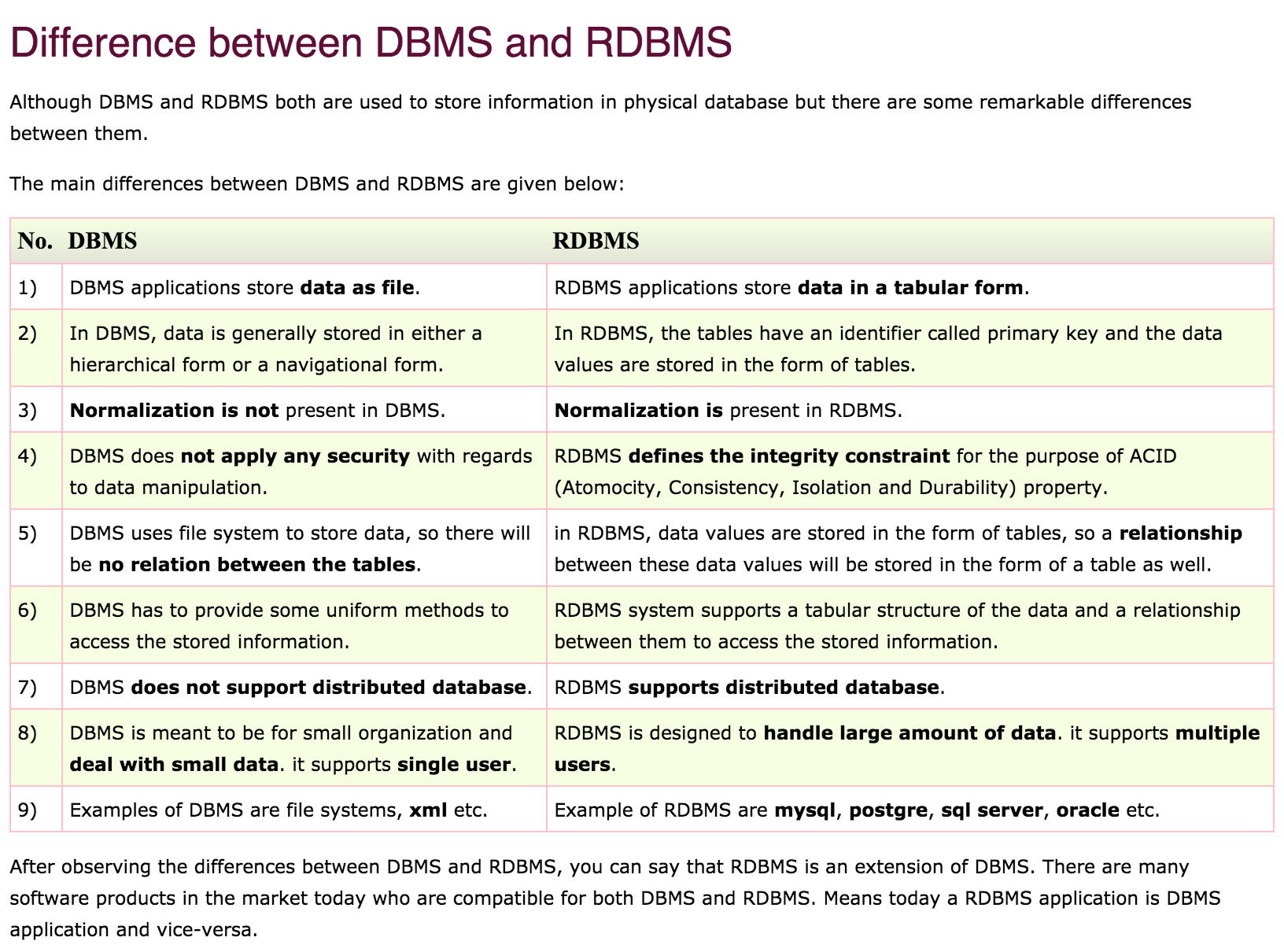
**DBMS vs RDBMS**

The key difference is that **RDBMS** (relational database management system) applications store data in a tabular form, while **DBMS** applications store data as files. In a **RDBMS**, the tables will have an identifier called primary key. Data values will be stored in the form of tables



**Normalization**

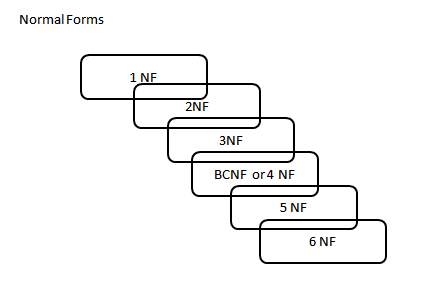
Normalization or database normalization is a process to organize the data into database tables. To make a good database design, you have to follow Normalization practices.

* Normalization reduces data redundancy and inconsistent data dependency
* Without normalization, a database system might be slow, inefficient and might not produce the expected result.

The process of taking a database design, and apply a set of formal criteria and rules, is called Normal Forms.

The database normalization process is further categorized into the following types:

1. First Normal Form (1 NF)
2. Second Normal Form (2 NF)
3. Third Normal Form (3 NF)
4. Boyce Codd Normal Form or Fourth Normal Form ( BCNF or 4 NF)
5. Fifth Normal Form (5 NF)
6. Sixth Normal Form (6 NF)

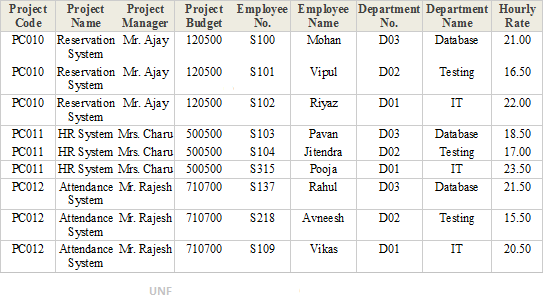


One of the driving forces behind database normalization is to streamline data by reducing redundant data. Redundancy of data means there are multiple copies of the same information spread over multiple locations in the same database.

The drawbacks of data redundancy include:

1. Data maintenance becomes tedious – data deletion and data updates become problematic
2. It creates data inconsistencies
3. Insert, Update and Delete anomalies become frequent. An update anomaly, for example, means that the versions of the same record, duplicated in different places in the database, will all need to be updated to keep the record consistent
4. Redundant data inflates the size of a database and takes up an inordinate amount of space on disk

To understand normal forms consider the following unnormalized database table. Now we will normalize the data of below table using normal forms.

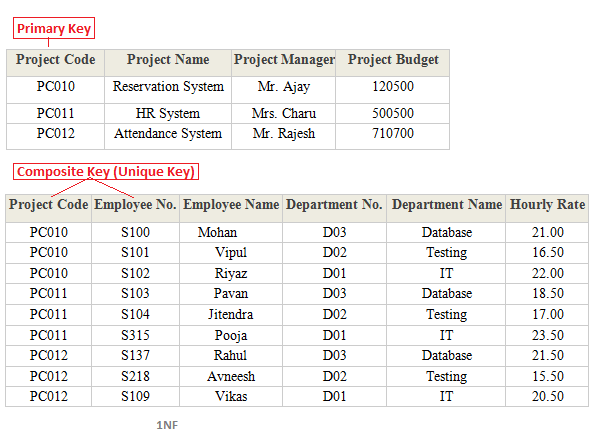


1. First Normal Form (1NF)

A database table is said to be in 1NF if it contains no repeating fields/columns. The process of converting the UNF table into 1NF is as follows:

* 1. Separate the repeating fields into new database tables along with the key from the unnormalized database table.
  2. The primary key of new database tables may be a composite key

1NF of above UNF table is as follows:



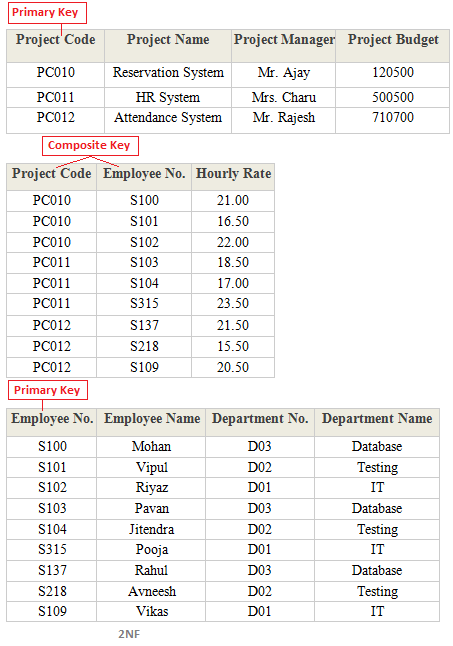
1. Second Normal Form (2NF)

A database table is said to be in 2NF if it is in 1NF and contains only those fields/columns that are functionally dependent(means the value of the field is determined by the value of another field(s)) on the primary key. In 2NF we remove the partial dependencies of any non-key field.

The process of converting the database table into 2NF is as follows:

* 1. Remove the partial dependencies(A type of functional dependency where a field is only functionally dependent on the part of primary key) of any non-key field.
  2. If field B depends on field A and vice versa. Also for a given value of B, we have only one possible value of A and vice versa, Then we put the field B into new database table where B will be the primary key and also marked as a foreign key in a parent table.

2NF of above 1NF tables is as follows:

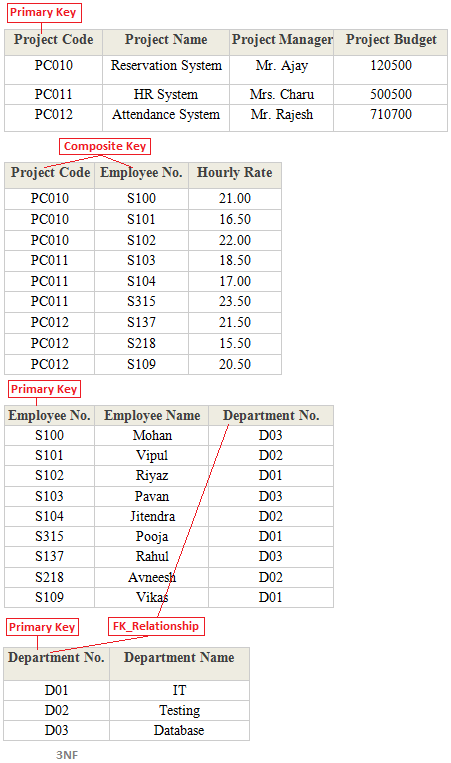


1. Third Normal Form (3NF)

A database table is said to be in 3NF if it is in 2NF and all non-keys fields should be dependent on primary key or We can also say a table to be in 3NF if it is in 2NF and no fields of the table are transitively functionally dependent on the primary key. The process of converting the table into 3NF is as follows:

* 1. Remove the transitive dependencies(A type of functional dependency where a field is functionally dependent on the Field that is not the primary key. Hence its value is determined, indirectly by the primary key )
  2. Make a separate table for transitive dependent Field.

3NF of above 2NF tables is as follows:

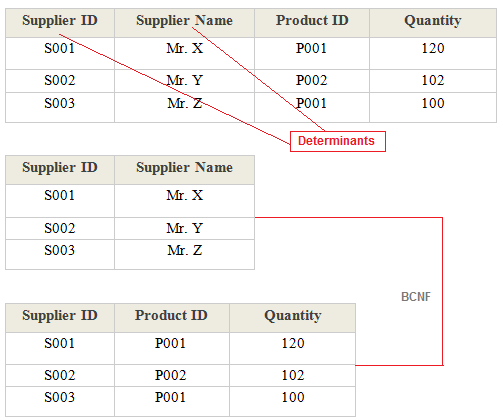


1. Boyce Code Normal Form (BCNF)

A database table is said to be in BCNF if it is in 3NF and contains each and every determinant as a candidate key. The process of converting the table into BCNF is as follows:

* 1. Remove the nontrivial functional dependency.
  2. Make a separate table for the determinants.

BCNF of below table is as follows:

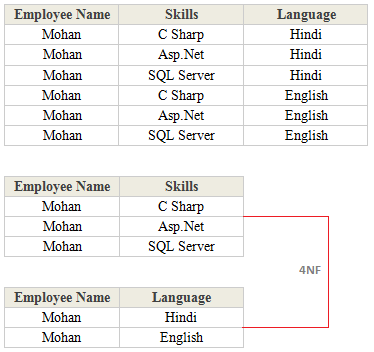


1. Fourth Normal Form (4NF)

A database table is said to be in 4NF if it is in BCNF and primary key has a one-to-one relationship to all non-keys fields or We can also say a table to be in 4NF if it is in BCNF and contains no multi-valued dependencies. The process of converting the table into 4NF is as follows:

* 1. Remove the multivalued dependency.
  2. Make a separate table for multivalued Fields.

4NF of below table is as follows:



1. Fifth Normal Form (5NF)

A database table is said to be in 5NF if it is in 4NF and contains no redundant values or We can also say a table to be in 5NF if it is in 4NF and contains no join dependencies. The process of converting the table into 5NF is as follows:

* 1. Remove the join dependency.
  2. Break the database table into smaller and smaller tables to remove all data redundancy.

5NF of below table is as follows:

