**Assignment 2**

**Overview:** This is a continuation to the previous assignment, In this assignment, you will translate your previously designed Entity-Relationship (ER) Diagram into a Relational Schema, normalize the schema, and ensure that all tables adhere to First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce-Codd Normal Form (BCNF). You will also define the foreign keys to enforce the relationships between entities and demonstrate how you normalize the schema step-by-step, providing clear explanations of the process.

1. Translate the ER diagram into relational schema:
   * For each entity in your ER diagram, create the corresponding relational schema (table) and define the primary and foreign keys.

Example for Student table:

StudentID (Primary Key), Name, DOB, Email, Phone

**Answer:**

The relational schema of each table is as follows:

|  |  |
| --- | --- |
| **Schema** | **Keys** |
| Student | StudentID (PK), Name, DOB, Email, Phone |
| Enrollment | EnrollmentID(PK), CourseID(FK), StudentID(FK), EnrollmentDate |
| Classroom | ClassroomID(PK), ClassroomName, Building, Capacity |
| Department | DepartmentID(PK), DepartmentName, Location |
| Course | CourseID(PK), DepartmentID(FK), Credits |
| Faculty | FacultyID(PK), DepartmentID(FK), Name, Email |

1. Define relationships:
   * Define the foreign keys for the relationships between the tables. For each relationship in your ER diagram, identify the foreign key columns in the corresponding tables.

Example for Enrollment table:

* EnrollmentID (Primary Key), StudentID (Foreign Key), CourseID (Foreign Key), EnrollmentDate

**Answer:**

The foreign keys of each table are as follows:

|  |  |
| --- | --- |
| **Table** | **Columns** |
| Enrollment | CourseID(FK), StudentID(FK) |
| Course | DepartmentID(FK) |
| Faculty | DepartmentID(FK) |

1. Normalize the schema:
   * Ensure that all tables follow the First Normal Form (1NF). Eliminate any repeating groups or arrays.

**Answer:**

**Rules for 1NF:**1. Each column must have atomic values.  
2. Each row must be unique and identifiable.

1. **Student Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **StudentID** | **Name** | **DOB** | **Email** | **Phone** |
| 1 | Amit Sharma | 12-05-2000 | amit.sharma@email.com | 9876543210 |
| 1 | Amit Sharma | 12-05-2000 | amit.sharma@email.com | 9123456789 |
| 2 | Priya Verma | 25-08-1999 | priya.verma@email.com | 9988776655 |
| 3 | Rahul Singh | 10-02-2001 | rahul.singh@email.com | 9876541230 |
| 3 | Rahul Singh | 10-02-2001 | rahul.singh@email.com | 8901234567 |
| 4 | Sneha Patil | 30-11-2002 | sneha.patil@email.com | 7788990011 |
| 5 | Ankit Mehta | 15-07-2000 | ankit.mehta@email.com | 8899776655 |
| 5 | Ankit Mehta | 15-07-2000 | ankit.mehta@email.com | 9900112233 |
| 6 | Komal Joshi | 05-12-1998 | komal.joshi@email.com | 7766554433 |
| 7 | Rohan Desai | 18-04-2003 | rohan.desai@email.com | 6655443322 |
| 8 | Neha Kapoor | 22-09-2001 | neha.kapoor@email.com | 9988001122 |
| 8 | Neha Kapoor | 22-09-2001 | neha.kapoor@email.com | 9900887766 |
| 9 | Vishal Thakur | 30-06-1997 | vishal.thakur@email.com | 8877665544 |
| 10 | Deepa Nair | 10-11-1999 | deepa.nair@email.com | 7766889900 |

1. **Enrollment Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **EnrollmentID** | **CourseID** | **StudentID** | **EnrollmentDate** |
| 101 | CSE101 | 1 | 10-01-2024 |
| 102 | CSE102 | 1 | 15-01-2024 |
| 103 | CSE101 | 2 | 12-01-2024 |
| 104 | CSE103 | 3 | 18-01-2024 |
| 105 | CSE101 | 3 | 20-01-2024 |
| 106 | CSE102 | 4 | 25-01-2024 |
| 107 | CSE104 | 5 | 02-02-2024 |
| 108 | CSE101 | 6 | 05-02-2024 |
| 109 | CSE103 | 7 | 10-02-2024 |
| 110 | CSE102 | 8 | 15-02-2024 |
| 111 | CSE104 | 8 | 20-02-2024 |
| 112 | CSE101 | 9 | 22-02-2024 |
| 113 | CSE103 | 10 | 25-02-2024 |

1. **Classroom Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ClassroomID** | **ClassroomName** | **Building** | **Capacity** |
| 101 | Einstein Hall | Science Block | 50 |
| 102 | Newton Room | Science Block | 40 |
| 103 | Aryabhata Lab | Tech Block | 60 |
| 104 | Kalam Auditorium | Main Block | 100 |
| 105 | Ramanujan Room | Math Block | 45 |

1. **Department Table:**

|  |  |  |
| --- | --- | --- |
| **DepartmentID** | **DepartmentName** | **Location** |
| 201 | Computer Science | Science Block |
| 202 | Mechanical Engineering | Tech Block |
| 203 | Electrical Engineering | Engineering Block |
| 204 | Mathematics | Math Block |
| 205 | Physics | Science Block |

1. **Faculty Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **FacultyID** | **Name** | **Email** | **DepartmentID** |
| 301 | Dr. Ramesh Patil | ramesh.patil@email.com | 201 |
| 302 | Prof. Sneha Desai | sneha.desai@email.com | 201 |
| 303 | Dr. Anil Sharma | anil.sharma@email.com | 201 |
| 304 | Dr. Priya Mehta | priya.mehta@email.com | 202 |
| 305 | Prof. Vikram Joshi | vikram.joshi@email.com | 202 |
| 306 | Dr. Kavita Rao | kavita.rao@email.com | 203 |
| 307 | Prof. Arjun Nair | arjun.nair@email.com | 204 |
| 308 | Dr. Neha Kapoor | neha.kapoor@email.com | 204 |
| 309 | Dr. Rajesh Verma | rajesh.verma@email.com | 205 |
| 310 | Prof. Sunita Kulkarni | sunita.kulkarni@email.com | 205 |

1. **Course Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **CourseID** | **CourseName** | **DepartmentID** | **Credits** |
| CSE101 | Data Structures & Algorithms | 201 | 4 |
| CSE102 | Machine Learning | 201 | 3 |
| CSE103 | Database Management Systems | 201 | 4 |
| ME201 | Thermodynamics | 202 | 4 |
| ME202 | Fluid Mechanics | 202 | 3 |
| EE301 | Circuit Analysis | 203 | 4 |
| EE302 | Power Systems | 203 | 3 |
| MTH401 | Linear Algebra | 204 | 4 |
| MTH402 | Probability & Statistics | 204 | 3 |
| PHY501 | Quantum Mechanics | 205 | 4 |

--------------------------------------------------------------------------------------------------------------------------------------

* + Convert the schema to Second Normal Form (2NF) by eliminating partial dependencies.

**Answer:**

**Rules:**  
1. The table should be in the first normal form.  
2. No any non-primary key is partially depend on the primary key. It should be fully dependent.

**Student Table:**

**Rules:**  
1. The table should be in the first normal form.  
2. No any non-primary key is partially depend on the primary key. It should be fully dependent.

**Problem in 1NF:**

* Phone is dependent only on StudentID, not on Name, DOB, Email.
* StudentID should be in one table, and Phone should be in a separate table.

**New Tables:**

**Student Table (2NF - Without Phone Numbers)**

| **StudentID** | **Name** | **DOB** | **Email** |
| --- | --- | --- | --- |
| 1 | Amit Sharma | 12-05-2000 | amit.sharma@email.com |
| 2 | Priya Verma | 25-08-1999 | priya.verma@email.com |
| 3 | Rahul Singh | 10-02-2001 | rahul.singh@email.com |
| 4 | Sneha Patil | 30-11-2002 | sneha.patil@email.com |
| 5 | Ankit Mehta | 15-07-2000 | ankit.mehta@email.com |
| 6 | Komal Joshi | 05-12-1998 | komal.joshi@email.com |
| 7 | Rohan Desai | 18-04-2003 | rohan.desai@email.com |

**Student Phone Table (2NF - Storing Phone Numbers Separately)**

| **StudentID** | **Phone** |
| --- | --- |
| 1 | 9876543210 |
| 1 | 9123456789 |
| 2 | 9988776655 |
| 3 | 9876541230 |
| 3 | 8901234567 |
| 4 | 7788990011 |
| 5 | 8899776655 |
| 5 | 9900112233 |
| 6 | 7766554433 |
| 7 | 6655443322 |

Now, there are no partial dependencies!

**Enrollment Table:**

Checking Dependencies:

* Primary Key: EnrollmentID (Unique for each row)
* CourseID, StudentID, and EnrollmentDate fully depend on EnrollmentID, so there are no partial dependencies.

The table is already in 2NF.

|  |  |  |  |
| --- | --- | --- | --- |
| **EnrollmentID** | **CourseID** | **StudentID** | **EnrollmentDate** |
| 101 | CSE101 | 1 | 10-01-2024 |
| 102 | CSE102 | 1 | 15-01-2024 |
| 103 | CSE101 | 2 | 12-01-2024 |
| 104 | CSE103 | 3 | 18-01-2024 |
| 105 | CSE101 | 3 | 20-01-2024 |
| 106 | CSE102 | 4 | 25-01-2024 |
| 107 | CSE104 | 5 | 02-02-2024 |
| 108 | CSE101 | 6 | 05-02-2024 |
| 109 | CSE103 | 7 | 10-02-2024 |
| 110 | CSE102 | 8 | 15-02-2024 |
| 111 | CSE104 | 8 | 20-02-2024 |
| 112 | CSE101 | 9 | 22-02-2024 |
| 113 | CSE103 | 10 | 25-02-2024 |

**Classroom Table:**

**Checking Dependencies:**

Primary Key: ClassroomID (each classroom has a unique ID).

ClassroomName, Building, and Capacity depend entirely on ClassroomID (no partial dependencies).

The table is already in 2NF.

|  |  |  |  |
| --- | --- | --- | --- |
| **ClassroomID** | **ClassroomName** | **Building** | **Capacity** |
| 101 | Einstein Hall | Science Block | 50 |
| 102 | Newton Room | Science Block | 40 |
| 103 | Aryabhata Lab | Tech Block | 60 |
| 104 | Kalam Auditorium | Main Block | 100 |
| 105 | Ramanujan Room | Math Block | 45 |

**Department Table:**

**Checking Dependencies:**

* Primary Key: DepartmentID (each department has a unique ID).
* DepartmentName and Location depend entirely on DepartmentID (no partial dependencies).

The table is already in 2NF.

|  |  |  |
| --- | --- | --- |
| **DepartmentID** | **DepartmentName** | **Location** |
| 201 | Computer Science | Science Block |
| 202 | Mechanical Engineering | Tech Block |
| 203 | Electrical Engineering | Engineering Block |
| 204 | Mathematics | Math Block |
| 205 | Physics | Science Block |

**Faculty Table:**

**Checking Dependencies:**

* **Primary Key:** FacultyID (each faculty member has a unique ID).
* **Name, Email, and DepartmentID** depend **entirely on FacultyID** (no partial dependencies).

The table is already in 2NF.

|  |  |  |  |
| --- | --- | --- | --- |
| **FacultyID** | **Name** | **Email** | **DepartmentID** |
| 301 | Dr. Ramesh Patil | ramesh.patil@email.com | 201 |
| 302 | Prof. Sneha Desai | sneha.desai@email.com | 201 |
| 303 | Dr. Anil Sharma | anil.sharma@email.com | 201 |
| 304 | Dr. Priya Mehta | priya.mehta@email.com | 202 |
| 305 | Prof. Vikram Joshi | vikram.joshi@email.com | 202 |
| 306 | Dr. Kavita Rao | kavita.rao@email.com | 203 |
| 307 | Prof. Arjun Nair | arjun.nair@email.com | 204 |
| 308 | Dr. Neha Kapoor | neha.kapoor@email.com | 204 |
| 309 | Dr. Rajesh Verma | rajesh.verma@email.com | 205 |
| 310 | Prof. Sunita Kulkarni | sunita.kulkarni@email.com | 205 |

**Course Table:**

**Checking Dependencies:**

Primary Key: CourseID (each course has a unique ID).

CourseName, DepartmentID, and Credits depend entirely on CourseID (no partial dependencies).

The table is already in 2NF.

|  |  |  |  |
| --- | --- | --- | --- |
| **CourseID** | **CourseName** | **DepartmentID** | **Credits** |
| CSE101 | Data Structures & Algorithms | 201 | 4 |
| CSE102 | Machine Learning | 201 | 3 |
| CSE103 | Database Management Systems | 201 | 4 |
| ME201 | Thermodynamics | 202 | 4 |
| ME202 | Fluid Mechanics | 202 | 3 |
| EE301 | Circuit Analysis | 203 | 4 |
| EE302 | Power Systems | 203 | 3 |
| MTH401 | Linear Algebra | 204 | 4 |
| MTH402 | Probability & Statistics | 204 | 3 |
| PHY501 | Quantum Mechanics | 205 | 4 |

--------------------------------------------------------------------------------------------------------------------------------------

* + Ensure the schema is in Third Normal Form (3NF) by eliminating transitive dependencies.

**Answers:**

**Rules for 3NF:**  
1. It should meet the requirements of the second normal form.  
2. There should not be any transitive dependencies.

**Student Table:**

Problems in 2NF:

* Email is dependent on StudentID, but it is also uniquely identifying students.
* We separate emails into a new table.

New Tables:

Student Table (3NF - Without Email)

| StudentID | Name | DOB |
| --- | --- | --- |
| 1 | Amit Sharma | 12-05-2000 |
| 2 | Priya Verma | 25-08-1999 |
| 3 | Rahul Singh | 10-02-2001 |
| 4 | Sneha Patil | 30-11-2002 |
| 5 | Ankit Mehta | 15-07-2000 |
| 6 | Komal Joshi | 05-12-1998 |
| 7 | Rohan Desai | 18-04-2003 |

Student Email Table (3NF - Storing Emails Separately)

| StudentID | Email |
| --- | --- |
| 1 | amit.sharma@email.com |
| 2 | priya.verma@email.com |
| 3 | rahul.singh@email.com |
| 4 | sneha.patil@email.com |
| 5 | ankit.mehta@email.com |
| 6 | komal.joshi@email.com |
| 7 | rohan.desai@email.com |

Now, there are no transitive dependencies!

**Enrollment Table:**

Checking Transitive Dependencies:

* CourseID depends on EnrollmentID, but CourseID contains information about courses.
* Solution: Split the table into two:
  + Enrollment Table (Without Course Details)
  + Course Table (Stores Course Details)

New Tables after 3NF

1. Enrollment Table (3NF - Without Course Details)

| EnrollmentID | StudentID | CourseID | EnrollmentDate |
| --- | --- | --- | --- |
| 101 | 1 | CSE101 | 10-01-2024 |
| 102 | 1 | CSE102 | 15-01-2024 |
| 103 | 2 | CSE101 | 12-01-2024 |
| 104 | 3 | CSE103 | 18-01-2024 |
| 105 | 3 | CSE101 | 20-01-2024 |
| 106 | 4 | CSE102 | 25-01-2024 |
| 107 | 5 | CSE104 | 02-02-2024 |
| 108 | 6 | CSE101 | 05-02-2024 |
| 109 | 7 | CSE103 | 10-02-2024 |
| 110 | 8 | CSE102 | 15-02-2024 |
| 111 | 8 | CSE104 | 20-02-2024 |
| 112 | 9 | CSE101 | 22-02-2024 |
| 113 | 10 | CSE103 | 25-02-2024 |

2. Course Table (3NF - Storing Course Details Separately)

| CourseID | CourseName | DepartmentID |
| --- | --- | --- |
| CSE101 | Data Structures & Algorithms | 201 |
| CSE102 | Machine Learning | 201 |
| CSE103 | Database Management Systems | 201 |
| CSE104 | Thermodynamics | 202 |

Now, there are no transitive dependencies!

**Classroom Table:**

Checking Transitive Dependencies:

* Building could potentially group multiple classrooms (e.g., multiple classrooms belong to "Science Block").
* Solution: Create a separate table for buildings to avoid redundancy.

New Tables after 3NF

1. Classroom Table (Without Building Details)

| ClassroomID | ClassroomName | BuildingID | Capacity |
| --- | --- | --- | --- |
| 101 | Einstein Hall | B1 | 50 |
| 102 | Newton Room | B1 | 40 |
| 103 | Aryabhata Lab | B2 | 60 |
| 104 | Kalam Auditorium | B3 | 100 |
| 105 | Ramanujan Room | B4 | 45 |

2. Building Table (Stores Building Details Separately)

| BuildingID | BuildingName |
| --- | --- |
| B1 | Science Block |
| B2 | Tech Block |
| B3 | Main Block |
| B4 | Math Block |

Now, there are no transitive dependencies!

**Department Table:**

**Checking Transitive Dependencies:**

* Location could potentially be redundant (e.g., multiple departments are located in "Science Block").
* Solution: Create a separate table for locations to avoid redundancy.

**New Tables after 3NF**

**1. Department Table (Without Location Details)**

| DepartmentID | DepartmentName | LocationID |
| --- | --- | --- |
| 201 | Computer Science | L1 |
| 202 | Mechanical Engineering | L2 |
| 203 | Electrical Engineering | L3 |
| 204 | Mathematics | L4 |
| 205 | Physics | L1 |

**2. Location Table (Stores Location Details Separately)**

| LocationID | LocationName |
| --- | --- |
| L1 | Science Block |
| L2 | Tech Block |
| L3 | Engineering Block |
| L4 | Math Block |

Now, there are no transitive dependencies!

**Faculty Table:**

**Issue:**

The DepartmentName and Location columns are functionally dependent on DepartmentID, not FacultyID.

Solution: Separate into a Faculty Table and a Department Table.

**1. Faculty Table (3NF - Without Redundant Department Details)**

| FacultyID | Name | Email | DepartmentID |
| --- | --- | --- | --- |
| 301 | Dr. Ramesh Patil | ramesh.patil@email.com | 201 |
| 302 | Prof. Sneha Desai | sneha.desai@email.com | 201 |
| 303 | Dr. Anil Sharma | anil.sharma@email.com | 201 |
| 304 | Dr. Priya Mehta | priya.mehta@email.com | 202 |
| 305 | Prof. Vikram Joshi | vikram.joshi@email.com | 202 |
| 306 | Dr. Kavita Rao | kavita.rao@email.com | 203 |
| 307 | Prof. Arjun Nair | arjun.nair@email.com | 204 |
| 308 | Dr. Neha Kapoor | neha.kapoor@email.com | 204 |
| 309 | Dr. Rajesh Verma | rajesh.verma@email.com | 205 |
| 310 | Prof. Sunita Kulkarni | sunita.kulkarni@email.com | 205 |

**2. Department Table (3NF - Stores Department and Location Separately)**

| DepartmentID | DepartmentName | Location |
| --- | --- | --- |
| 201 | Computer Science | Science Block |
| 202 | Mechanical Engineering | Tech Block |
| 203 | Electrical Engineering | Engineering Block |
| 204 | Mathematics | Math Block |
| 205 | Physics | Science Block |

Now, there are no transitive dependencies!

**Course Table:**

**Checking Transitive Dependencies:**

* DepartmentID is related to DepartmentName from the Department Table, so DepartmentID should be stored separately.
* Solution: Remove DepartmentName from this table (if present) and link via a separate Department Table.

**1. Course Table (3NF - Without Redundant Department Details)**

| CourseID | CourseName | DepartmentID | Credits |
| --- | --- | --- | --- |
| CSE101 | Data Structures & Algorithms | 201 | 4 |
| CSE102 | Machine Learning | 201 | 3 |
| CSE103 | Database Management Systems | 201 | 4 |
| ME201 | Thermodynamics | 202 | 4 |
| ME202 | Fluid Mechanics | 202 | 3 |
| EE301 | Circuit Analysis | 203 | 4 |
| EE302 | Power Systems | 203 | 3 |
| MTH401 | Linear Algebra | 204 | 4 |
| MTH402 | Probability & Statistics | 204 | 3 |
| PHY501 | Quantum Mechanics | 205 | 4 |

**2. Department Table (3NF - Stores Department Details Separately)**

| DepartmentID | DepartmentName |
| --- | --- |
| 201 | Computer Science |
| 202 | Mechanical Engineering |
| 203 | Electrical Engineering |
| 204 | Mathematics |
| 205 | Physics |

Now, there are no transitive dependencies!

--------------------------------------------------------------------------------------------------------------------------------------

* + Finally, check the schema for Boyce-Codd Normal Form (BCNF) if necessary.

**Answers:**

**Rules for BCNF:**  
1. It should meet the requirements of the third normal form.  
2. Every determinant (a column that determines another column) must be a candidate key.

**Student Table:**

1. Final Tables (BCNF)

1. Student Table
   * (StudentID, Name, DOB)

| StudentID | Name | DOB |
| --- | --- | --- |
| 1 | Amit Sharma | 12-05-2000 |
| 2 | Priya Verma | 25-08-1999 |
| 3 | Rahul Singh | 10-02-2001 |
| 4 | Sneha Patil | 30-11-2002 |
| 5 | Ankit Mehta | 15-07-2000 |
| 6 | Komal Joshi | 05-12-1998 |
| 7 | Rohan Desai | 18-04-2003 |

1. **Student Phone Table**
   * (StudentID, Phone)

| **StudentID** | **Phone** |
| --- | --- |
| 1 | 9876543210 |
| 1 | 9123456789 |
| 2 | 9988776655 |
| 3 | 9876541230 |
| 3 | 8901234567 |
| 4 | 7788990011 |
| 5 | 8899776655 |
| 5 | 9900112233 |
| 6 | 7766554433 |
| 7 | 6655443322 |

1. **Student Email Table**
   * (StudentID, Email)

| StudentID | Email |
| --- | --- |
| 1 | amit.sharma@email.com |
| 2 | priya.verma@email.com |
| 3 | rahul.singh@email.com |
| 4 | sneha.patil@email.com |
| 5 | ankit.mehta@email.com |
| 6 | komal.joshi@email.com |
| 7 | rohan.desai@email.com |

Now, the table follows BCNF, ensuring data integrity and eliminating redundancy.

**Enrollment Table:**

Checking BCNF Violations:

* Primary Key in Enrollment Table: EnrollmentID (No functional dependency violations).
* Primary Key in Course Table: CourseID (Unique for each course).

The tables are already in BCNF, meaning no further decomposition is required.

Final Normalized Tables (BCNF)

1. Enrollment Table (BCNF - Holds Student Enrollments)
   * (EnrollmentID, StudentID, CourseID, EnrollmentDate)

| EnrollmentID | StudentID | CourseID | EnrollmentDate |
| --- | --- | --- | --- |
| 101 | 1 | CSE101 | 10-01-2024 |
| 102 | 1 | CSE102 | 15-01-2024 |
| 103 | 2 | CSE101 | 12-01-2024 |
| 104 | 3 | CSE103 | 18-01-2024 |
| 105 | 3 | CSE101 | 20-01-2024 |
| 106 | 4 | CSE102 | 25-01-2024 |
| 107 | 5 | CSE104 | 02-02-2024 |
| 108 | 6 | CSE101 | 05-02-2024 |
| 109 | 7 | CSE103 | 10-02-2024 |
| 110 | 8 | CSE102 | 15-02-2024 |
| 111 | 8 | CSE104 | 20-02-2024 |
| 112 | 9 | CSE101 | 22-02-2024 |
| 113 | 10 | CSE103 | 25-02-2024 |

1. Course Table (BCNF - Holds Course Details)
   * (CourseID, CourseName, DepartmentID)

| CourseID | CourseName | DepartmentID |
| --- | --- | --- |
| CSE101 | Data Structures & Algorithms | 201 |
| CSE102 | Machine Learning | 201 |
| CSE103 | Database Management Systems | 201 |
| CSE104 | Thermodynamics | 202 |

Now, the tables follow BCNF, ensuring data integrity and eliminating redundancy.

**Classroom Table:**

Checking BCNF Violations:

Primary Key in Classroom Table: ClassroomID (each classroom is uniquely identified).

Primary Key in Building Table: BuildingID (each building has a unique ID).

**Final Normalized Tables (BCNF)**

**Classroom Table (BCNF - Holds Classroom Details)**

(ClassroomID, ClassroomName, BuildingID, Capacity)

| ClassroomID | ClassroomName | BuildingID | Capacity |
| --- | --- | --- | --- |
| 101 | Einstein Hall | B1 | 50 |
| 102 | Newton Room | B1 | 40 |
| 103 | Aryabhata Lab | B2 | 60 |
| 104 | Kalam Auditorium | B3 | 100 |
| 105 | Ramanujan Room | B4 | 45 |

**Building Table (BCNF - Holds Building Details)**

(BuildingID, BuildingName)

| BuildingID | BuildingName |
| --- | --- |
| B1 | Science Block |
| B2 | Tech Block |
| B3 | Main Block |
| B4 | Math Block |

Now, the tables follow BCNF, ensuring data integrity and eliminating redundancy.

**Department Table:**

**Checking BCNF Violations:**

* Primary Key in Department Table: DepartmentID (each department is uniquely identified).
* Primary Key in Location Table: LocationID (each location has a unique ID).

The tables are already in BCNF, meaning no further decomposition is required.

**Final Normalized Tables (BCNF)**

1. **Department Table (BCNF - Holds Department Details)**
   * (DepartmentID, DepartmentName, LocationID)

| DepartmentID | DepartmentName | LocationID |
| --- | --- | --- |
| 201 | Computer Science | L1 |
| 202 | Mechanical Engineering | L2 |
| 203 | Electrical Engineering | L3 |
| 204 | Mathematics | L4 |
| 205 | Physics | L1 |

1. **Location Table (BCNF - Holds Location Details)**
   * (LocationID, LocationName)

| LocationID | LocationName |
| --- | --- |
| L1 | Science Block |
| L2 | Tech Block |
| L3 | Engineering Block |
| L4 | Math Block |

**Faculty Table:**

Issue:

The Department Table still has Location dependent on DepartmentID.

Solution: Further normalize the department and location into a separate Location Table.

**1. Faculty Table (BCNF - Same as 3NF)**

| FacultyID | Name | Email | DepartmentID |
| --- | --- | --- | --- |
| 301 | Dr. Ramesh Patil | ramesh.patil@email.com | 201 |
| 302 | Prof. Sneha Desai | sneha.desai@email.com | 201 |
| 303 | Dr. Anil Sharma | anil.sharma@email.com | 201 |
| 304 | Dr. Priya Mehta | priya.mehta@email.com | 202 |
| 305 | Prof. Vikram Joshi | vikram.joshi@email.com | 202 |
| 306 | Dr. Kavita Rao | kavita.rao@email.com | 203 |
| 307 | Prof. Arjun Nair | arjun.nair@email.com | 204 |
| 308 | Dr. Neha Kapoor | neha.kapoor@email.com | 204 |
| 309 | Dr. Rajesh Verma | rajesh.verma@email.com | 205 |
| 310 | Prof. Sunita Kulkarni | sunita.kulkarni@email.com | 205 |

**2. Department Table (BCNF - Without Location)**

| DepartmentID | DepartmentName | LocationID |
| --- | --- | --- |
| 201 | Computer Science | L1 |
| 202 | Mechanical Engineering | L2 |
| 203 | Electrical Engineering | L3 |
| 204 | Mathematics | L4 |
| 205 | Physics | L1 |

**3. Location Table (BCNF - Stores Location Separately)**

| LocationID | LocationName |
| --- | --- |
| L1 | Science Block |
| L2 | Tech Block |
| L3 | Engineering Block |
| L4 | Math Block |

Now, the tables follow BCNF, ensuring no redundancy and data integrity.

**Course Table:**

**Issue:**

* The Department Table still has an indirect dependency on the Location.
* **Solution:** Further normalize Location into a separate **Location Table**.

**1. Course Table (BCNF - Same as 3NF)**

| **CourseID** | **CourseName** | **DepartmentID** | **Credits** |
| --- | --- | --- | --- |
| CSE101 | Data Structures & Algorithms | 201 | 4 |
| CSE102 | Machine Learning | 201 | 3 |
| CSE103 | Database Management Systems | 201 | 4 |
| ME201 | Thermodynamics | 202 | 4 |
| ME202 | Fluid Mechanics | 202 | 3 |
| EE301 | Circuit Analysis | 203 | 4 |
| EE302 | Power Systems | 203 | 3 |
| MTH401 | Linear Algebra | 204 | 4 |
| MTH402 | Probability & Statistics | 204 | 3 |
| PHY501 | Quantum Mechanics | 205 | 4 |

**2. Department Table (BCNF - Without Location Details)**

| **DepartmentID** | **DepartmentName** | **LocationID** |
| --- | --- | --- |
| 201 | Computer Science | L1 |
| 202 | Mechanical Engineering | L2 |
| 203 | Electrical Engineering | L3 |
| 204 | Mathematics | L4 |
| 205 | Physics | L1 |

**3. Location Table (BCNF - Stores Location Details Separately)**

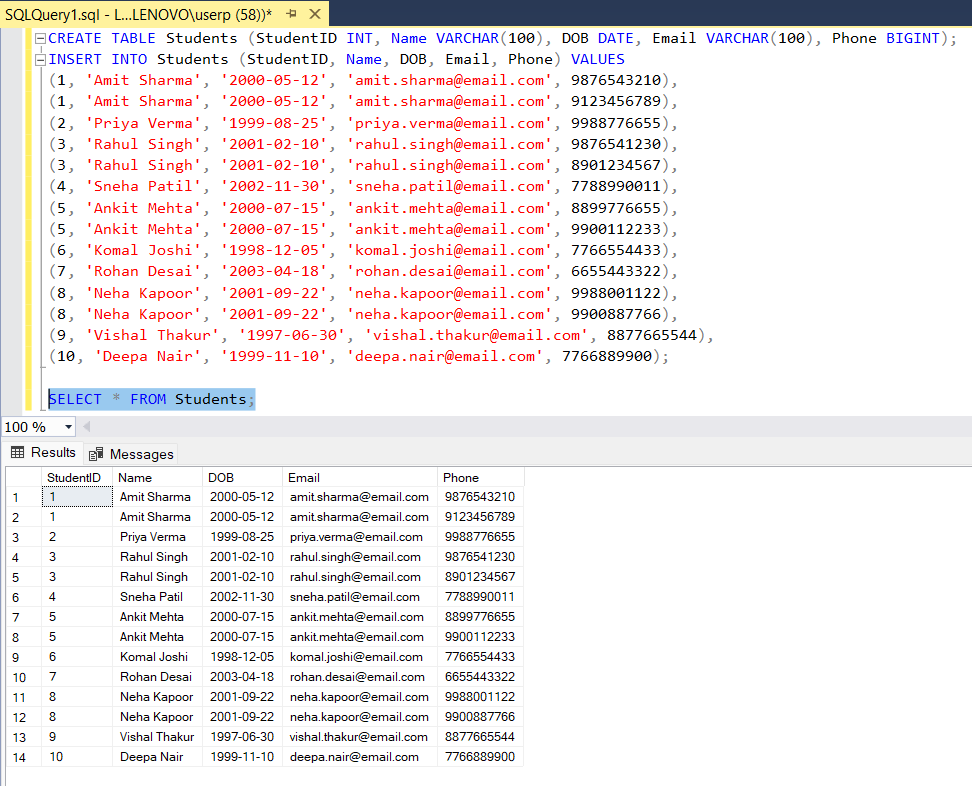
| **LocationID** | **LocationName** |
| --- | --- |
| L1 | Science Block |
| L2 | Tech Block |
| L3 | Engineering Block |
| L4 | Math Block |

**Now, the tables follow BCNF, ensuring no redundancy and data integrity.**

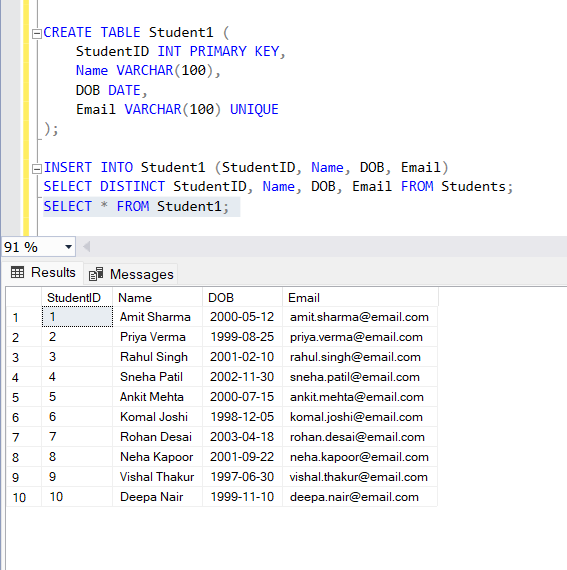
1. Deliverables (How and what to submit):
   * Submit the relational schema in tabular form, clearly indicating primary and foreign keys.
   * Provide a step-by-step explanation of the normalization process:
     + Show how each table is transformed from 1NF to 2NF, 2NF to 3NF, and 3NF to BCNF.
     + Explain the rationale behind every change made to eliminate partial or transitive dependencies, and other issues that violate normal forms.
   * For the schema you have designed, Write SQL queries for basic operations such as:
     + Insert data into each table.
     + Select and show data from each table.
     + Join tables based on relationships (e.g., SELECT \* FROM Enrollment JOIN Student ON Enrollment.StudentID = Student.StudentID).

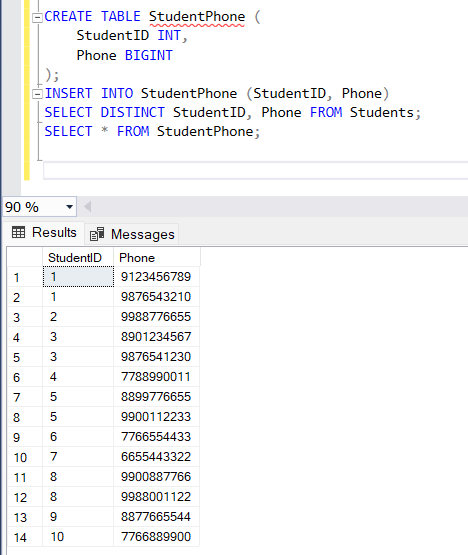
Provide **complete screenshots** for each query and result/output to demonstrate that your schema works as expected.

**Student Table:**



First Normal Form:

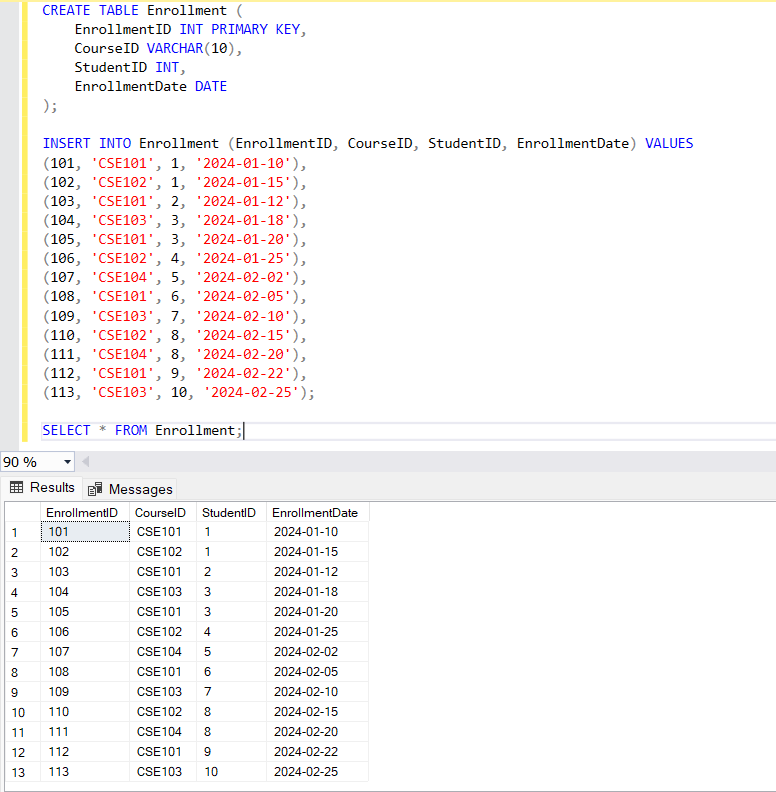




The above codes are enough for the other forms of the normalization.

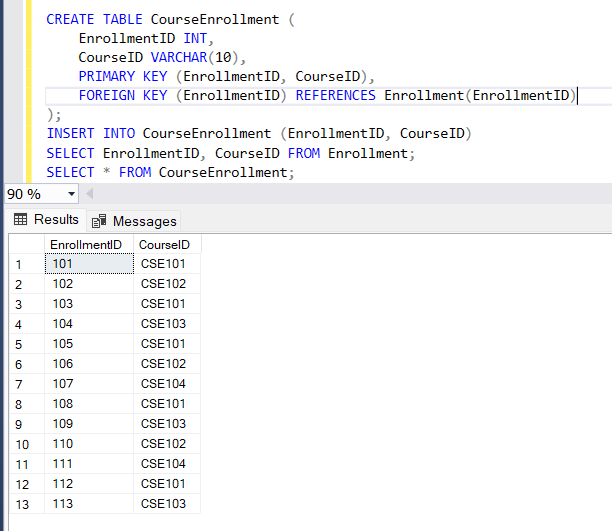
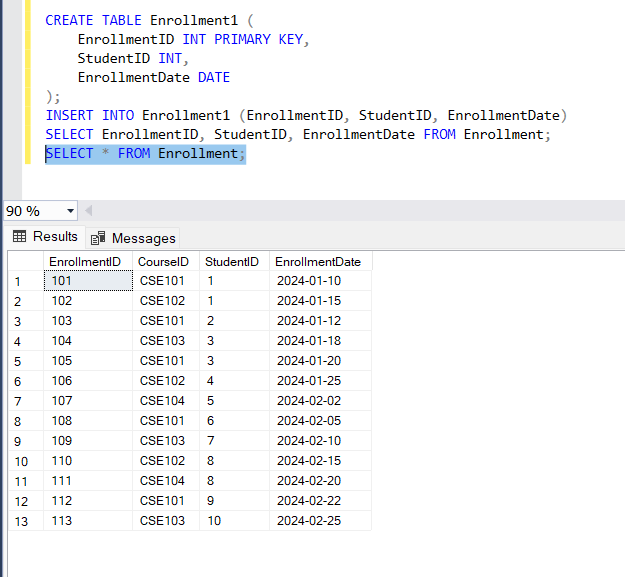
**Enrollment Table:**

Creating the data:

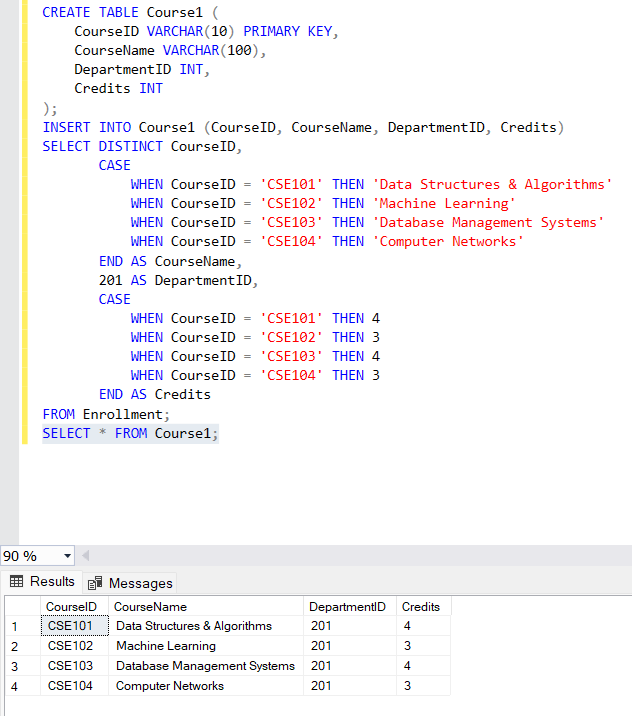


Normal Forms:

* + 1. The above table is in the first normal forms so there is no any change required.
    2. Second normal forms:



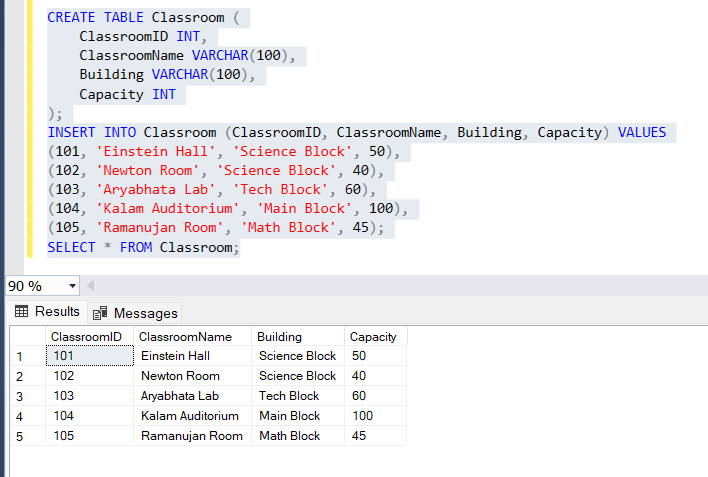
The above table can be used for the third normal form. The one required table is as follows:



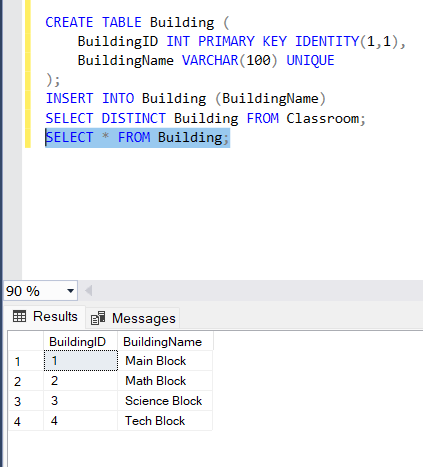
The above all the tables are the Boyce-Codd Normal form.

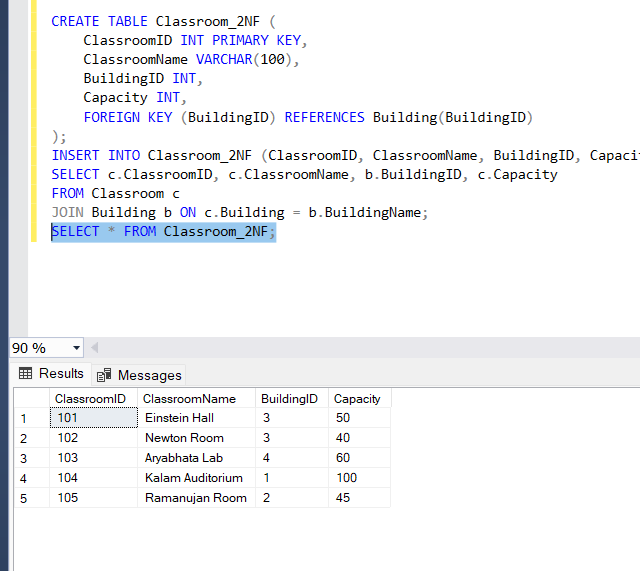
**Classroom Table:**

First Normal form:

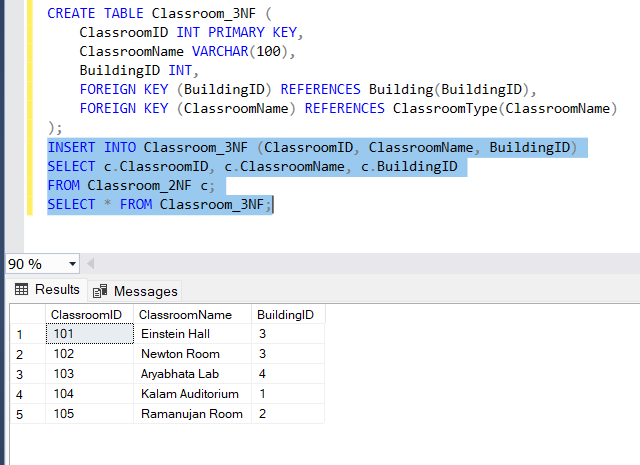
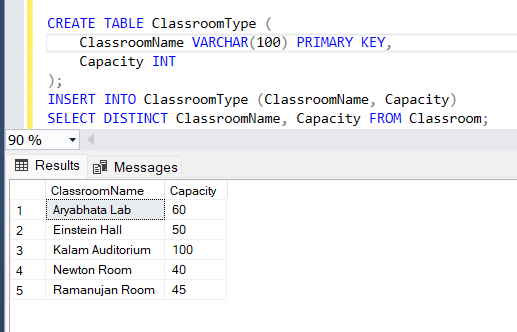


Second Normal Form:

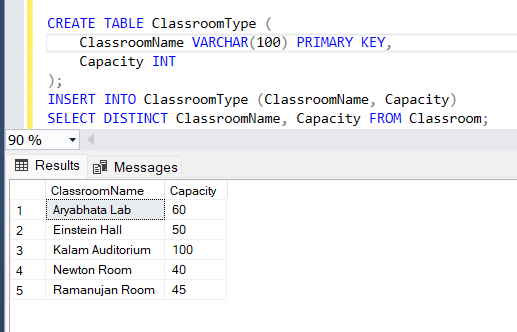
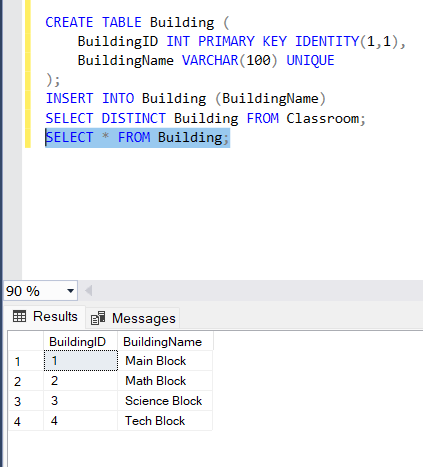




Third Normal Form:

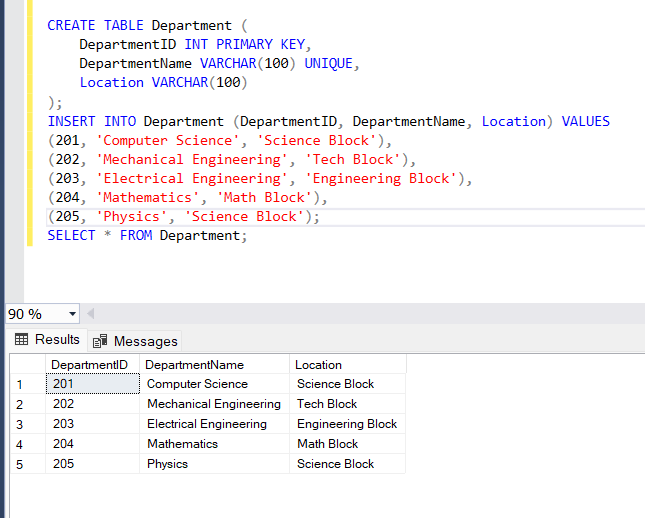


BCNF:

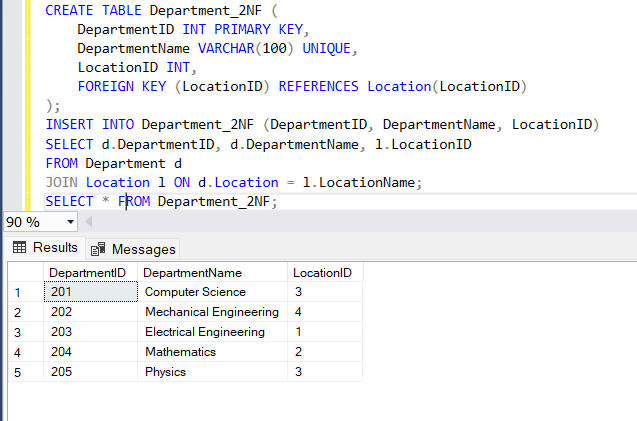
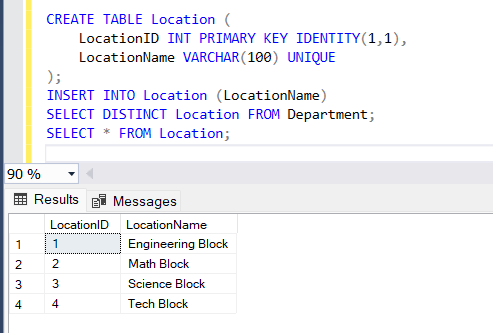


**Department Table:**

First Normal form:



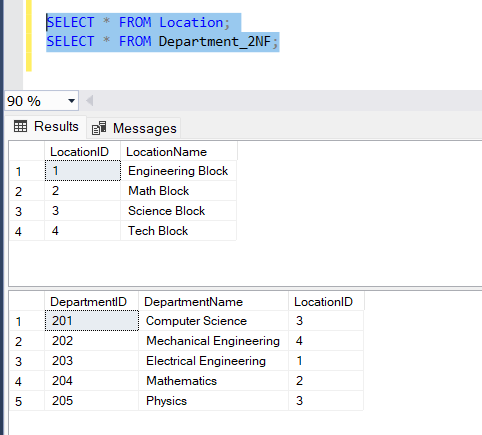
Second Normal Form:



Third Normal Form:

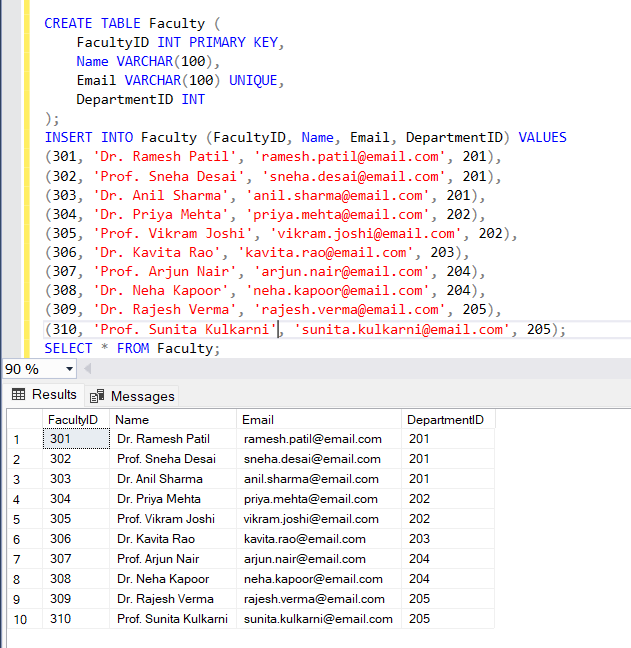
The second normal form is also be a third normal form for this table.

BNCF:

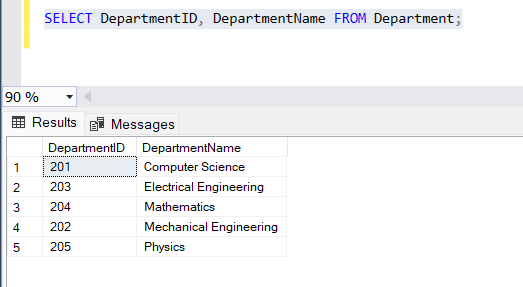


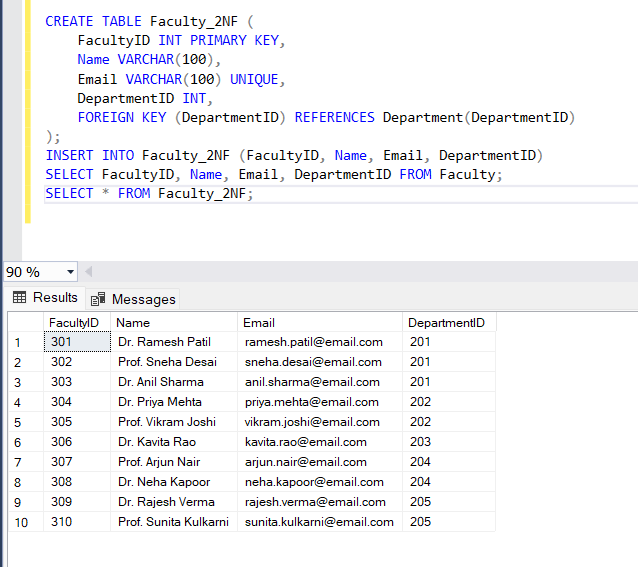
**Faculty Table:**

First Normal Form:



Second Normal Form:

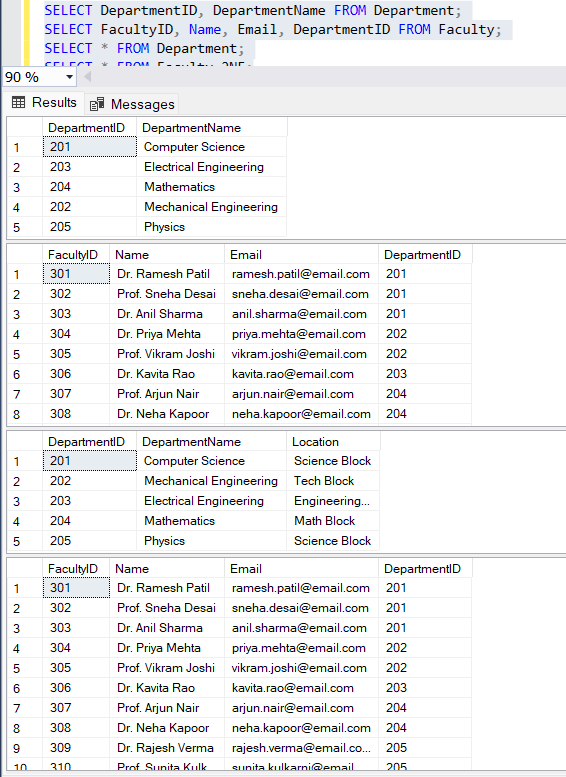




Third Normal Form:

The third normal form is the same as the second normal form.

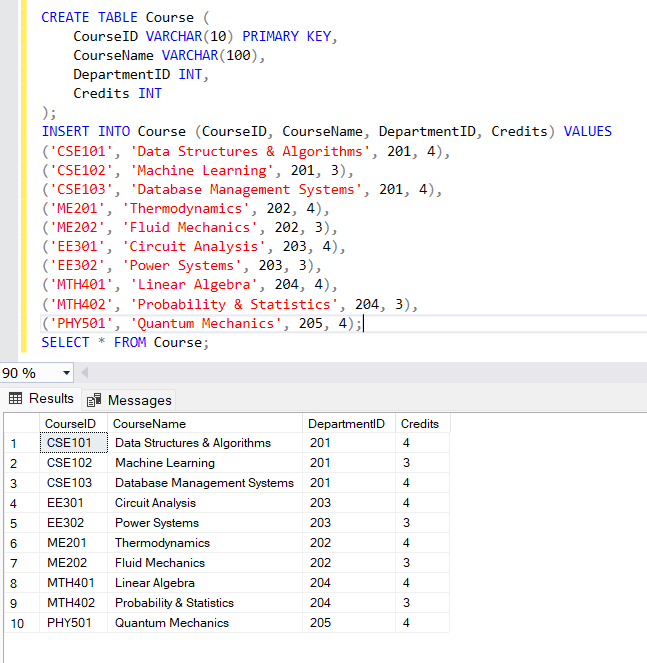
BCNF:



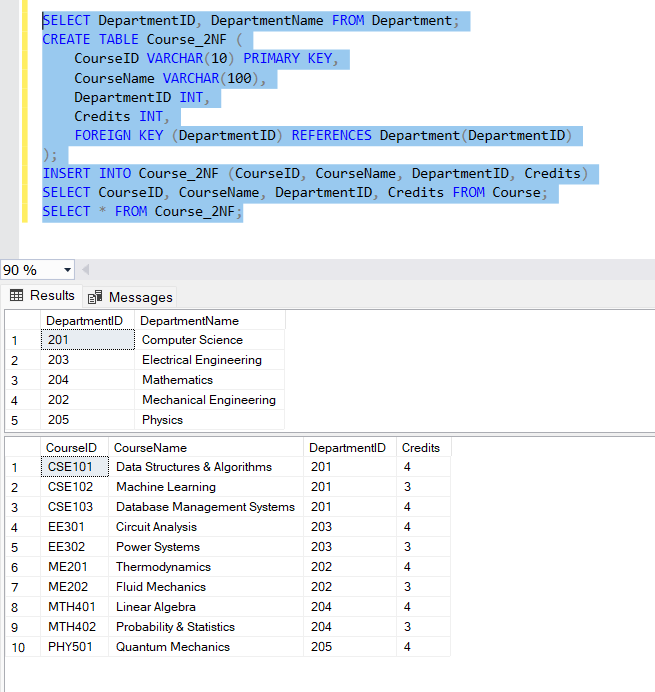
I have used the above tables to convert it into the required form.

**Course Table:**

First Normal Form:



Second Normal Form:



Third Normal Form:

The third normal form and BCNF are the same as the second normal form.