Interpreting ER Diagrams

Table of contents

The checklist . . . 1
Cardinality: read it from the far side . . . 2
Participation: absence of a double line is a fact . . . 2
Keys: what isn't underlined can repeat . . . 3
Weak entities: the partial key is not the key . . . 3
Checking a candidate relationship set . . . 4
Subclasses: `d` and `o` are per circle . . . 4
Degree: count the lines . . . 5

Drawing a diagram and *reading* one are different skills, and the applied classes test the second at least as hard as the first: given an ER/EER diagram and a plain-English claim about the world, decide whether the diagram actually says it. Practised in the statement-evaluation questions in week2-tutorial-applied-class-1-intro-to-dbms-and-basic-erd[1] and week3-tutorial-applied-class-2-entity-relationship-diagrams[2], and in the questions in 2026-02-23-conceptual-database-design-and-the-er-model[3] and 2026-03-02-weak-entities-the-eer-model-and-design-choices[4]. See er-diagram-notation[5] for what each symbol is.

The checklist

If the claim is about…	Check	Trap
How many X per Y	The cardinality ratio, **from the far side**	Reading 1:N in the wrong direction

[1] courses/infs1200/week2-tutorial-applied-class-1-intro-to-dbms-and-basic-erd.pdf
[2] courses/infs1200/week3-tutorial-applied-class-2-entity-relationship-diagrams.pdf
[3] courses/infs1200/2026-02-23-conceptual-database-design-and-the-er-model.pdf
[4] courses/infs1200/2026-03-02-weak-entities-the-eer-model-and-design-choices.pdf
[5] courses/infs1200/er-diagram-notation.pdf

If the claim is about…	Check	Trap
Whether every X does something	Participation (double vs single line)	Assuming total participation because the relationship exists
Uniqueness of an attribute	Whether that attribute is underlined	Treating a non-key attribute as if it were unique
"Only identifiable by…"	Whether a *second* key oval exists	Missing a second candidate key
A weak entity's identity	Owner's key + partial key	Treating the partial key as a full key
An instance being two kinds at once	`d` vs `o` on the specialisation circle	Reading two separate circles as one
A relationship's degree	Count the lines into the diamond	Calling a pair of binaries "ternary"
Whether a given relationship *set* is possible	Every cardinality ratio **and** every participation constraint, tuple by tuple	Checking the ratios and forgetting the double lines

Cardinality: read it from the far side

A 1:N relationship constrains *both* ends, and the interesting constraint is usually the one on the "many" side. `HEALTHWORKER` —1:N `OVERSEEN`→ `HEALTH` means one health worker can oversee many participants, and therefore each participant has **at most one** overseeing health worker. "A participant can be overseen by several healthcare workers" is *incorrect* against that diagram — a claim that reads naturally in English and is contradicted by the diagram in the direction most people don't check. See relationship-constraints[6].

An M:N relationship with no distinguishing attribute also says something: each pair appears at most once. An athlete/event pair in a plain `PARTICIPATES` relationship cannot be recorded twice, so "an athlete cannot participate at the same event more than once" is *correct*. Making repeats possible requires a distinguishing key attribute (a `Timestamp`, say) or a weak entity — see er-design-choices[7].

Participation: absence of a double line is a fact

Total participation (double line) says every instance *must* participate; partial participation (single line) says it need not. Both are claims, and the partial one is easy to skip over:

[6] courses/infs1200/relationship-constraints.pdf
[7] courses/infs1200/er-design-choices.pdf

- The recursive `WITH` relationship on `VACCINE` is partial on both sides, so "each vaccine must have a second dose or a preceding dose" is *incorrect* — the relationship allows a two-dose program, it doesn't require one.
- `OVERSEEN` has total participation on the `HEALTH` side, so "every participant in the HEALTH subclass is overseen by a health worker" is *correct*.
- Some venues are backups and never host an event, which is exactly why `VENUE`'s participation in `HOSTS` is partial — "all venues must host an event" is *incorrect*.

See relationship-constraints[8].

Keys: what isn't underlined can repeat

If an attribute isn't marked as a key, the diagram is asserting nothing about its uniqueness — so two different venues *may* share an `Address` when only `ID` is underlined. Conversely, `HOTEL.Address` being the key attribute is what makes "two hotels can share an address" false.

Claims of the form "X can *only* be identified by Y" need the whole attribute list checked, because entity types routinely have more than one candidate key: `VACCINE` has both `ID` and the composite `Brand` + `Name`, so "a vaccine can only be identified by the combination of its brand and name" is *incorrect*. See entity-types-and-attributes[9].

Weak entities: the partial key is not the key

A weak entity's partial key is unique only *within* its owner, so any claim that treats it as globally unique is false. The hotel/room question is the canonical case: `ROOM`'s real key is `(HOTEL.Address, ROOM.Number)`, so neither "no two rooms share a number" nor "no two hotels have rooms with the same number" holds — room numbers repeat freely across hotels, which is the entire reason `ROOM` is weak. The same logic makes "a BATCH is uniquely identified by only its number" incorrect, and makes an emergency contact named "Mum" registrable by two different users.

The hotel/room question is also a reminder that "none of the above" is a live option: with `HOTEL.Address` as the key and `ROOM.Number` as a partial key, *every* uniqueness claim on offer — two hotels sharing an address, no two rooms sharing a number, no two hotels having rooms with the same number — is false at once.

See relationship-constraints[10].

[8] courses/infs1200/relationship-constraints.pdf
[9] courses/infs1200/entity-types-and-attributes.pdf
[10] courses/infs1200/relationship-constraints.pdf

Checking a candidate relationship set

A second question format hands you a small diagram plus several candidate relationship sets written out as tables of tuples, and asks which one is *possible*. It is mechanical once you know the two things to check:

Notation on the diagram	What it forbids in the table
A `1` on E's side of R	An entity of E appearing in more than one row of R
An `N` or `M` on E's side of R	Nothing — that side is unconstrained
A double line between E and R	An entity of E appearing in *no* row of R

Worked instance. Take `A` —N— `R` —1— `B` —N— `S` M `C`, with entity sets `{a , a }`, `{b , b }` and `{c , c }`. The `1` on `B`'s side of `R` means each `a` may appear at most once in `R`; the double line on `C`'s side of `S` means every `c` must appear somewhere in `S`; `M:N` on `S` constrains nothing else. So:

- `R = ` , `S = ` fails — `c` and `c` are missing from `S`.
- `R = {(a ,b )}`, `S = {(b ,c )}` fails — `c` is still missing.
- `R = {(a ,b ), (a ,b )}` fails — `a` appears twice, violating the `1`.
- `R = {(a ,b )}`, `S = {(b ,c ), (b ,c ), (b ,c )}` is possible — no `a` repeats and both `c`s appear.

Note that an *empty* relationship set is perfectly legal wherever participation is partial; it is only the double line that forces rows to exist. That is the half of the check people skip.

Subclasses: `d` and `o` are per circle

An **overlapping** (`o`) specialisation permits an instance to be in several subclasses at once, so "a participant can be in both the HEALTH subclass and the AGE subclass" is *correct* when those two hang off an `o` circle. A **disjoint** (`d`) specialisation forbids it. The per-circle part matters: a superclass with two separate specialisation circles constrains each one independently, and membership in a subclass under one circle says nothing about membership under the other. See specialisation-and-generalisation[11].

[11] courses/infs1200/specialisation-and-generalisation.pdf

Degree: count the lines

"EVENT has a ternary relationship" is *incorrect* when `EVENT`'s relationships are `PARTICIPATES` (to `ATHLETE`) and `HOSTS` (to `VENUE`) — two separate binary relationships, not one three-way one. An entity type participating in several relationships is not the same as participating in a high-degree relationship. See relationship-types[12].

[12] courses/infs1200/relationship-types.pdf