Students

|  |  |
| --- | --- |
| stud\_id | stud\_name |
| 1234 | John Smith |
| 9555 | Peter |

Subjects

|  |  |
| --- | --- |
| subj\_code | subj\_title |
| ITM2005 | System Architecture |
| ACS1005 | Data Mgmt |
| PRG1001 | Programming I |

Convenor

|  |  |
| --- | --- |
| Staff\_id | Convenor |
| 111 | Bob Hauser |
| 222 | Jane Collins |
| 111 | Bob Hauser |
| 333 | Ahmad Singh |

Results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| stud\_id | subj\_code | Sem | Year | staff\_id | grade | Mark |
| 1234 | ITM2005 | 1 | 2015 | 111 | 83 | D |
| 1234 | ACS1005 | 2 | 2015 | 222 | 44 | N |
| 9555 | ITM20005 | 1 | 2015 | 111 | 95 | HD |
| 9555 | PRG1001 | 2 | 2015 | 333 | 65 | C |

To achieve third normal form, all fields (columns) can be determined only by the key in the table and no other key. In this task, staff\_id and convenor were used to make a new table called Convenor. This is because they had an indirect relationship that causes functional dependency, known as transitive dependency. What this essentially means is that changes to one non-primary attribute will influence another non-primary attribute. For example, if Bob Hauser stopped System Architecture, you will have to make changes to staff\_id column, and the convenor column because it’s dependent on staff\_id. It becomes harder to make such changes, especially if the convenor teaches more than one subject. To avoid these complications, a new table is created, leaving staff\_id in the Results table as a foreign key. Now if the convenor of a subject changes, we only need to make changes to the staff\_id in the Results table.