ER Diagram Notation

Table of contents

Core symbols 1
Relationship constraints 2
Subclasses / superclasses (EER) 2
Naming conventions (INFS1200/7900 style guide) 3
How diagrams are drawn in these notes 3

Reference legend for the symbols used across every ER/EER diagram in this course — introduced across 2026-02-23-conceptual-database-design-and-the-er-model[1] and 2026-03-02-weak-entities-the-eer-model-and-design-choices[2].

The notation below is the **mandatory** standard: it is what the lectures' own notation guide specifies, and it is what assignments and the exam are marked against. (The separately published *INFS1200/7900 EER & ER Notation Guide* handout only standardises *naming* — see the style guide at the bottom of this note — and explicitly defers the symbols themselves to the lecture material.)

Core symbols

Symbol	Meaning
Rectangle	**Entity type**
Double-line rectangle	**Weak entity type** (no key attribute of its own)
Oval	**Attribute**
Oval, name underlined	**Key attribute**
Oval, name underlined with a dotted line	**Partial key attribute** (weak entity's own distinguishing attribute)

[1] courses/infs1200/2026-02-23-conceptual-database-design-and-the-er-model.pdf
[2] courses/infs1200/2026-03-02-weak-entities-the-eer-model-and-design-choices.pdf

Symbol	Meaning
Double-line oval	**Multivalued attribute** (can hold more than one value)
Dashed oval, attached by a dashed line	**Derived attribute** (computed from another stored attribute, e.g. `Age` from `BirthDate`)
Diamond	**Relationship type**
Double-line diamond	**Identifying relationship type** (connects a weak entity to its owner entity)

- **Composite attribute**: an attribute with its own sub-attributes attached (e.g. `Name` splitting into `FirstName`/`MiddleName`/`LastName`) — the parent attribute oval isn't underlined/dashed/doubled itself unless it's also a key/derived/multivalued attribute.
- **Value sets** (the domain of legal values for an attribute, e.g. "integers 21-65") are never shown on the diagram itself.

Relationship constraints

Symbol	Meaning
`1`, `N`, `M` on the lines into a diamond	**Cardinality ratio** — `E1 -1- R -N- E2` reads as ratio 1:N for E1:E2 in R
Double line between an entity type and a diamond	**Total participation** of that entity type in that relationship
Single line	**Partial participation**

See relationship-constraints[3] for what each ratio and each participation constraint actually asserts.

Subclasses / superclasses (EER)

A subclass is drawn as an ordinary rectangle. What makes it a subclass is the symbol on the line joining it to its superclass, not the shape of its box — a subclass box is a plain single-line rectangle, never the double-line rectangle that means *weak entity*. The two are entirely different ideas (a subclass inherits its superclass's key; a weak entity has no key of its own at all) and the notation keeps them apart.

[3] courses/infs1200/relationship-constraints.pdf

Symbol	Meaning
on the line between two entity types	The entity type on the *closed* (narrow) side of the is a **subclass** of the one on the *open* (wide) side — read it as the set-inclusion symbol, `subclass superclass`
A circle joining several subclass lines to one superclass, each line carrying its own opening towards the circle	Several entity types are all **subclasses of the same superclass**
`d` inside that circle	**Disjoint** specialisation — an entity instance can belong to at most one of the subclasses
`o` inside that circle	**Overlapping** specialisation — an entity instance can belong to more than one subclass at once
Single line from superclass to the subclass circle	**Partial** specialisation — not every superclass instance needs to belong to a subclass
Double line from superclass to the subclass circle	**Total** specialisation — every superclass instance must belong to at least one subclass

Naming conventions (INFS1200/7900 style guide)

Course-specific naming standard for entities/relationships/attributes on ER and EER diagrams (distinct from — and in addition to — the mandatory notation above):

- **Entity names**: capitalised, no spaces, ideally one word (e.g. `STUDENT`, `POLICEOFFICER` — not `PoliceOfficer`).
- **Relationship names**: capitalised, no spaces, ideally one word, preferably a verb (e.g. `CREATES`, `ACTSIN` — not `LeadActor`).
- **Attribute names**: UpperCamelCase — first letter of each word capitalised, no spaces, acronym letters stay capitalised (e.g. `ComputerIP`, `DateOfBirth` — not `Date of Birth`).

How diagrams are drawn in these notes

Diagrams in this course's notes are drawn as Mermaid `flowchart` graphs (renders in both PDF and HTML), using shapes chosen to match the Chen notation above as closely as Mermaid's flowchart shapes allow:

Mermaid shape	Used for
`[Entity]` (rectangle)	Entity type — **and subclasses**, which are ordinary rectangles

Mermaid shape	Used for
`[[WeakEntity]]` (subroutine, double vertical bars)	Weak entity type, and nothing else
`{Relationship}` (diamond/rhombus)	Relationship type
`([Attribute])` (stadium)	Attribute
`((d))` / `((o))` (small circle)	Disjoint/overlapping specialisation marker

The `[[X]]` shape is **reserved for weak entity types**. Drawing an EER subclass with it is a notation error, not a stylistic one: it renders the double-line box that the official notation uses for a weak entity, and so asserts the opposite of what a subclass is. Subclasses are plain `[X]` rectangles; their subclass-ness is carried by the edge into the `((d))` or `((o))` circle, exactly as the carries it in the hand-drawn notation.

Since Mermaid can't reliably underline/dash node text across renderers, attribute qualifiers are written as a plain-text suffix instead: `(key)`, `(partial key)`, `(derived)`, `(multi)`. Composite attributes are drawn as a parent attribute node connected to its component attribute nodes. Cardinality is a text label on the edge (`"1"`, `"N"`, `"M"`); **total** participation/identifying relationships use a thick edge (`===`/`==>`), **partial** participation uses a thin edge (`---`/`-->`).

Any diagram with more than ~6-8 nodes needs an explicit `%%| fig-width:` cell option (in inches) or it runs off the right edge of the PDF page - mermaid sizes itself from the diagram's own SVG bounding box, and there's no project-wide default that reaches it (a document/project `fig-width` only applies to executed-code-cell figures, e.g. matplotlib). `5.5` fits this vault's PDF page width (`scrartcl`, `DIV=11`, letter) comfortably:

````
```{mermaid}
%%| fig-width: 5.5
flowchart TD
    ...
```
````