Tutorial 5 : Normalisation

Functional Dependency

Definition - What does Functional Dependency mean?
Functional dependency is a relationship that exists when one attribute uniquely determines another attribute.

If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X. Here X is a determinant set and Y is a dependent attribute. Each value of X is associated with precisely one Y value.

Functional dependency in a database serves as a constraint between two sets of attributes. Defining functional dependency is an important part of relational database design and contributes to aspect normalization.

Why normalise ?

1. Increased consistency. Information is stored in one place and one place only, reducing the possibility of inconsistent data.
2. Easier object-to-data mapping. Highly-normalized data schemas in general are closer conceptually to object-oriented schemas because the object-oriented goals of promoting high cohesion and loose coupling between classes results in similar solutions (at least from a data point of view).

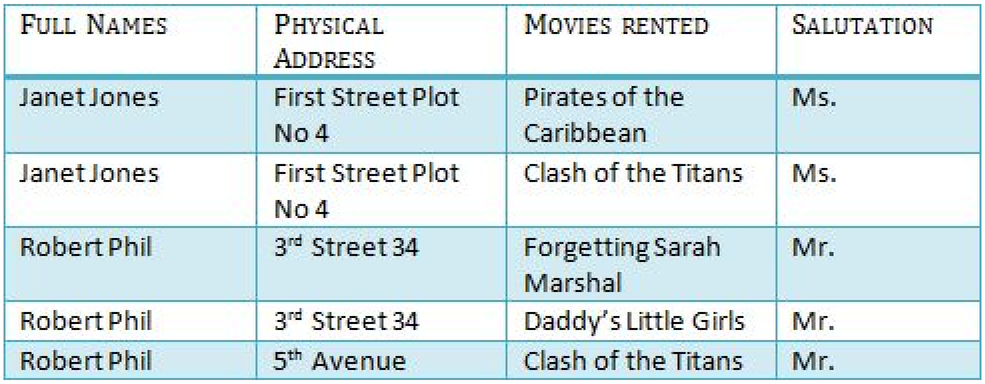
Hierarchy

Generally the levels of normalisation are 1NF, 2NF, 3NF, BCNF (**Boyce-Codd Normal Form).**

Normal forms are **inclusive** in nature.

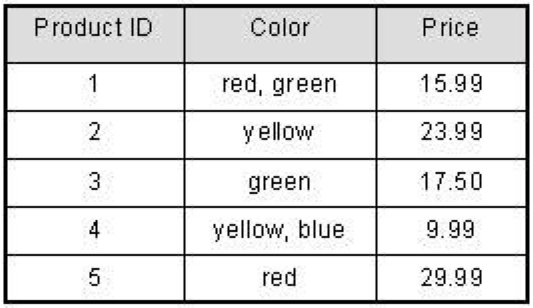
1 NF

An entity type is in first normal form (1NF) when it contains no repeating groups of data.



Full Names	Physical Address	Movies rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

TABLE_PRODUCT



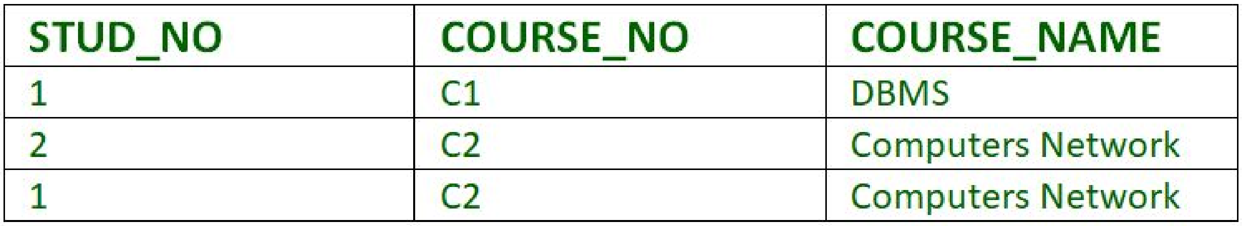
Product ID	Color	Price
1	red, green	15.99
2	yellow	23.99
3	green	17.50
4	yellow, blue	9.99
5	red	29.99

The above table isn't in 1NF. (Here each of the color is considered as a value)

2 NF

A relation is in 2NF iff it has **No Partial Dependency,** i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

Partial Dependency – If proper subset of candidate key determines non-prime attribute, it is called partial dependency.



STUD_NO	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	Computers Network
1	C2	Computers Network

Table 3

3 NF

- A relation is in third normal form, if there is **no transitive dependency** for non-prime attributes is it is in second normal form.A relation is in 3NF iff **at least one of the following condition holds** in every non-trivial function dependency X –> Y
    1. X is a super key.
    2. Y is a prime attribute (each element of Y is part of some candidate key).

The decomposition is both **lossless-join** and **dependency-preserving.**

Boyce-Codd Normal Form (BCNF)

A relation R is in BCNF if R is in Third Normal Form and for every FD, LHS is super key. A relation is in BCNF iff in every non-trivial functional dependency X –> Y, X is a super key.

The decomposition is **lossless-join** but **may not be dependency-preserving.**

Notes:

1. BCNF is free from redundancy.
2. If a relation is in BCNF, then 3NF is also also satisfied.

3. If all attributes of relation are prime attribute, then the relation is always in 3NF.
4. A relation in a Relational Database is always and at least in 1NF form.
5. Every Binary Relation (a Relation with only 2 attributes) is always in BCNF.
6. If a Relation has only singleton candidate keys(i.e. every candidate key consists of only 1 attribute), then the Relation is always in 2NF(because no Partial functional dependency possible).
7. Sometimes going for BCNF form may not preserve functional dependency. In that case go for BCNF only if the lost FD(s) is not required, else normalize till 3NF only.
8. There are many more Normal forms that exist after BCNF, like 4NF and more. But in real world database systems it's generally not required to go beyond BCNF.

References:

https://www.geeksforgeeks.org/database-normalization-normal-forms/

http://agiledata.org/essays/dataNormalization.html

https://tutorialink.com/dbms/dependency-preserving-decomposition.dbms

https://www.youtube.com/playlist?list=PLeNFpOhruv2iM5EFv04SH4d84AO9WD2bA

http://fac.ksu.edu.sa/sites/default/files/E-%20Decomposition.pdf