# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

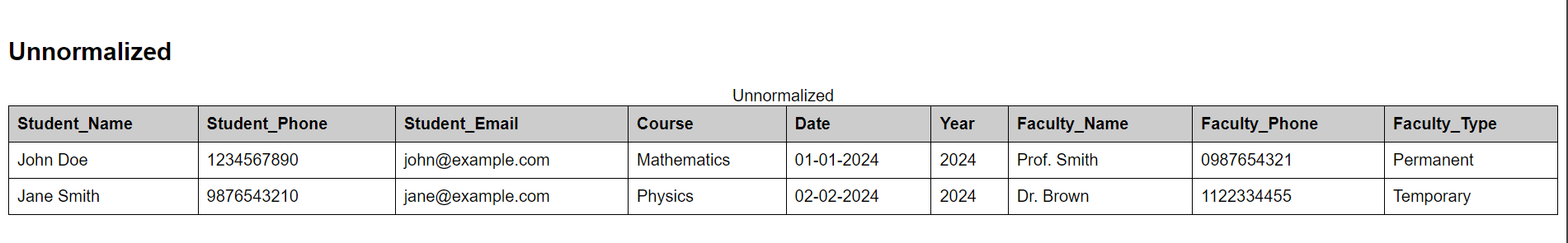
Assignment No: 2

**Topic:** College Management System Normalization

Normalization in database design ensures that the data is organized efficiently by reducing redundancy and improving data integrity. The following is a step-by-step normalization of a College Management System.

Step 1: Unnormalized Table

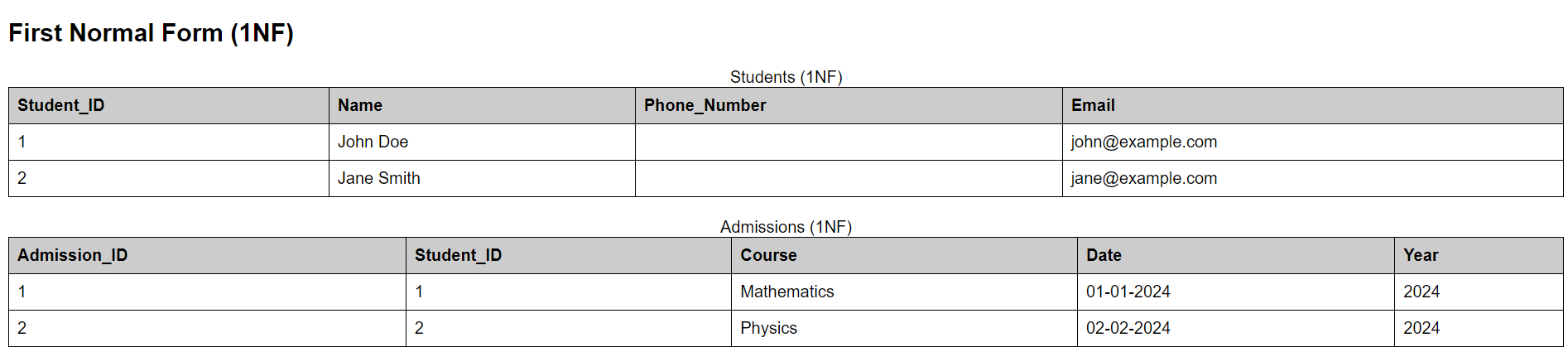
In the unnormalized table, there are redundancies and grouped data. Here, data related to students, courses, faculty, and dates are all stored in a single table, leading to repetition and difficulty in managing updates.



**Step 2: First Normal Form (1NF)**

To display the data in **First Normal Form (1NF)**, we ensure that each piece of information is stored in its most atomic form without repeating groups. In this case, we can create several separate tables to represent the normalized data:

* **Students Table**: Contains information about students.
* **Admissions Table**: Contains information about students and their enrollment details.
* **Faculty Table**: Contains information about instructors.



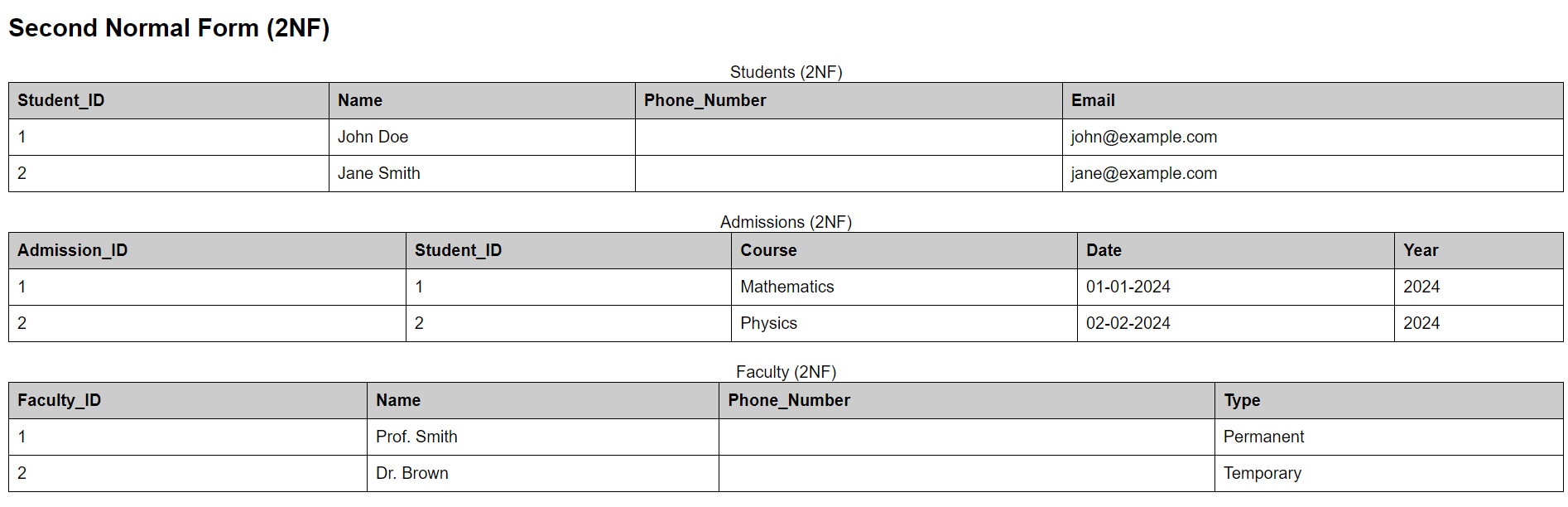
**Step 3: Second Normal Form (2NF)**

To display the data in **Second Normal Form (2NF)**, we must ensure that:

* The tables are already in 1NF.
* All non-key attributes are fully functionally dependent on the primary key.

We make further modifications:

* The **Admissions Table** references the **Students** and **Courses** tables through their IDs.
* **Faculty** information is completely separated from courses, as instructors can teach multiple courses.



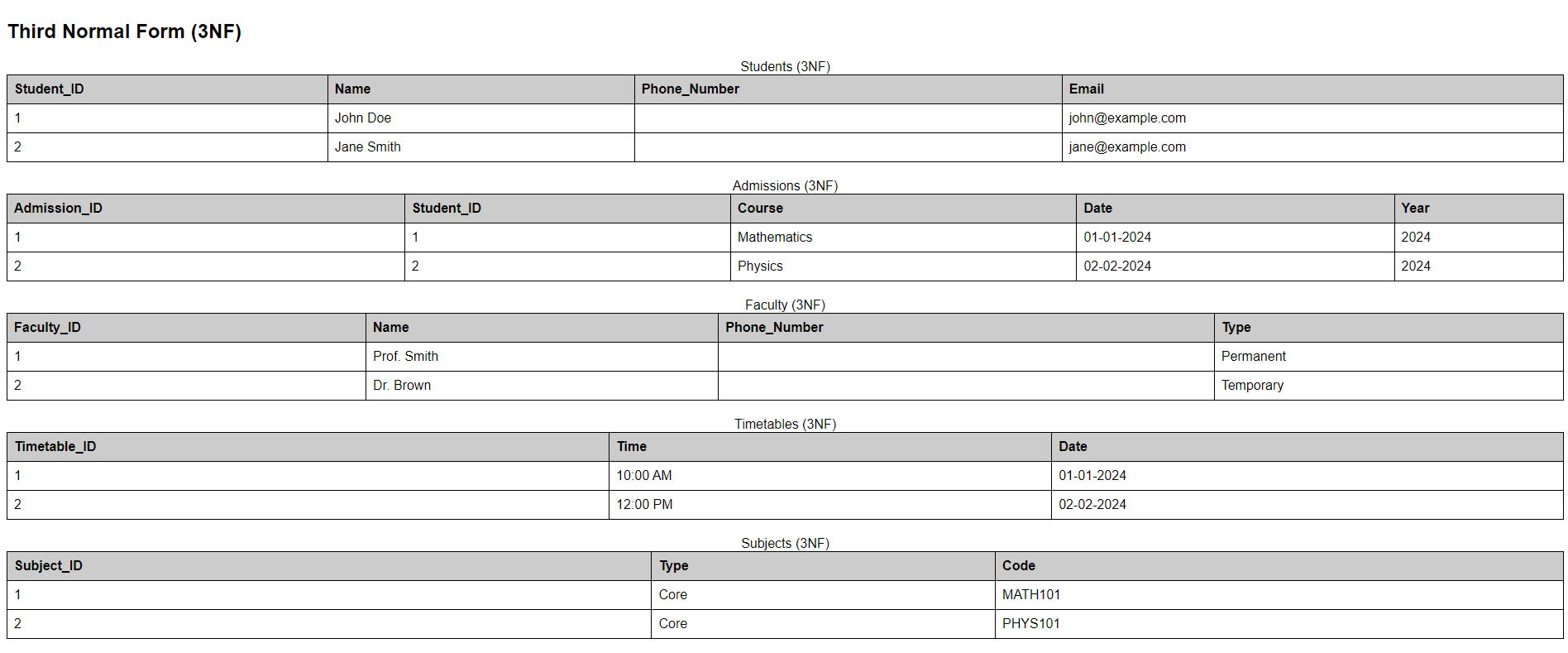
**Step 3: Third Normal Form (3NF)**

To convert the data structure into **Third Normal Form (3NF)**, we ensure that:

* The table is in 2NF.
* There are no transitive dependencies (all attributes are functionally dependent only on the primary key).

In this case, we create a separate **Timetable** and **Subjects** table, further reducing redundancy:

* **Students** and **Admissions** remain as before.
* **Faculty** is referenced by **CourseID** to remove transitive dependencies.



Step 5: Fourth and Fifth Normal Forms (4NF & 5NF)

In **Fourth Normal Form (4NF)** and **Fifth Normal Form (5NF)**, we eliminate multi-valued and complex join dependencies, ensuring that our database design adheres to strict normalization principles. At this stage, our tables have reached the final level of normalization, optimizing data integrity and minimizing redundancy.

Conclusion

In this document, we have normalized a **College Management System** database through various stages: from **1NF** to **5NF**. By applying these normalization techniques, we have reduced redundancy, enhanced data integrity, and organized the information into efficient and meaningful tables. This structured approach ensures that the database is optimized for storage, retrieval, and maintenance, while mitigating potential anomalies and inconsistencies that arise from poor design.