## Given Table:

| StudentID | StudentName | Course1 | Course2 | Course3 |
|---|---|---|---|---|
| 101 | Alice | Math | History | Physics |
| 102 | Bob | Physics | NULL | NULL |
| 103 | Carol | Chemistry | Biology | NULL |

**1)**
**Normalization Issues:**

1) Repetition of the column in the table for the same kinds of data. This redundancy can cause data inconsistency and storage problems. It violates the 1NF.
2) If any student wants to enroll in more than 3 subjects, we need to alter the structure of the table.
3) It is difficult to add, update, or remove the courses without altering the table.
4) Writing queries to get the data can be challenging
5) Hard to aggregate and analyze the data properly

**2) Normalized table**

**STUDENT TABLE:**

| StudentID | StudentName |
|---|---|
| 101 | Alice |
| 102 | Bob |
| 103 | Carol |

**Course Table**

| CourseID | CourseName |
|---|---|
| 1 | Math |
| 2 | Physics |
| 3 | Chemistry |
| 4 | History |
| 5 | Biology |

**Enrollment Table**

| EnrollmentID | StudentID | CourseID |
|---|---|---|
| **1** | **101** | **1** |
| **2** | **101** | **2** |
| **3** | **101** | **4** |
| **4** | **102** | **2** |
| **5** | **103** | **3** |
| **6** | **103** | **5** |

**How it solves the normalization problem?**

1) The given table violates 1NF by adding non-atomic values. We separated the table of courses and students and now it is atomic.
2) The Enrollment table has studentID and CourseID foreign keys and has a unique value in each row. So no Duplication.