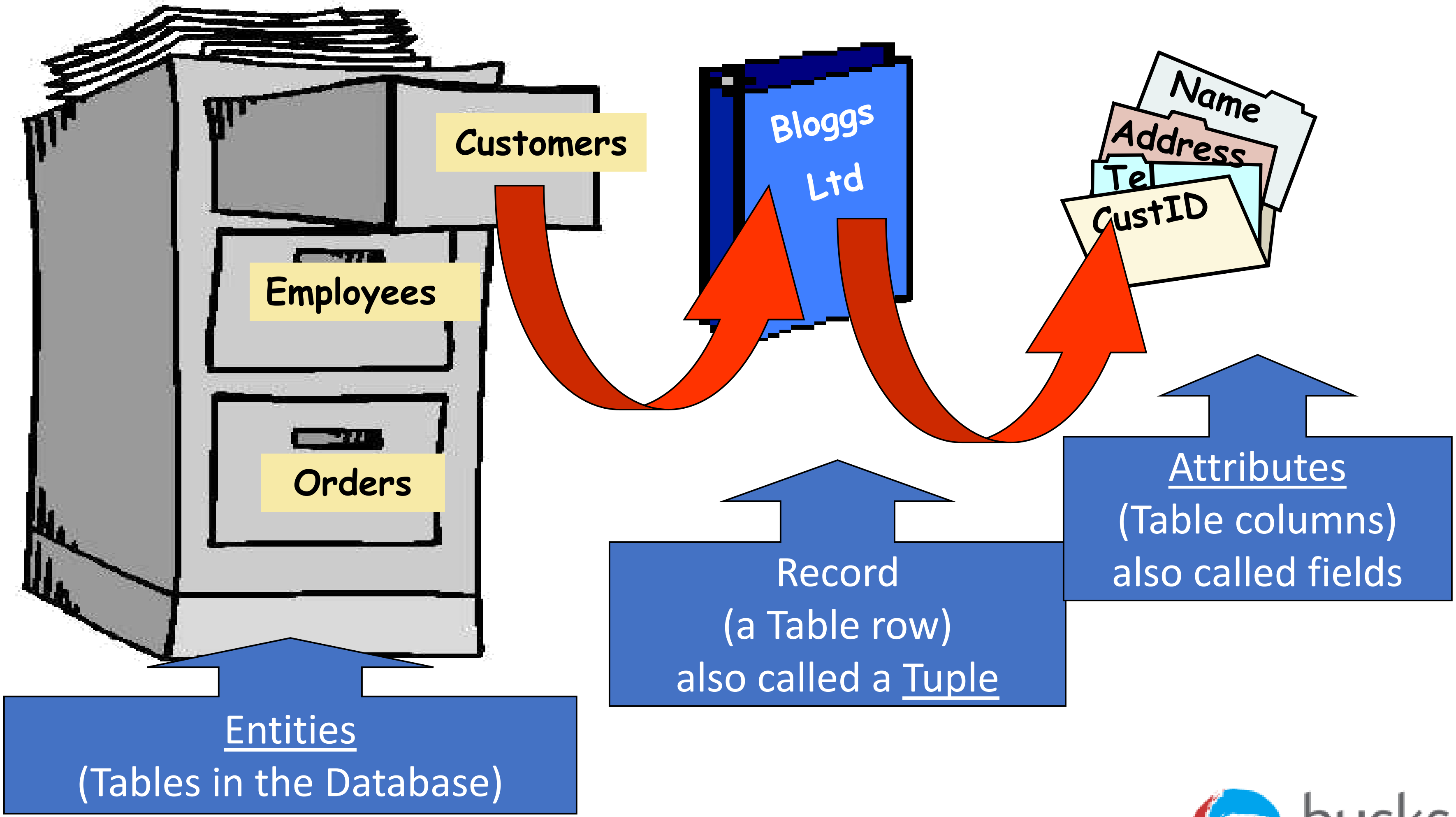
CO550 – Web Applications

UNIT 3 – DATA MODELLING

What is Data Modelling?

- Process of creating logical representation of structure of things or events in real world
- Could be argued that it’s most important task in software development
- Real-world structures and relationships can be described diagrammatically
- In web applications different techniques used e.g. Entity Relationship diagrams or Unified Modelling Language

What is a Database?



Entities
(Tables in the Database)

Record
(a Table row)
also called a Tuple

Attributes
(Table columns)
also called fields

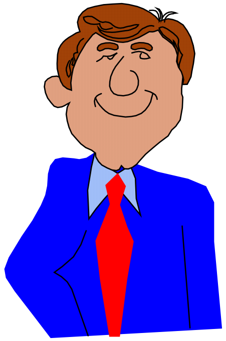
Entity-Relationship Model

- E-R diagram /model First described by Chen P. (1976)
- Shows relationship between entities in a database
- “In software engineering, an entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them, and databases” Wikipedia

What are Entities?

- Anything you are “tracking”
- A discrete object or item about which you are storing information

Stock

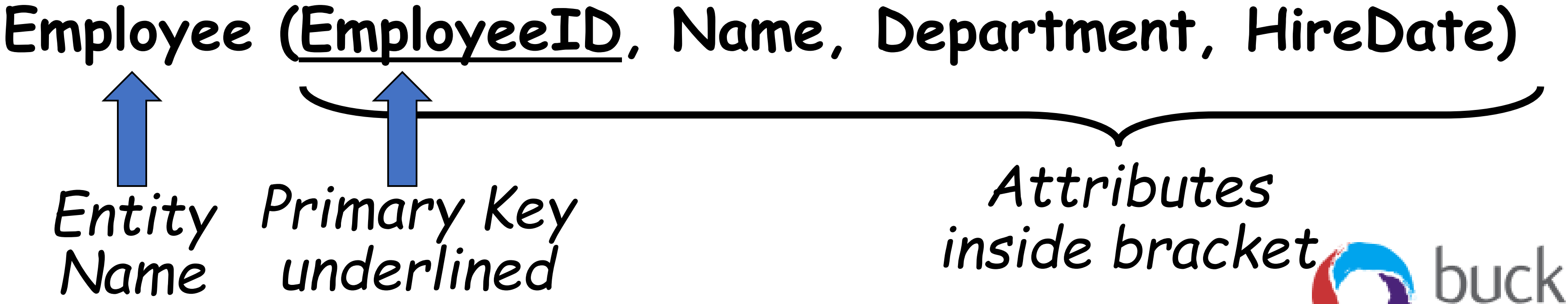


Customers

Students

What is an Attribute?

- Attributes describe the characteristics of an entity
- Can be described in shorthand:



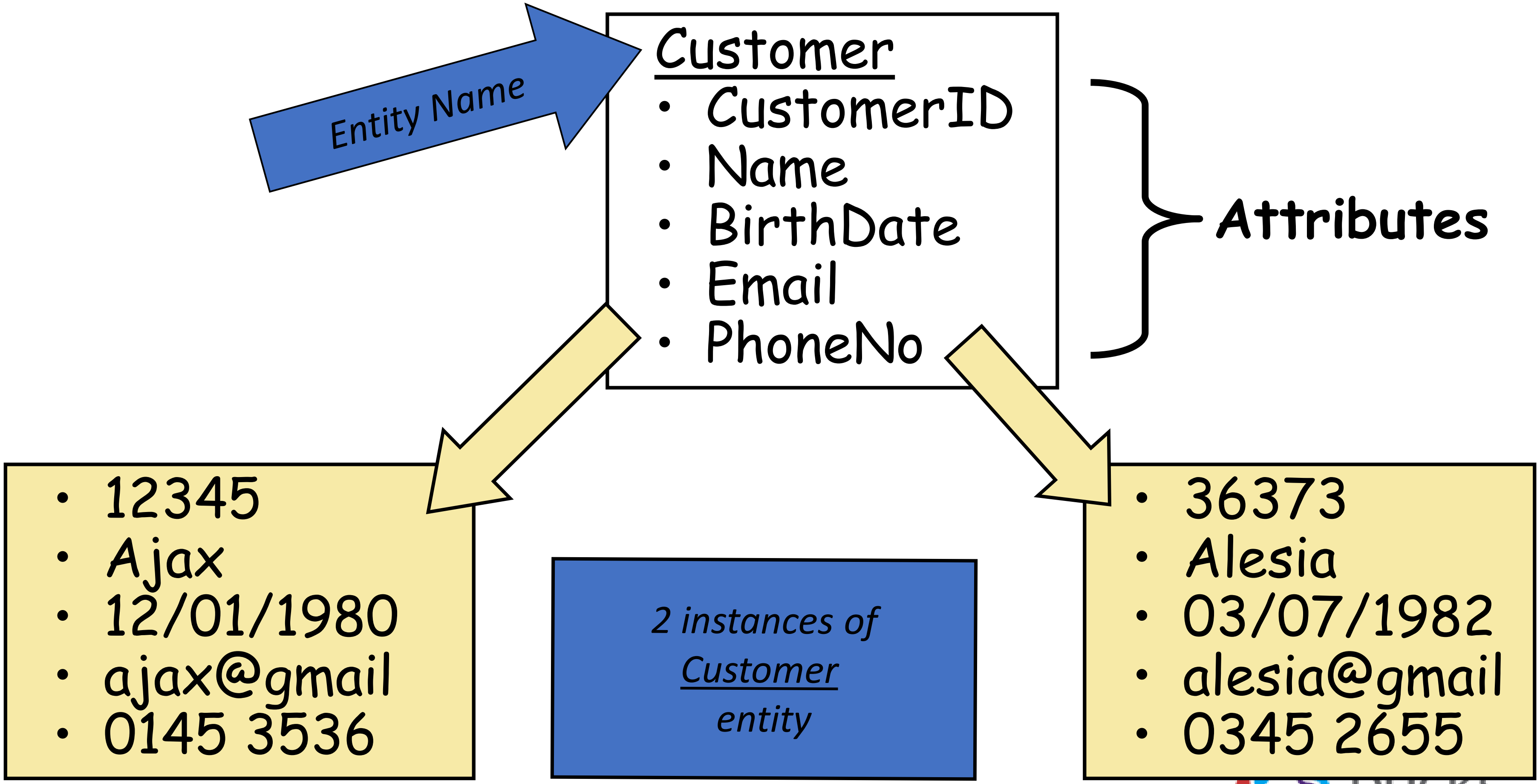
Employee (EmployeeID, Name, Department, HireDate)

Entity Name

Primary Key underlined

Attributes inside bracket

Entities & Attributes



Entity Name

Customer

- CustomerID
- Name
- BirthDate
- Email
- PhoneNo

Attributes

- 12345
- Ajax
- 12/01/1980
- ajax@gmail
- 0145 3536

2 instances of
Customer
entity

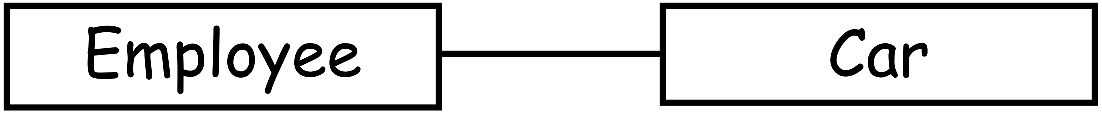
- 36373
- Alesia
- 03/07/1982
- alesia@gmail
- 0345 2655

Entity Relationships (ER)

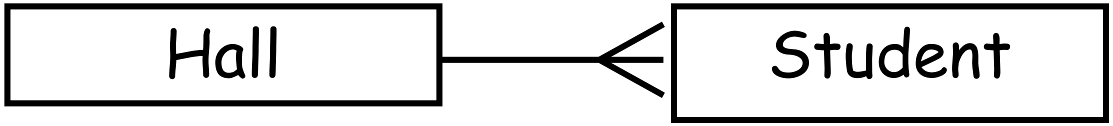
- A database may have many entities
  - e.g. a hospital database may have entities such as Staff, Patients, Doctors, Wards, etc.
- It is also very important to know how these entities interact with or relate to one another
- There will be relationships that link the various entities together
  - e.g. Patients will be linked to a particular ward
- During database development the various entities will form Tables and the relationships will be used to link them

ER Notation

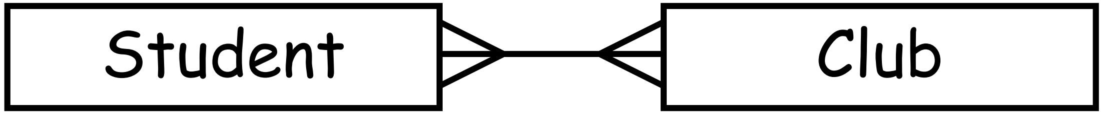
- One to One
  - An employee has one car and a car belongs to only one employee



- One to Many
  - A Hall of Residence has many students but a student lives in only one hall

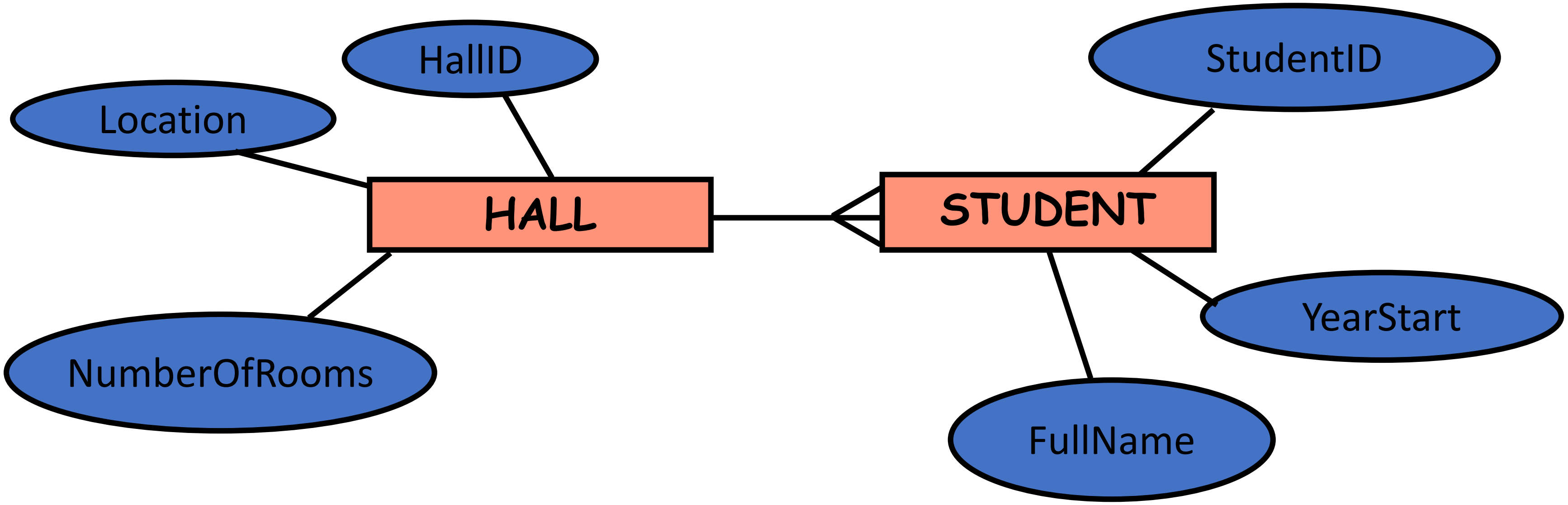


- Many to Many
  - A student may go to many clubs and a club can have many students

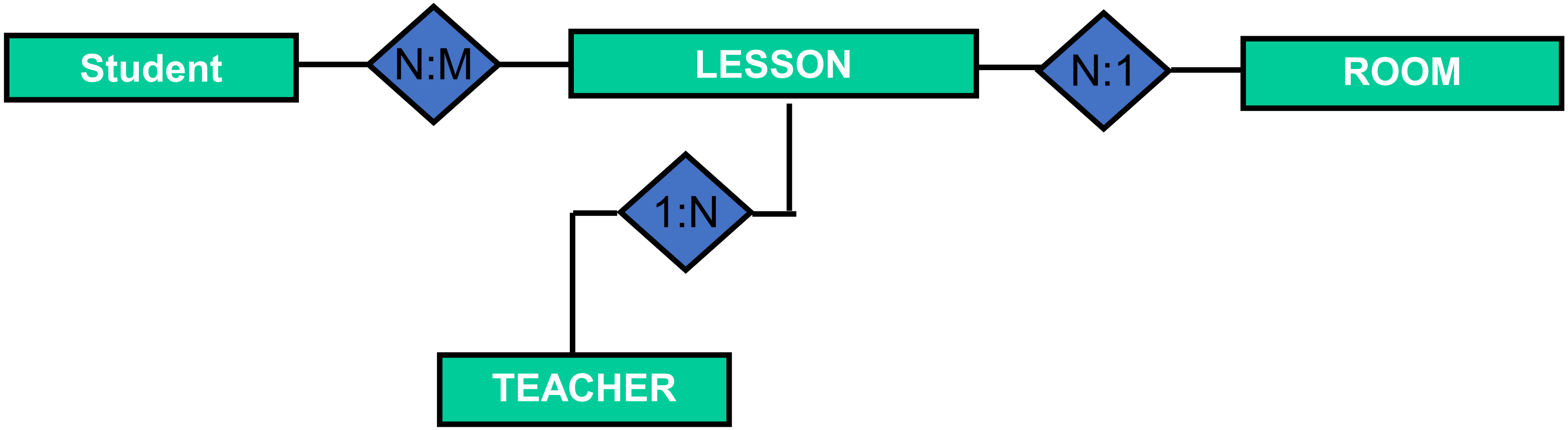


ER Diagrams including attributes

- In this case we have also included the attributes



A More Complete ER Diagram



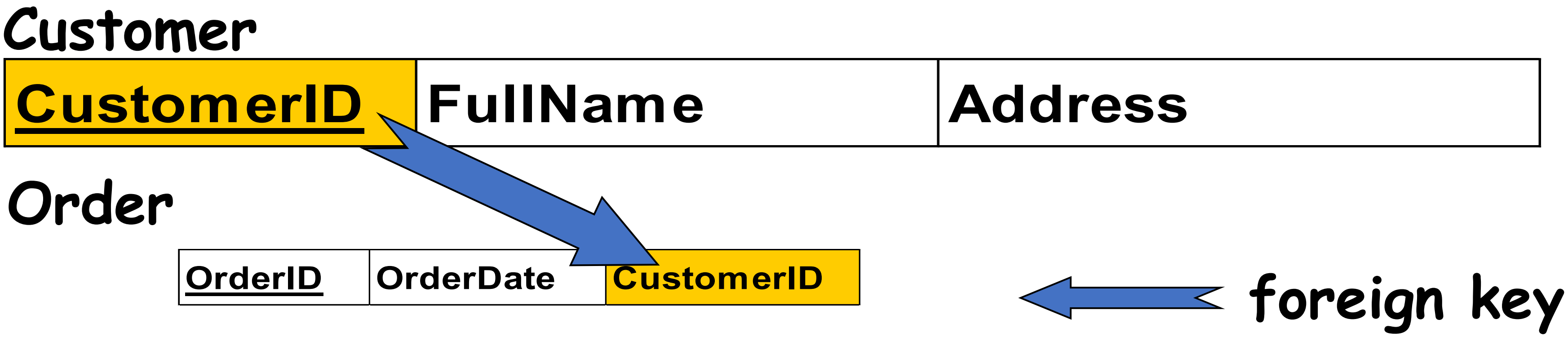
- In this example, there are 4 entities
- There are 2 one-many and 1 many-many relationships
- The attributes have not been shown

Identifiers & Primary Keys

- An identifier is an attribute that names or identifies an entity
  - Often this is a number e.g....
  - CustomerNo, StudentNo, EmployeeNo, StockCode, etc.
- If this is always unique, it can be the Primary Key for the entity
  - the primary key facilitates fast searching and linking of tables in the database
- Microsoft Naming Convention is to use ID even if the attribute is a number. Strings can be used but are harder to manage.

Foreign Keys

- A Foreign Key is the Primary Key of one entity that is stored again as an attribute of another entity
- The Foreign key serves as a link between the two entities



Customer

CustomerID	FullName	Address

Order

OrderID	OrderDate	CustomerID

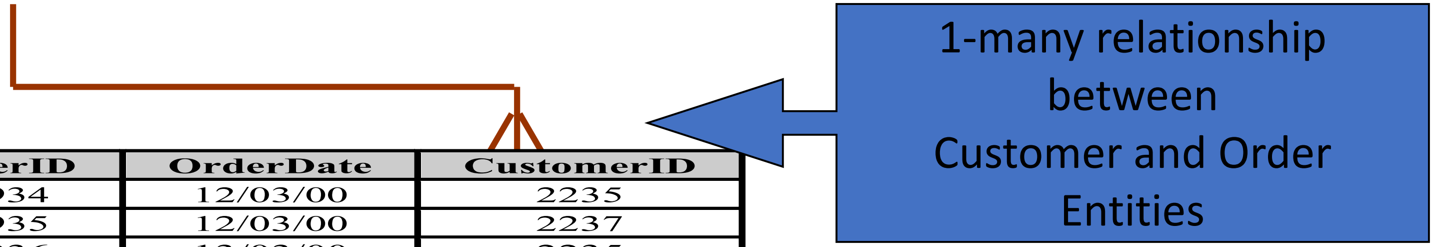
foreign key

Conversion into Tables

- From the E-R diagram we create tables with correct attributes and linkages:

Customer

CustomerID	Name	Address
2234	Bloggs Computers	2 Spon St, Watford
2235	Sissy Systems	3 Foo Lane, Marlow
2236	Black Box Ltd.	12 Laff Rd, Wycombe
2237	Reject Bros	8 Rot Row, Marlow



1-many relationship between Customer and Order Entities

Order

OrderID	OrderDate	CustomerID
99934	12/03/00	2235
99935	12/03/00	2237
99936	13/03/00	2235
99937	13/03/00	2237

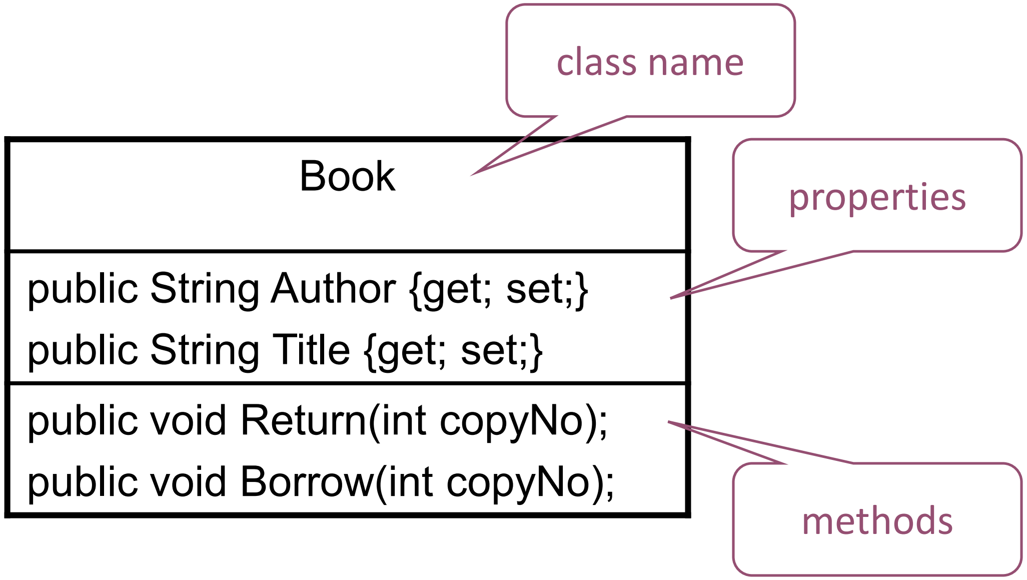
UML Class Models

- Class models are used to document the static structure of the system e.g. what the entities are and how they are related (not how they interact to achieve particular behaviours)
- Classes represent enduring objects in the real world
- A rectangle is used as a class icon (which can show Analysis, Implementation details or have the details suppressed)
- For CO550 CW1 we are really only concerned with ‘entity’ classes

Class Attributes and Operations

- A class has properties and methods.
- Properties are public data contained in a class with a getter and setter.
- Operations (i.e., methods) define the way classes interact (by sending messages).
- During OOA, focus more on attributes.

Example:



class name

Book

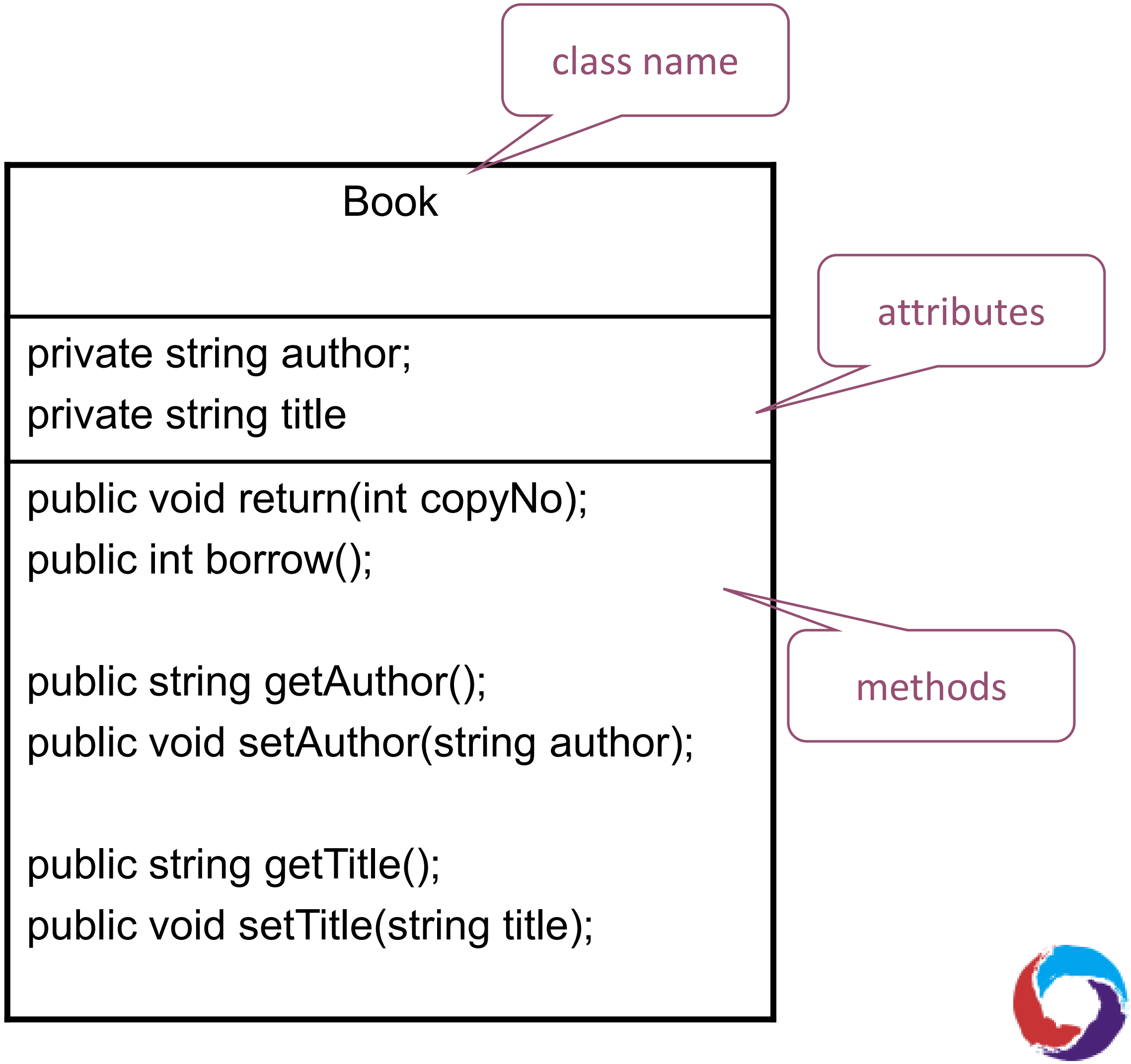
```
public String Author {get; set;}
public String Title {get; set;}
```

```
public void Return(int copyNo);
public void Borrow(int copyNo);
```

properties

methods

Java Equivalent



```
Book

private string author;
private string title

public void return(int copyNo);
public int borrow();

public string getAuthor();
public void setAuthor(string author);

public string getTitle();
public void setTitle(string title);
```

Identifying Classes - Nouns

Two stages

- Identify candidate classes by picking all the nouns and phrases out of a requirements specification.
- Discard candidates which are inappropriate for any reason, renaming the remaining classes if necessary.

Class Model Case Study: Library System - Requirements

- Scenario: develop computer system for a university library that handles bookkeeping and user browsing.

- Books and journals
  - Library contains books and journals.
  - May have several copies of a book.
- Users
  - Students
  - Librarians
- Borrowing
  - Must keep track of when books and journals are borrowed and returned.
  - System should produce reminders when a book is overdue.
- Browsing
  - User can search for a book by subject, author, title, etc.
  - User can reserve book.

Class Model Case Study: Library System – Noun identification

- Typical examples of classes (tangible or real-world things)
  - book, copy, journal, author
  - roles: library member, student, librarians, tutors

- Inappropriate examples

• library	(outside system scope)
• short term loan	(an event)
• member of library	(redundant, same as library member)
• week	(measures time, not a *thing*)
• item	(vague)
• time	(outside system scope)
• system	(meta-language)

Class Model Case Study: Library System - Noun Identification

- Nouns that are retained (first cut list of probable classes):
  - Book
  - Journal
  - Copy (of book)
  - Student
  - Staff

Class Model Case Study: Library System - Associations

- Associations express the relationship between classes
- In general, classes correspond to nouns, associations correspond to verbs.
- Just as there are instances of classes (*objects*) there are instances of associations (*links* in UML)
- For example:
- ‘a library member borrows a book’
- ‘an account generates a statement’
- ‘a customer orders a product’
- ‘a lecturer teaches a student’

Class Model Case Study: Library System

Relations/Associations between Classes

- We can identify the relations or associations between classes in a number of ways…

- Class A and class B are associated if:
  - a class A object sends a message to a class B object,
  - a class A object creates a class B object,
  - a class A object has a class B object as a component (composition),
  - a class A object receives a message with a class B object as an argument,

- In other words, we attempt to model the interaction between two (or more) classes.

Case Study: Library System Class Model - Multiplicities

- It would also be useful to model the cardinality of the relationship between classes
- Multiplicities represent this
- They tell us number of objects of one class that relate to a single object of an associated class
- It is possible to specify:
  - an exact number
  - a range of numbers
  - an arbitrary, unspecified number

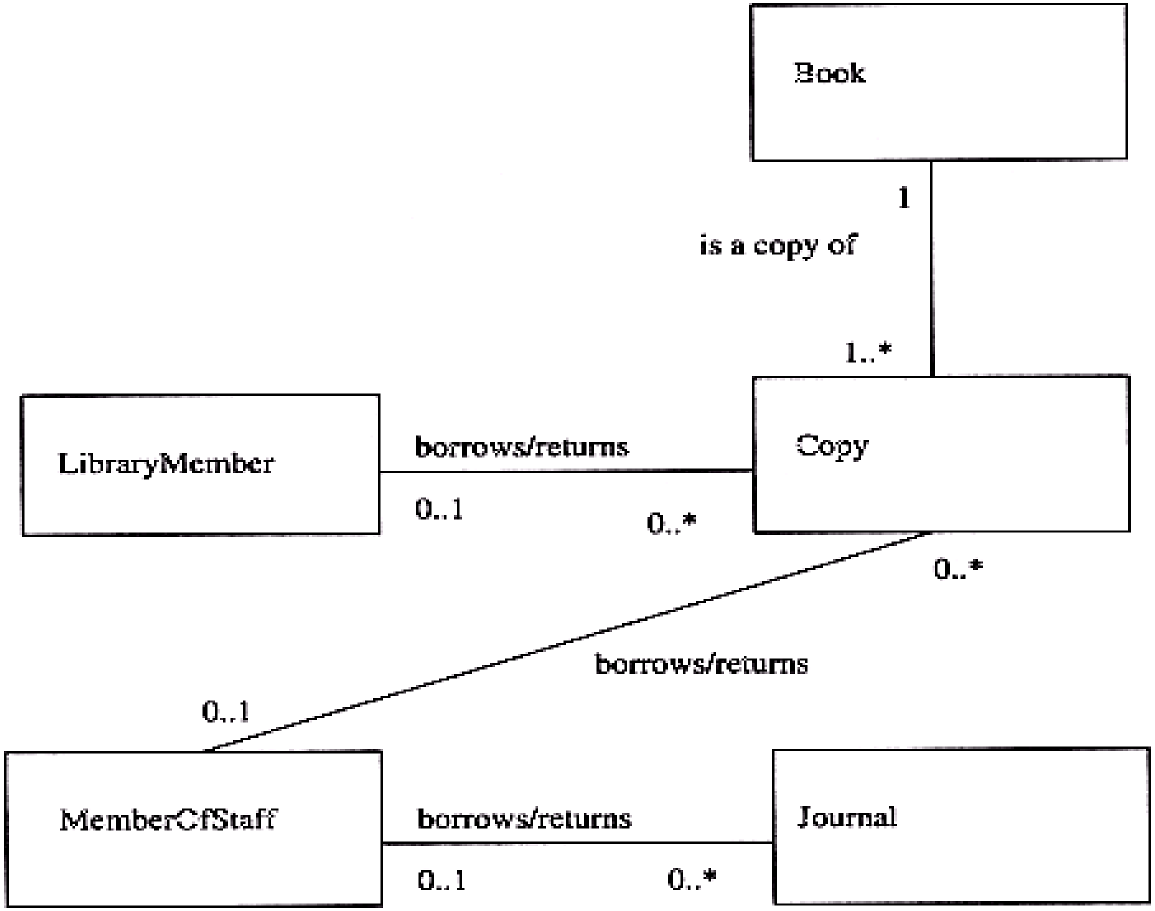
Class Model Case Study: Library System – Multiplicities

Example associations:

- is a copy of
- borrows/returns

Multiplicities

1	one
0..1	0 or 1
0..*	0 or many
1..*	1 or many



SUMMARY

- Reviewed Data Modelling and its purpose
- Entities, Attributes, and Relationships
- ER Diagrams and Class Models

NOW: Assignment Workshop

- Review project ideas in teams
- Start identifying your entities

Next UNIT

First step of ASP.NET Core Razor Pages tutorial

https://docs.microsoft.com/en-us/aspnet/core/data/ef-rp/intro?view=aspnetcore-2.1&tabs=visual-studio