

Lecture 13 - Introduction to Databases

Computer Systems, Data Structures and Data Management
(4CM508)

Dr Sam O'Neill

What should I have done to date?

- To date, you should have attempted the 9 MCQ Quizzes (you have unlimited attempts).
- There are going to be 12 assessed MCQ quizzes (originally 15)

I have reopened the first 9 tests.

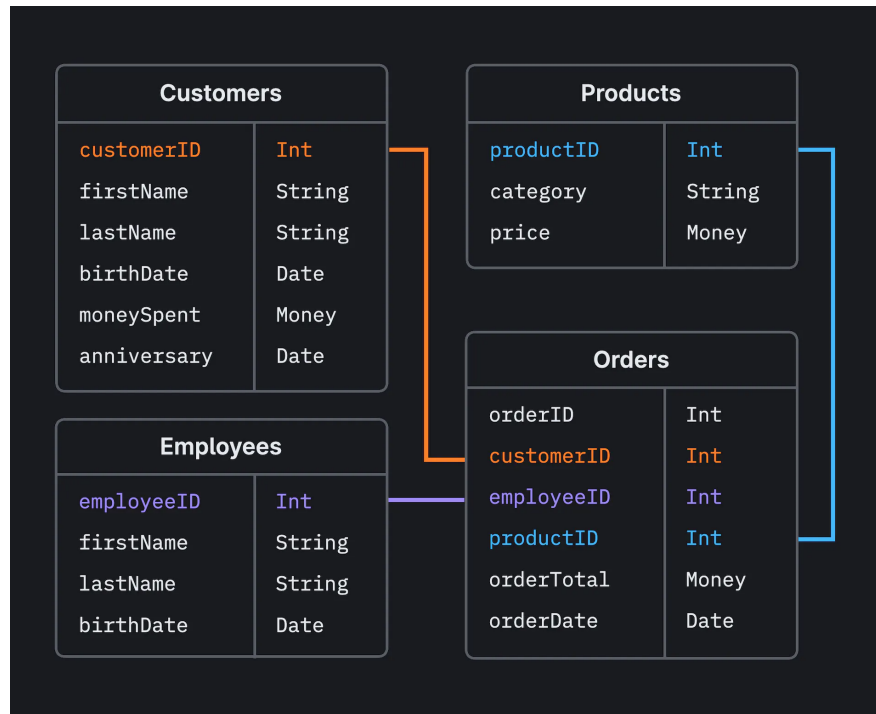
Relational Database Management System (RDBMS)

Organises data into tables with rows and columns; relationships between the tables.

- Based on relational model of data.
- Relational database first used by [E. F. Codd at IBM in 1970](#).

Examples

- MySQL
- PostgreSQL
- Microsoft SQL Server
- Oracle Database



Relational Database Management System (RDBMS)

Advantages:

- **Data Integrity:** Well defined schema.
- **Robust Transactions:** They follow ACID (Atomicity, Consistency, Isolation, Durability).
- **Standardised Query Language:** SQL used for querying (well established standard).
- **Maturity:** Around for a long time and have a large ecosystem of tools and best practices.

Disadvantages:

- **Scalability:** Scaled vertically by adding more powerful hardware, not designed to scale horizontally across multiple servers.
- **Flexibility:** Predefined schema, which can limit flexibility when dealing with unstructured or semi-structured data.
- **Complexity:** The strict schema and relationships can make them complex to set up and manage.

Entity Relationship Modelling

Entity Relationship Modelling

Before we start building a database we will look at the data model that is standard for relational databases.

This can be viewed diagrammatically which is very useful for designing a database.

Entities, Attributes and Values

- An **entity** is used to represent an object in the real world
 - physical (e.g. a **person**) or a concept (e.g. a **task**)
- An **attribute** is used to represent a characteristic of an entity
 - e.g. the **name** of a person
- An **entity** has a **value** for each of its **attributes**.
 - e.g. "Sam" would be the value of the **name** attribute of the **person** entity.

Instances and Entity Identifiers

- An **instance** is a specific set of values for an **entity**.
 - e.g. Li Wang is an **instance** of **Teacher** entity
- Each instance has a *unique* **entity identifier** (key attribute) to distinguish between different instances.

Entity Identifier



TeacherId	FirstName	MiddleName	LastName	DateofBirth	HireDate	Email
208	Li	Bella	Wang	05/12/1980	03/07/2000	li.wang@example.com
209	Simon		Bennett	19/11/1990	05/09/2000	simon.bennett@example.com

Composite Entity Identifier

- Sometimes a single attribute is not enough to create a *unique* entity identifier.
- Instead combine a *minimum* set of attributes to create a unique **composite entity identifier**.

Composite Entity Identifier



TeacherId	StartDate	EndDate	Reason
208	14/09/2022	15/09/2022	Sick Leave
208	25/10/2022	01/11/2022	Holiday
209	25/10/2022	25/10/2022	Jury Duty

Entity Descriptions

You can also write entities as follows:

```
EntityName (EntityIdentifier, attribute1, attribute2, attribute3, ...)
```

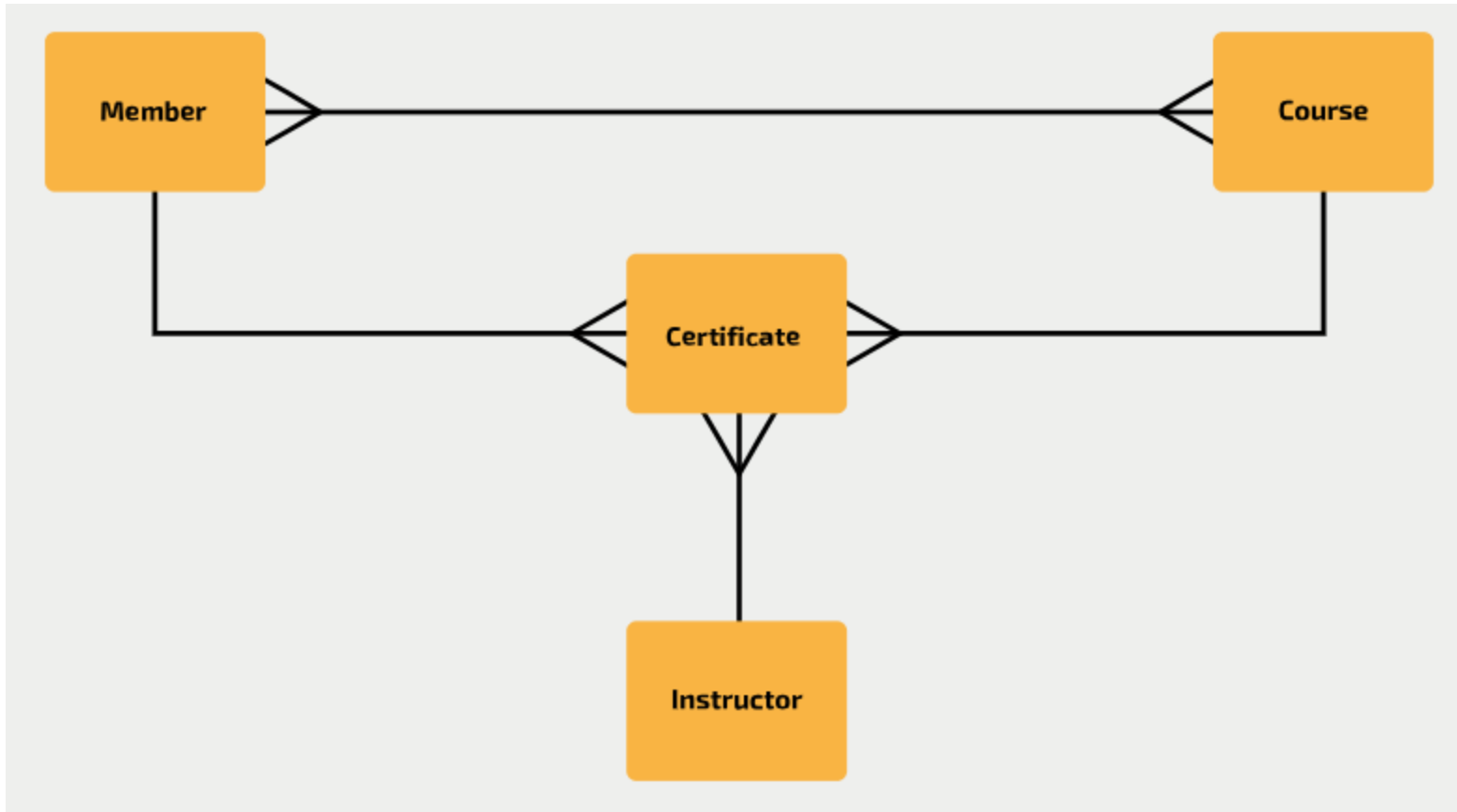
Note that the **entity identifier** is underlined.

e.g.

```
Teacher (TeacherId, FirstName, MiddleName, LastName, DateofBirth, HireDate, Email)
```

```
TimeOff (TeacherId, StartDate, EndDate, Reason)
```

Entity Relationship Diagram



Relationship Categories

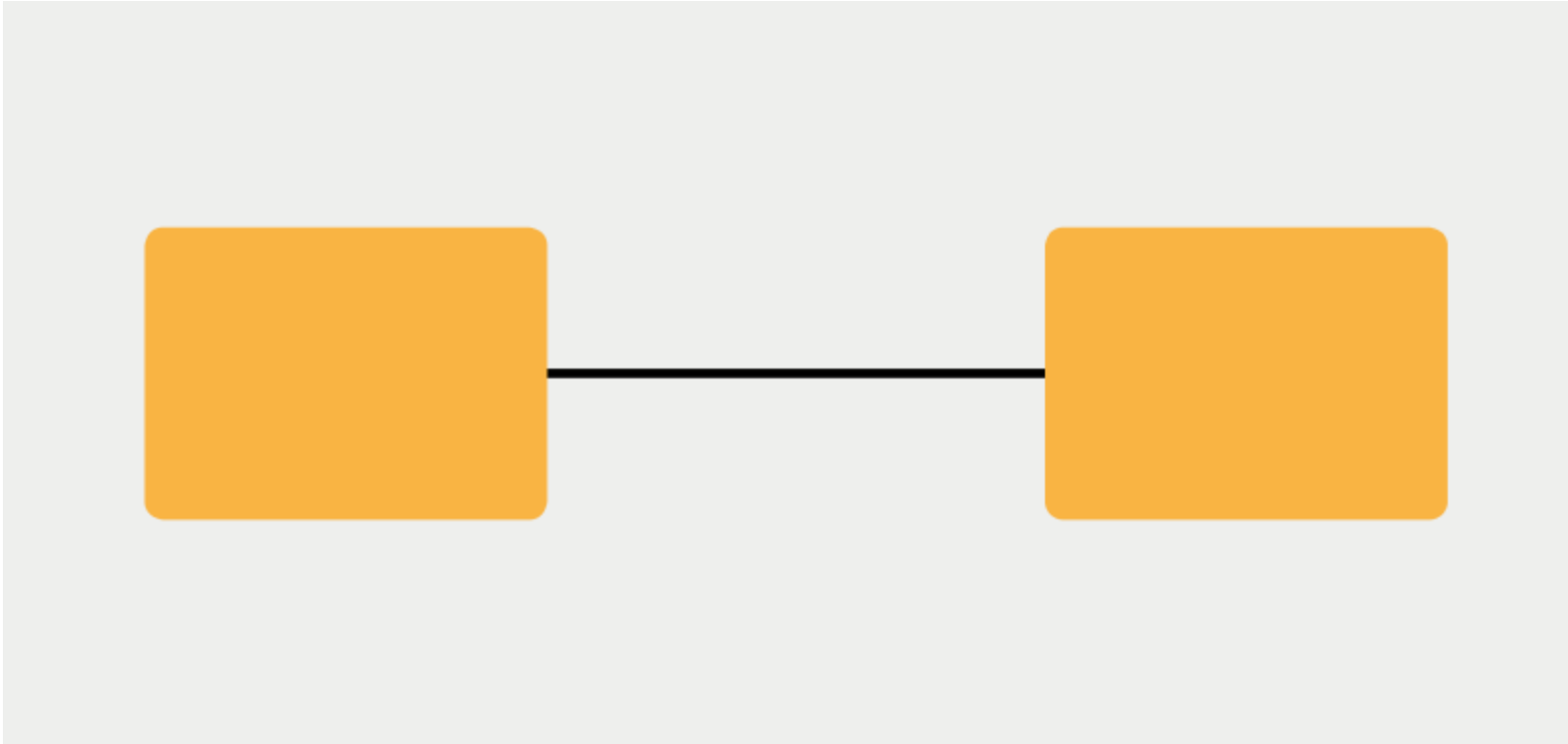
A **relationship** between entities is used to represent an interaction or association.

- A **one-to-one** relationship - One instance of an entity is associated with only one instance of another entity.
- A **one-to-many** - One instance of an entity is associated with more than one instance of another entity.
- A **many-to-many** - More than one instance of an entity is associated with more than one instance of another entity.

For a **cardinality** of **one** the end of the relationship line is straight.

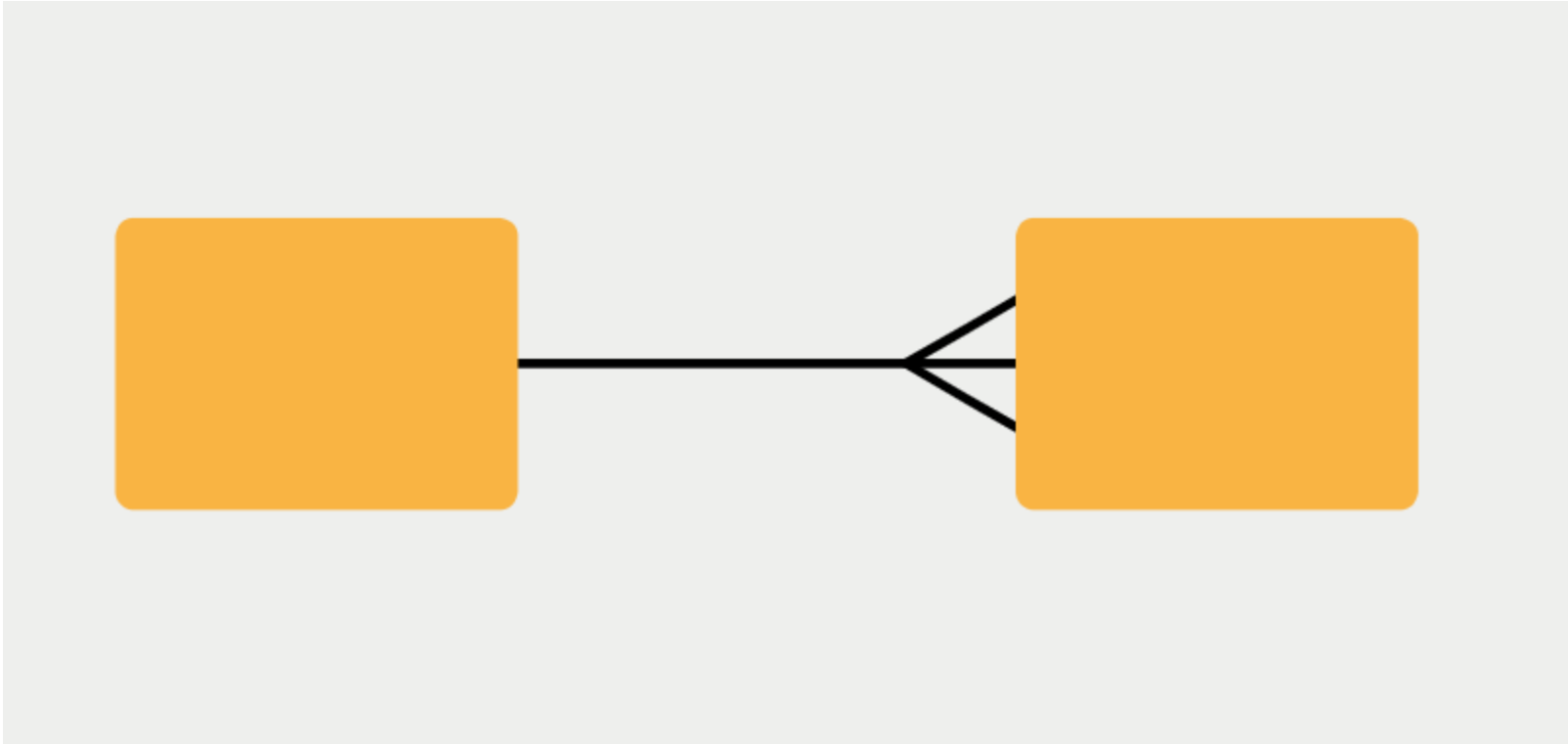
For a **cardinality** of **many** the end of the relationship line is a splayed line (**crow's foot**).

one-to-one



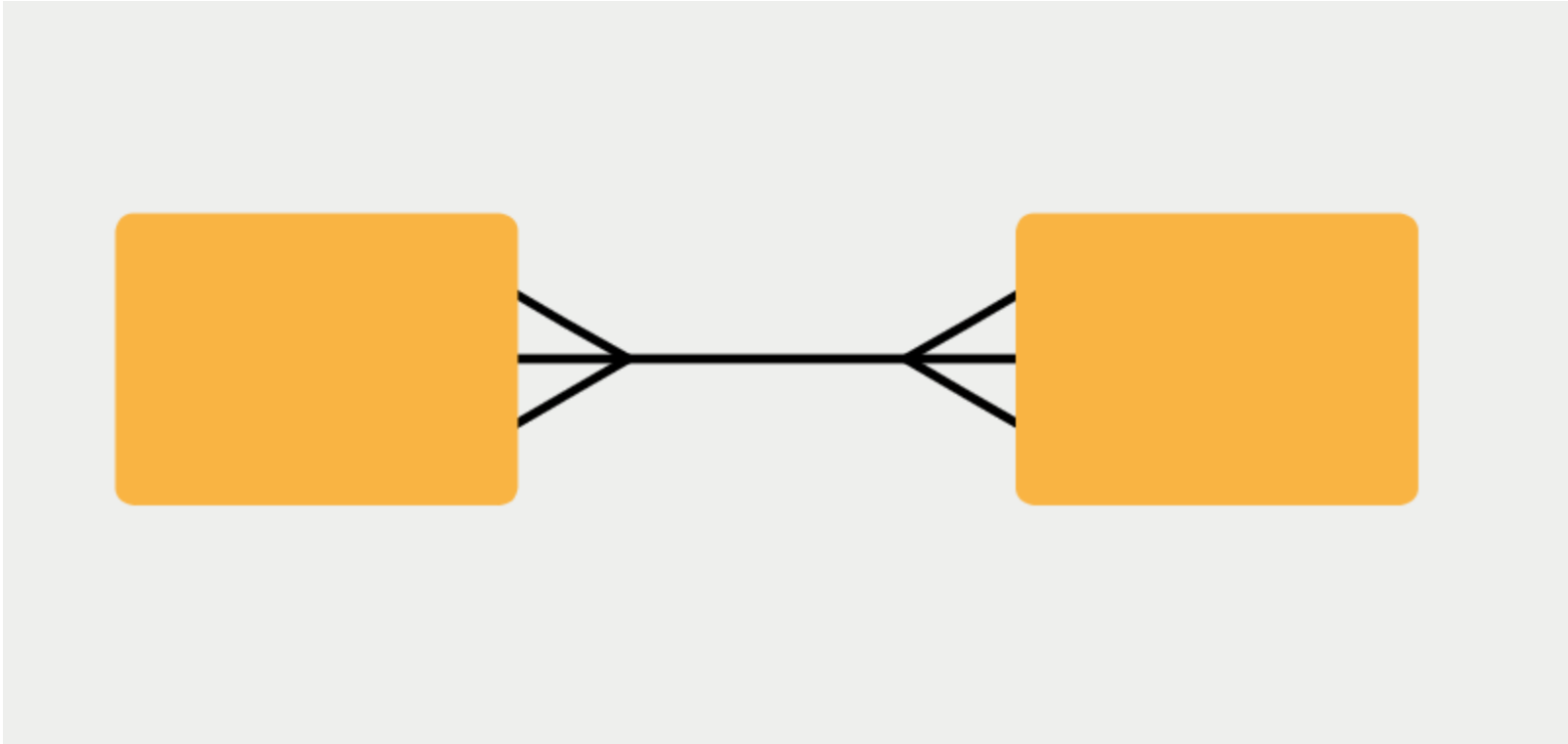
- A single line represents a **one-to-one** relationship - One instance of an entity is associated with only one instance of another entity.

one-to-many



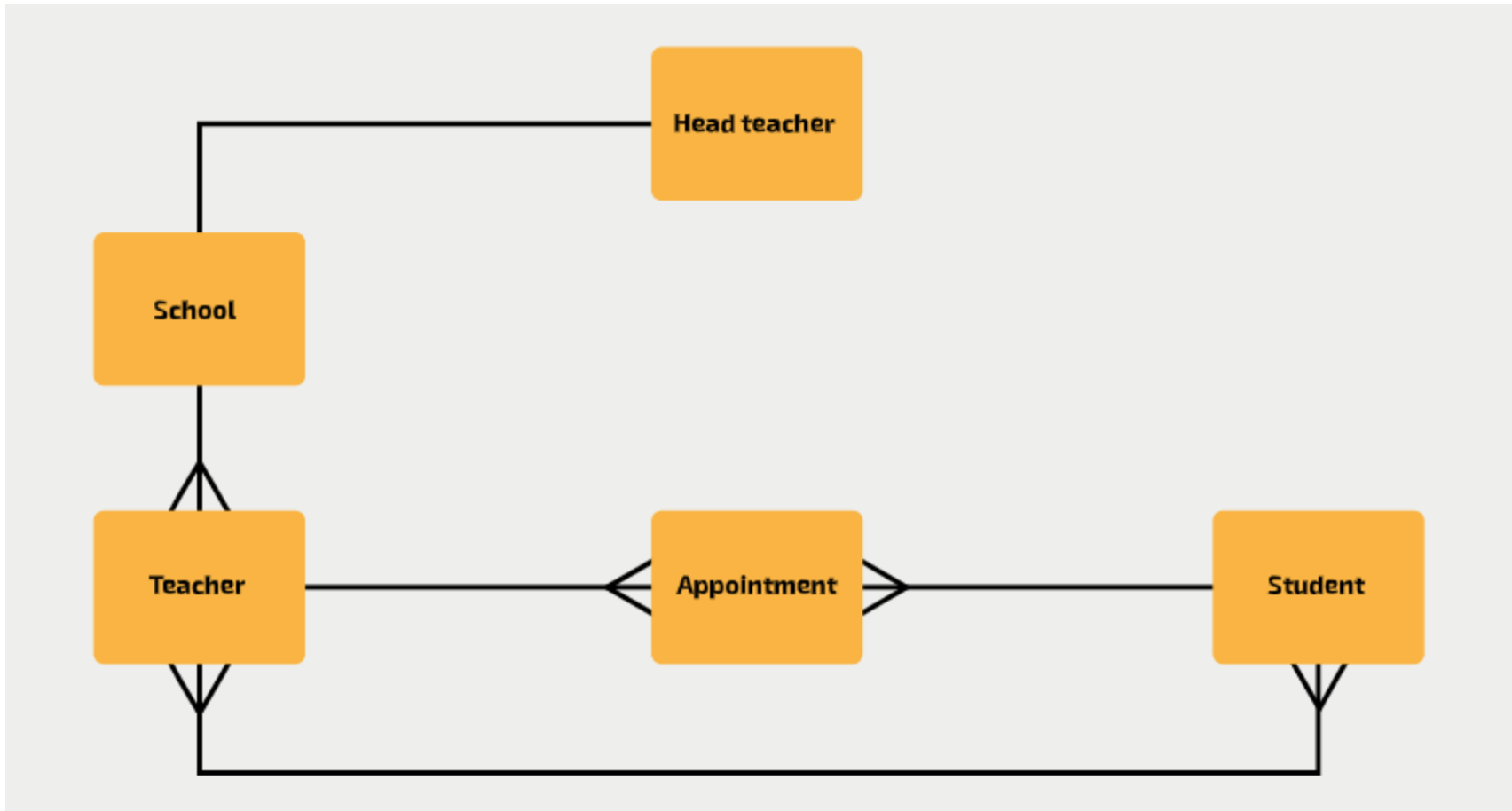
- A **one-to-many** - One instance of an entity is associated with more than one instance of another entity.

many-to-many



- A **many-to-many** - More than one instance of an entity is associated with more than one instance of another entity.

Example Entity Relationship Diagram



You should be able to understand the relationships based on this diagram.

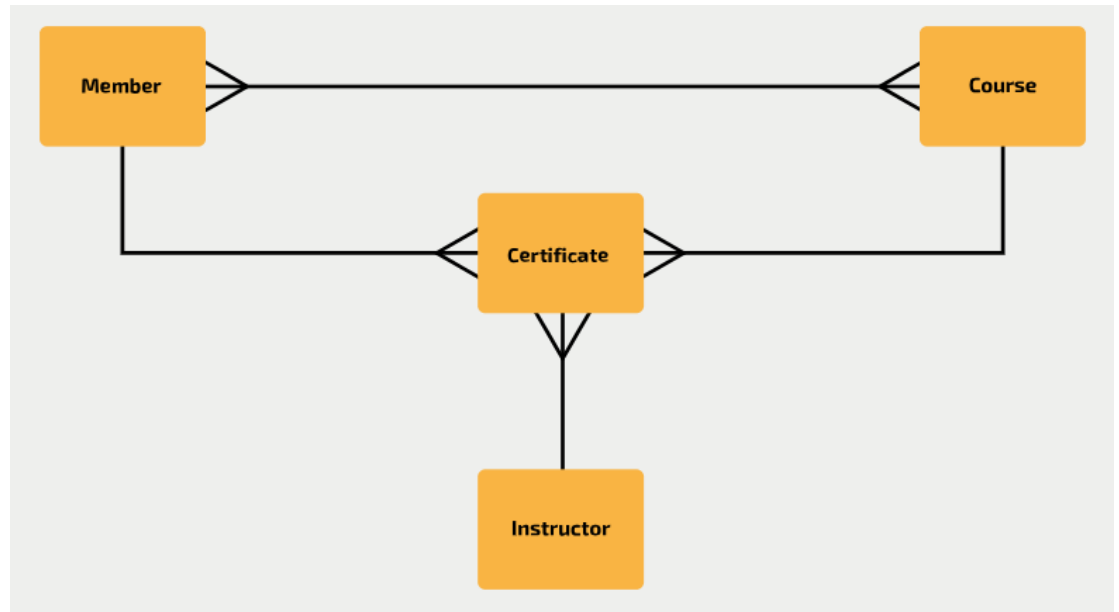
Relational Databases

Blue Peter Example

Consider the following example:

Here we represent a Sports club with 4 entities.

- What are their relationships?
- How do we implement this?



Tables

- A **table** is a collection of related data made up of columns and rows.
- It is an implementation of an **entity**. e.g. a Person
- Columns are known as **fields** and rows as **records**.
- **Fields** represent the attributes of the entity. e.g. name of a Person.
- A **record** is an instance of an entity. i.e. a given Person. Each **row** is a single **record**.

Tables

- Each table represents a database **entity**. e.g. A *member* of a sports club.

The diagram illustrates a database table structure. A horizontal line labeled "Fields" has four arrows pointing down to the column headers of a table. To the left of the table, a vertical line labeled "Records" has five arrows pointing right to the rows of the table. The table has four columns: MemberId, FirstName, LastName, and Phone. It contains five rows of data.

MemberId	FirstName	LastName	Phone
012010	Emily	Marr	01632 961743
131092	Joe	Donald	01632 960007
132099	Abdel	Patel	01632 967267
145543	Precious	Jones	01632 962816
148765	Jack	Marr	01632 961743

Member (MemberId, FirstName, LastName, Phone)

Primary Key

- A primary key is a **field** (or set of **fields**) that *uniquely identifies* each **record**.
- It is an implementation of an entity identifier.

Primary key →


MemberId	FirstName	LastName	Phone
012010	Emily	Marr	01632 961743
131092	Joe	Donald	01632 960007
132099	Abdel	Patel	01632 967267
145543	Precious	Jones	01632 962816
148765	Jack	Marr	01632 961743

Member (MemberId, FirstName, LastName, Phone)

Composite Key

Composity primary keys are made up of more than one field.

Composite key



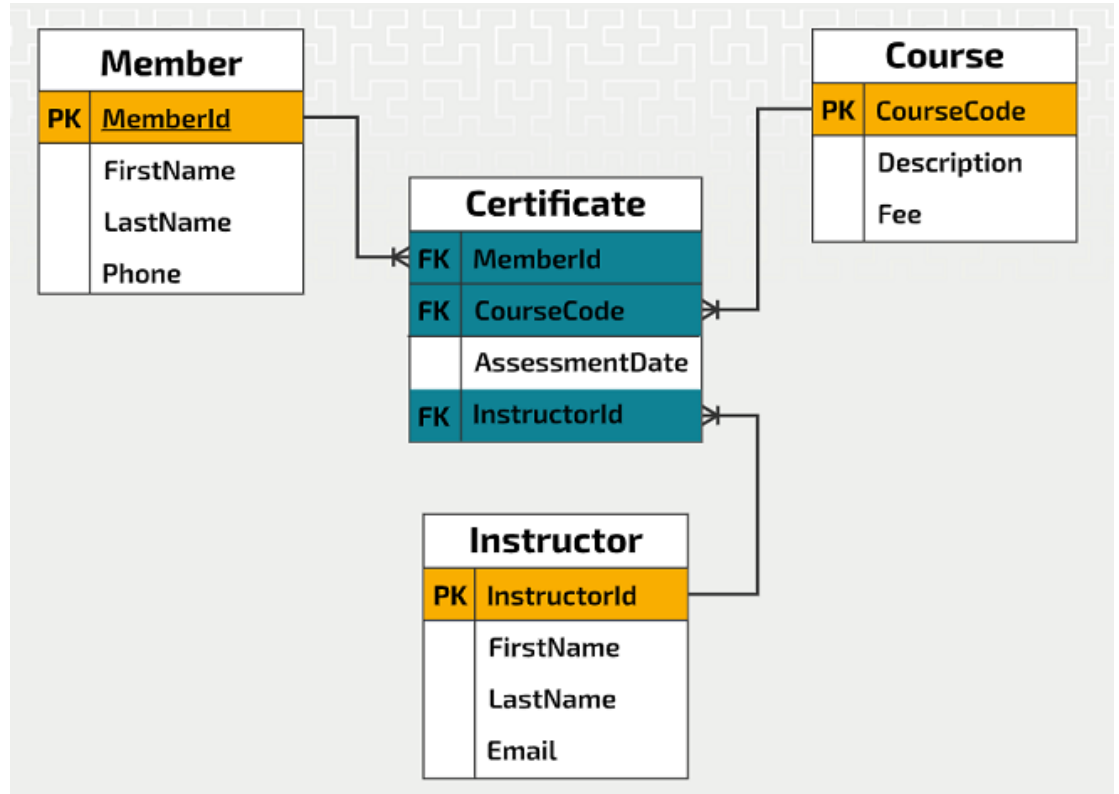
MemberId	CourseCode	AssessmentDate	InstructorId
145543	DG0011	21/02/2019	1
012010	DG0011	21/02/2019	1
132099	DG3002	01/03/2019	2
131092	CR0001	15/04/2019	3

Certificate (MemberId, CourseCode, AssessmentDate, InstructorId)

Foreign Key

A foreign key is a field (or set of fields) in a table *that appears as the **primary** key of another table*.

It is used to represent **relationships**.



Microsoft Access Demonstration

Demonstration. If reading these slides please view the Sports Club Access Demo -
`Sports Club Demo.accdb` .