

Boyce Codd Normal Form



First Normal Form

- All columns have a unique name
- All columns have a well defined domain
 - All data in a column must be of the same type
- All columns contain one distinct value per row
 - Single Valued Attribute
- The order in which you store data in a table does not matter

Second Normal Form

- Your table is currently in First Normal Form
 - Note each table, on its own, needs to be in first normal form for the database schema to be in first normal form
- There are no partial dependencies
 - If there is **a column that depends on only some of the columns contained in the primary key**, then this is a partial dependency
 - Candidate keys are a great way to identify partial dependencies

Third Normal Form

- Your table is currently in Second Normal Form
 - Note that every table, on its own, needs to be in second normal form for the database schema to be in second normal form
- There are no transitive dependencies in the table
 - When a column in a table depends on **something other than the primary key of the table for uniqueness and context**, we call this a transitive dependency
- It is always possible to preserve dependencies and ensure lossless decomposition with 3NF
 - Meaning, we can still write a query and get the desired result sets, based on our known dependencies

Boyce Codd Normal Form

- Your table is currently in Third Normal Form
 - Note that every table needs to be in third normal form for the database schema to be in third normal form
- For every dependency $A \rightarrow B$, A should be a candidate key
 - We know that whenever a dependency exists, it needs to depend on the primary key. Second and third normal forms ensure this. But, we want a stronger assertion.
 - If **A is not a candidate key, then it should not depend on the primary key for uniqueness and context.**
 - Note we allow for trivial dependencies, where B is equal to or contained in A
- It is not always possible to ensure lossless decomposition with BCNF