**CTS Preparation**

* The “bubble” sort is called so **because the list elements with greater value than their surrounding elements “bubble” towards the end of the list**
* **Assembler –** Assembler is a program which converts assembly language into machine code
* **Compiler –** Compiler is a program that converts a number of statement of program into binary language, but it is more intelligent than interpreter because it goes through the entire code at once and can tell the possible errors and limits and ranges
* **Interpretor –** An interpreter is also a program like a compiler that converts assembly language into binary but an interpreter goes through one line of code at a time
* **Linker –** Linker is a program that holds one or more object files which is created by compiler, combines them into one executable file
* **Loader –** Loader is a program that loads machine codes of a program into the system memory.It is part of the OS of the computer that is responsible for loading the program. It is the bare beginning of the execution of a program.
* There are 4 major principles of oop’s these are **Encapsulation, Data Abstraction, Polymorphism and Inheritance**
* Encapsulation is defined as wrapping up of data and information under a single unit. Encapsulation is defined as binding together the data and the functions that manipulate them.

Consider a real-life example of encapsulation, in a company, there are different sections like the accounts section, finance section, sales section, etc. The finance section handles all the financial transactions and keeps records of all the data related to finance. Similarly, the sales section handles all the sales-related activities and keeps records of all the sales. Now there may arise a situation when for some reason an official from the finance section needs all the data about sales in a particular month. In this case, he is not allowed to directly access the data of the sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data. This is what encapsulation is. Here the data of the sales section and the employees that can manipulate them are wrapped under a single name “sales section”.

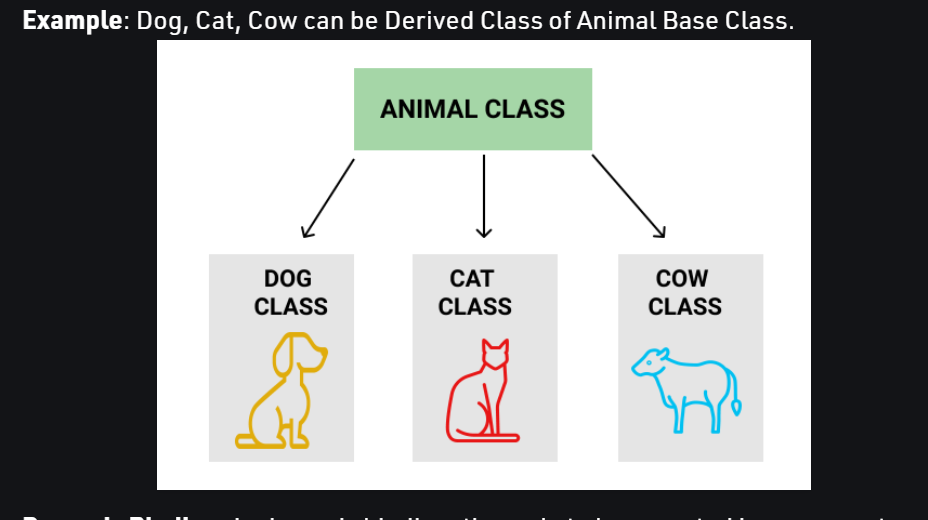
* Abstraction is one of the most essential and important features of object-oriented programming . Abstraction means displaying only essential information and hiding the Internal details.

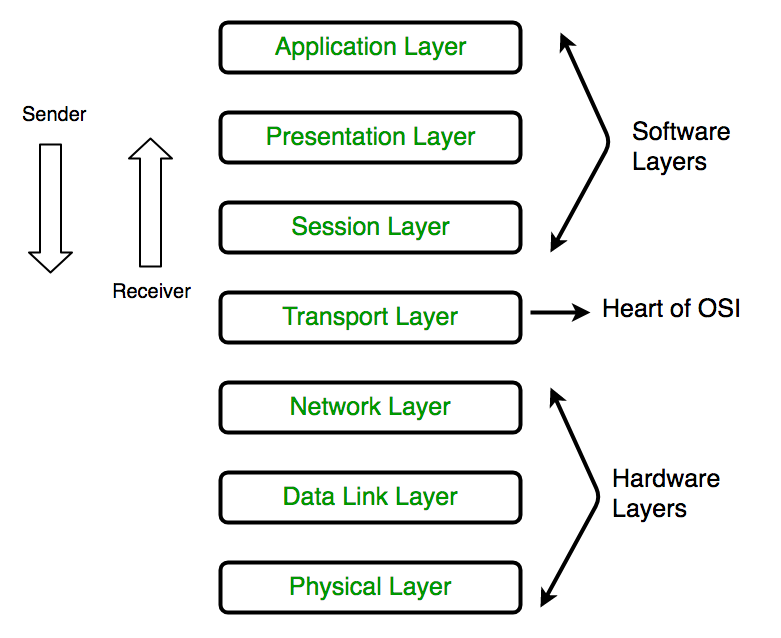
Consider a real-life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of the car or applying brakes will stop the car, but he does not know about how on pressing the accelerator the speed is increasing, he does not know about the inner mechanism of the car or the implementation of the accelerator, brakes, etc in the car. This is what abstraction is

* Polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.

For example, A person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee. So the same person posses different behavior in different situations. This is called polymorphism.

* Inheritance when one object acquires all the properties and behaviour of parent object it is known as Inheritance





* The HAVING clause is used instead of WHERE with aggregate functions. While the GROUP BY Clause groups rows that have the same values.

**Normalization In DBMS:-**

**Normalization** is a process of organizing the data in database to avoid data redundancy,

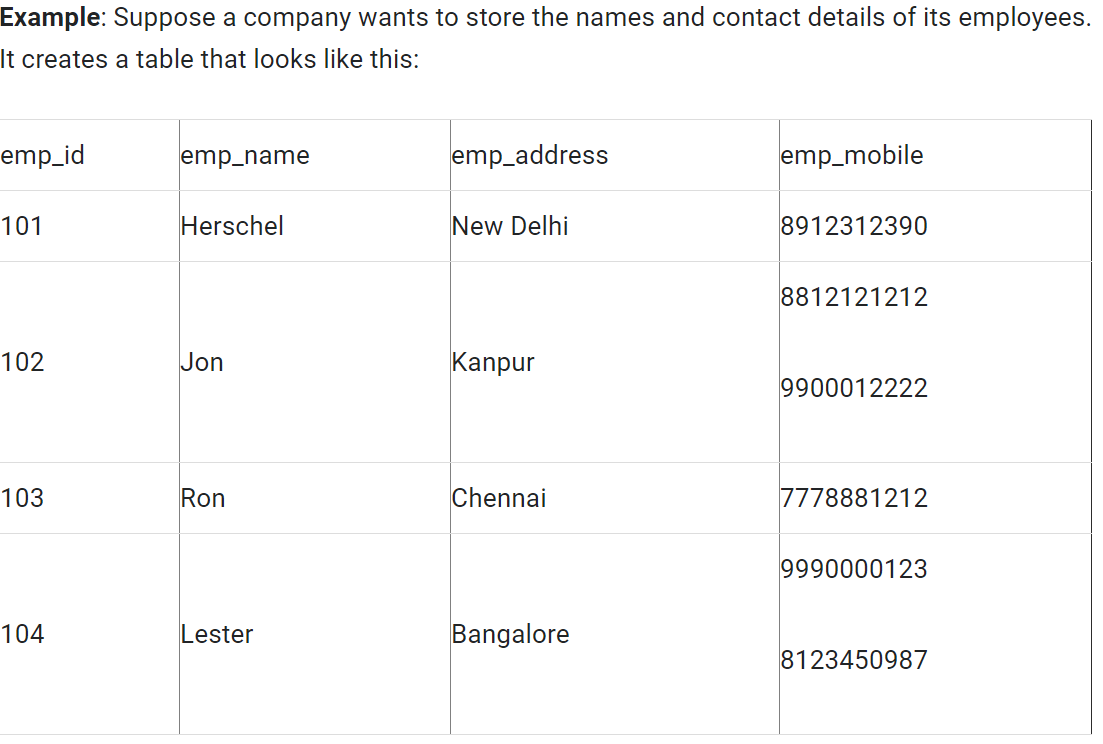
**Normalization**

Here are the most commonly used normal forms:

* First normal form(1NF)
* Second normal form(2NF)
* Third normal form(3NF)
* Boyce & Codd normal form (BCNF)

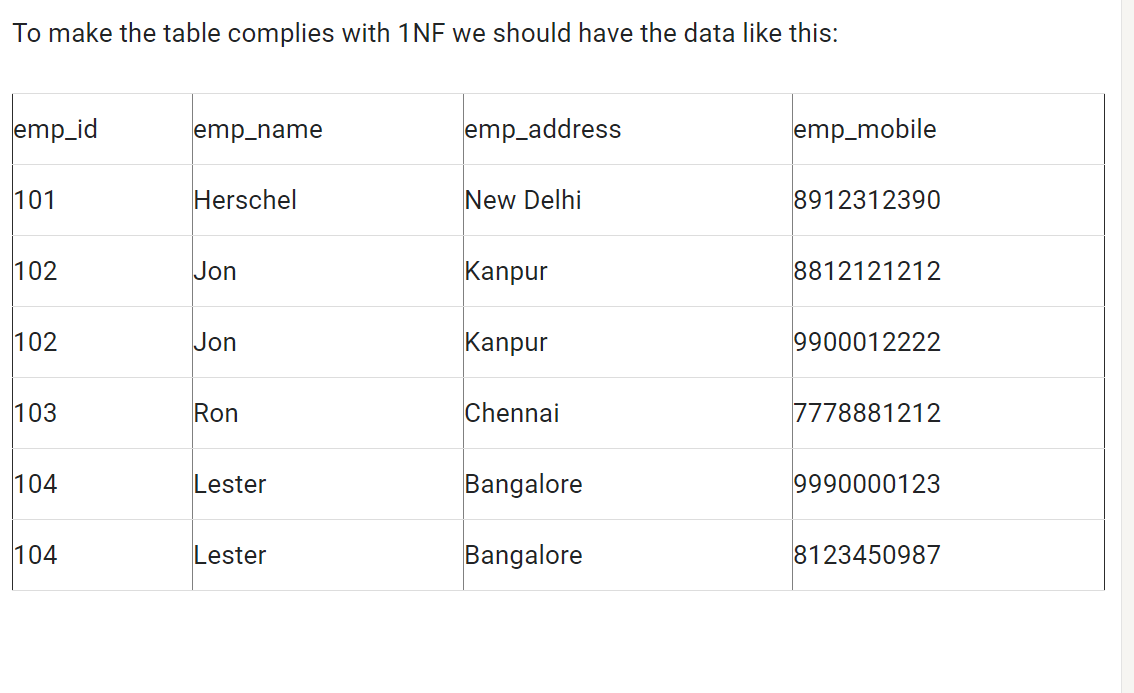
**First normal form (1NF)**

A Table is said to be in 1NF if column of a table cannot hold multiple values. It should hold only atomic values.



Two employees (Jon & Lester) are having two mobile numbers so the company stored them in the same field as you can see in the table above.

This table is **not in 1NF**as the rule says “each attribute of a table must have atomic (single) values”, the emp\_mobile values for employees Jon & Lester violates that rule.



## Second normal form (2NF)

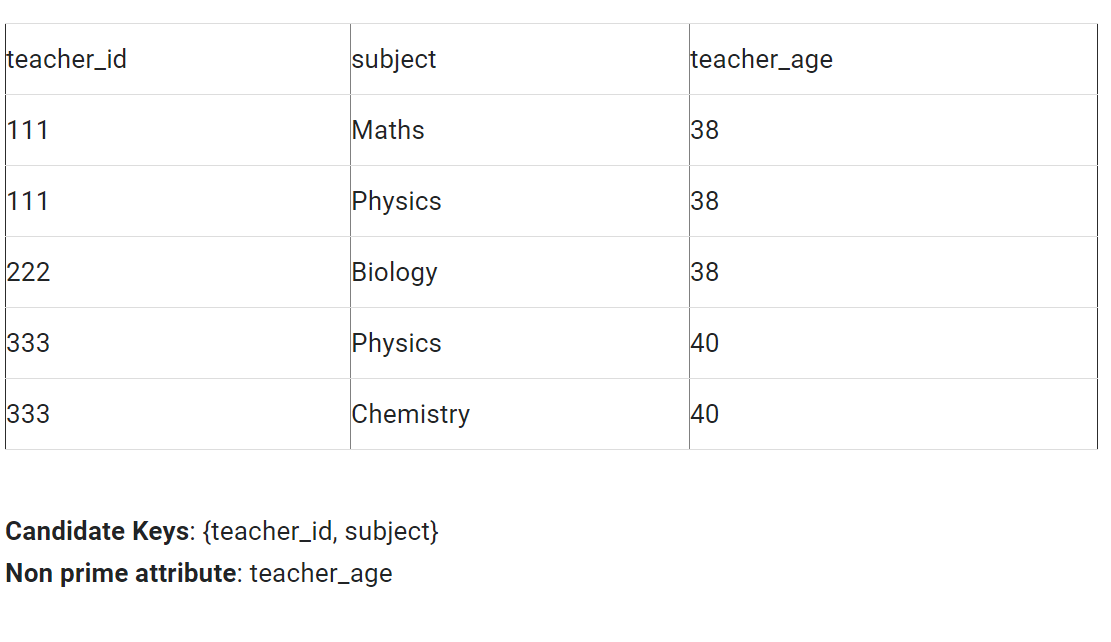
A table is said to be in 2NF if both the following conditions hold:

* Table is in 1NF (First normal form)
* Table shouldn’t contain partial dependency.

Partial Dependency **occurs when a non-prime attribute is functionally dependent on part of a candidate key**

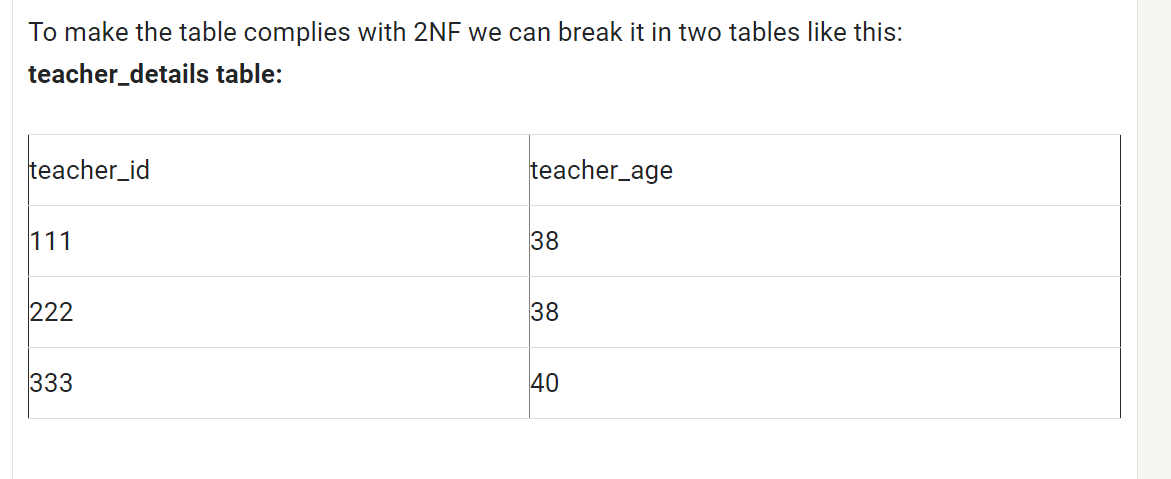
An attribute that is not a part of one of the candidate keys is known as non prime attribute.

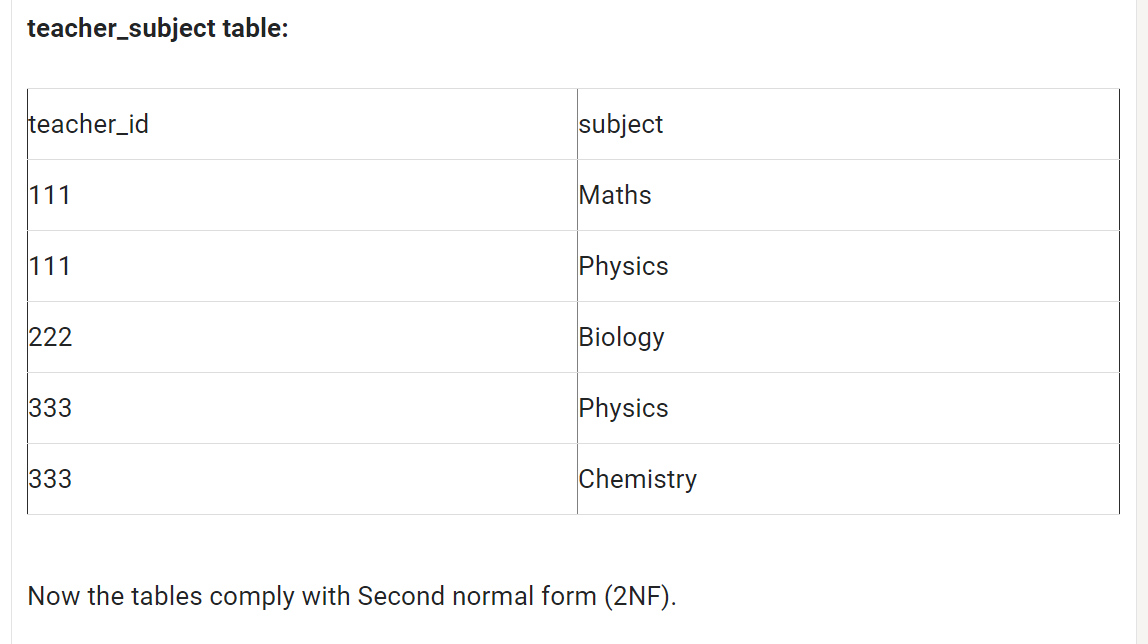
**Example**: Suppose a school wants to store the data of teachers and the subjects they teach. They create a table that looks like this: Since a teacher can teach more than one subjects, the table can have multiple rows for a same teacher.

Candidate Key: **The minimal set of attributes that can uniquely identify a records** is known as a candidate key.

Primary Key: It’s the single column value used to identify records uniquely.

The table is in 1 NF because each attribute has atomic values. However, it is not in 2NF because non prime attribute teacher\_age is dependent on teacher\_id alone which is a proper subset of candidate key. This violates the rule for 2NF as the rule says “**no** non-prime attribute is dependent on the proper subset of any candidate key of the table”.





## Third Normal form (3NF)

A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:

* X is a [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/) of table
* Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as prime attribute.

 A super key is a set of one or more columns, which can uniquely identify a row in a table



**Super keys**: {emp\_id}, {emp\_id, emp\_name}, {emp\_id, emp\_name, emp\_zip}…so on  
**Candidate Keys**: {emp\_id}  
**Non-prime attributes**: all attributes except emp\_id are non-prime as they are not part of any candidate keys.

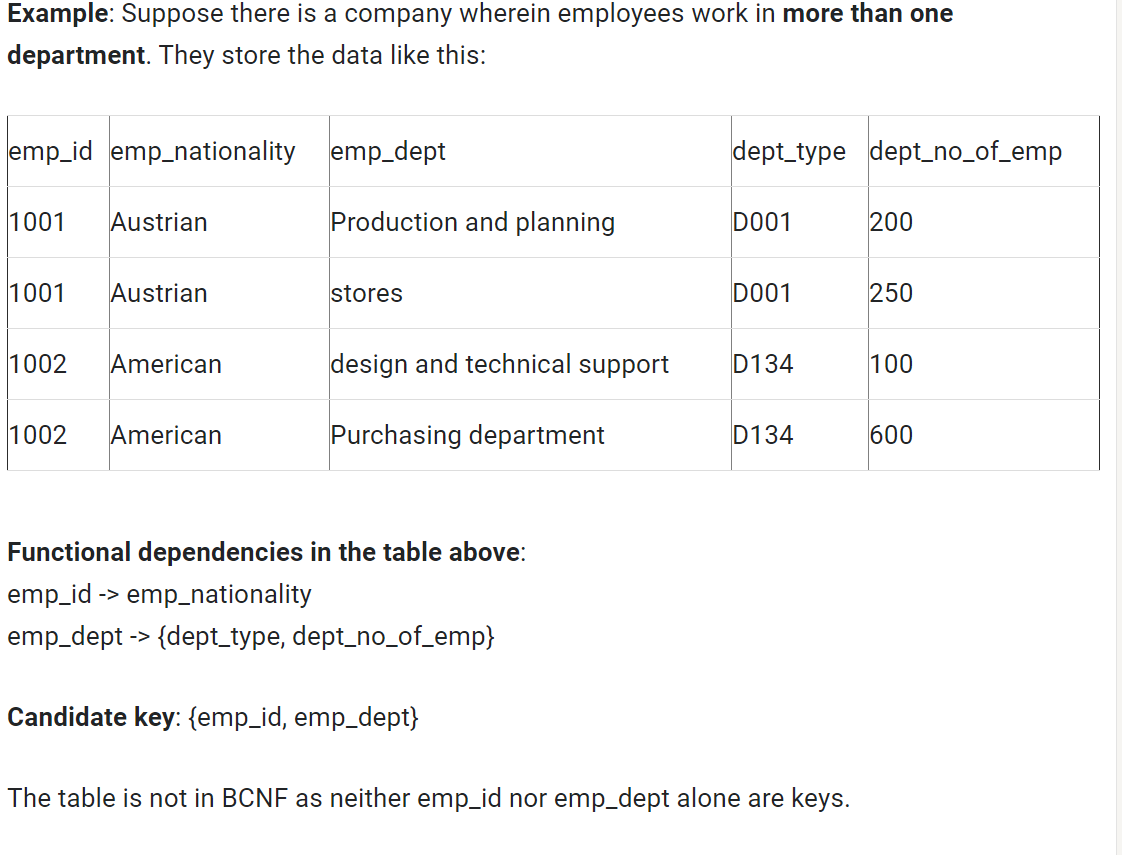
Here, emp\_state, emp\_city & emp\_district dependent on emp\_zip. And, emp\_zip is dependent on emp\_id that makes non-prime attributes This violates the rule of 3NF.

To make this table complies with 3NF we have to break the table into two tables to remove the transitive dependency:



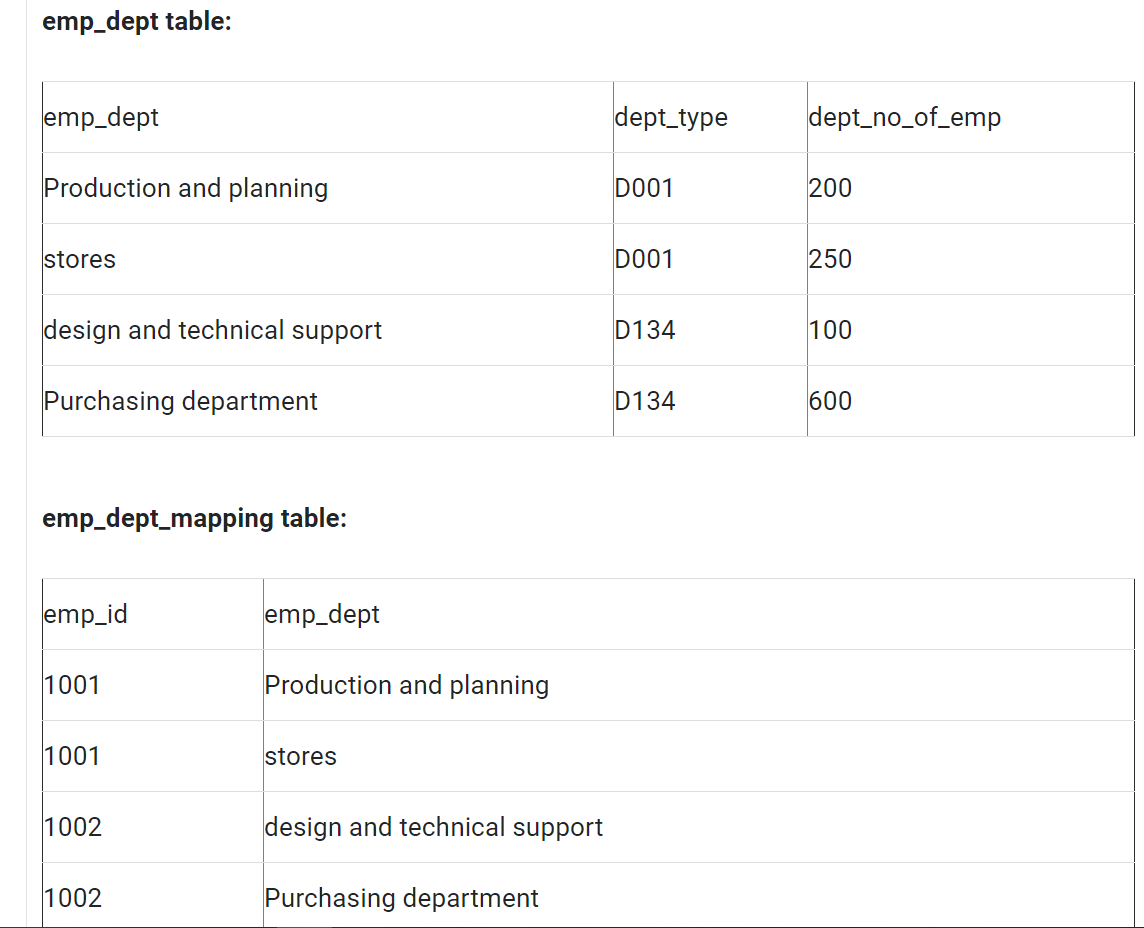
## Boyce Codd normal form (BCNF)

* It is an advance version of 3NF that’s why it is also referred as 3.5NF. BCNF is stricter than 3NF.
* A table complies with BCNF if it is in 3NF and for every [functional dependency](https://beginnersbook.com/2015/04/functional-dependency-in-dbms/) X->Y, X should be the super key of the table.



The table is not in BCNF as neither emp\_id nor emp\_dept alone are keys.





**Functional dependencies**:  
emp\_id -> emp\_nationality  
emp\_dept -> {dept\_type, dept\_no\_of\_emp}

**Candidate keys**:  
For first table: emp\_id  
For second table: emp\_dept  
For third table: {emp\_id, emp\_dept}

This is now in BCNF as in both the functional dependencies left side part is a key.

