

What is Second Normal Form?

If you want you can skip the video, as the concept is covered in detail below the video.

For a table to be in the Second Normal Form, it must satisfy two conditions:

1. The table should be in the First Normal Form.
2. There should be no Partial Dependency.

What is **Partial Dependency**? Do not worry about it. First let's understand what is **Dependency** in a table?

What is Dependency?

Let's take an example of a **Student** table with columns `student_id`, `name`, `reg_no`(registration number), `branch` and `address`(student's home address).

<code>student_id</code>	<code>name</code>	<code>reg_no</code>	<code>branch</code>	<code>address</code>

In this table, `student_id` is the primary key and will be unique for every row, hence we can use `student_id` to fetch any row of data from this table

Even for a case, where student names are same, if we know the `student_id` we can easily fetch the correct record.

<code>student_id</code>	<code>name</code>	<code>reg_no</code>	<code>branch</code>	<code>address</code>
10	Akon	o7-WY	CSE	Kerala
11	Akon	o8-WY	IT	Gujarat

Hence we can say a **Primary Key** for a table is the column or a group of columns(composite key) which can uniquely identify each record in the table.

I can ask from branch name of student with `student_id` **10**, and I can get it. Similarly, if I ask for name of student with `student_id` **10** or **11**, I will get it. So all I need is `student_id` and every other column **depends** on it, or can be fetched using it.

This is **Dependency** and we also call it **Functional Dependency**.

What is Partial Dependency?

Now that we know what dependency is, we are in a better state to understand what partial dependency is.

For a simple table like Student, a single column like `student_id` can uniquely identify all the records in a table.

But this is not true all the time. So now let's extend our example to see if more than 1 column together can act as a primary key.

Let's create another table for **Subject**, which will have `subject_id` and `subject_name` fields and `subject_id` will be the primary key.

subject_id	subject_name
1	Java
2	C++
3	Php

Now we have a **Student** table with student information and another table **Subject** for storing subject information.

Let's create another table **Score**, to store the **marks** obtained by students in the respective subjects. We will also be saving **name of the teacher** who teaches that subject along with marks.

score_id	student_id	subject_id	marks	teacher
1	10	1	70	Java Teacher
2	10	2	75	C++ Teacher
3	11	1	80	Java Teacher

In the score table we are saving the **student_id** to know which student's marks are these and **subject_id** to know for which subject the marks are for.

Together, `student_id` + `subject_id` forms a **Candidate Key**(learn about [Database Keys](#)) for this table, which can be the **Primary key**.

Confused, How this combination can be a primary key?

See, if I ask you to get me marks of student with `student_id` 10, can you get it from this table? No, because you don't know for which subject. And if I give you `subject_id`, you would not know for which student. Hence we need

`student_id` + `subject_id` to uniquely identify any row.

But where is Partial Dependency?

Now if you look at the **Score** table, we have a column names `teacher` which is only dependent on the subject, for Java it's Java Teacher and for C++ it's C++ Teacher & so on.

Now as we just discussed that the primary key for this table is a composition of two columns which is `student_id` & `subject_id` but the teacher's name only depends on subject, hence the `subject_id`, and has nothing to do with `student_id`.

This is **Partial Dependency**, where an attribute in a table depends on only a part of the primary key and not on the whole key.

How to remove Partial Dependency?

There can be many different solutions for this, but our objective is to remove teacher's name from Score table.

The simplest solution is to remove columns `teacher` from Score table and add it to the Subject table. Hence, the Subject table will become:

<code>subject_id</code>	<code>subject_name</code>	<code>teacher</code>
1	Java	Java Teacher
2	C++	C++ Teacher
3	Php	Php Teacher

And our Score table is now in the second normal form, with no partial dependency.

<code>score_id</code>	<code>student_id</code>	<code>subject_id</code>	<code>marks</code>
1	10	1	70

2	10	2	75
3	11	1	80

Quick Recap

1. For a table to be in the Second Normal form, it should be in the First Normal form and it should not have Partial Dependency.
2. Partial Dependency exists, when for a composite primary key, any attribute in the table depends only on a part of the primary key and not on the complete primary key.
3. To remove Partial dependency, we can divide the table, remove the attribute which is causing partial dependency, and move it to some other table where it fits in well.