**Normalization Example**

We will use the **Student\_Grade\_Report**table below, from a School database, as our example to explain the process for Normalization.

**Student\_Grade\_Report**(StudentNo, StudentName, Major, CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation, Grade)



Figure Dependency diagram

The abbreviations used in Figure1 are as follows:

* PD: partial dependency
* TD:  transitive dependency
* FD:  full dependency (Note: FD typically stands for **functional** dependency. Using FD as an abbreviation for full dependency is only used in Figure1)

**First Normal Form (1NF)**

In the *first normal form*, only single values are permitted at the intersection of each row and column; hence, there are no repeating groups.

To normalize a relation that contains a repeating group, remove the repeating group and form two new relations.

The PK of the new relation is a combination of the PK of the original relation plus an attribute from the newly created relation for unique identification.

We will use the **Student\_Grade\_Report**table below, from a School database, as our example to explain the process for 1NF.

**Student\_Grade\_Report**(StudentNo, StudentName, Major, CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation, Grade)

In the Student Grade Report table, the repeating group is the course information

* Remove the repeating group. In this case, it’s the course information for each student.
* Identify the PK for your new table.
* The PK must uniquely identify the attribute value (StudentNo and CourseNo).
* After removing all the attributes related to the course and student, you are left with the student course table (**StudentCourse**).
* The Student table (**Student**) is now in first normal form with the repeating group removed.
* The two new tables are shown below.

**Student** (StudentNo, StudentName, Major)

**StudentCourse** (StudentNo, CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation, Grade)

## Second Normal Form (2NF)

For the second normal form, the relation must first be in 1NF. The relation is automatically in 2NF if, and only if, the PK comprises a single attribute.

If the relation has a composite PK, then each non-key attribute must be fully dependent on the entire PK and not on a subset of the PK (i.e., there must be no partial dependency or augmentation).

To move to 2NF, a table must first be in 1NF.

### **Process for 2NF**

* The Student table is already in 2NF because it has a single-column PK.
* When examining the Student Course table, we see that not all the attributes are fully dependent on the PK; specifically, all course information. The only attribute that is fully dependent is grade.
* Identify the new table that contains the course information.
* Identify the PK for the new table.
* The three new tables are shown below.

**Student** (StudentNo, StudentName, Major)

**CourseGrade** (StudentNo, CourseNo, Grade)

**CourseInstructor** (CourseNo, CourseName, InstructorNo, InstructorName, InstructorLocation)

## Third Normal Form (3NF)

To be in third normal form, the relation must be in second normal form. Also all transitive dependencies must be removed; a non-key attribute may not be functionally dependent on another non-key attribute.

### **Process for 3NF**

* Eliminate all dependent attributes in transitive relationship(s) from each of the tables that have a transitive relationship.
* Create new table(s) with removed dependency.
* Check new table(s) as well as table(s) modified to make sure that each table has a determinant and that no table contains inappropriate dependencies.
* See the four new tables below.

**Student** (StudentNo, StudentName, Major)

**CourseGrade** (StudentNo, CourseNo, Grade)

**Course** (CourseNo, CourseName, InstructorNo)

**Instructor** (InstructorNo, InstructorName, InstructorLocation)

At this stage, there should be no anomalies in third normal form.

## Boyce-Codd Normal Form (BCNF)

**Student** (StudentNo, StudentName, Major)

**CourseGrade** (StudentNo, CourseNo, Grade)

**Course** (CourseNo, CourseName, InstructorNo)

**Instructor** (InstructorNo, InstructorName, InstructorLocation)

These relations in BCNF; every determinant is a key.