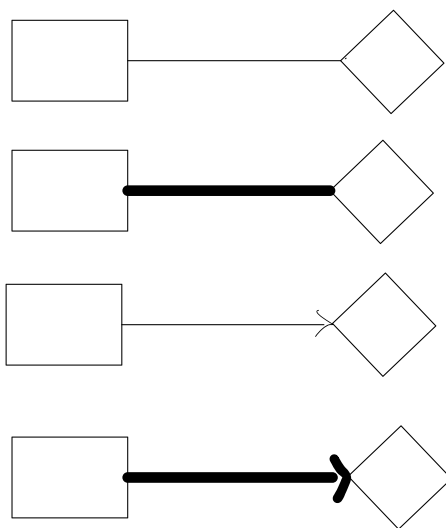


Chen's notation

v.s.

Crow's foot notation



participation

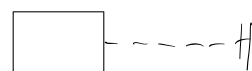
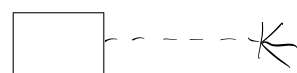
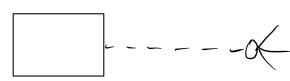
key constraint

0 ... m

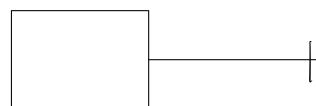
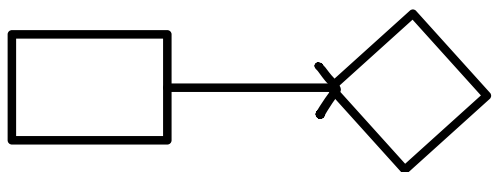
1 ... m

0 ... 1

1 ... 1



Weak entity & identifying relationship



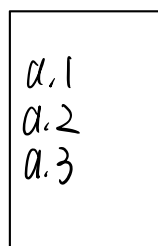
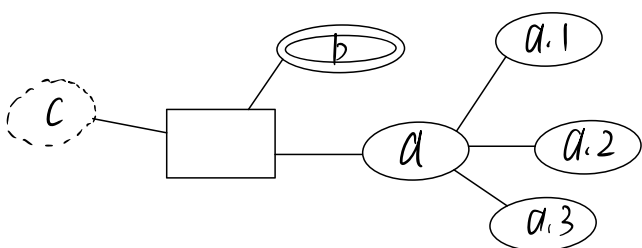
From conceptual (w/ chen's) to logical & physical (w/ crow's foot)

Note: Crow's foot notation can be used in conceptual as well!

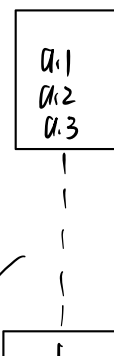
Conceptual

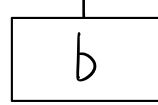
Logical

Step 1: Flatten composite & multi-value attribute, remove derived attribute



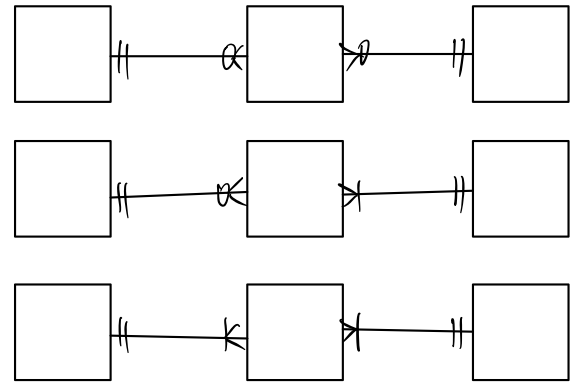
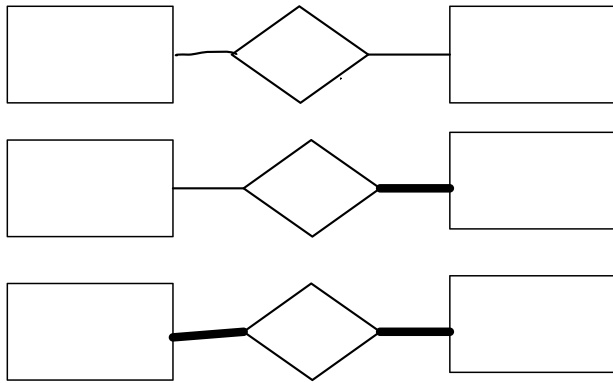
or





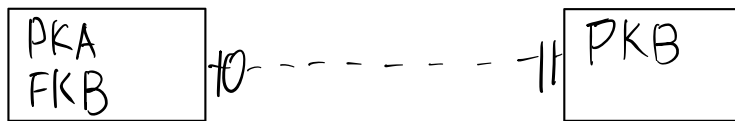
weak entity or not and constraints are decided by scenario

Step 2: Resolve many-to-many relationships by creating associative entity



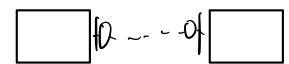
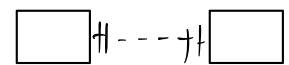
Step 3: Add foreign keys (at Crow's foot notation), place relationship attributes to correct table

Rules: **One-to-one** relationships are resolved by adding a foreign key on either table, giving preference to the table that has mandatory participation in the relationship if there is only one.



FKB refers to PKB

Random choose



**One-to-many** relationships are resolved by adding a foreign key on the **one** side of the relationship. (in Chen's)

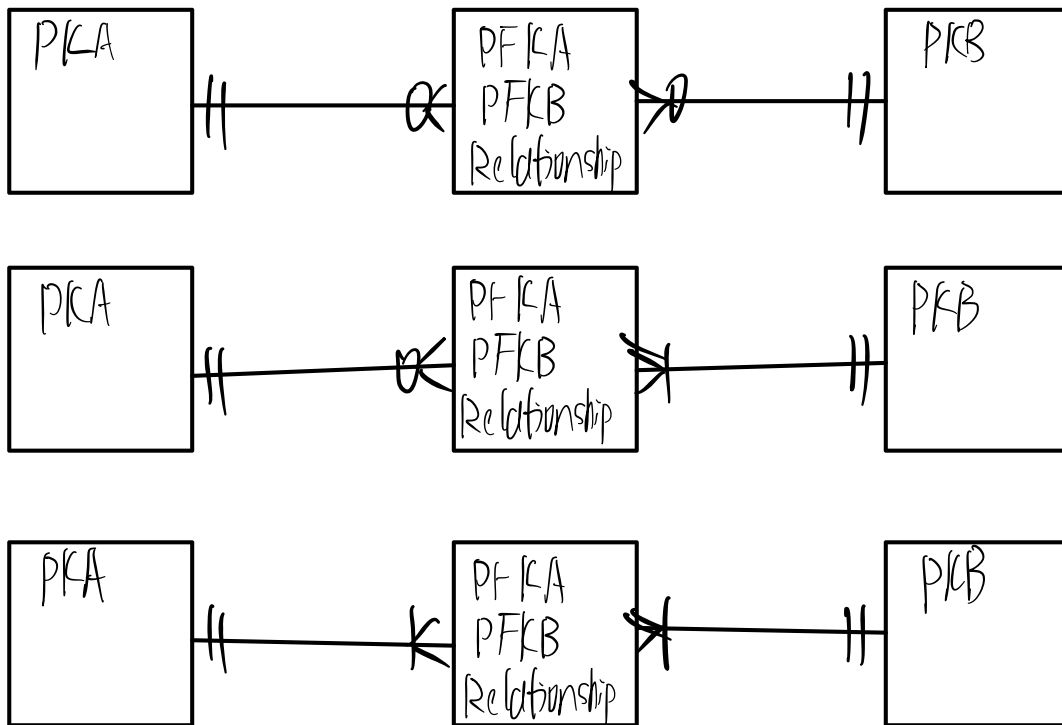


FKA refers to PKA

Many side in crow's foot notation

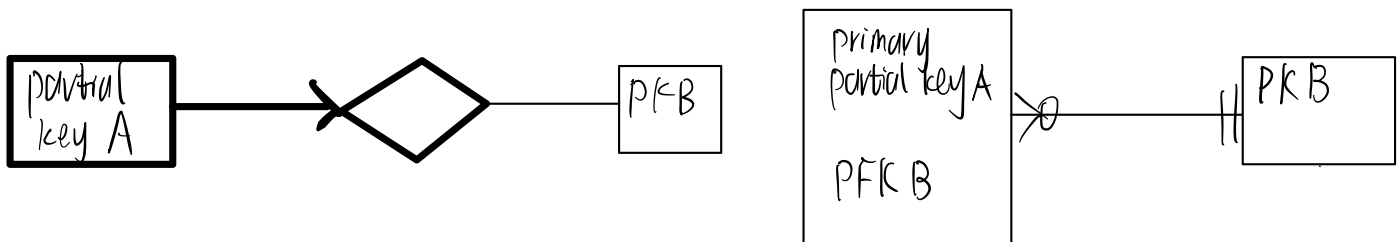
Relationship attribute: placed in the same table as foreign keys

**Many-to-many** relationships are resolved by creating a new entity called an “associative entity”. This entity contains primary foreign keys for each table in the relationship.



Note: relationship means the relationship attributes

Weak entity:



Physical design:

step 1: Decide data types

For data types:

- Go back to the business case and look for any ideas as to which data type should be used.
- Foreign key columns must have the same data type as the primary key column they refer to.
- If it is not clear which data type to use (for example, “model number”), pick any data type that seems reasonable. Different people might choose different data types for some columns.

Step 2: Decide Null / Not null

For NULL/NOT NULL constraints:

- Primary key columns must always be NOT NULL.
- For foreign keys, look at the conceptual model, specifically the participation constraints. Is the table's participation in the relationship mandatory? If so, the foreign key must be NOT NULL. For example, every Screen must be located in a Cinema, so the CinemaID foreign key on the Screen table is NOT NULL. On the other hand, the ProjectorSerialNumber column on Screen is optional (may be NULL), because not every Screen is associated with a Projector.
- For other columns, think carefully about whether the data in that column is required or optional.