JOBSHEET WEEK 3 ENTITY RELATIONSHIP DIAGRAM

## Learning Objective

The objective of this jobsheet is to master how to design a database in conceptual model, spesifically on Entity Relationship Diagram (ERD)

## Entity Relationship Diagram

You have to make jobsheet report and submit it with this format “**DATABASE\_2\_SIB1G\_YOUR NAME.pdf”.**

### Database Design

Every system needs to be designed well, especially software systems, it has some process from requirement analysis, design, coding, testing, and maintenance. In requirement analysis phase, a system analyst do an interview to users and stakeholders, this interview will describe data requirements and functional requirements. After the requirements received, a database designer or perhaps system analyst too will design the database based on the requirements. Phases of database design can be derived into three parts:

* + - 1. Conceptual design

This phase transforms data requirements into conceptual model that describes data entities, relationships, constraints, attributes, and cardinalities. Entity relationship diagram is a part of this phase.

* + - 1. Logical design

This phase maps the conceptual data model to the logical data model that used by the DBMS (Database Management Systems) like relational model of the database design.

* + - 1. Physical design

This phase creates internal structures that needed to efficiently store or manage the data. In this phase, we will define some elements like table spaces, indexes, etc.

### Entity Relationship Diagram

Entity relationship diagram is a traditional approach in database conceptual modelling. This diagram introduced in 1976 by Peter Chen. In ERD, there’s some elements of it:

* + - 1. Entity

Entity represents a “thing” in the real world an independents existence. It has its own identity or characteristics and just represents one thing. We can call an entity with objects that its data needed to be recorded in a system.

Example: you have showroom that sells vehicles like motorcycle and car. You want to make your own sales management system. Then in that system needs data for sold item, which is the motorcycle and car. They have same characteristics, so you can make “vehicle” entity.

There’s two kinds of entity:

* + - * 1. Strong entity

Strong entity is an entity that independence to another entity. It represented with a rectangle like this below.

**employee**

* + - * 1. Weak entity

Weak entity is an entity that depends to another entity. It represented with a shape like this below.

**parents**

* + - 1. Attribute

Attribute is a property or characteristics of an entity. Example: name of an employee, color of a car, balance of an account, location of a house.

The attribute can be classified into these type:

* + - * 1. Simple attribute

Simple attribute is an attribute that composed of a single component. Example: Price of an item in sales information system.

ERD notation:

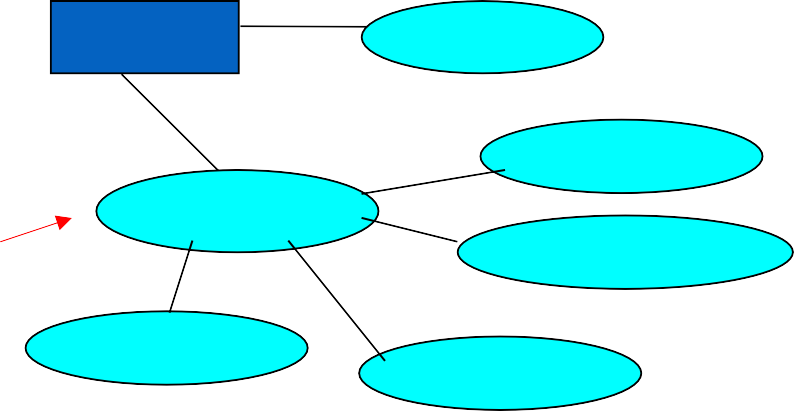
**name**

**employee**

* + - * 1. Composite attribute

Composite attribute is an attribute that composed of multiple components. Example: address of a company, it composed of street, building number, zip code, city, and country.

ERD notation:



**employee**

**name**

**street**

**address**

**building\_no**

**city**

**zip\_code**

* + - * 1. Multi-valued attribute

Multi-valued attribute is an attribute that has possibility to have more than one value. Example: phone number of a student. One student can have more than one phone number.

ERD notation:

**phone**

**student**

* + - * 1. Derived attribute

Derived attribute is an attribute that has value derived from another attribute. Example: age attribute of a student. Age value can be calculated from birth date and existing date. ERD notation:

**age**

**student**

* + - * 1. Key attribute

Key attribute is an attribute that identify data of entity. Key attribute must have a unique value.

ERD notation:

**student\_no**

**student**

* + - 1. Relationship

Relationship relate an entity to another entity or itself. For example, there’s entity person and vehicle. Person data may have relation with vehicle data, or we can say maybe a person have a vehicle. Thus, person entity related to vehicle. It notated with diamond shape and filled with a verb that illustrate its relationship like this below.

owns

vehicle

person

A weak entity that related to strong entity should have total participation constraint. Total participation constraint denoted with double line. And every relationship between strong and week entity related with identifying relationship like this example below.

has

student

parent

Depends on data dependency between entities, there’s three kinds of cardinality:

* + - * 1. One to one (1:1)

One to one cardinality happened when there’s one data related only with one data. Example there’s lecturer and department entity. Every department headed by only one head of department. The notation like this below.

# lecturer 1

lead

1

department

* + - * 1. One to many/many to one (1:N or N:1)

One to many cardinality is when one data related with more than one data. Example there’s person entity and vehicle entity. Each person may have one or more than one vehicle, so cardinality between person and vehicle is one to many.

person 1

owns

M

vehicle

* + - * 1. Many to many (N:M)

Many to many cardinality is when one data in entity A related with more than one data in entity B, and so that one data in entity B may related to more than one data in entity

A. So, cardinality between entity A and B is many to many. For example, a student can take some course and a course can be taken by many students.

# student N

take

M

subject

Relationship has its degree, it derived as three kind of relationship degree:

1. Unary

1

marry

M

person

1. Binary

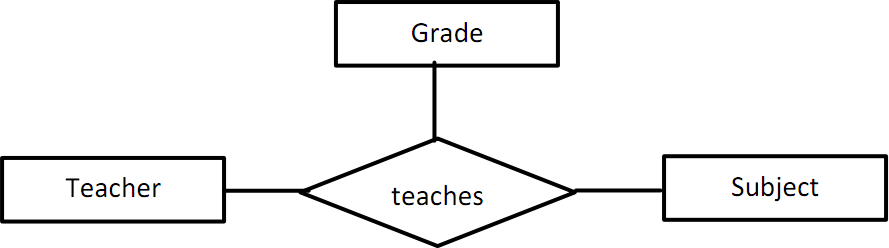
person 1

owns

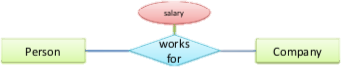
M

vehicle

1. Ternary



Similar to entity, relationship may even have attributes like this picture below.

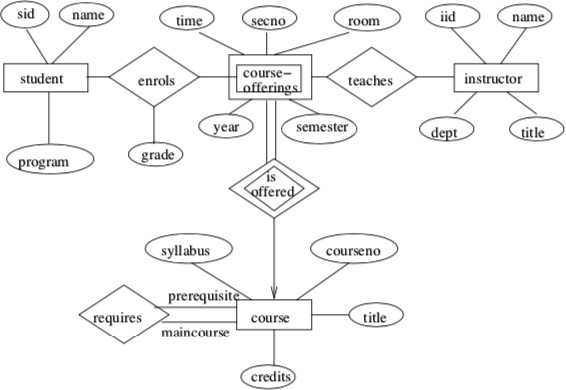


And then, how we design a ERD? You can follow these steps below:

1. Find the basic entity types
2. Find the attributes of entities
   * Decide to which entity an attribute should be assigned
   * Which attributes are key attributes?
   * Some attributes are better modeled as own entities, which ones?
3. Define the relationship types
   * Which role do entities play?
   * Do relationships require additional entity types?
   * Are the relationships total? Identifying? Are weak entities involved?
   * What are the cardinalities of the relationship type?

### Lab Exercise

* + - 1. Identify entity, attribute (including key attribute), and relationship from this ERD below.



Answer :

* Entitas yang dimiliki yaitu Student, instructor, course
* Weak entity = course offering
* Atribut (including key attributes):

1.Course :

- Key Attribute: courseno

- Other Attributes: title, credits,syllbus

2.Instructor :

- Key Attribute: iid

- Other Attributes: year, semester, room, secno

3.Student :

- Key Attribute: sid

- Other Attributes: name, year

* Hubungan:

1. Mengajar : Menghubungkan instruktur ke kursus yang mereka ajar(many to many)

2. Pendaftaran : Menghubungkan siswa dengan mata kuliah yang diikutinya(many to many)

3. Memerlukan : Menghubungkan mata kuliah dengan mata kuliah prasyarat yang dibutuhkan( one to many)

4. Ditawarkan : Menghubungkan kursus dengan penawaran kursus(one to many)

* + - 1. A large bank operates several divisions. Information Technology (IT) is operated as one of these divisions. Within the IT division are many departments that are managed by one manager, and all IT employees belong to one of these departments.

The IT division assigns its employees to one or more ongoing projects in the bank. A project may be planned, but not have any employees assigned to it for several months. Each project will have a single employee assigned who acts as a project leader.

* + - * 1. Identify entity, attribute, relationship, and cardinality for description above!

1.Entitas:

* Bank (Bank)
* Divisi (Division)
* Departemen (Department)
* Karyawan (Employee)
* Proyek (Project)

2.Atribut:

* + Bank:

Nama Bank

* + Divisi:

Nama Divisi

* + Departemen:

Nama Departemen

* + Karyawan:

Nama Karyawan

* + Jabatan
  + Proyek:

Nama Proyek

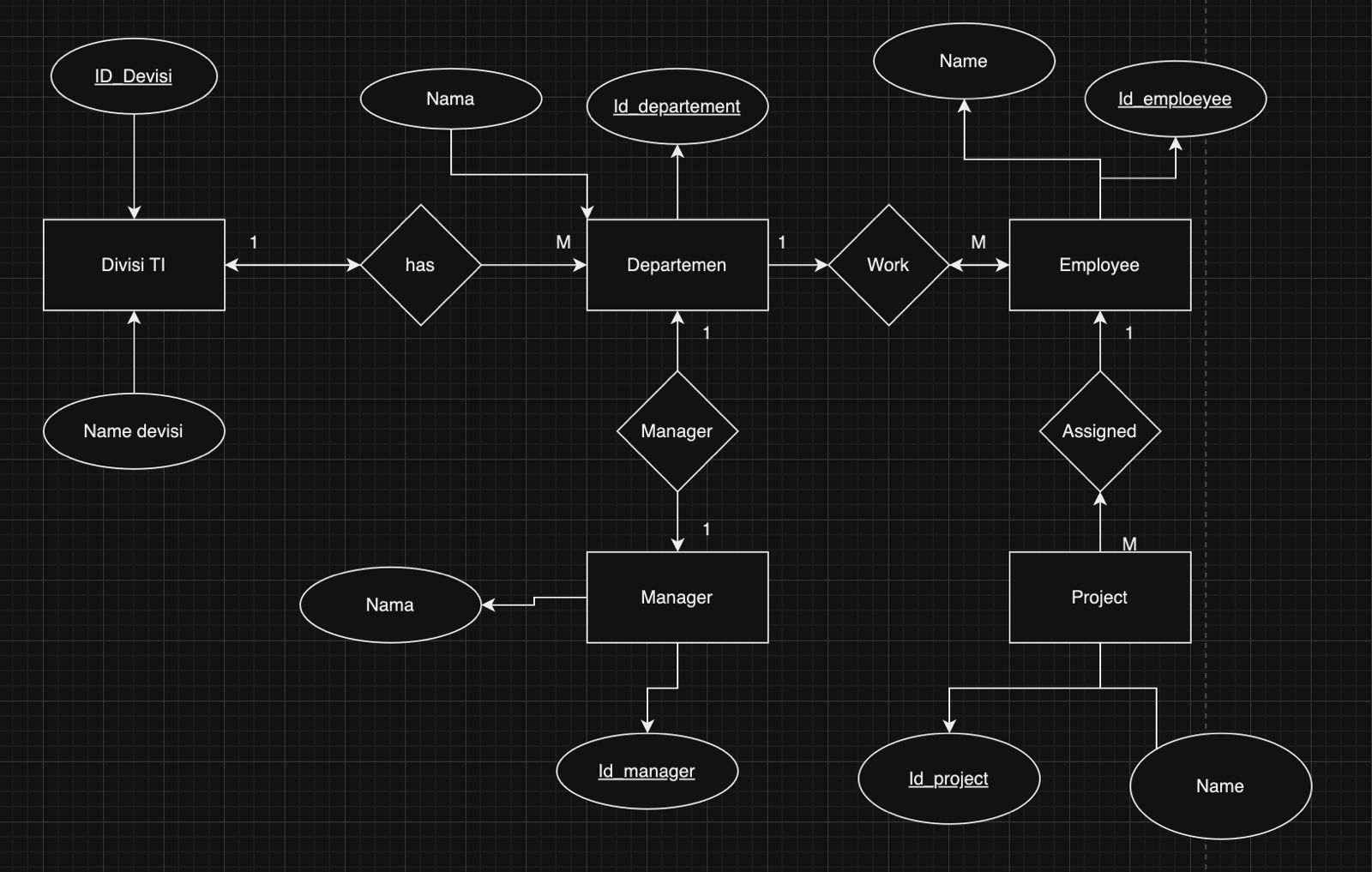
Status Proyek

3.Relasi:

* Divisi memiliki banyak Departemen (1:N)
* Departemen dikelola oleh satu Manajer (1:1)
* Karyawan tergabung dalam satu Departemen (N:1)
* Divisi menugaskan Karyawan untuk satu atau lebih Proyek (N:M)
* Setiap Proyek memiliki satu Karyawan yang bertindak sebagai Pemimpin Proyek (1:1)

4.Kardinalitas:

* 1 Divisi memiliki banyak Departemen
* 1 Departemen memiliki satu Manajer
* 1 Departemen memiliki banyak Karyawan
* 1 Karyawan dapat tergabung dalam satu atau lebih Departemen
* 1 Divisi menugaskan banyak Karyawan untuk satu atau lebih Proyek
* 1 Karyawan dapat ditugaskan untuk satu atau lebih Proyek
* Setiap Proyek memiliki satu Karyawan yang bertindak sebagai Pemimpin Proyek
  + - * 1. Design an Entity-Relationship diagram for this case above!



-- good luck everyone  --