Normalisation: quick guide

1 NF: all attributes must be atomic.

- no multivalued attributes allowed!
- Each attribute is **considered** to be a whole (i e you **use** an ID-number as a **whole**, not as a conglomerate of its constituent parts: day-month-year-sign-number-controlNumber)

2 NF: partial dependencies by non-key attributes on the primary key are not allowed.

- That means: if you have a primary key consisting of two attributes A and B, and a functional dependency (FD) B -> C, and C is **not part of any candidate key for this relation**, then this relation does not satisfy 2 NF.
- B alone determines C, so that information should be put into a separate table.
- Example: exerc. 5, budo-data. The primary key is membership-nr, art, trainer , but membership-nr alone determines memberName:
  membership-nr -> memberName
  memberName is **partially dependent** on the primary key. Moreover, memberName is not part of any candidate key for the relation, it is not a key-attribute. A relation that has such FD:s does **not** satisfy 2 NF.
- Example: think of a table
  Course = (courseCode, courseName, teacherID, teacherName)
  We assume that there is one teacher only for each course, and his/her ID and name are included in this table. The FD:s for this table are:

  courseCode -> * (all the attributes - just as a primary key should!)
  teacherID -> teacherName

  This means storing the same information (the teacher's name) several times (each time he/she gives a course). However, *teacherID* is NOT a part of the primary key (or any other candidate key), so the non-key attribute *teacherName* is not partially dependent on the primary key. This is a dependency among non-key attributes and satisfies 2 NF.

3 NF: dependencies among key attributes are allowed:

- If we have a FD of the form A -> B, one of the following must be true:
- A is a **superkey** for the relation, or
- B is contained in **some** candidate key for the relation.
- Example: consider the table Rental in the slide series:
  Rental = (clientNumber, propertyNumber, rentStart, rentFinish)

  The designers want the following FD:s to hold:

  clientNumber, propertyNumber -> * (all the attributes)
  clientNumber, rentStart -> propertyNumber, rentFinish
  propertyNumber, rentStart -> clientNumber, rentFinish.

  As you can see, both the combinations *clientNumber, rentStart* and *propertyNumber, rentStart* are candidate keys for this relation (they can uniquely identify any row in it.)

  The key attributes for this relation are: clientNumber, propertyNumber, rentStart. They are included in **some** candidate key for the relation. The only non-key attribute is rentFinish. All of the FD: s are such that the determinant part is a candidate key for the relation:

  clientNumber, propertyNumber -> * (all the attributes) (PK)
  clientNumber, rentStart -> propertyNumber, rentFinish
  propertyNumber, rentStart -> clientNumber, rentFinish.

  So this table is in 3 NF. There are no partial dependencies between the key attributes either (where e g client-number alone would determine another key attribute) so the table satisfies even Boyce-Codd NF.

BCNF (Boyce-Codd normal form): the primary key (or another candidate key) alone determines everything. Nothing else may determine anything. No partial dependencies!

- **The following is in 3 NF, but NOT in BCNF:**
- StreetInfo = (streetname, zipCode, city, lenght)
- alternative candidate key: streetName, city -> zipCode, lenght
  (assumes that the whole street is inside the same zip code area)
- problematic FD: zipCode -> city. Both of these are key attributes.
  city is **partially dependent** on one candidate key. => **not BCNF**.