

**First Normal Form (1NF)**

A relation schema is in 1NF if domains of attributes include only atomic (simple, indivisible) values and the value of an attribute is a single value from the domain of that attribute

1NF disallows

* having a set of values, a tuple of values, or a combination of both as an attribute value for a single tuple
* “relations within relations” and “relations as attributes of tuples

**Second Normal Form (2NF)**

A relation is in 2NF, if for every FD X→Y where X is a minimal key and Y is a non-prime attribute, then no proper subset of X determines Y

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

A relation R is in BCNF if for all non-trivial dependencies in R: If X 🡪 b then X is a superkey for R

**Trivial**: If an FD X → Y holds where Y subset of X, then it is called a trivial FD. Trivial FDs are always hold.

**Non-trivial**: If an FD X → Y holds where Y is not subset of X, then it is called non-trivial FD.

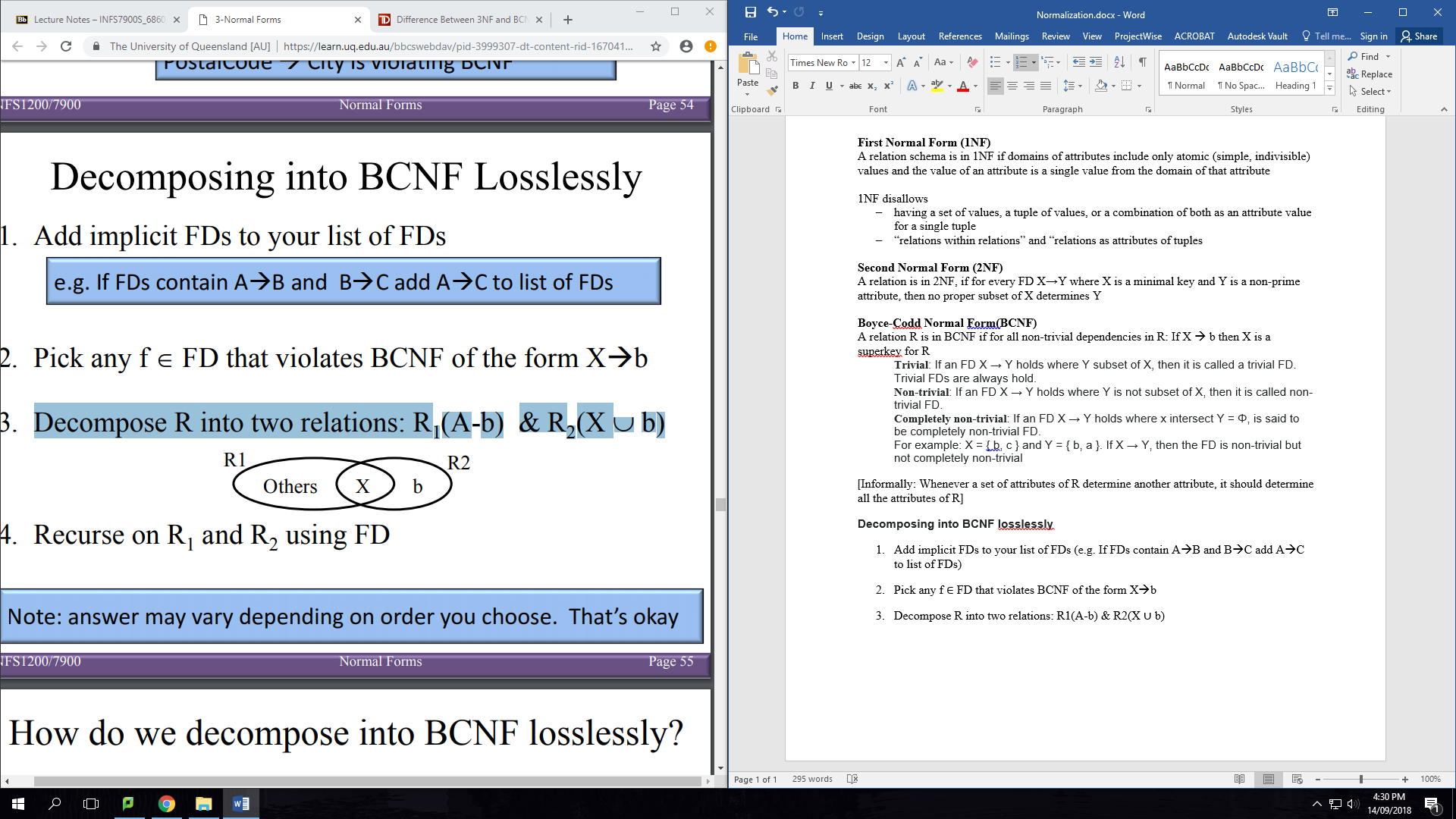
**Completely non-trivial**: If an FD X → Y holds where x intersect Y = Φ, is said to be completely non-trivial FD.

For example: X = { b, c } and Y = { b, a }. If X → Y, then the FD is non-trivial but not completely non-trivial

[Informally: Whenever a set of attributes of R determine another attribute, it should determine all the attributes of R]

**Decomposing into BCNF losslessly**

1. Add implicit FDs to your list of FDs (e.g. If FDs contain A🡪B and B🡪C add A🡪C to list of FDs)
2. Pick any f FD that violates BCNF of the form X🡪b
3. Decompose R into two relations: R1(A-b) & R2(X b)



1. Recurse on R1 and R2 using FD

**Third Normal Form (3NF)**

A relation R is in 3NF if for all non-trivial dependencies in R:

* If X 🡪 b then X is a superkey for R
* OR b is a prime attribute of R