UNIVERSITY OF GONDAR

COLLAGE OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

DATABASE PROJECT

TITLE: Student Registration System

GROUP MEMBER	ID
1. AMANUAL AZANAW	02595/15
2. NATNAEL GETNET	01640/15
3. ASHENAFI HABTE	02264/15
4. GETAHUN NIGUSSIE	02621/15
5. ENCHALEW AMSALU	02208/15

SUBMITTE TO: Mr. GETNET
SUBMISSION DATE: 09/04/2017 E.C

GONDAR, ETHIOPIA

ROLE OF THE STUDENT

GROUP MEMBER **ID**

1. AMANUAL AZANAW 02595/15---------------------writes the SQL codes
2. NATNAEL GETNET 01640/15------------------draw ER diagram
3. ASHENAFI HABTE 02264/15------------------- relational model
4. GETAHUN NIGUSSIE 02621/15------------------- data collection
5. ENCHALEW AMSALU 02208/15------------------- normalize the database

Other tasks are performed by all group members together.

CONTENT LIST

PAGE NO.

Chapter 1: Introduction

Chapter 2: Database designing

Student Registration System Project

Chapter 1: Introduction

1.1 Background

The student registration system project aims to revolutionize the process of enrolling students in educational institutions by transitioning from a manual to an automated system. Traditional registration methods often involve cumbersome paperwork, leading to inefficiencies, errors, and delays in processing. The need for a modern solution is evident, as educational institutions require a streamlined process that can handle the complexities of student registration efficiently.

Technology: The system will utilize modern technologies such as relational databases (e.g., MySQL, PostgreSQL), programming languages (e.g., Java, Python), and web frameworks (e.g., Django, Spring). These technologies ensure the system is robust, scalable, and user-friendly.

Title-Based Area: The project focuses on educational institutions, including schools, colleges, and universities. It aims to address the challenges faced by these institutions in managing student registrations.

Problem: The current manual registration process is inefficient, leading to delays, errors, and dissatisfaction among students and staff. There is a lack of real-time access to registration data, making it difficult to manage and track student information.

Proposed Solution: The student registration system will automate the registration process, providing a seamless experience for students and staff. It will offer features such as online registration, real-time data access, and secure data storage..

1.2 Statement of Problem

The current manual registration process is fraught with inefficiencies, leading to significant delays in processing registrations. Errors during data entry can result in incorrect student information, causing frustration for both students and administrative staff. Furthermore, the lack of real-time access to registration data hampers the ability to manage and track student information effectively, making it difficult to respond to inquiries and support student needs.

1.3 Objectives

1.3.1 General Objective

To develop an efficient and user-friendly student registration system that streamlines the registration process and enhances data accuracy.

1.3.2 Specific Objectives

- Automate the registration process, reducing the time and effort required for manual data entry.
- Provide real-time access to registration data, enabling staff to manage and track student information effectively.
- Ensure data security and privacy, protecting sensitive student information from unauthorized access.
- Enhance the user experience by providing a seamless and intuitive interface for students and staff.

1.4 Methodology

1.4.1 Data Collection Technique

Data collection will involve surveys, interviews, and document analysis. Surveys will be conducted with students and staff to gather feedback on the current registration process and identify areas for improvement. Interviews with administrative personnel will provide insights into the challenges faced in managing student registrations. Document analysis will entail reviewing existing registration forms and procedures for further context.

1.4.2 Database Modeling Approach

A relational database model will be employed due to its capability to handle structured data and support complex queries. This approach ensures data integrity and consistency, making it suitable for managing student registration data effectively.

1.4.3 Hardware and Software Requirements

Hardware: The system will require servers for hosting the database and application, as well as computers and network devices to provide access for students and staff.

Software: The implementation will utilize database management systems (e.g., MySQL, PostgreSQL), development tools (e.g., Eclipse, Visual Studio), and operating systems (e.g., Windows, Linux) to facilitate development, deployment, and maintenance.

Chapter 2: Database Designing

2.1 Overview

Database modeling is a critical step in designing the student registration system. It involves creating a structured representation of the data and defining relationships between various data entities. Effective database modeling ensures data integrity, consistency, and efficient data retrieval, which are essential for addressing the problems outlined in the statement of the problem.

2.2 Business Rules/Requirements

The business rules and requirements for the student registration system include:

- **Unique Student ID:** Each student must have a unique identifier to ensure distinct and easily retrievable records.
- **Course Prerequisites:** Courses should have prerequisites to ensure that students meet necessary requirements before enrolling in advanced courses.
- **Multiple Registrations:** Students can register for multiple courses, and each course can have multiple students. This many-to-many relationship must be accurately represented in the database.
- **Secure Data Storage:** Registration data must be stored securely and accessible only to authorized users, implementing access controls and encryption to protect sensitive information.

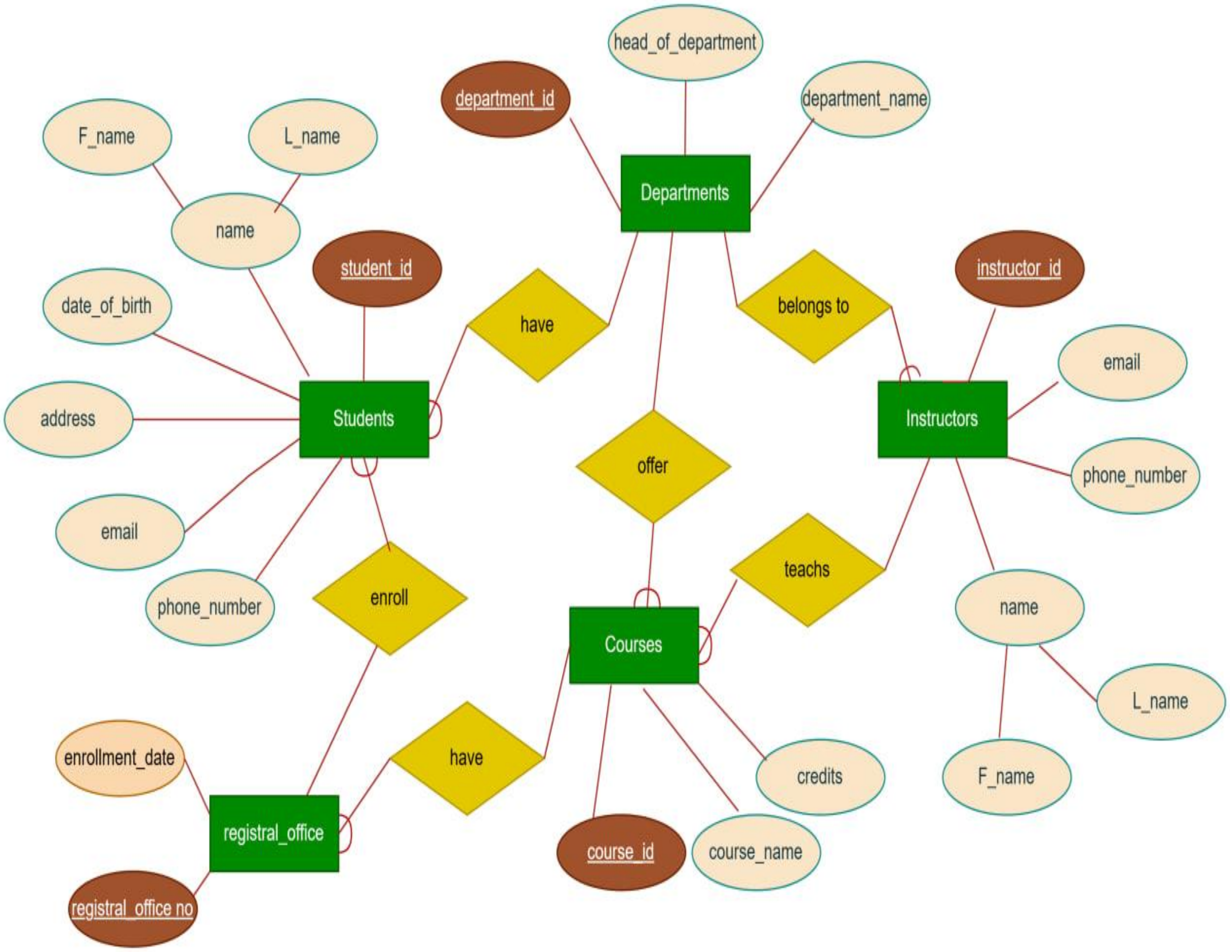
2.3 Building Blocks of the Database

The main components of the database will include:

- **Students Table:** Stores information about students (student ID, name, date of birth, address, email, phone number).
- **Courses Table:** Contains information about courses (course ID, course name, credits, department ID, instructor ID).
- **Registrar office Table:** Records student registrations (Registrar office no, student ID, course ID, Enrollment date).
- **Instructors Table:** Stores information about instructors (instructor ID, name, department ID, email, phone number).
- **Departments Table:** Contains information about departments (department ID, department name, head of department).

2.4 ER Diagram

An Entity-Relationship (ER) diagram visually represents the database structure, illustrating entities like Students, Courses, Registrations, Instructors, and Departments, along with their relationships.



2.5 Relational Model

The relational model maps the ER diagram to a set of tables and relationships:

Department relation

Department id	Department name	Department head
1	Computer science	Dr. Sisay
2	mathematics	Dr. Jone
3	physics	Dr. Belay

Instructor relation

Instructor id	Name	Department id	Email	Phone number
101	Abebe Worku	1	Abebe.wiams@gmail.com	+251 908765678
201	Belay Mulugeta	2	Belay.mr@ gmail.com	+251 908765456
323	Chale Dawit	3	Chale.dav@ gmail.com	+251 987654569

Student relation

Student id	name	Date of birth	Gender	address	Phone_number	Email	Enrollment date	Department id
145	Jone Kebede	1995/01/01	M	Gondar	+251 945678967	John.doe@gmail.com	2017/09/01	1
245	Beti Sisay	1996/02/07	F	Mota	+251 956789012	Beti.th@gmail.com	2017/09/03	2
364	Emebet Adise	1994/03/03	F	Debark	+251 967890123	Embet.son@gmail.com	2017/09/05	3

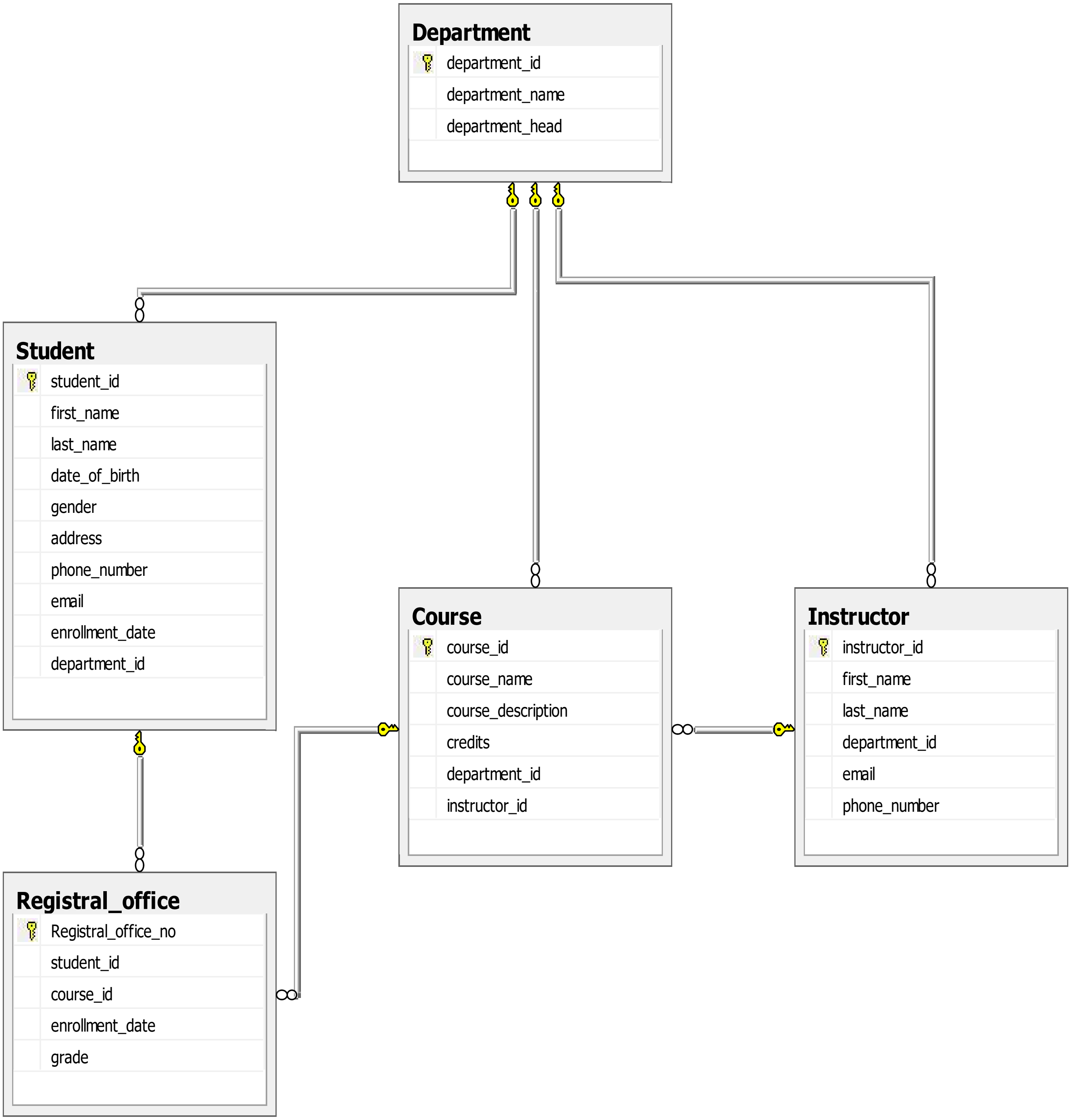
Course relation

Course id	Course name	Course description	credit	Department id	Instructor id
167	Introduction to programming	Basic of programming	3	1	101
309	Database	Introduction to DBMS	4	2	323
412	Calculus I	Introduction to differential and integral calculus	4	3	201

Registrar office

Registrar office no	Student id	Course id	Enrollment date	grade
190	145	167	2017/09/01	A
209	245	412	2017/09/03	C
312	354	309	2017/09/05	B

Relational diagrams



2.6 Normalization

The database will be normalized to reduce redundancy and improve data integrity through the following steps:

- **First Normal Form (1NF):** Ensure each table has a primary key and that each column contains atomic values.

Department relation

Department id	Department name	Department head
1	Computer science	Dr. Sisay
2	mathematics	Dr. Jone
3	physics	Dr. Belay

Course relation

Course id	Course name	Course description	credit	Department id	Instructor id
167	Introduction to programming	Basic of programming	3	1	101
309	Database	Introduction to DBMS	4	2	323
412	Calculus I	Introduction to differential and integral calculus	4	3	201

Registrar office

Registrar office no	Student id	Course id	Enrollment date	grade
190	145	167	2017/09/01	A
209	245	412	2017/09/03	C
312	354	309	2017/09/05	B

They are in First Normal Form because each attribute contains atomic values.

Instructor relation

Instructor id	Name	Department id	Email	Phone number
101	Abebe Worku	1	Abebe.wiams@gmail.com	+251 908765678
201	Belay Mulugeta	2	Belay.mr@ gmail.com	+251 908765456
323	Chale Dawit	3	Chale.Dav@ gmail.com	+251 987654569

It is in not in First Normal Form because it contains composite value we need to decompose it to be in 1NF.like below

Instructor id	F name	L name	Department id	Email	Phone number
101	Abebe	Worku	1	Abebe.wiams@gmail.com	+251 908765678
201	Belay	Mulugeta	2	Belay.mr@ gmail.com	+251 908765456
323	Chale	Dawit	3	Chale. dav@ gmail.com	+251 987654569

Students relation

Student id	name	Date of birth	Gender	address	Phone_number	Email	Enrollment date	Department id
145	Jone Kebede	1995/01/01	M	Gondar	+251 945678967	John.doe@gmail.com	2017/09/01	1
245	Beti Sisay	1996/02/07	F	Mota	+251 956789012	Beti.th@gmail.com	2017/09/03	2
364	Emebet Adise	1994/03/03	F	Debark	+251 967890123	Embet.son@gmail.com	2017/09/05	3

It is in not in First Normal Form because it contains composite value we need to decompose it to be in 1NF.like below

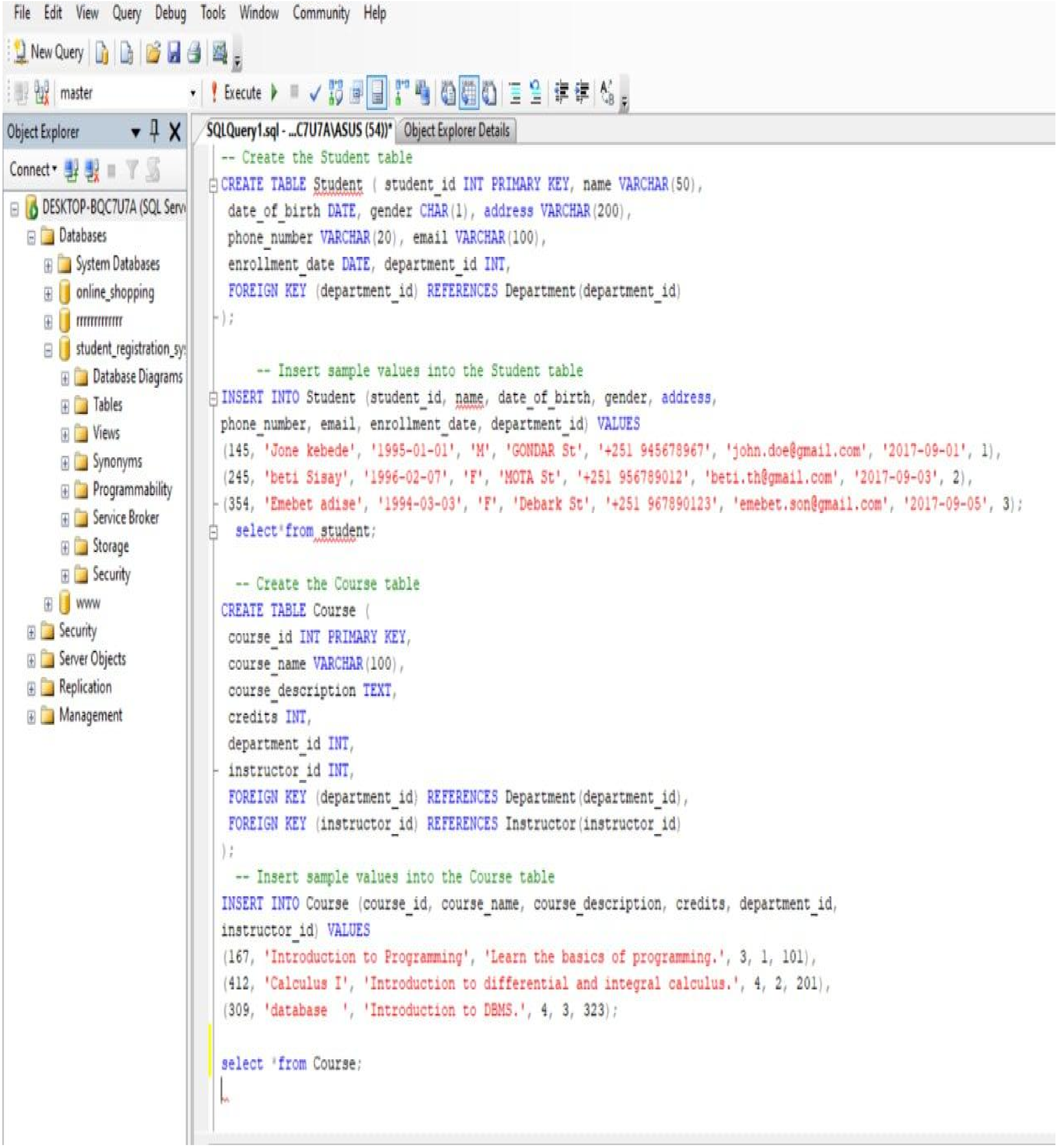
Student id	F name	L name	Date of birth	Gender	address	Phone number	Email	Enrollment date	Department id
145	Jone	Kebede	1995/01/01	M	Gondar	+251 945678967	John.doe@gmail.com	2017/09/01	1
245	Beti	Sisay	1996/02/07	F	Mota	+251 956789012	Beti.th@gmail.com	2017/09/03	2
364	Emebet	Adise	1994/03/03	F	Debark	+251 967890123	Embet.son@gmail.com	2017/09/05	3

- **Second Normal Form (2NF):** Remove partial dependencies while ensuring the table is in 1NF.

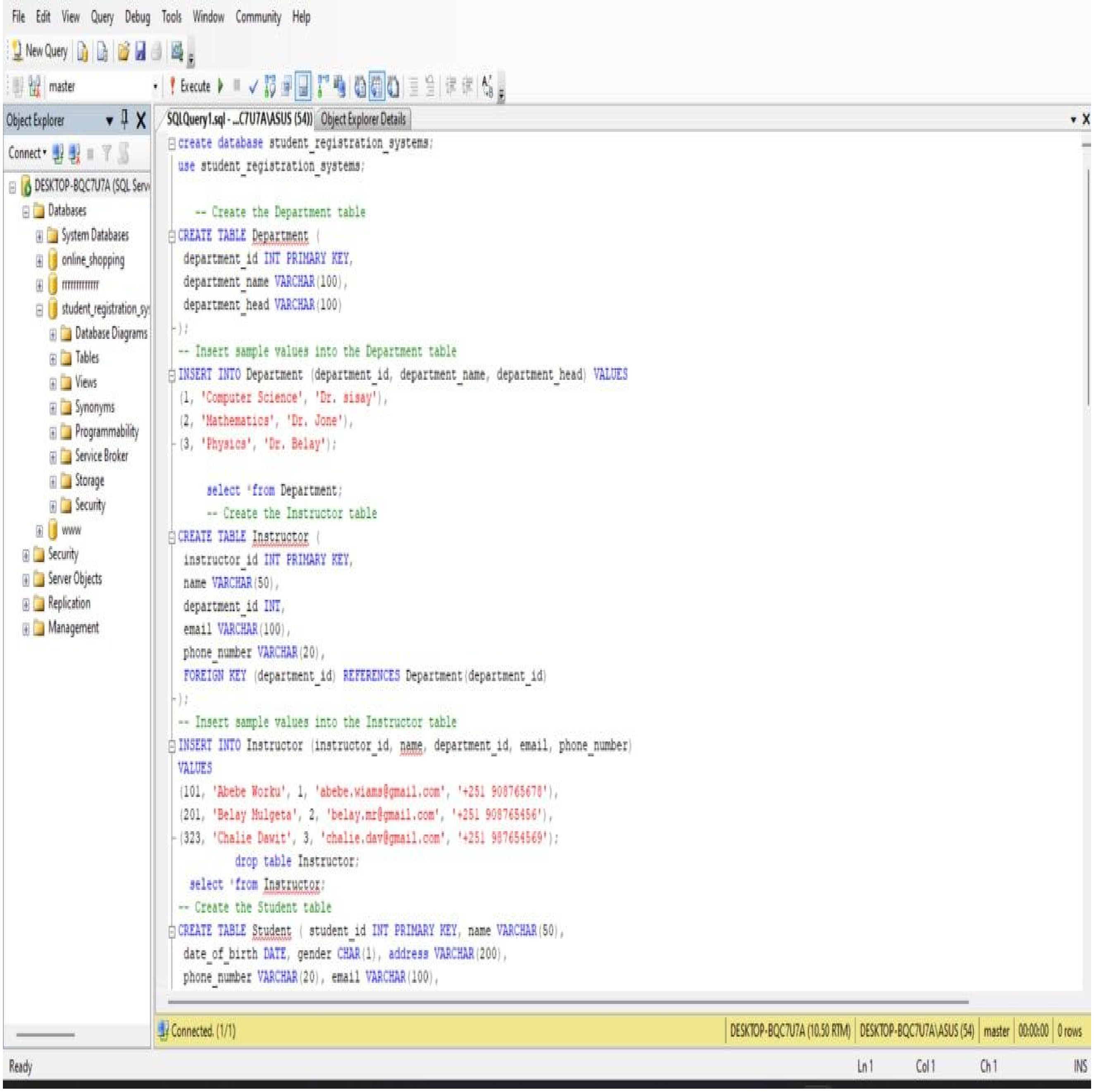
The above relations are all in 2NF because there are in 1NFand there is no partial dependency on them

Chapter 3: Database Implementation

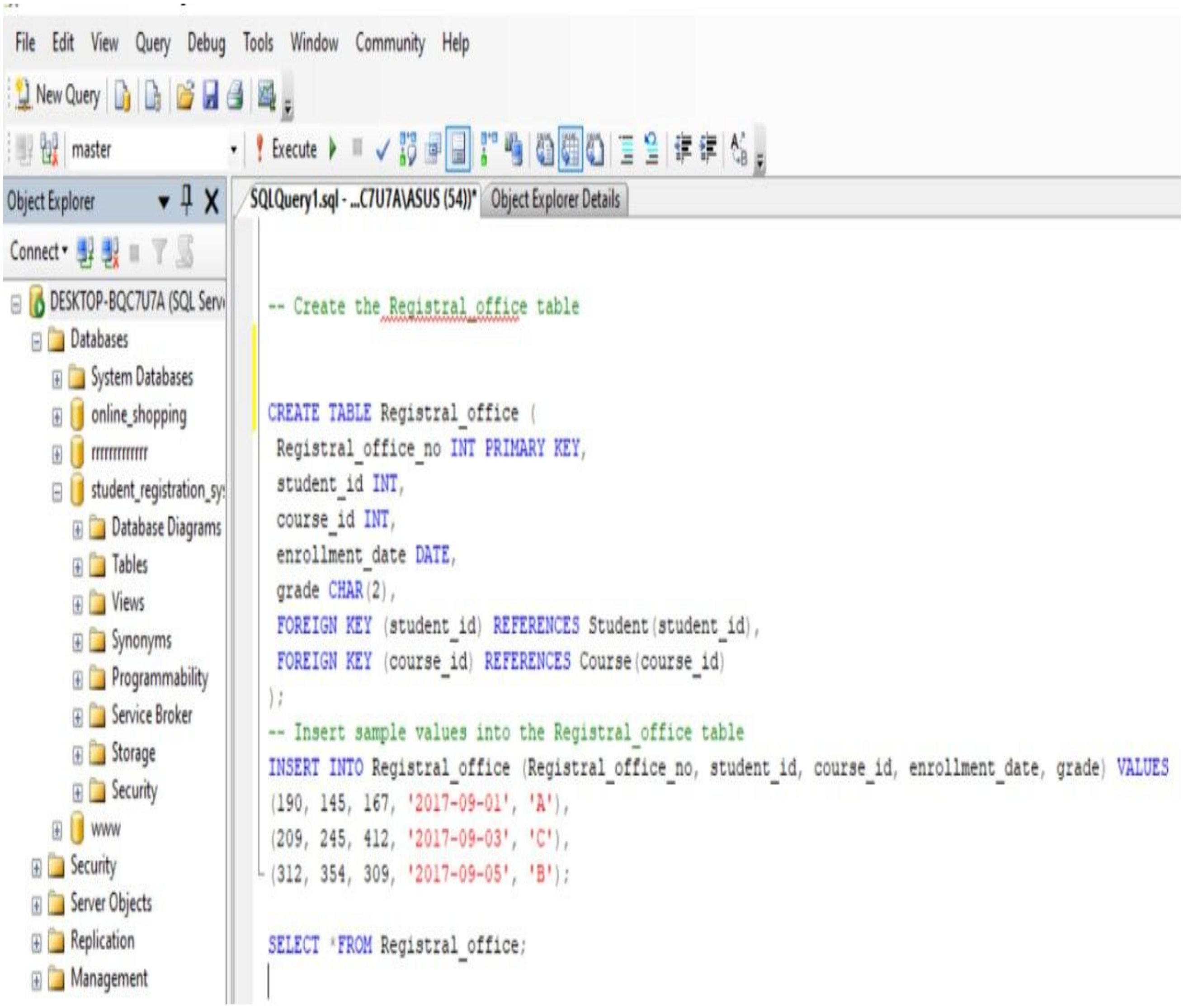
SQL Commands



```sql
-- Create the Student table
CREATE TABLE Student ( student_id INT PRIMARY KEY, name VARCHAR(50),
 date_of_birth DATE, gender CHAR(1), address VARCHAR(200),
 phone_number VARCHAR(20), email VARCHAR(100),
 enrollment_date DATE, department_id INT,
 FOREIGN KEY (department_id) REFERENCES Department(department_id)
);


     -- Insert sample values into the Student table
INSERT INTO Student (student_id, name, date_of_birth, gender, address,
phone_number, email, enrollment_date, department_id) VALUES
(145, 'Jone kebede', '1995-01-01', 'M', 'GONDAR St', '+251 945678967', 'john.doe@gmail.com', '2017-09-01', 1),
(245, 'beti Sisay', '1996-02-07', 'F', 'MOTA St', '+251 956789012', 'beti.th@gmail.com', '2017-09-03', 2),
(354, 'Emebet adise', '1994-03-03', 'F', 'Debark St', '+251 967890123', 'emebet.son@gmail.com', '2017-09-05', 3);
  select*from student;

  -- Create the Course table
CREATE TABLE Course (
 course_id INT PRIMARY KEY,
 course_name VARCHAR(100),
 course_description TEXT,
 credits INT,
 department_id INT,
 instructor_id INT,
 FOREIGN KEY (department_id) REFERENCES Department(department_id),
 FOREIGN KEY (instructor_id) REFERENCES Instructor(instructor_id)
);
  -- Insert sample values into the Course table
INSERT INTO Course (course_id, course_name, course_description, credits, department_id,
instructor_id) VALUES
(167, 'Introduction to Programming', 'Learn the basics of programming.', 3, 1, 101),
(412, 'Calculus I', 'Introduction to differential and integral calculus.', 4, 2, 201),
(309, 'database  ', 'Introduction to DBMS.', 4, 3, 323);

select *from Course;
```

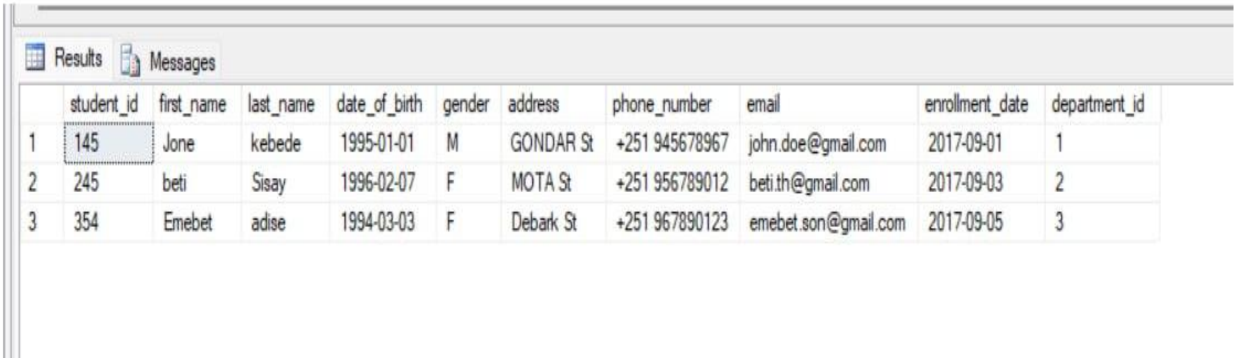


```sql
create database student_registration_systems;
use student_registration_systems;

   -- Create the Department table
CREATE TABLE Department (
 department_id INT PRIMARY KEY,
 department_name VARCHAR(100),
 department_head VARCHAR(100)
);
-- Insert sample values into the Department table
INSERT INTO Department (department_id, department_name, department_head) VALUES
(1, 'Computer Science', 'Dr. sisay'),
(2, 'Mathematics', 'Dr. Jone'),
(3, 'Physics', 'Dr. Belay');

     select *from Department;
     -- Create the Instructor table
CREATE TABLE Instructor (
 instructor_id INT PRIMARY KEY,
 name VARCHAR(50),
 department_id INT,
 email VARCHAR(100),
 phone_number VARCHAR(20),
 FOREIGN KEY (department_id) REFERENCES Department(department_id)
);
-- Insert sample values into the Instructor table
INSERT INTO Instructor (instructor_id, name, department_id, email, phone_number)
VALUES
(101, 'Abebe Worku', 1, 'abebe.wiams@gmail.com', '+251 908765678'),
(201, 'Belay Mulgeta', 2, 'belay.mr@gmail.com', '+251 908765456'),
(323, 'Chalie Dawit', 3, 'chalie.dav@gmail.com', '+251 987654569');
          drop table Instructor;
  select *from Instructor;
-- Create the Student table
CREATE TABLE Student ( student_id INT PRIMARY KEY, name VARCHAR(50),
 date_of_birth DATE, gender CHAR(1), address VARCHAR(200),
 phone_number VARCHAR(20), email VARCHAR(100),
```



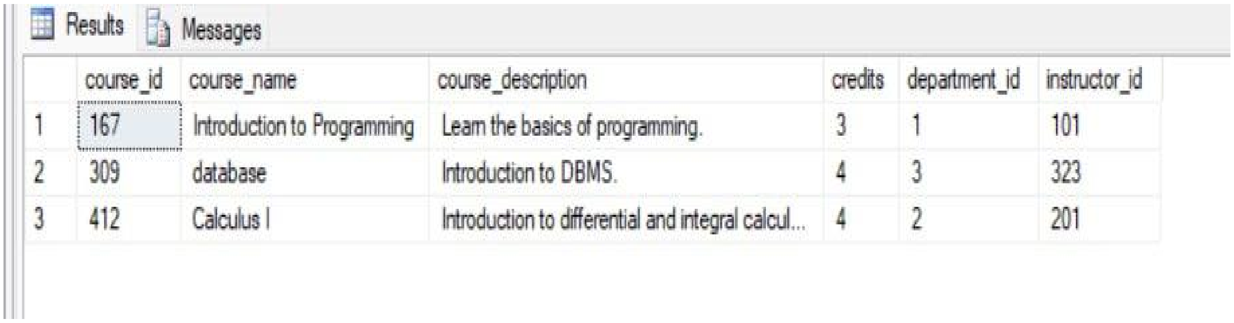
```
-- Create the Registral_office table


CREATE TABLE Registral_office (
 Registral_office_no INT PRIMARY KEY,
 student_id INT,
 course_id INT,
 enrollment_date DATE,
 grade CHAR(2),
 FOREIGN KEY (student_id) REFERENCES Student(student_id),
 FOREIGN KEY (course_id) REFERENCES Course(course_id)
);
-- Insert sample values into the Registral_office table
INSERT INTO Registral_office (Registral_office_no, student_id, course_id, enrollment_date, grade) VALUES
(190, 145, 167, '2017-09-01', 'A'),
(209, 245, 412, '2017-09-03', 'C'),
(312, 354, 309, '2017-09-05', 'B');

SELECT *FROM Registral_office;
```

Student relation



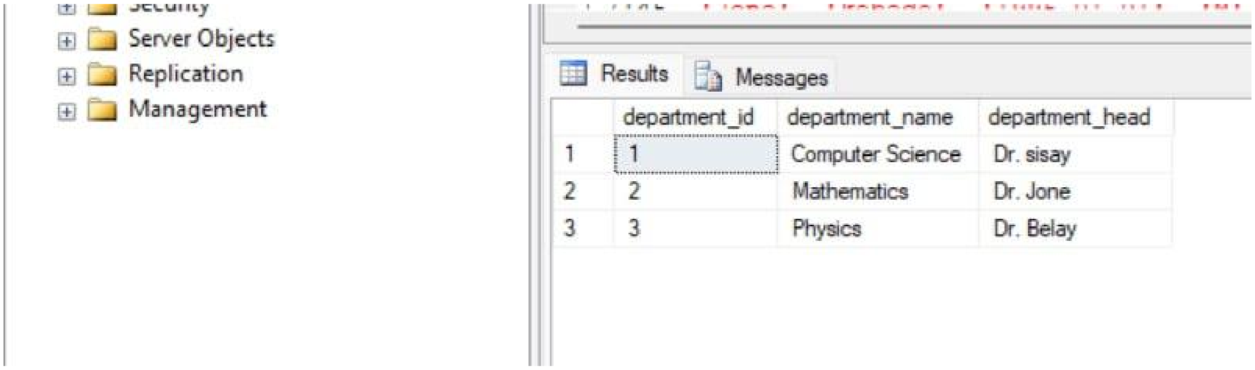
	student_id	first_name	last_name	date_of_birth	gender	address	phone_number	email	enrollment_date	department_id
1	145	Jone	kebede	1995-01-01	M	GONDAR St	+251 945678967	john.doe@gmail.com	2017-09-01	1
2	245	beti	Sisay	1996-02-07	F	MOTA St	+251 956789012	beti.th@gmail.com	2017-09-03	2
3	354	Emebet	adise	1994-03-03	F	Debark St	+251 967890123	emebet.son@gmail.com	2017-09-05	3

Course relation



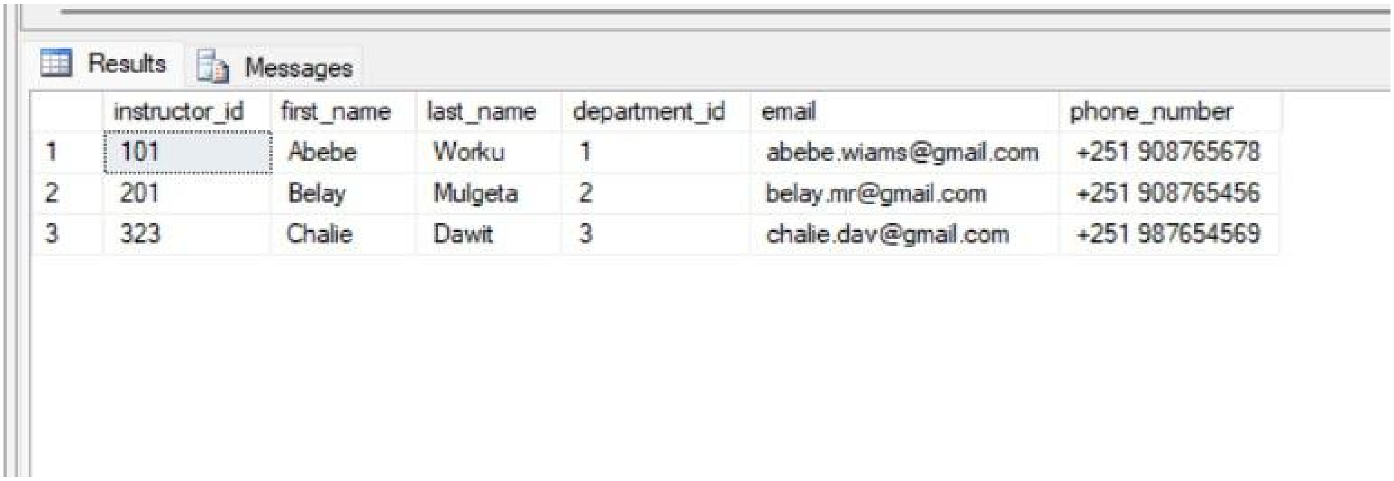
	course_id	course_name	course_description	credits	department_id	instructor_id
1	167	Introduction to Programming	Learn the basics of programming.	3	1	101
2	309	database	Introduction to DBMS.	4	3	323
3	412	Calculus I	Introduction to differential and integral calcul...	4	2	201

Department table



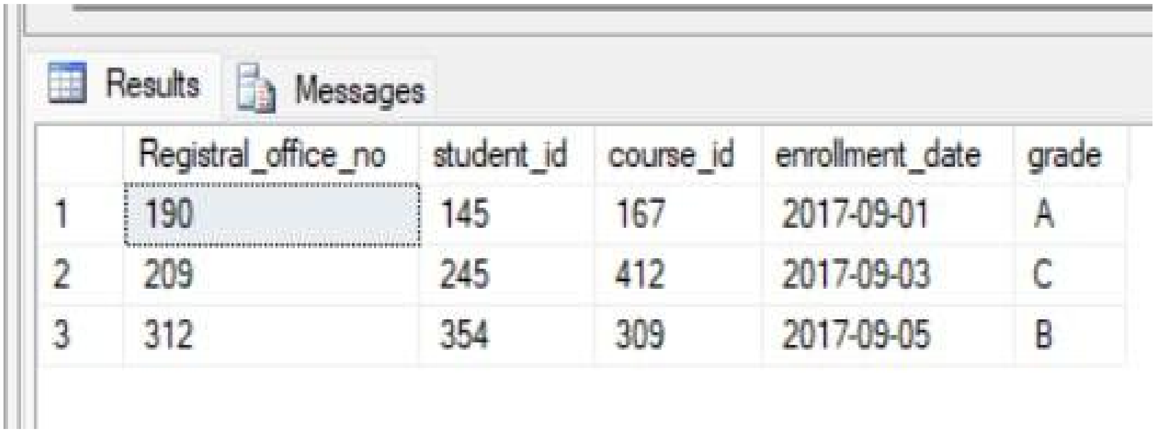
	department_id	department_name	department_head
1	1	Computer Science	Dr. sisay
2	2	Mathematics	Dr. Jone
3	3	Physics	Dr. Belay

Instructor relation



	instructor_id	first_name	last_name	department_id	email	phone_number
1	101	Abebe	Worku	1	abebe.wiams@gmail.com	+251 908765678
2	201	Belay	Mulgeta	2	belay.mr@gmail.com	+251 908765456
3	323	Chalie	Dawit	3	chalie.dav@gmail.com	+251 987654569

Registral office



	Registral_office_no	student_id	course_id	enrollment_date	grade
1	190	145	167	2017-09-01	A
2	209	245	412	2017-09-03	C
3	312	354	309	2017-09-05	B