships

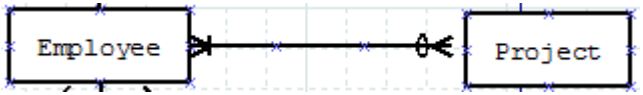


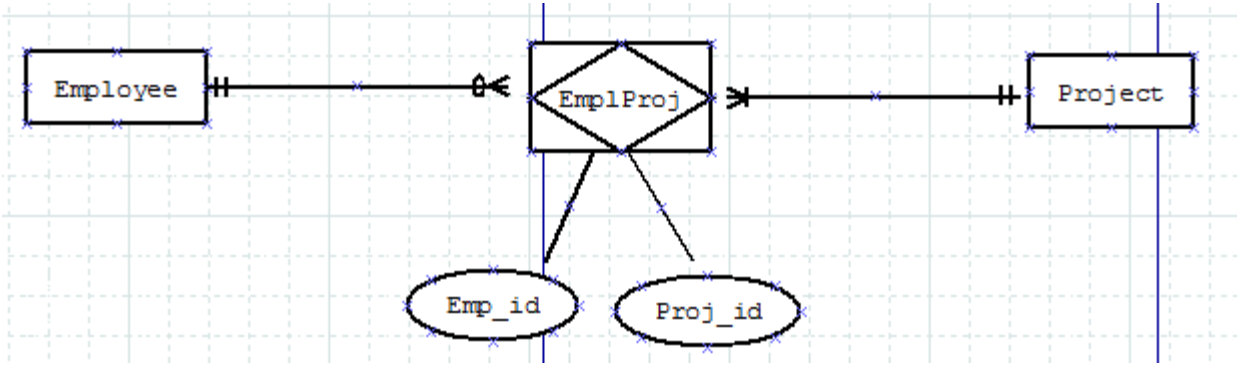




The new entity must contain the primary keys of both Department and Employee

Many to Many Relationships

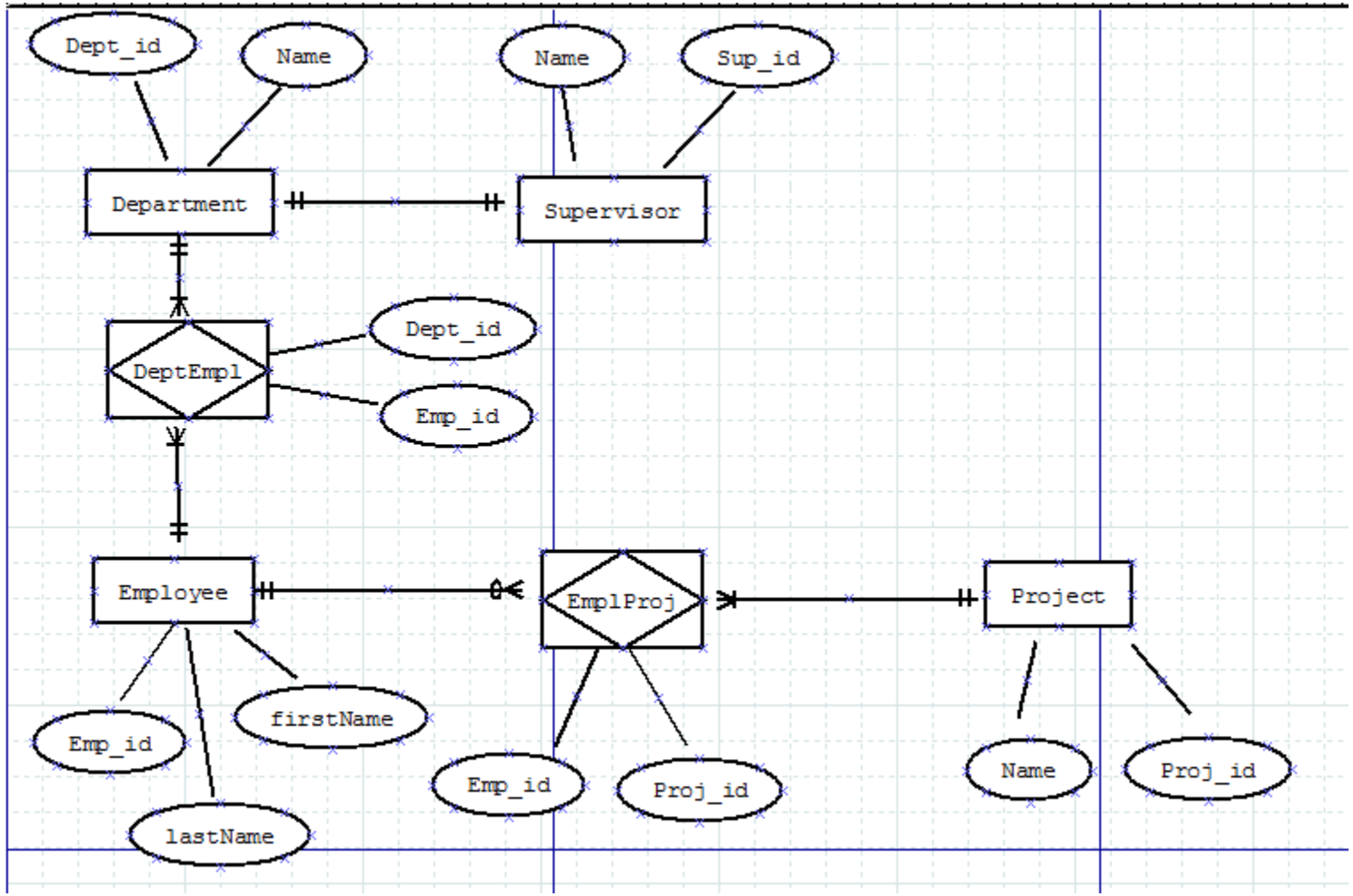




The new entity must contain the primary keys of both Employee and Project

At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects.

Complete ERD in Dia



Benefits of ERD modelling.

1. **Helps the analyst to understand the requirements:** The process of creating an ERD helps the database analyst understand the information needs of the end users

2. **Visual Representation**: Having an effective design provides assistance to the database designers to determine the flow of data and working of the complete system.

Benefits of ERD modelling ctd.

3. **Effective communication**: The clear representation of the data listed under proper headings and tables results in the effective flow of information and communication. The readers can easily understand the relationship between different fields. The information is represented via different symbols . These symbols allow the designer to have a proper understanding of the working of the database after completion.

3. **Easy To Understand**: ERDs are designed in a simple manner so all the individuals can understand it easily. Before actually designing the database, the designers are required to get the design confirm approved the representatives who are to use this data

Consequences of poor modelling:

1. If you don't take the time to map out the needs of the project and how the database is going to meet them, then the whole project may veer off course and lose direction.
2. If you don't take the time at the start to get the database design right, then you'll find that any substantial changes in the database structures that you need to make further down the line could have a huge impact on the whole project, and greatly increase the likelihood of the project timeline slipping.