Normalization (cont)

Boyce-Codd Normal Form

- A relation is in BCNF if every determinant is a candidate key.
- Sometimes called 3 and 1/2 Normal Form as it is really and extension of 3NF.
- BCNF is stronger than 3NF
- Deals with the case where a nonkey attribute is the determinant of a key attribute.

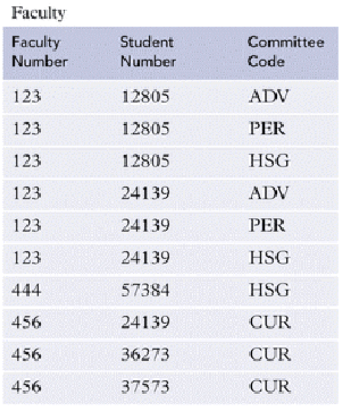
4NF & 5NF

- Once tables are in BCNF, all remaining normalization problems really come up only when we are dealing with "key only" tables
- Usually only apply to tables with 3 or more columns (hence a 3 or more part key)
- At the ERD level, look for 3 (or more)-way resolution entities.
  - Note: these may be perfectly legitimate. Problems arise when they are derivable from simpler, more fundamental relationships.

Fourth Normal Form

Interested in Faculty members, the students they advise, and the committees on which they serve.

Primary key = FacultyNumber, StudentNumber, CommitteeCode



Faculty

Faculty Number	Student Number	Committee Code
123	12805	ADV
123	12805	PER
123	12805	HSG
123	24139	ADV
123	24139	PER
123	24139	HSG
444	57384	HSG
456	24139	CUR
456	36273	CUR
456	37573	CUR

Modification anomalies

- Update -changing the code of a committee for faculty member 123 requires more than one change

- Additions -new faculty member who advises a student but does not serve on any committees

- Deletions -if faculty member stops advising a student (eg faculty member 444 advising student 57384) loose information that that faculty member is on a committee (eg HSG).

Multivalued Dependence

- In a table with columns A, B, and C, there is a multivalued dependence of column B on column A, if each value for A is associated with a specific collection of values for B and, furthermore, this collection is independent of any values for C.

- A table is in fourth normal form (4NF) if it is in 3NF and there are no multivalued dependencies.

Fifth Normal Form

- Should always be able to reconstruct the original table by joining (matching values in) the new table
- Normalization splits each table into underlying tables from which the original table can be derived if necessary
- 5NF essentially says to keep up splitting process until we can go no further. Only stop splitting when:
  - Any further splitting would lead to tables that could not be joined to produce the original table, or
  - The only splits left to us are trivial.

- 5NF defines an end-point after which any further “normalization” would cause us to lose information.

4NF & 5NF

- In simple terms, the definition of 4NF says that two, many-to-many relationships cannot be resolved with one table.
- Satisfying 5NF requires that two **or more** many-to-many relationships are not resolved by a single table.
- Check for 4NF & 5NF with the Business Specialist
- Ask questions such as:
  - On what (business) basis do we add a row to this table?
  - On what basis do we delete rows?
  - Do we apply any rules?
- Ultimately want to discover if any further splitting of the table makes sense.

Domain/Key Normal Form

- R. Fagin in 1981 defines DK/NF
- Involves only the concepts of key and domain.
- A relation is in DK/NF if every constraint on the relation is a logical consequence of the definition of keys and domains.
- Informally, a relation is in DK/NF if enforcing key and domain restrictions causes all of the constraints to be met.
- Relations in DK/NF cannot have modification anomalies so the DBMS can prohibit them by enforcing key and domain restrictions.
- No known algorithm for converting a relation to DK/NF.
- Finding or designing, DK/NF is more of an art than a science.

The Quick Version

- Does any attribute have more than one value?
  - No, in 1NF
- Does any attribute just need part of the key?
  - No, in 2NF
- Is any attribute dependent on another attribute and not the key?
  - No, in 3NF
- Is a key attribute dependent on a non-key attribute?
  - No, in BCNF
- Does any part of the key depend on another part of the key?
  - No, in 4NF

The Even Quicker Version

- Does each item of data depend upon the Key, the whole Key and nothing but the Key?

- If no, then are not in 3NF

Optimization

- Join together groups of data where the key is identical
- Be careful of introducing transitive dependencies during optimization
- Beware of similarly named data that have different meanings (homonyms)