# Relational Theory

## Functional Dependencies

- Constraint on the set of legal relations
  - Mapping from one set of attributes to another as many-to-one relation
- `X -> Y` means that `X` functionally determines `Y`

#### Example

- for element 'a' and 'b' is subset of relation schema `R` and `a -> b`
  - So, for tuple `t1` and `t2` in `R`
  - if `t1[a] = t2[a]` then `t1[b] = t2[b]`

#### Trivial FD

- `X -> Y` is trivial if `Y` is subset of `X`
- `X -> X` is trivial FD

#### Application of FD

- Definition of keys

  - `K` is a key of `R` if `K -> R`
  - `K` is a candidate key of `R` if `K -> R`
    - and no proper subset of `K -> R`

- Trivial FD

  - `X -> Y` is trivial if Y is subset of X
  - `X -> X` is trivial FD

#### Closure of FD

- Set of all FDs that can logically implied by a given set of FDs

- How to logically imply FDs? : Armstrong's Axioms

  - Reflexivity : `X -> Y` for any `Y` subset of `X`
  - Augmentation : `X -> Y` then `XZ -> YZ`
  - Transitivity : `X -> Y` and `Y -> Z` then `X -> Z`
  - Union : `X -> Y` and `X -> Z` then `X -> YZ`
  - Decomposition : `X -> YZ` then `X -> Y` and `X -> Z`
  - Pseudotransitivity : `X -> Y` and `WY -> Z` then `WX -> Z`

- Example

  - `R = (A, B, C, G, H, I)` and `F = {A -> B, A -> C, CG -> H, CG -> I, B -> H}`
  - `A+` = `{A, B, C, H}`

# Normal Forms

#### First Normal Form (1NF)

- single & single valued attributes
- domains of all attributes must be atomic

#### Second Normal Form (2NF)

- 1NF + no partial dependencies
- partial dependencies : non-prime attribute depends on proper subset of candidate key
  - e.g. `R = (A, B, C, D)` and `F = {AB -> C, B -> D}`
  - violates 2NF because `B -> D` and `B` is subset of candidate key `AB`

#### Third Normal Form (3NF)

- 2NF + no transitive dependencies
  - e.g. `R = (A, B, C)` and `F = {A -> B, B -> C}`
  - decompose `R` into `R1 = (A, B)` and `R2 = (B, C)`
    - because `A -> C` is non-direct dependency(transitive dependency)

#### Boyce-Codd Normal Form (BCNF)

- 3NF + determinant must subset of candidate key

  - e.g. `R = (A, B, C)` and `F = {AB -> C, C -> B}`
  - violates BCNF because `C -> B` and `C` is not subset of candidate key `AB`

- In other words, all relations in BCNF follow one of 2 conditions

  - FD is trivial
  - every non-trivial determinant is a superkey
  - e.g. `R = (A, B, C)` and `F = {A -> B, B -> C}`
    - violates BCNF because `B -> C` and `B` is not superkey

# Decomposition

#### Lossy Decomposition

- `R1` and `R2` are lossy decomposition of `R` if `R1 ∩ R2` is not subset of `R1` or `R2`
  - e.g. `R = (A, B, C)` and `F = {A -> B, B -> C}`
  - `R1 = (A, B)` and `R2 = (B, C)` is lossy decomposition of `R`
    - because `R1 ∩ R2 = (B)` and `B` is not subset of `R1` or `R2`

#### Lossless Decomposition

- For relation `R` and it's decomposition `(R1, R2)`
  - `R` must equal to natural join of `R1` and `R2`
  - $R = \Pi_{R1}(R) \bowtie \Pi_{R2}(R)$

- $R1 \cap R2 \rightarrow R1$ or $R1 \cap R2 \rightarrow R2$

**Dependency Preservation**

- For relation `R` and it's decomposition `(R1, ..., Ri ..., Rn)`
  - $F_i$ is restriction of `F` to `Ri`
  - then $F^{'} = F_1 \cup .. \cup F_n$
  - the decomposition is dependency preserving if $F^{'}+ = F+$