Software Engineering
Week # 3

LECTURER: ABDULRAHMAN QAIM

Class Diagram

The class diagram is the main building block of object-oriented modelling.

It is used for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Thus, class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

Class Diagram

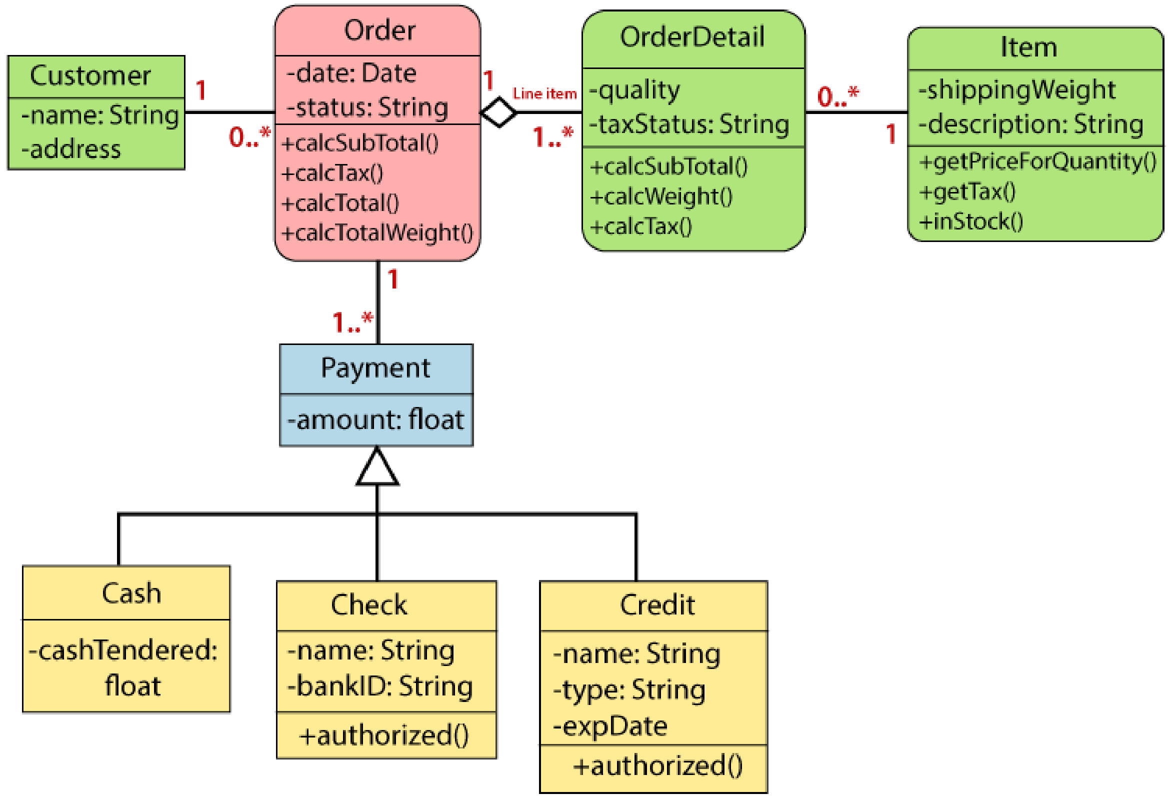
Purpose of Class Diagrams

1. Shows static structure of classifiers in a system
2. Diagram provides basic notation for other structure diagrams prescribed by UML
3. Helpful for developers and other team members too
4. Business Analysts can use class diagrams to model systems from business perspective

A UML class diagram is made up of:

- A set of classes and
- A set of relationships between classes

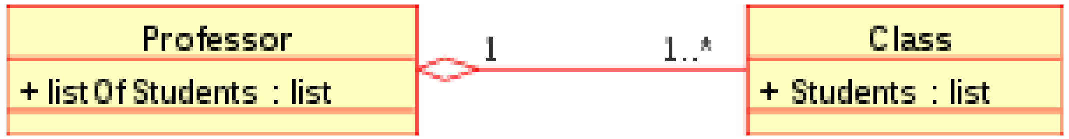
Class Diagram



Class Diagram

Instance-level relationships

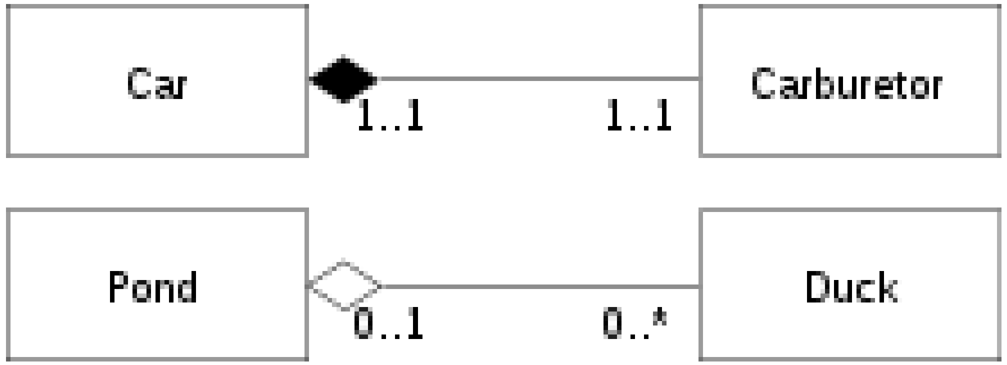
- **Association**
  - **Aggregation**
    - Aggregation is a variant of the "has a" association relationship; aggregation is more specific than association. It is an association that represents a part-whole or part-of relationship. As shown in the image, a Professor 'has a' class to teach.



Class Diagram

Composition

The composite aggregation relationship is a stronger form of aggregation where the aggregate controls the lifecycle of the elements it aggregates. The graphical representation is a *filled* diamond shape on the containing class end of the line that connect contained class(es) to the containing class



Class Diagram

Differences between Composition and Aggregation

Composition relationship

1. When the container is destroyed, the contents are also destroyed, e.g. a university and its departments

Aggregation relationship

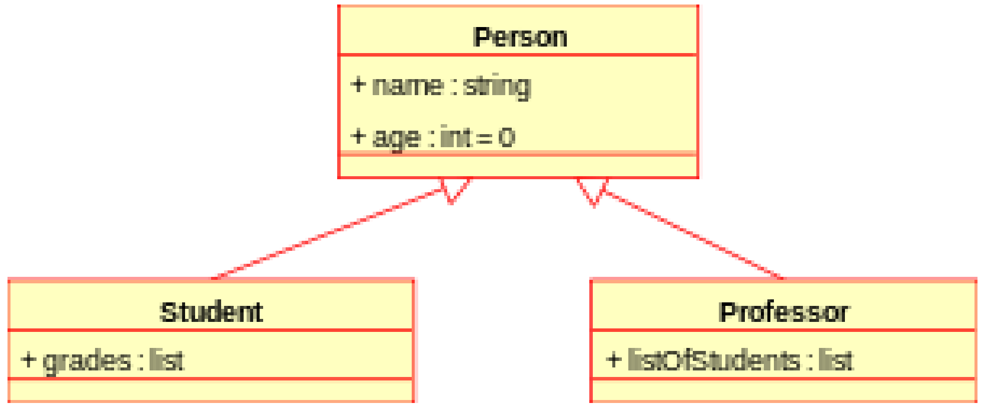
1. When the container is destroyed, the contents are usually not destroyed, e.g. a professor has students; when the professor leaves the university the students do not leave along with the professor.

Class Diagram

Class-level relationships

Generalization/Inheritance

It indicates that one of the two related classes (the *subclass*) is considered to be a specialized form of the other (the *super type*) and the superclass is considered a Generalization of the subclass.



Entity Relationship Diagram

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.

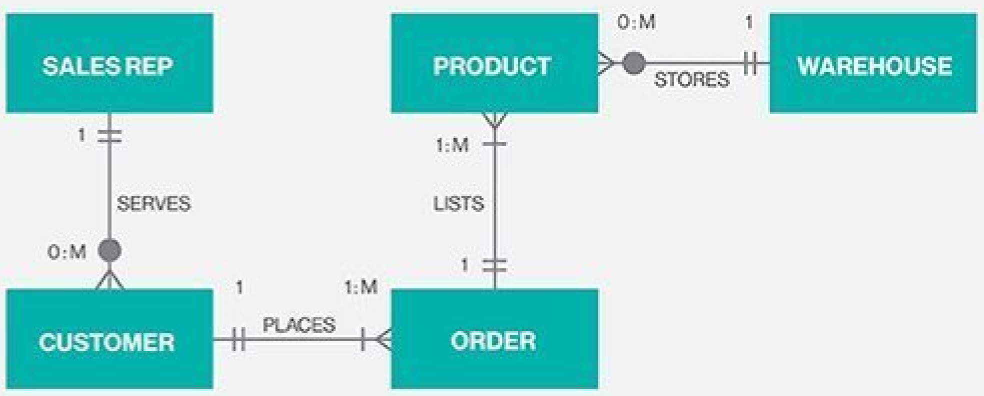
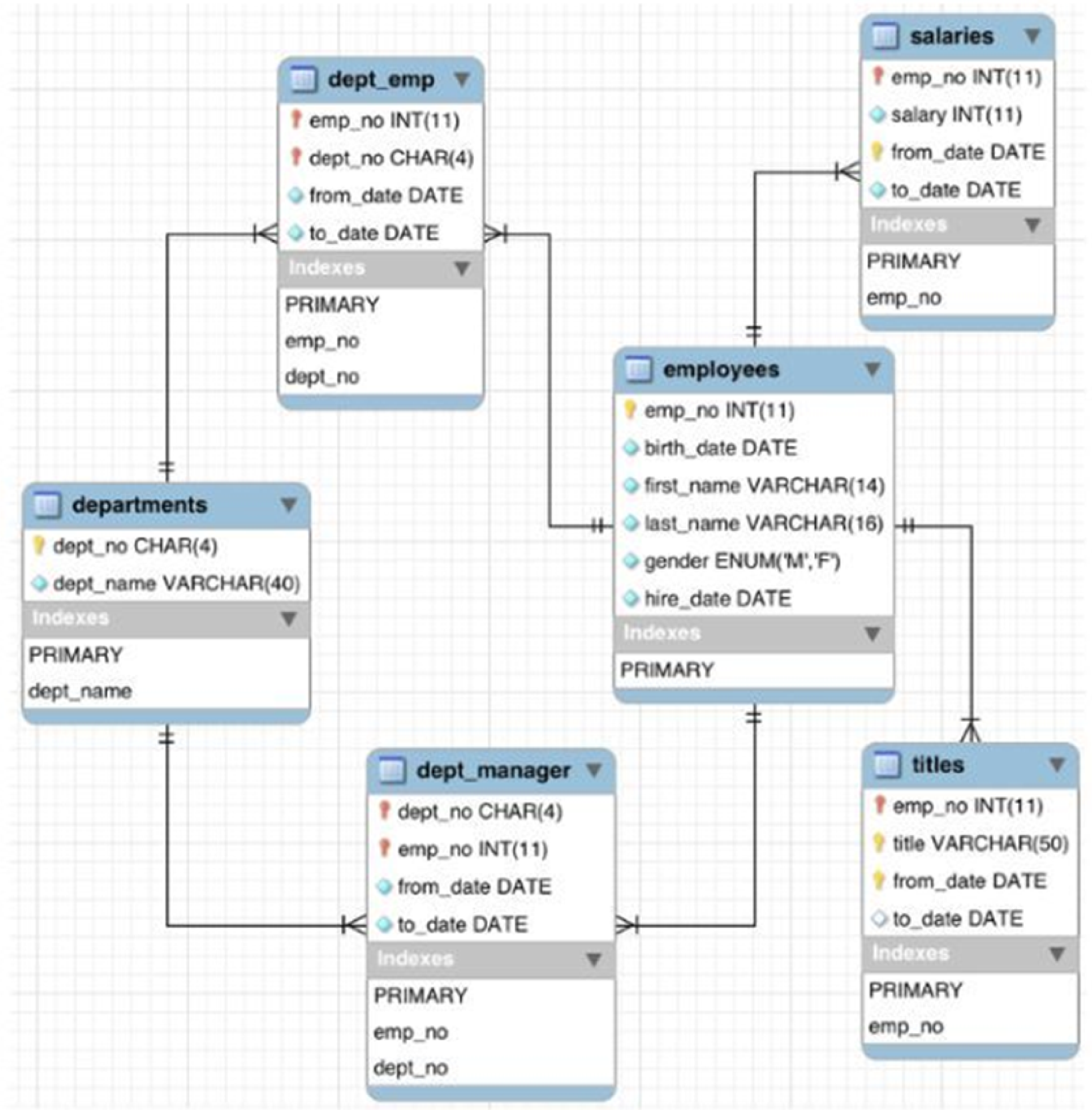


Figure 1. Entity-Relationship Diagram

*1 INSTANCE OF A SALES REP SERVES 1 TO MANY CUSTOMERS
*1 INSTANCE OF A CUSTOMER PLACES 1 TO MANY ORDERS
*1 INSTANCE OF AN ORDER LISTS 1 TO MANY PRODUCTS
*1 INSTANCE OF A WAREHOUSE STORES 0 TO MANY PRODUCTS


salaries
emp_no INT(11)
salary INT(11)
from_date DATE
to_date DATE
Indexes
PRIMARY
emp_no
dept_emp
emp_no INT(11)
dept_no CHAR(4)
from_date DATE
to_date DATE
Indexes
PRIMARY
emp_no
dept_no
employees
emp_no INT(11)
birth_date DATE
first_name VARCHAR(14)
last_name VARCHAR(16)
gender ENUM('M','F')
hire_date DATE
Indexes
PRIMARY
departments
dept_no CHAR(4)
dept_name VARCHAR(40)
Indexes
PRIMARY
dept_name
dept_manager
dept_no CHAR(4)
emp_no INT(11)
from_date DATE
to_date DATE
Indexes
PRIMARY
emp_no
dept_no
titles
emp_no INT(11)
title VARCHAR(50)
from_date DATE
to_date DATE
Indexes
PRIMARY
emp_no

ERD

ERD

The three main cardinalities are:

1. A one-to-one relationship (1:1). For example, if each customer in a database is associated with one mailing address.
2. A one-to-many relationship (1:M). For example, a single customer might place an order for multiple products. The customer is associated with multiple entities, but all those entities have a single connection back to the same customer.
3. A many-to-many relationship (M:N). For example, at a company where all call center agents work with multiple customers, each agent is associated with multiple customers, and multiple customers might also be associated with multiple agents.

Quiz

1st Quiz will be conducted next week.

Quiz will be conducted on LMS.

MCQs