**FSDM / CPCM – 2023S**

**Database Design & SQL**

A screen shot of a computer

Description automatically generated with low confidence**Student ID:** 901142  
**Student Name:** Roshan Shrestha  
**Practical Exercise 9\_10\_11\_12**

**Practical Activity #1**

1. **Criteria for 1NF**

* There shouldn't be any repeating groupings in the table.
* There shouldn't be any multivalued columns in any table.
* A primary key need to be specified.

1. **Criteria for 2NF**

* The table must satisfy all requirements to be in 1NF.
* A table should not be partially dependent on any other tables.

1. **Criteria for 3NF**

* Table must meet all requirements to be in 2NF.
* The value of any non-key columns should not depend on any other non-key columns.

**Practical Activity #2**

**Apply the criteria and normalize the above table into 1NF:**

The table shouldn't have two or more related or similar-data-containing columns, as per the requirements listed for the table to be in 1NF. Two columns in the supplied table, **“Major”1** and **“Major2”**, appear to contain the same type of data, indicating that they include repeated groups of data. To alleviate this problem and fulfil the first requirement for 1NF, a new table can be created to record the **“Major”** In a similar way, the second requirement for a table to be in 1NF is that it cannot contain any multivalued columns. Since the given table does not contain any multivalued columns, we can move on to the final requirement, which is defining a primary key for the table. We can utilise **StudentID**, a unique primary key, in the table that is provided. After applying all the requirements for 1NF, the new updated table can be visualized as below:

# Create a separate table to combine and store value of “Major1” and “Major2”.

|  |  |
| --- | --- |
| **MajorID** | **Major** |
| M1 | Programming |
| M2 | Web Development |
| M3 | Programming |
| M4 | Web Development |
| M5 | Accounting |
| M6 | Web Development |
| M7 | Auto Technology |
| M8 | Web Development |
| M9 | Welding |
| M10 | Web Development |

# Create table to store other remaining data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **StudentID** | **Student** | **ProfID** | **ProfFirst** | **ProfLast** | **Department** |
| 003-112-224 | Alexia Carnell | AKJ | Akram | Aziz | Computer Science |
| 013-112-229 | Savanna  Sanderlin | AKJ | Akram | Aziz | Computer  Science |
| 009-112-227 | Buddy Pothier | DOM | Donnell | Meaders | Business |
| 012-111-228 | Verline Papadopoulos | DOM | Donnell | Meaders | Business |
| 002-111-223 | Buddy Pothier | JJJ | James | Justus | Technology |

1. **Create table to link the remaining data in the table and newly created table to store major using “StudentID” as primary key.**

|  |  |
| --- | --- |
| **MajorID** | **StudentID** |
| M1 | 003-112-224 |
| M2 | 003-112-224 |
| M3 | 013-112-229 |
| M4 | 013-112-229 |
| M5 | 009-112-227 |
| M6 | 009-112-227 |
| M7 | 012-111-228 |
| M8 | 012-111-228 |
| M9 | 002-111-223 |
| M10 | 002-111-223 |

**Practical Activity #3**

**Apply the criteria and normalize the above table into 2NF:**

A table must initially fulfil every prerequisite for being in **1NF** before it can be in **2NF**. The given table has already been normalised to **1NF** in this case, therefore we may move on to the next criterion. The final standard for a table. The table in **2NF** should not have any partial dependencies; instead, we must ensure that all the non-key columns already in existence are solely dependent on the primary key. According to the provided table, it appears that there are partial dependencies; the non-key characteristics linked to professor information, such as **ProfID**, **ProfFirst**, and **ProfLast**, should be reliant on the full primary key, i.e., **StudentID**, but they are only dependent on **ProfID**. So, to address this issue we can create a separate table to store information of professor and link with the existing table using **ProfID** as the primary key. After applying all the criteria for 2NF, the new updated table can be visualized as below:

1. **Create separate table to store information related to professor.**

|  |  |  |
| --- | --- | --- |
| **ProfID** | **ProfFirst** | **ProfLast** |
| AKJ | Akram | Aziz |
| DOM | Donnell | Meaders |
| JJJ | James | Justus |

1. **Link the existing data table using ProfID as primary key.**

|  |  |  |  |
| --- | --- | --- | --- |
| **StudentID** | **Student** | **Department** | ***ProfID*** |
| 003-112-224 | Alexia Carnell | Computer Science | AKJ |
| 013-112-229 | Savanna  Sanderlin | Computer  Science | AKJ |
| 009-112-227 | Buddy Pothier | Business | DOM |
| 012-111-228 | Verline Papadopoulos | Business | DOM |
| 002-111-223 | Buddy Pothier | Technology | JJJ |

1. **List down other remaining data table.**

|  |  |
| --- | --- |
| **MajorID** | **Major** |
| M1 | Programming |
| M2 | Web Development |
| M3 | Programming |
| M4 | Web Development |
| M5 | Accounting |
| M6 | Web Development |
| M7 | Auto Technology |
| M8 | Web Development |
| M9 | Welding |
| M10 | Web Development |

|  |  |
| --- | --- |
| **MajorID** | **StudentID** |
| M1 | 003-112-224 |
| M2 | 003-112-224 |
| M3 | 013-112-229 |
| M4 | 013-112-229 |
| M5 | 009-112-227 |
| M6 | 009-112-227 |
| M7 | 012-111-228 |
| M8 | 012-111-228 |
| M9 | 002-111-223 |
| M10 | 002-111-223 |

**Practical Activity #4**

**Apply the criteria and normalize the above table into 3NF:**

According to the normalisation principles, a table must first meet every need of **2NF** to satisfy the conditions of **3NF**. Furthermore, no non-key value column should be used to determine the value of another non-key value column. Here, after normalising the given table to **2NF**, we can observe that there is a transitive relationship between the **"Department"** column and the professor detail, **"ProfID"** which is not a primary key. As a result, we can make a new table with **"DepartmentID**" as its primary key to contain information about departments to end this reliance. And after completing all the steps applying all criteria for **3NF** the final tables can be visualised as shown:

1. **Students table to store information related to student.**

|  |  |  |  |
| --- | --- | --- | --- |
| **StudentID** | **Student** | ***ProfID*** | **DepartmentID** |
| 003-112-224 | Alexia Carnell | AKJ | C100 |
| 013-112-229 | Savanna Sanderlin | AKJ | C100 |
| 009-112-227 | Buddy Pothier | DOM | B101 |
| 012-111-228 | Verline Papadopoulos | DOM | B101 |
| 002-111-223 | Buddy Pothier | JJJ | T102 |

1. **Major table to store information about majors.**

|  |  |
| --- | --- |
| **MajorID** | **Major** |
| M1 | Programming |
| M2 | Web Development |
| M3 | Programming |
| M4 | Web Development |
| M5 | Accounting |
| M6 | Web Development |
| M7 | Auto Technology |
| M8 | Web Development |
| M9 | Welding |
| M10 | Web Development |

1. **StdMajor table to link students table to major table.**

|  |  |
| --- | --- |
| **MajorID** | **StudentID** |
| M1 | 003-112-224 |
| M2 | 003-112-224 |
| M3 | 013-112-229 |
| M4 | 013-112-229 |
| M5 | 009-112-227 |
| M6 | 009-112-227 |
| M7 | 012-111-228 |

1. **Professors table to store information about professor.**

|  |  |  |
| --- | --- | --- |
| **ProfID** | **ProfFirst** | **ProfLast** |
| AKJ | Akram | Aziz |
| DOM | Donnell | Meaders |
| JJJ | James | Justus |

1. **Department table to store information related to departments.**

|  |  |
| --- | --- |
| **DepartmentID** | **Department** |
| C100 | Computer Science |
| B101 | Business |
| T10Z | Technology |