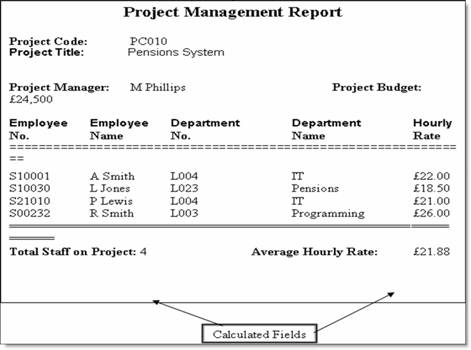
**Normalisation Example**

We will demonstrate the process of normalisation (to 3NF) by use of an example. Normalisation is a bottom-up technique for database design, normally based on an existing system (which may be paper-based). We start by analysing the documentation, eg reports, screen layouts from that system. We will begin with the Project Management Report, which describes projects being worked upon by employees. This report is to be 'normalised'. Each of the first four normalisation steps is explained.



**Next:** [Step 1](http://www.sqa.org.uk/e-learning/MDBS01CD/page_27.htm#Step1)

**Step 1**

Select the data source (ie the report from the previous page) and convert into an unnormalised table (UNF). The process is as follows:

* Create column headings for the table for each data item on the report (**ignoring any calculated fields**). A calculated field is one that can be derived from other information on the form. In this case **total staff** and **average hourly rate**.
* Enter sample data into table. (This data is not simply the data on the report but a representative sample. In this example it shows several employees working on several projects. In this company the same employee can work on different projects and at a different hourly rate.)
* Identify a **key** for the table (and underline it).
* Remove duplicate data. (In this example, for the chosen key of Project Code, the values for Project Code, Project Title, Project Manager and Project Budget are duplicated if there are two or more employees working on the same project. **Project Code** chosen for the key and duplicate data, associated with each project code, is removed. Do not confuse duplicate data with repeating attributes which is descibed in the next step.

****

**UNF: unnormalised table**

**Next:** [Step 2](http://www.sqa.org.uk/e-learning/MDBS01CD/page_28.htm#Step2)

**Step 2**

Transform a table of unnormalised data into first normal form (1NF). any repeating attributes to a new table. A repeating attribute is a data field within the UNF relation that may occur with multiple values for a single value of the key. The process is as follows:

* Identify repeating attributes.
* Remove these repeating attributes to a new table together with a **copy** of the key from the UNF table.
* Assign a key to the new table (and underline it). The key from the original unnormalised table **always** becomes **part** of the key of the new table. A **compound key** is created. The value for this key must be unique for each entity occurrence.

**Notes:**

* After removing the duplicate data the repeating attributes are easily identified.
* In the previous table the Employee No, Employee Name, Department No, Department Name and Hourly Rate attributes are repeating. That is, there is potential for more than one occurrence of these attributes for each project code. These are the repeating attributes and have been to a new table together with a copy of the original key (ie: Project Code).
* A key of Project Code and Employee No has been defined for this new table. This combination is unique for each row in the table.

**Next:** [1NF Tables](http://www.sqa.org.uk/e-learning/MDBS01CD/page_29.htm#NF1Tables)

**1NF Tables: Repeating Attributes Removed**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Code** | **Project Title** | **Project Manager** | **Project Budget** |
| PC010 | Pensions System | M Phillips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC064 | HR System | K Lewis | 12250 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Code** | **Employee No.** | **Employee Name** | **Department No.** | **Department Name** | **Hourly Rate** |
| PC010 | S10001 | A Smith | L004 | IT | 22.00 |
| PC010 | S10030 | L Jones | L023 | Pensions | 18.50 |
| PC010 | S21010 | P Lewis | L004 | IT | 21.00 |
| PC045 | S10010 | B Jones | L004 | IT | 21.75 |
| PC045 | S10001 | A Smith | L004 | IT | 18.00 |
| PC045 | S31002 | T Gilbert | L028 | Database | 25.50 |
| PC045 | S13210 | W Richards | L008 | Salary | 17.00 |
| PC064 | S31002 | T Gilbert | L028 | Database | 23.25 |
| PC064 | S21010 | P Lewis | L004 | IT | 17.50 |
| PC064 | S10034 | B James | L009 | HR | 16.50 |

**Next:** [Step 3](http://www.sqa.org.uk/e-learning/MDBS01CD/page_30.htm#Non)

**Step 3**

Transform 1NF data into second normal form (2NF). Remove any -key attributes (partial Dependencies) that only depend on part of the table key to a new table.

What has to be determined "is field A dependent upon field B or vice versa?" This means: "Given a value for A, do we then have only one possible value for B, and vice versa?" If the answer is yes, A and B should be put into a new relation with A becoming the primary key. A should be left in the original relation and marked as a foreign key.

Ignore tables with (a) a simple key or (b) with no non-key attributes (these go straight to 2NF with no conversion).

The process is as follows:

Take each non-key attribute in turn and ask the question: is this attribute dependent on **one part** of the key?

* If yes, remove the attribute to a new table with a **copy** of the **part** of the key it is dependent upon. The key it is dependent upon becomes the key in the new table. Underline the key in this new table.
* If no, check against other part of the key and repeat above process
* If still no, ie: not dependent on either part of the key, keep attribute in current table.

**Notes:**

* The first table went straight to 2NF as it has a simple key (Project Code).
* Employee name, Department No and Department Name are dependent upon Employee No only. Therefore, they were moved to a new table with Employee No being the key.
* However, Hourly Rate is dependent upon both Project Code and Employee No as an employee may have a different hourly rate depending upon which project they are working on. Therefore it remained in the original table.

**Next:** [2NF Tables](http://www.sqa.org.uk/e-learning/MDBS01CD/page_31.htm#NF2Tables)

**2NF Tables: Partial Key Dependencies Removed**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Code** | **Project Title** | **Project Manager** | **Project Budget** |
| PC010 | Pensions System | M Phillips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC064 | HR System | K Lewis | 12250 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Code** | **Employee No.** | **Hourly Rate** |  | **Employee No.** | **Employee Name** | **Department No.** | **Department Name** |
| PC010 | S10001 | 22.00 |  | S10001 | A Smith | L004 | IT |
| PC010 | S10030 | 18.50 |  | S10030 | L Jones | L023 | Pensions |
| PC010 | S21010 | 21.00 |  | S21010 | P Lewis | L004 | IT |
| PC045 | S10010 | 21.75 |  | S10010 | B Jones | L004 | IT |
| PC045 | S10001 | 18.00 |  | S31002 | T Gilbert | L028 | Database |
| PC045 | S31002 | 25.50 |  | S13210 | W Richards | L008 | Salary |
| PC045 | S13210 | 17.00 |  | S10034 | B James | L009 | HR |
| PC064 | S31002 | 23.25 |  |  |  |  |  |
| PC064 | S21010 | 17.50 |  |  |  |  |  |
| PC064 | S10034 | 16.50 |  |  |  |  |  |

**Next:** [Step 4](http://www.sqa.org.uk/e-learning/MDBS01CD/page_32.htm#Transform)

**Step 4**

data in second normal form (2NF) into third normal form (3NF).

Remove to a new table any non-key attributes that are more dependent on other non-key attributes than the table key.

What has to be determined is "is field A dependent upon field B or vice versa?" This means: "Given a value for A, do we then have only one possible value for B, and vice versa?" If the answer is yes, then A and B should be put into a new relation, with A becoming the primary key. A should be left in the original relation and marked as a foreign key.

Ignore tables with zero or only one non-key attribute (these go straight to 3NF with no conversion).

The process is as follows: If a non-key attribute is more dependent on another non-key attribute than the table key:

* Move the **dependent** attribute, together with a **copy** of the non-key attribute upon which it is dependent, to a new table.
* Make the non-key attribute, upon which it is dependent, the key in the new table. Underline the key in this new table.
* **Leave** the non-key attribute, upon which it is dependent, in the original table and mark it a **foreign key** (\*).

**Notes:**

* The project team table went straight from 2NF to 3NF as it only has one non-key attribute.
* Department Name is more dependent upon Department No than Employee No and therefore was moved to a new table. Department No is the key in this new table and a foreign key in the Employee table.

Next: [3NF Tables](http://www.sqa.org.uk/e-learning/MDBS01CD/page_33.htm#NF3Tables)

**3NF Tables: Non-Key Dependencies Removed**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Code** | **Project Title** | **Project Manager** | **Project Budget** |
| PC010 | Pensions System | M Phillips | 24500 |
| PC045 | Salaries System | H Martin | 17400 |
| PC064 | HR System | K Lewis | 12250 |

|  |  |  |
| --- | --- | --- |
| **Project Code** | **Employee No.** | **Hourly Rate** |
| PC010 | S10001 | 22.00 |
| PC010 | S10030 | 18.50 |
| PC010 | S21010 | 21.00 |
| PC045 | S10010 | 21.75 |
| PC045 | S10001 | 18.00 |
| PC045 | S31002 | 25.50 |
| PC045 | S13210 | 17.00 |
| 064 | S31002 | 23.25 |
| PC064 | S21010 | 17.50 |
| PC064 | S10034 | 16.50 |

|  |  |  |
| --- | --- | --- |
| **Employee No.** | **Employee Name** | **Department No. \*** |
| S10001 | A Smith | L004 |
| S10030 | L Jones | L023 |
| S21010 | P Lewis | L004 |
| S10010 | B Jones | L004 |
| S31002 | T Gilbert | L023 |
| S13210 | W Richards | L008 |
| S10034 | B James | L0009 |

|  |  |
| --- | --- |
| **Department No.** | **Department Name** |
| L004 | IT |
| L023 | Pensions |
| L028 | Database |
| L008 | Salary |
| L009 | HR |

**Next**: [Summary of Normalisation Rules](http://www.sqa.org.uk/e-learning/MDBS01CD/page_34.htm#Summary)

**Summary of Normalisation Rules**

That is the complete process. Having started off with an unnormalised table we finished with four normalised tables in 3NF. You will notice that duplication has been removed (apart from the keys needed to establish the links between those tables).

The process may look complicated. However, if you follow the rules **completely**, and **donot** miss out any steps, then you should arrive at the correct solution. If you omit a rule there is a high probability that you will end up with too few tables or incorrect keys.

The following normal forms were discussed in this section:

1. **First normal form:** A table is in the first normal form if it contains no repeating columns.
2. **Second normal form:** A table is in the second normal form if it is in the first normal form and contains only columns that are dependent on the whole (primary) key.
3. **Third normal form:** A table is in the third normal form if it is in the second normal form and all the non-key columns are dependent only on the primary key. If the value of a non-key column is dependent on the value of another non-key column we have a situation known as transitive dependency. This can be resolved by removing the columns dependent on non-key items to another table.