IN
PARTNERSHIP
WITH
PLYMOUTH
UNIVERSITY

Name: Dewmith Kiriella

Student Reference Number: 10748147

Module Code: **PUSL2019**	Module Name: **Information Management & Retrieval**

Coursework Title: **Coursework 2021-2022 Group Project - Group 26**

Deadline Date: **14-01-2022**	Member of staff responsible for coursework: **Mr. Naji Saravanabhavan**

Programme: **BSc (Hons) Software Engineering, BSc (Hons) Computer Science**

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

Dewmith Thenura Kiriella	- 10748147
Mohamed Yesneen Mohamed Yusry	- 10749082
Singhalage Matheesha Akash Dharmasena	- 10749195
Edirisinghe Appuhamillage Yenuka Indrajith	- 10749143
Passikku Hannadige Nipuni Kavindya	- 10748162
Mahindu Poorna Vithanage Bandaranayake	- 10749841

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group: Kiriella

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed :

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software………………………………………………………………

Overall mark _____% **Assessors Initials ______** **Date_________**

MANAGEMENT AND RETRIEVAL

STUDENT MANAGEMENT SYSTEM

Abstract

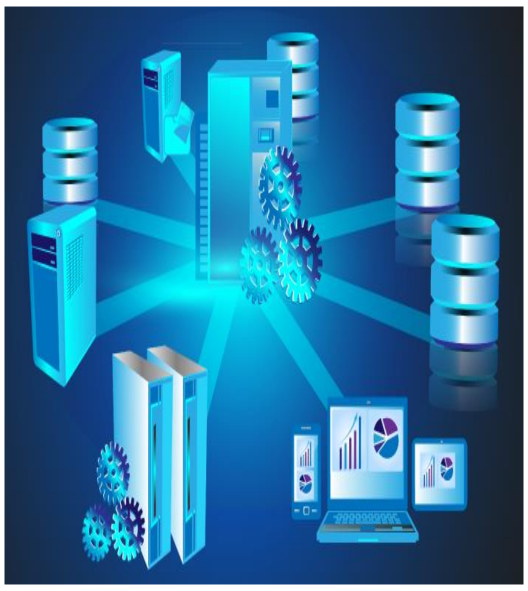
TABLES CREATED USING THE MICROSOFT SQL SERVER MANAGEMENT

GROUP PROJECT- GROUP NO 26
TEAM LEADER:DEWMITH KIRIELLA

PUSL 2019 INFORMATION MANAGEMENT AND RETRIEVAL

BATCH 09

STUDENT MANAGEMENT SYSTEM



PREPARED BY

DT KIRIELLA, MPV BANDARANAYAKE, MYM YUSRY, EAYI EDIRISINGHE, PHN KAVINDYA, SMA DHARMASENA

SUPERVISED BY

MR. SARAVANABAVAN NASIKETHA

TABLE OF CONTENTS

TOPIC	PAGE
INTRODUCTION	4
ER DIAGRAM	5
RELATIONAL MAPPING	7
NORMALIZATION	8
DATA DICTIONARY	13
Table Creation SQL Commands	17
Insert Data Into Tables Queries	24
Database Diagram	26
SQL Triggers	27
SQL Views	29
SQL Procedures	31
SQL Functions	33
Critical Evaluation	34

INTRODUCTION

SCENARIO

Students at the university are required to fill out forms and register for exams in order to face them. It is required to submit the completed forms to the ‘Examination Department’ once the students have handed them over to the ‘Registration Department’ on or before the due date. All the students must be enrolled to a particular degree program either to a local degree or for an affiliated program. Exams will be conducted either physically or online based on the prevailing situation in the country. All the students will receive exam papers relevant to the study year and according to the paper structure given by the foreign universities. University provides library facility to the students, students can either visit the online store which is visible in the ‘Learning Management System’ or visit physically to the library at the university premises. In order to gain access to the ‘Library’ privileges, the students are required to obtain the membership card of the library. Also, the university provides free Wi-Fi for the students once students enter to the university premises, every faculty is providing free Wi-Fi. Through the Learning Management System, users can view the specific degree program and all the essential details including the relevant modules as well. All the lecture materials are also accessible through the system and submission too can be done through the platform. Admins of the system will update the course materials, timetables, announcements, exam schedules and other necessary things as well.

Surrounding- University

1. Students must be registered to a particular degree program
2. Also, to face the examinations, students are required to register under the Examination Department
3. The registration details will be confirmed by the ‘Administration Department’

Important factors to be considered

- Students must be enrolled to face an exam
- Exams are held for almost for each and every module, some modules may be there with a different evaluation criteria
- Modules taught in the university will based on the course chosen by the students
- There are variety of courses available in the university some programs are affiliated programs as well and others are the local degrees offered by the university itself, where the major difference is the affiliated programs will be conducted for only 3 years’ time period while the local degree are up to 4 years’ time period
- Each and every student must belong to a faculty
- Each faculty contains sub departments where the department represent the relevant authorities accountable for each and every course offering in the university
- Lecturers deliver the module content
- Each and every lecturer belong to a specific faculty based on the specialization he/she has
- All the students will receive exam results
- Student’s profile must contain details of a guardian person
- All the students given the Wi-Fi facility
- Through the Wi-Fi students are able to access the online library repository
- Admins, lecturer and students are given the opportunity to access the Learning Management System
- Admins who have access to the LMS and the Student Management System has different job roles
- Admins who are in charge in the library section handles the fine payments
- Book details needs to be mentioned
- Books are being borrowed by students
- Students are required to enter through the main gate
- Students must login with the correct credentials to gain access to the LMS
- Also, admins and lecturers too can use the Wi-Fi facility

Entities and Attributes relevant to the scenario

Gate - Gate_Id, Date, Attendance

Wi-Fi - Std_pwd, Std_Username, Security, Sub_net_mask, Router, IP_Address, Signal_Strength, Mac, Connection_speed

Registration - Enroll_No, Std_Name

Admin – A_Id, A_Name, A_Pwd, Job_Role, A_Username

Exam – E_Id, Attempt_No, GPA, Year_of_Study, Class, Index_No, Grade

Student – Std_Id, Std_FName, Std_LName, Std_Address, Std_DOB, Pre-requisities, Contact-No, Std_PersonalEmail, Std_UniversityEmail, Age

Faculty – Fac_Id, Fac_Desc, Fac_Dean, Fac_Name

Department – Dep_No, Dep_Name, Dep_HeadsName, Location

Lecturer – Lec-Id, Lec_FName, Lec_LName, Lec_Username, Specialization, Category, Contact_No, Lec_PersonalEmail, Lec_UniversityEmail

Course – C_Id, C_Name, Syllabus, Credits, Duration, Offered_University, Fee

Module – M_Code, M_Name, Evaluation_type

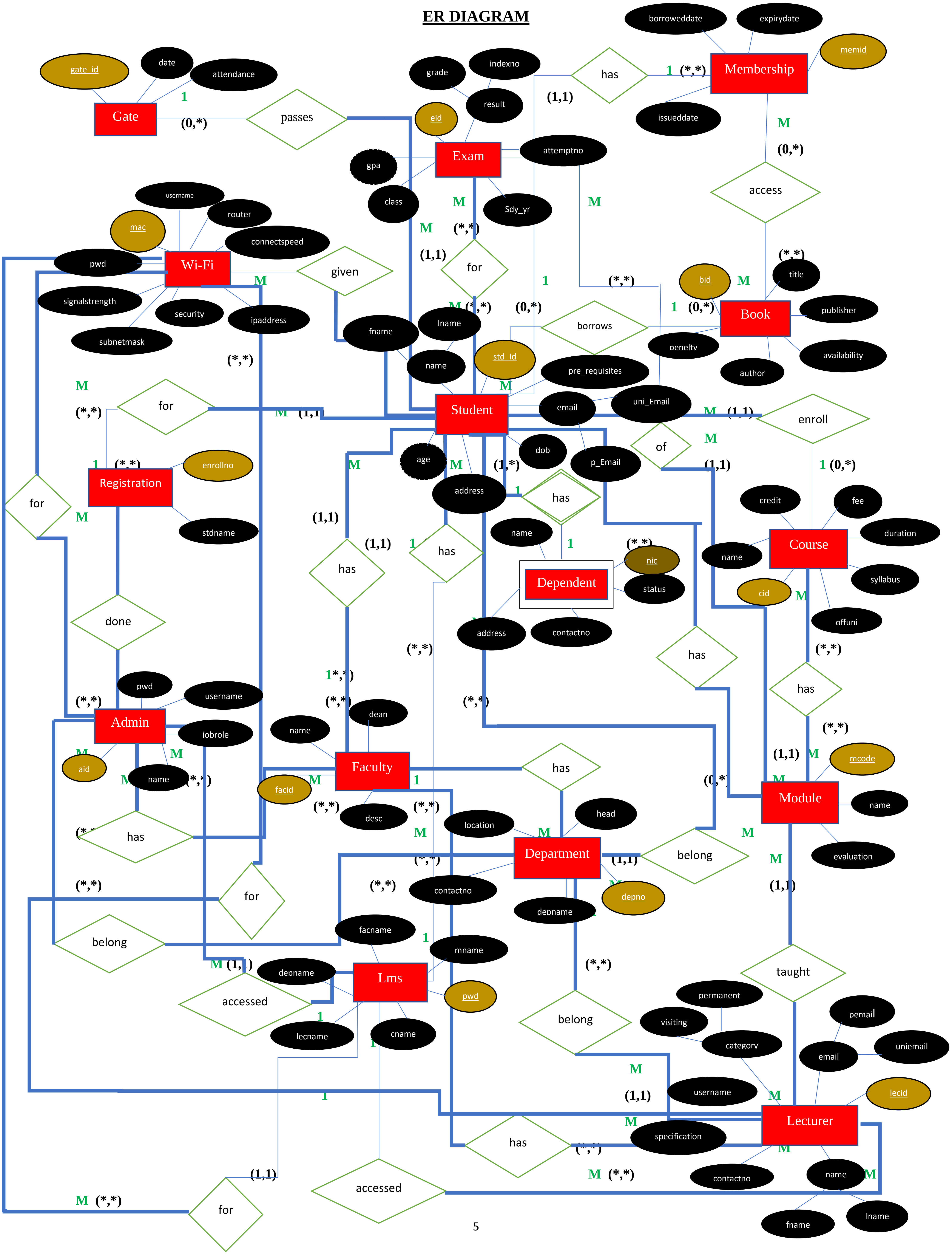
LMS – Dep_Name, Fac_Name, Lec_Name, C_Name, M_Name, Password, Username

Membership – Mem_Id, Issued-Date, Borrowed_Date, Expiry_Date

Book – Book_id, B_title, B_Author, Availability, Penalty, Publisher

Dependent – Name, Address, Contact_No, Relationship_Status, NIC_No

ER DIAGRAM



borroweddate
expirydate
memid
Membership
has
1 (*,*)
(1,1)
issueddate
M
(0,*)
access
(*,*)
title
gate_id
date
attendance
Gate
1
(0,*)
passes
grade
indexno
result
eid
Exam
attemptno
gpa
class
Sdy_yr
M
M
(*,*)
(1,1)
for
M (*,*)
(0,*)
1
(*,*)
bid
M
Book
publisher
availability
author
peneltv
1 (0,*)
borrows
username
router
mac
connectspeed
pwd
Wi-Fi
M
given
signalstrength
security
ipaddress
subnetmask
fname
lname
name
std_Id
pre_requisites
M
uni_Email
email
Student
(*,*)
M
(*,*)
for
M (1,1)
M (1,1)
enroll
of
M
(1,1)
1 (0,*)
1 (*,*)
Registration
enrollno
M
stdname
for
age
M
M
address
(1,*)
dob
p_Email
has
1
name
1
(1,1)
(1,1)
has
1
has
(*,*)
nic
Dependent
status
address
contactno
credit
fee
duration
Course
name
syllabus
cid
M
offuni
(*,*)
has
has
done
(*,*)
(*,*)
1*,*)
(*,*)
(*,*)
pwd
username
Admin
jobrole
M
aid
name
M
M
(*,*)
dean
name
Faculty
facid
M
1
has
(*,*)
desc
(*,*)
has
(0,*)
(1,1)
M
mcode
M
Module
name
evaluation
location
head
M
Department
(1,1)
belong
M
M
(1,1)
(*,*)
contactno
(*,*)
depname
depno
for
belong
facname
1
mname
M (1,1)
depname
Lms
pwd
accessed
1
lecname
cname
1
(*,*)
belong
taught
permanent
visiting
category
pemail
email
uniemail
M
(1,1)
username
lecid
M
Lecturer
1
has
M
specification
(*,*)
M
contactno
name
M
(1,1)
M (*,*)
accessed
M (*,*)
for
fname
lname

Assumptions

- The main gate provide access to all the students, also the presence is marked at that point
- All the students must face exam and pass in order to progress further
- All the students as well as the staff members (Admins and Lecturers) will receive free Wi-Fi facility
- All the students must register to face an exam
- All the admins who belong to a certain department under a specific degree program is in charge of enrolling students for the examinations
- All the students will receive exam results
- Students must be enrolled to a degree program offered by the university out of the available list
- One course may contain similar modules under a specific degree program and all the modules must belong to a certain degree program
- All the lecturers must belong to a department
- All the students must belong to a particular faculty
- There will be many departments in a faculty
- All the modules are being conducted by lecturers; one lecturer can teach more than one module based on the year of study
- Many students can borrow the same book in the library, while one student can borrow a variety of books
- Students may have a membership card of the library or not
- Each and every course might not contain exams for every module, there may be few modules with a different evaluation type
- Each and every student had access to the LMS, all the students must use the LMS privilege for assignment submission and for other important activities
- Admins must maintain the LMS
- Through the Wi-Fi facility students and lecturers can login to the LMS
- Lecturers who belong to a department must be under a certain faculty
- All the students must be registered under a certain course (Affiliated or Local)
- There may be some courses with no student enrollment
- All the students must have at least one guardian
- Also, admins too belong to a certain department under a particular faculty

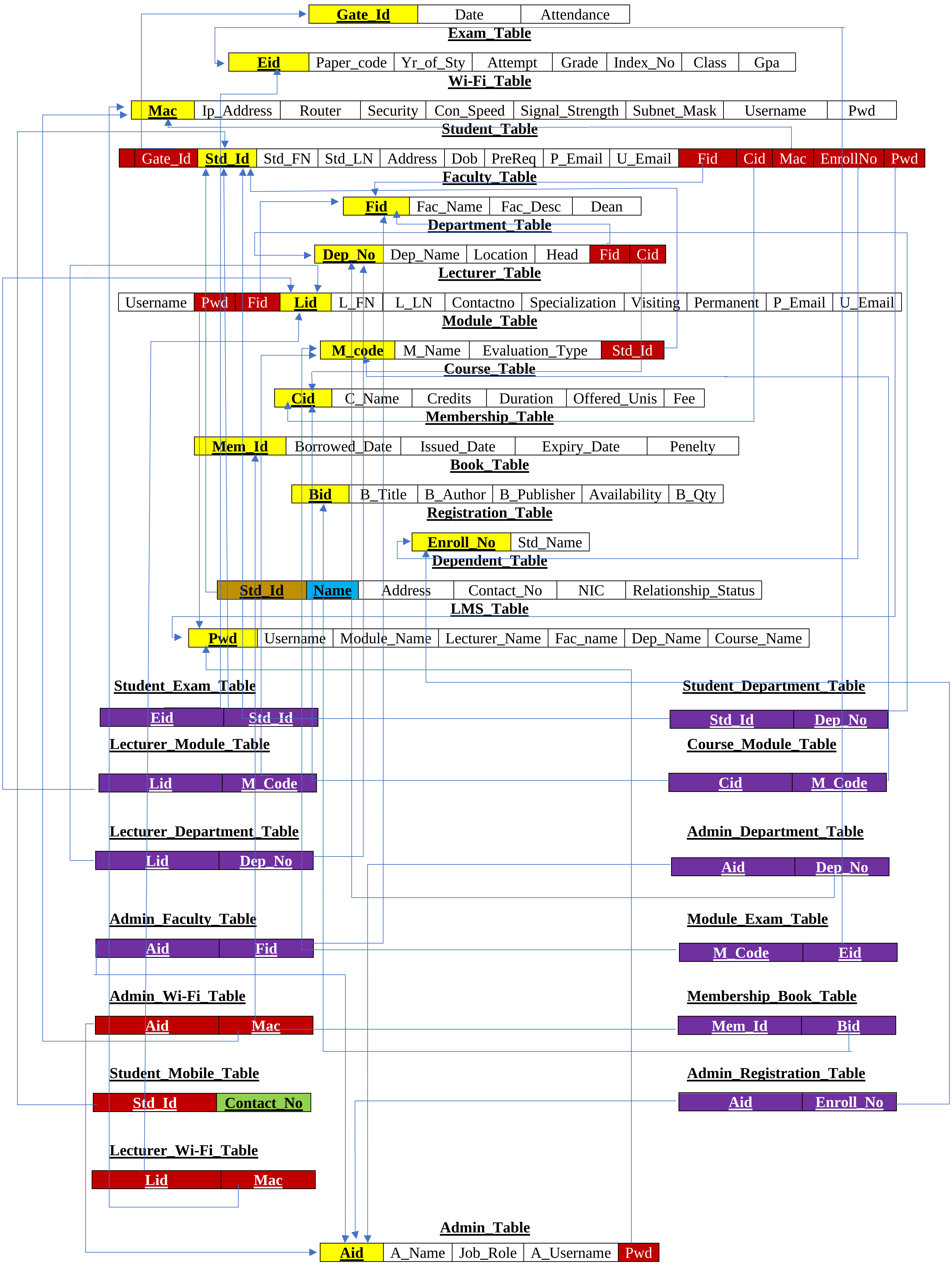
━━━ = Total Participation **─── = Partial Participation**

⬮ = Primary Key **⬮ = Partial Key**

Composite Primary Key 🟪 Partial 🟫 Weak 🟦

RELATIONAL MAPPING

🟨 PK 🟥 FK 🟩 Multi-Valued



Entrance_Table

Gate_Id	Date	Attendance

Exam_Table

Eid	Paper_code	Yr_of_Sty	Attempt	Grade	Index_No	Class	Gpa

Wi-Fi_Table

Mac	Ip_Address	Router	Security	Con_Speed	Signal_Strength	Subnet_Mask	Username	Pwd

Student_Table

Gate_Id	**Std_Id**	Std_FN	Std_LN	Address	Dob	PreReq	P_Email	U_Email	Fid	Cid	Mac	EnrollNo	Pwd

Faculty_Table

Fid	Fac_Name	Fac_Desc	Dean

Department_Table

Dep_No	Dep_Name	Location	Head	Fid	Cid

Lecturer_Table

Username	Pwd	Fid	**Lid**	L_FN	L_LN	Contactno	Specialization	Visiting	Permanent	P_Email	U_Email

Module_Table

M_code	M_Name	Evaluation_Type	Std_Id

Course_Table

Cid	C_Name	Credits	Duration	Offered_Unis	Fee

Membership_Table

Mem_Id	Borrowed_Date	Issued_Date	Expiry_Date	Penelty

Book_Table

Bid	B_Title	B_Author	B_Publisher	Availability	B_Qty

Registration_Table

Enroll_No	Std_Name

Dependent_Table

Std_Id	**Name**	Address	Contact_No	NIC	Relationship_Status

LMS_Table

Pwd	Username	Module_Name	Lecturer_Name	Fac_name	Dep_Name	Course_Name

Student_Exam_Table

Eid	**Std_Id**

Student_Department_Table

Std_Id	**Dep_No**

Lecturer_Module_Table

Lid	**M_Code**

Course_Module_Table

Cid	**M_Code**

Lecturer_Department_Table

Lid	**Dep_No**

Admin_Department_Table

Aid	**Dep_No**

Admin_Faculty_Table

Aid	**Fid**

Module_Exam_Table

M_Code	**Eid**

Admin_Wi-Fi_Table

Aid	**Mac**

Membership_Book_Table

Mem_Id	**Bid**

Student_Mobile_Table

Std_Id	**Contact_No**

Admin_Registration_Table

Aid	**Enroll_No**

Lecturer_Wi-Fi_Table

Lid	**Mac**

Admin_Table

Aid	A_Name	Job_Role	A_Username	Pwd

NORMALIZATION

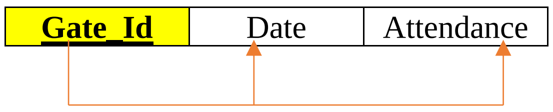
1ST NORMALIZATION FORM

- In the 1st Normalization form all the multivalued attributes, nested relations, composite attributes in a single individual tuple must be removed
- Multivalued attributes are taken to another new table along with the primary key of the desired entity in the relationship
- Tables above do not consist of any nested relations or composite attributes in a single row, therefore all the tables can be moved towards the 2nd Normalization
- Table below was taken to relational mapping according to the step 6 where the multivalued attributes are taken to a new table along with the primary key of the other entity and this was already normalized during the process since the table do not consist of any data
- Therefore 1st Normalization isn't required for the above tables

Std_Id	**Contact_No**

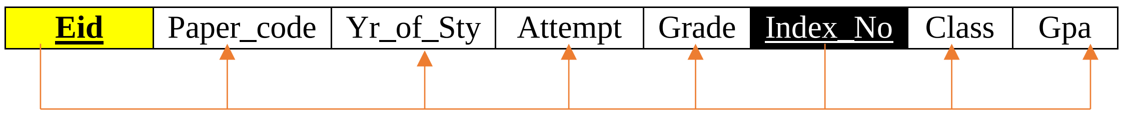
Identifying Partial, Full-Functional and Transitive Dependencies

Entrance_Table ✓



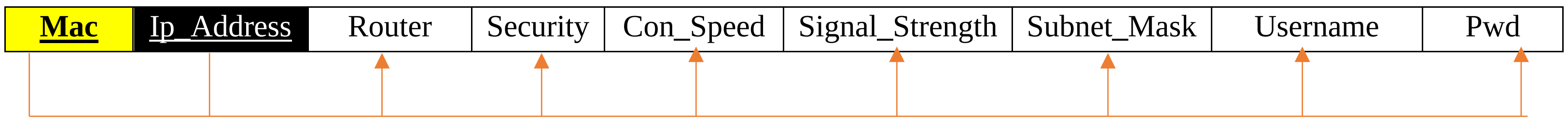
Gate_Id	Date	Attendance

Exam_Table ✓



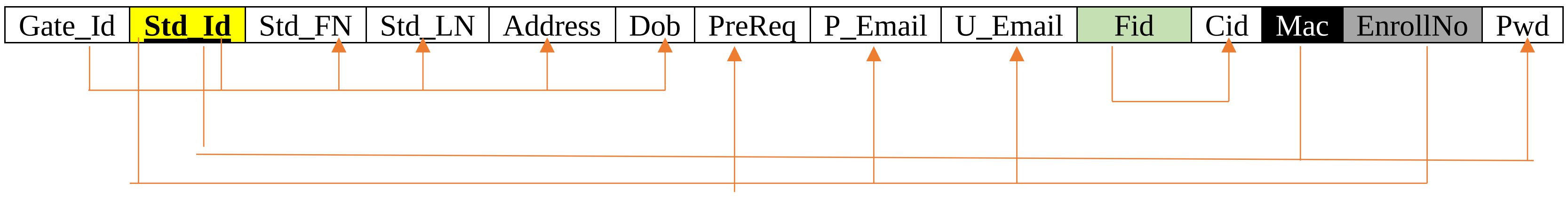
Eid	Paper_code	Yr_of_Sty	Attempt	Grade	Index_No	Class	Gpa

Wi-Fi_Table ✓



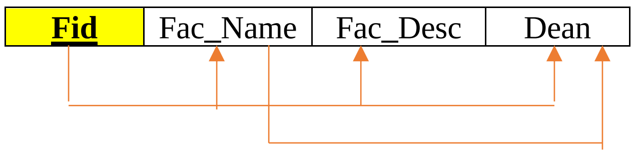
Mac	Ip_Address	Router	Security	Con_Speed	Signal_Strength	Subnet_Mask	Username	Pwd

Student_Table (2nd Normalization is required)



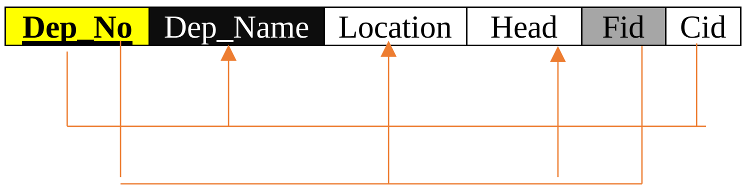
Gate_Id	**Std_Id**	Std_FN	Std_LN	Address	Dob	PreReq	P_Email	U_Email	Fid	Cid	Mac	EnrollNo	Pwd

Faculty_Table (2nd and 3rd Normalization is required)



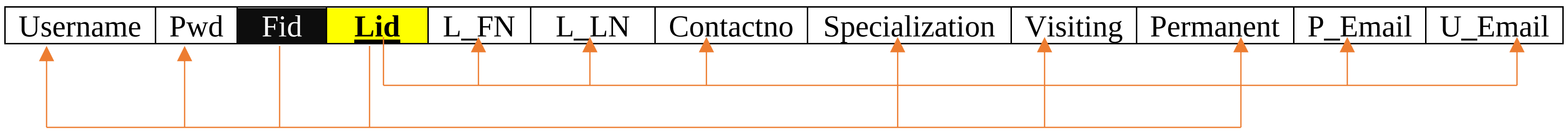
Fid	Fac_Name	Fac_Desc	Dean

Department_Table (2nd Normalization is required)



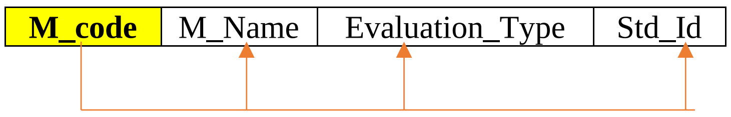
Dep_No	Dep_Name	Location	Head	Fid	Cid

Lecturer_Table (2nd Normalization is required)



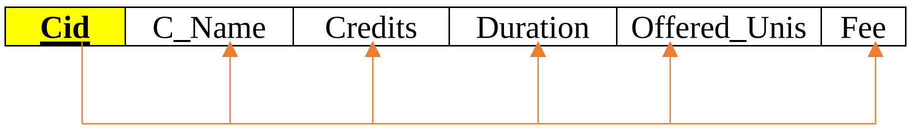
Username	Pwd	Fid	**Lid**	L_FN	L_LN	Contactno	Specialization	Visiting	Permanent	P_Email	U_Email

Module_Table ✓



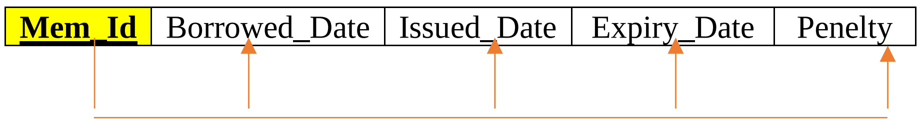
M_code	M_Name	Evaluation_Type	Std_Id

Course_Table ✓



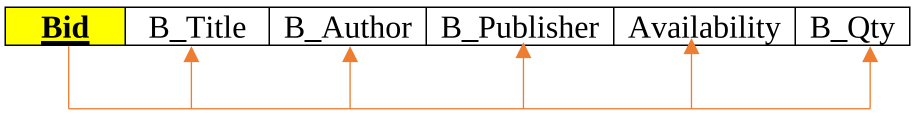
Cid	C_Name	Credits	Duration	Offered_Unis	Fee

Membership_Table ✓



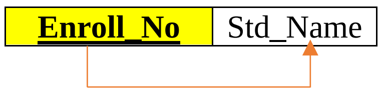
Mem_Id	Borrowed_Date	Issued_Date	Expiry_Date	Penelty

Book_Table ✓



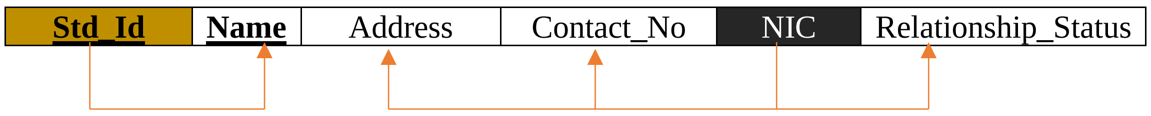
Bid	B_Title	B_Author	B_Publisher	Availability	B_Qty

Registration_Table ✓



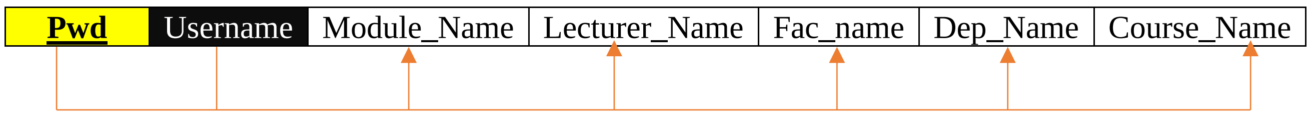
Enroll_No	Std_Name

Dependent_Table (3rd Normalization is required)



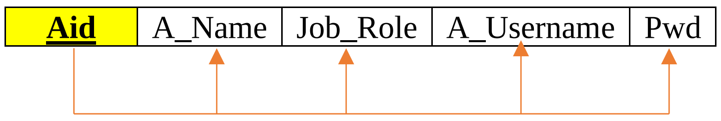
Std_Id	**Name**	Address	Contact_No	NIC	Relationship_Status

LMS_Table ✓



Pwd	Username	Module_Name	Lecturer_Name	Fac_name	Dep_Name	Course_Name

Admin_Table ✓

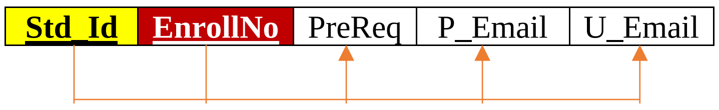


Aid	A_Name	Job_Role	A_Username	Pwd

2nd NORMALIZATION FORM

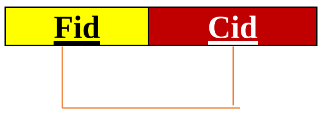
- All the non-prime attributes should depend on a primary key
- All the partial dependencies and the full-functional dependencies will be clearly identified
- All the tables created for the many-to-many relationship are recognized as full-functional dependencies, where the primary key and the foreign key of the table together will make a unique key to identify the particular information
- Tables which are in the 2nd Normalization form is also in the 1st Normalized form
- ✓ Indicates all the full-functional dependency tables

Student_Enroll_Table



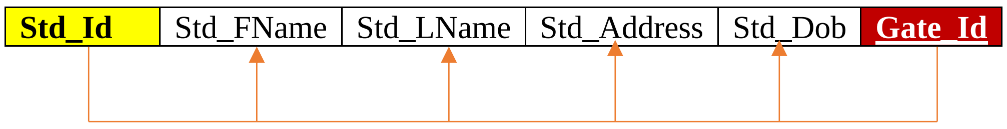
Std_Id	**EnrollNo**	PreReq	P_Email	U_Email

Faculty_Course_Table



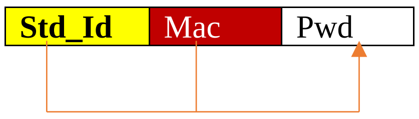
Fid	**Cid**

Student_Table



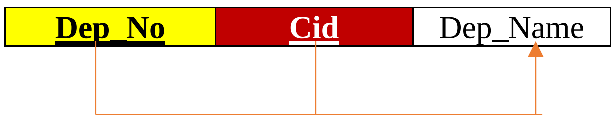
Std_Id	Std_FName	Std_LName	Std_Address	Std_Dob	**Gate_Id**

Student_Wi-Fi_Table



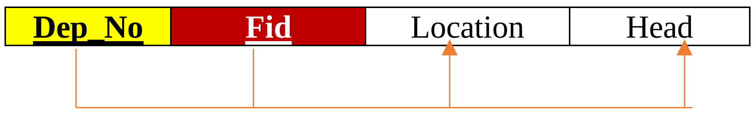
Std_Id	Mac	Pwd

Departrment_Course_Table



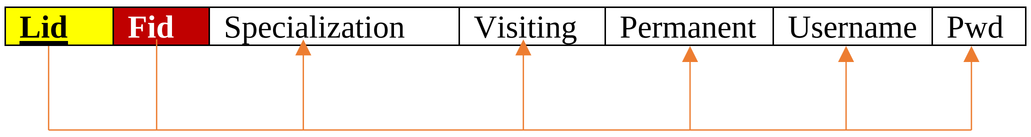
Dep_No	**Cid**	Dep_Name

Department_Faculty_Table



Dep_No	**Fid**	Location	Head

Lecturer_Faculty_Table



Lid	**Fid**	Specialization	Visiting	Permanent	Username	Pwd

Lecturer_Table

Lid	L_FName	L_FName	Contact_No	P_Email	Uni_Email

Dependent_Table

Std_Id	Dependent_Name

3rd NORMALIZATION FORM

- All the transitive dependencies are clearly identified (non-prime attributes which depend on other non-prime attributes)
- All the tables which are in 3rd Normalization form is in the 2nd Normalized form as well the 1st Normalized form

Student_Dependent_Table

NIC	Address	Contact_No	Relationship_Status

Faculty_Head_Table

Fac_Name	Dean_Name

Final tables available for the database after Normalization was carried out

Entrance_Table

Gate_Id	Date	Attendance

Exam_Table

Eid	Index_No	Yr_of_Sty	Attempt	Grade	Paper_code	Class	Gpa

Wi-Fi_Table

Mac	**Ip_Address**	Router	Security	Con_Speed	Signal_Strength	Subnet_Mask	Username	Pwd

Module_Table

M_code	**Std_Id**	Evaluation_Type	M_Name

Course_Table

Cid	C_Name	Credits	Duration	Offered_Unis	Fee

Membership_Table

Mem_Id	Borrowed_Date	Issued_Date	Expiry_Date	Penelty

Book_Table

Bid	B_Title	B_Author	B_Publisher	Availability	B_Qty

Registration_Table

Enroll_No	Std_Name

LMS_Table

Pwd	**Username**	Module_Name	Lecturer_Name	Fac_name	Dep_Name	Course_Name

Admin_Table

Aid	**Pwd**	A_Name	Job_Role	A_Username

Student_Enroll_Table

Std_Id	**EnrollNo**	PreReq	P_Email	U_Email

Faculty_Course_Table

Fid	**Cid**

Student_Table

Std_Id	**Gate_Id**	Std_FName	Std_LName	Std_Address	Std_Dob

Student_Wi-Fi_Table

Std_Id	**Mac**	**Pwd**

Departrment_Course_Table

Dep_No	**Cid**	Dep_Name

Department_Faculty_Table

Dep_No	**Fid**	Location	Head

Lecturer_Faculty_Table

Lid	**Fid**	**Pwd**	Visiting	Permanent	Username	Specialization

Lecturer_Table

Lid	L_FName	L_FName	Contact_No	P_Email	Uni_Email

Dependent_Table

Dependent_Name	**Std_Id**

Student_Dependent_Table

NIC	Address	Contact_No	Relationship_Status

Faculty_Head_Table

Dean_Name	Fid

Student_Exam_Table

Eid	Std_Id

Student_Department_Table

Std_Id	Dep_No

Lecturer_Module_Table

Lid	M_Code

Course_Module_Table

Cid	M_Code

Lecturer_Department_Table

Lid	Dep_No

Admin_Department_Table

Aid	Dep_No

Admin_Faculty_Table

Aid	Fid

Module_Exam_Table

M_Code	Eid

Admin_Wi-Fi_Table

Aid	Mac

Membership_Book_Table

Mem_Id	Bid

Admin_Registration_Table

Aid	Enroll_No

Lecturer_Wi-Fi_Table

Lid	Mac

🟨 = Primary Key ⬛ = Foreign Key

DATA DICTIONARY

TABLE NAME	FIELD NAME	DATA TYPE	DESCRIPTION
Entrance_Table	Gate_Id	int	Unique gate identification
	Date	date	Date and time when the cards were punched
	Attendance	var	No of students entered the premises
	Entrance_Id	int	Unique entrance identification
Exam_Table	Exam_Id	int	Unique number to represent the all the exams
	Index_No	var	Unique code to identify the students who are enrolled to the exams
	Yr_of_study	var	Student’s year of study
	Attempt	int	Student’s attempt number
	Grade	var	Letter to represent the score
	Paper_code	var	Degree program and the code of the module
	Class	var	Considering the gpa
	Gpa	decimal	Value calculated using the mark of each module
Wi-Fi_Table	Mac	var	To identify devices uniquely
	Pwd	var	To login
	Router	var	To identify access points
	Security	var	Security level
	Connection_Speed	int	Indicates upload and download speed
	Signal_Strength	var	Signify the nature
	Username	var	Unique Email to identify the user
	Ip_Address	var	To identify device location
Module_Table	module_code	var	Unique way of identifying all the modules
	Std_Id	int	To access student details
	Evaluation_type	var	Method-Written, course work
	module_name	var	Name of the module
Course_Table	Course_Id	var	Unique way of identifying the course
	Course_name	var	Name of the course

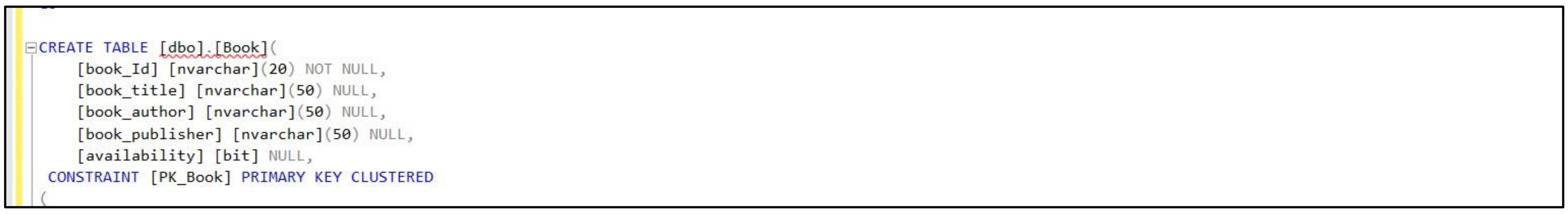
	Credits	int	The total value required to get the degree
	Duration	var	Time period (3 or 4)
	Offered-University	var	Available universities (foreign or local)
	Fee	money	Value in dollars
Membership_Table	Mem_Id	int	To identify library users
	Borrowed_Date	date	Date of borrowing
	Issued_Date	date	Membership card issuing
	Expiry_Date	date	Membership card renewal date
	Penalty	money	Fine charges
Book_Table	Book_Id	var	Unique way to identify specific category of books
	Book_Name	var	The name of the book
	Book_Author	var	Mentioning the author
	Publisher	var	Publication information
	Availability	bit	Store
	B_Qty	Int	Number of books
Registration_Table	Enroll_No	int	Unique way of identifying the people who registered for the specific academic year
	Std_Name	var	Full name of the person
LMS_Table	Pwd	var	For login
	Username	var	For login
	Module_Name	var	Indicates one’s specific modules relevant to the degree program
	Lecturer_Name	var	Display all the information
	Fac_Name	var	Display all the information
	Dep_Name	var	Display all the information
	Course_Name	var	Display all the information
Admin_Table	Aid	int	Using the Aid all the admins can be identified
	Pwd	var	For Login
	A_Name	var	Admin’s name
	Job_Role	var	Specialization

	A_Username	var	For login
Student_Enroll_Table	Std_Id	int	To recognize each and every student uniquely
	Enroll_No	var	Specifically, to identify the students who registered for the exam
	PreReq	var	To check eligibility
	P_Email	var	Personal Email
	U_Email	var	University Email
Faculty_Course_Table	Fid	var	To separately identify the courses under a faculty
	Cid	var	The course Id's
Student_Table	Std_Id	int	Unique Id
	Gate_Id	int	Record of entering inside the university premises
	Std_FName	var	First Name of the student
	Std_LName	var	Last name of the student
	Std_Address	var	Home address
	Dob	date	Date of birth
Student_Wi-Fi_Table	Std_Id	int	To create a unique new connection
	Mac	var	Relating to the device connected
	Pwd	var	Personally created
Departement_Course_Table	Dep_No	var	Uniquely identify the department
	Cid	var	Course Id
	Dep_Name	var	Department name
Department_Faculty_Table	Dep_No	var	Department number
	Fid	var	Faculty number
	Location	var	Specific location of the relevant department
	Head	var	Name of the head of the department
Lecturer_Faculty_Table	Lid	int	Lecturer Id number
	Fid	var	Faculty Id number
	Pwd	var	Lecturer's Login
	Visiting	bit	Tick if he/she is a visiting lecturer

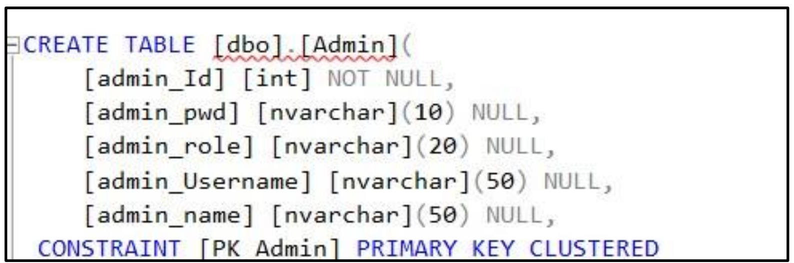
	Permanent	bit	Same as above
	Username	var	Lecturer’s Username
	Specialization	var	Qualifications
Lecturer_Table	Lid	int	Lecturer Id
	L_FName	var	First name of the lecturer
	Contact_No	int	Last name of the lecturer
	P_Email	var	Personal Email of the lecturer
	U_Email	var	University Email of the lecturer
Dependent_Table	Dependent_Name	var	Surname or the full name relevant to the student
	Std_Id	int	Forming a unique key to identify the guardian
Student_Dependent_Table	Address	var	Address of the guardian
	NIC	var	National identity card
	Relationship_Status	var	Guardian relationship
Faculty_Head_Table	Dean	var	Name of the faculty dean(Who is in charge with the matters)
	Fid	var	To identify the faculty uniquely
Student_Exam_Table	Eid	int	Unique table to identify the information relating to the student and the exams
	Std_Id	int	To identify the student
Student_Department_Table	Std_Id	int	Unique way to separately identify the departments of each and every student
	Dep_No	var	Department number
Lecturer_Module_Table	Lid	int	To separately identify the lecturers who deliver the modules
	M_Code	var	To uniquely identify the module
Course_Module_Table	Cid	var	To separately retrieve the modules relevant to the course
	M_Code	var	Module code
Lecturer_Department_Table	Lid	int	To list down all the lecturers belonging to a certain department
	Dep_No	var	Department number
Admin_Department_Table	Aid	int	To separately identify all the admins who belong to a certain department
	Dep_No	var	Department number
Module_Exam_Table	M_Code	var	Exam procedure for the modules table

	Eid	int	Exam Id
Admin_Wi-Fi_Table	Aid	int	Admin login details
	Mac	var	Admin login devices details
Membership_Book_Table	Mem_Id	int	Borrowed book information
	Bid	var	To identify the book uniquely
Admin_Registration_Table	Aid	int	Admin who belongs to a certain department undertakes the registration process
	EnrollNo	var	Number of enrollments done with the enroll number
Lecturer_Wi-Fi_Table	Lid	int	Lecturer login details
	Mac	var	Lecturer login devices

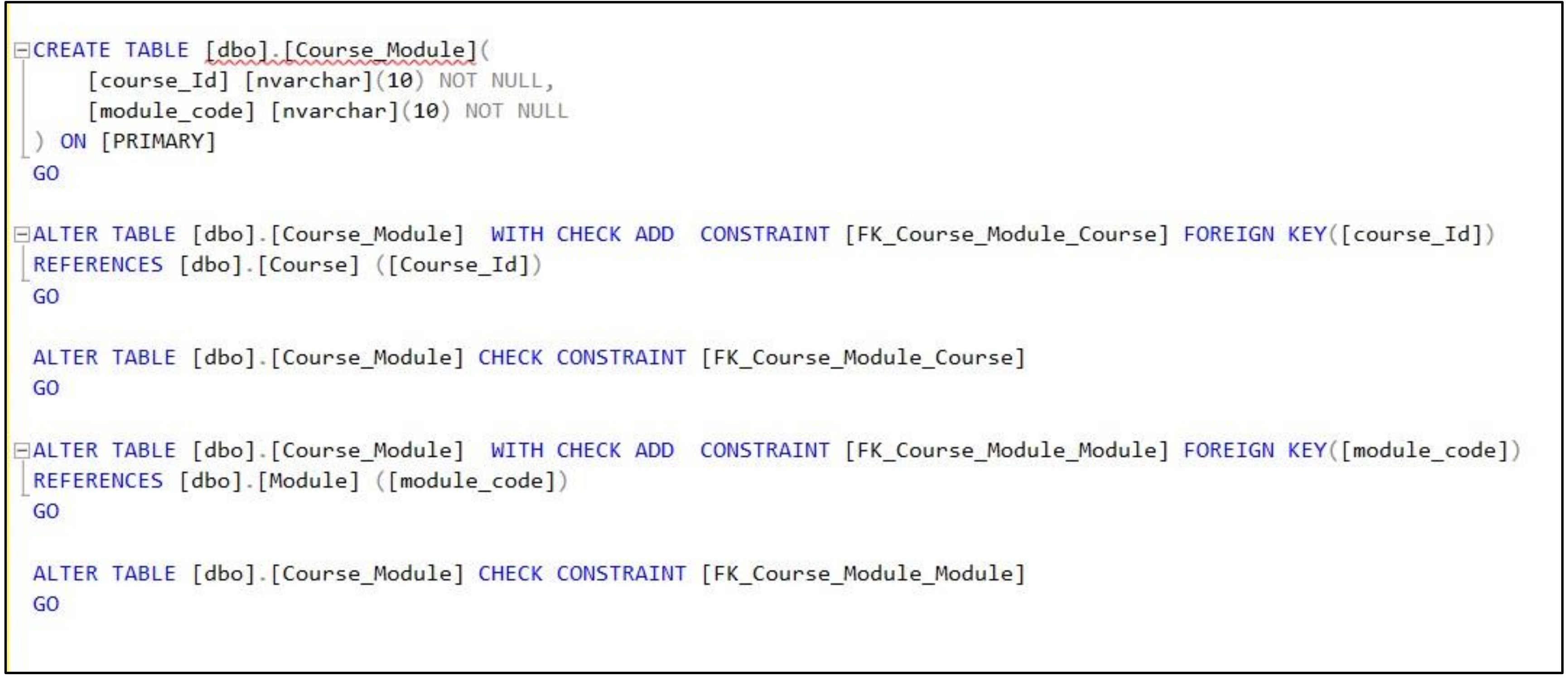
Table Creation SQL Commands



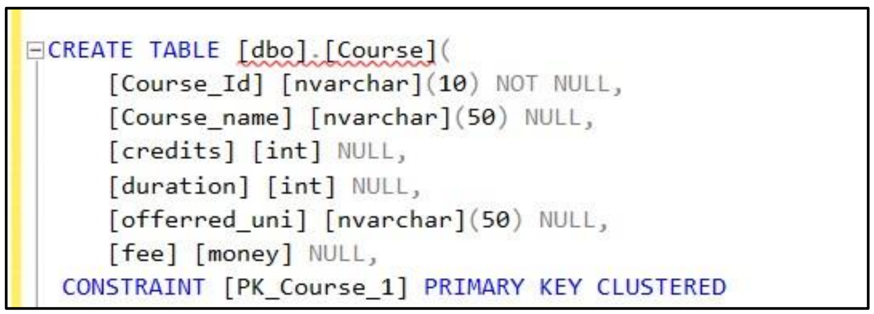
```
CREATE TABLE [dbo].[Book](
     [book_Id] [nvarchar](20) NOT NULL,
     [book_title] [nvarchar](50) NULL,
     [book_author] [nvarchar](50) NULL,
     [book_publisher] [nvarchar](50) NULL,
     [availability] [bit] NULL,
 CONSTRAINT [PK_Book] PRIMARY KEY CLUSTERED
(
```



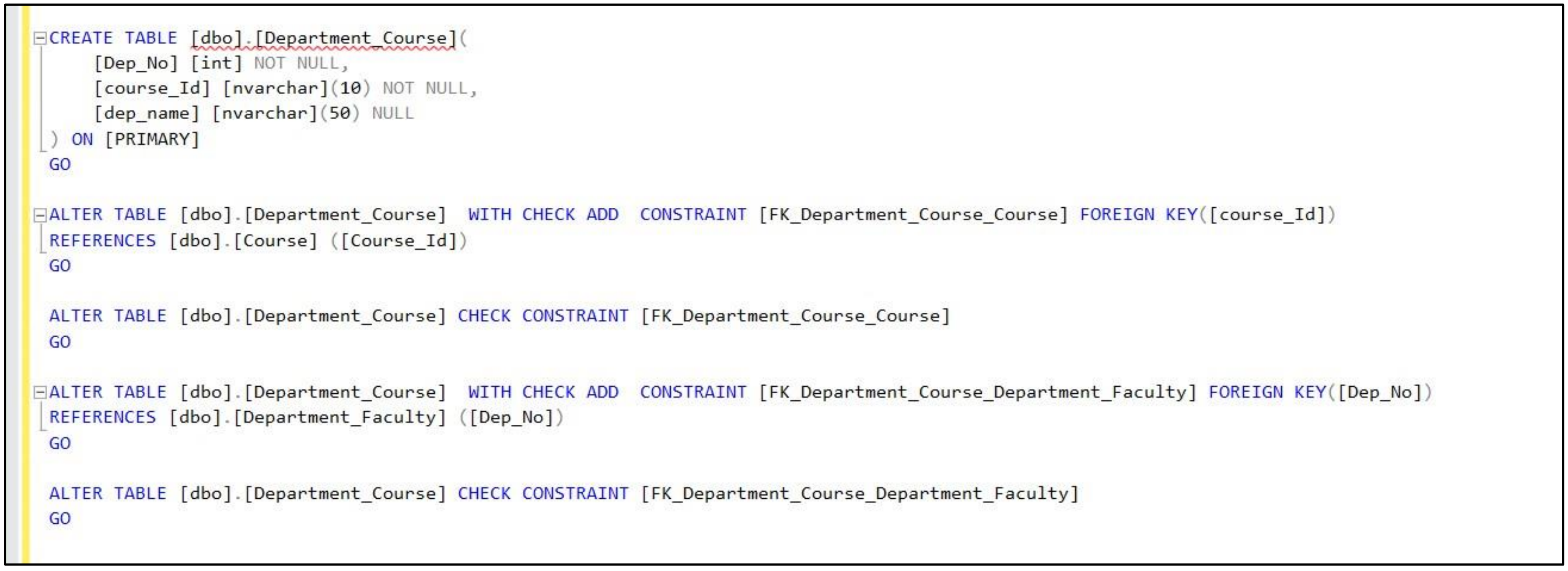
```
CREATE TABLE [dbo].[Admin](
    [admin_Id] [int] NOT NULL,
    [admin_pwd] [nvarchar](10) NULL,
    [admin_role] [nvarchar](20) NULL,
    [admin_Username] [nvarchar](50) NULL,
    [admin_name] [nvarchar](50) NULL,
 CONSTRAINT [PK_Admin] PRIMARY KEY CLUSTERED
```



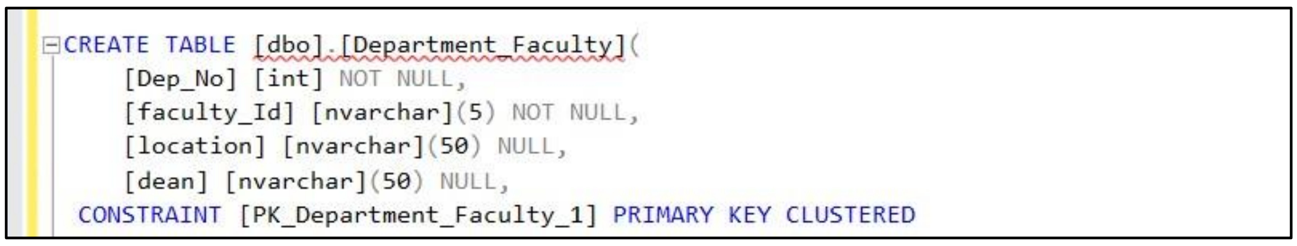
```
CREATE TABLE [dbo].[Course_Module](
    [course_Id] [nvarchar](10) NOT NULL,
    [module_code] [nvarchar](10) NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_Module]  WITH CHECK ADD  CONSTRAINT [FK_Course_Module_Course] FOREIGN KEY([course_Id])
REFERENCES [dbo].[Course] ([Course_Id])
GO

ALTER TABLE [dbo].[Course_Module] CHECK CONSTRAINT [FK_Course_Module_Course]
GO

ALTER TABLE [dbo].[Course_Module]  WITH CHECK ADD  CONSTRAINT [FK_Course_Module_Module] FOREIGN KEY([module_code])
REFERENCES [dbo].[Module] ([module_code])
GO

ALTER TABLE [dbo].[Course_Module] CHECK CONSTRAINT [FK_Course_Module_Module]
GO
```



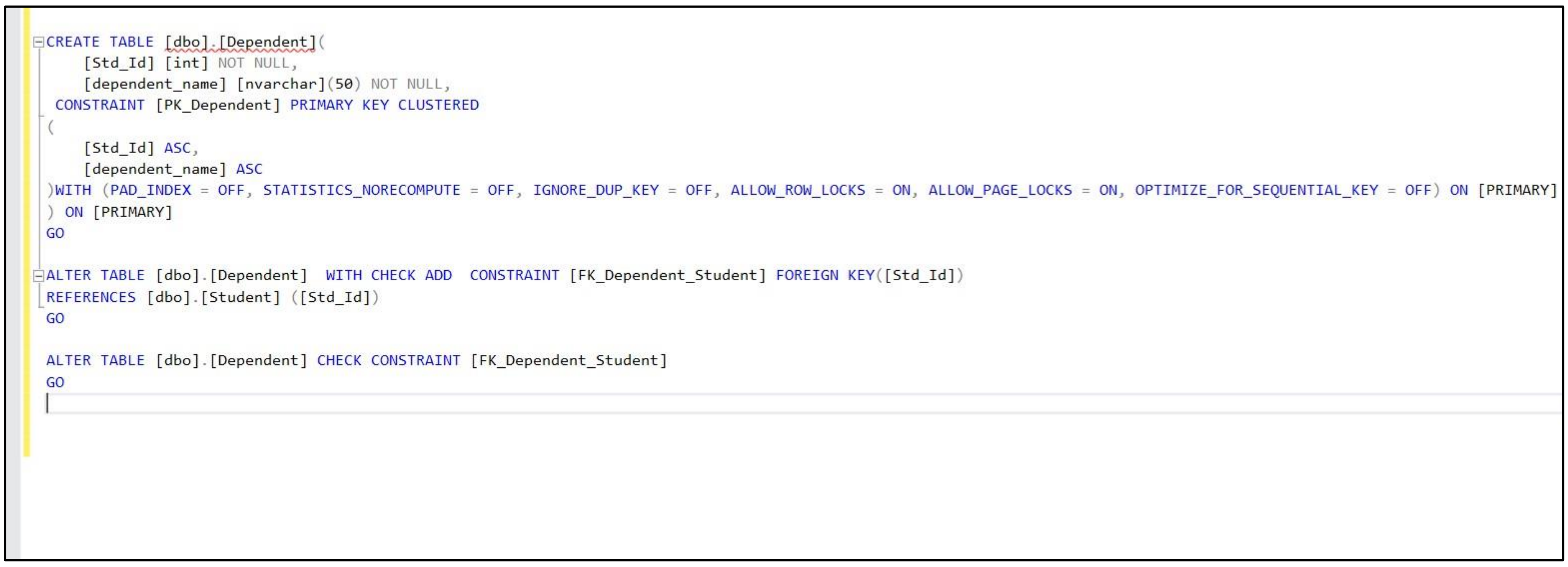
```
CREATE TABLE [dbo].[Course](
    [Course_Id] [nvarchar](10) NOT NULL,
    [Course_name] [nvarchar](50) NULL,
    [credits] [int] NULL,
    [duration] [int] NULL,
    [offerred_uni] [nvarchar](50) NULL,
    [fee] [money] NULL,
 CONSTRAINT [PK_Course_1] PRIMARY KEY CLUSTERED
```



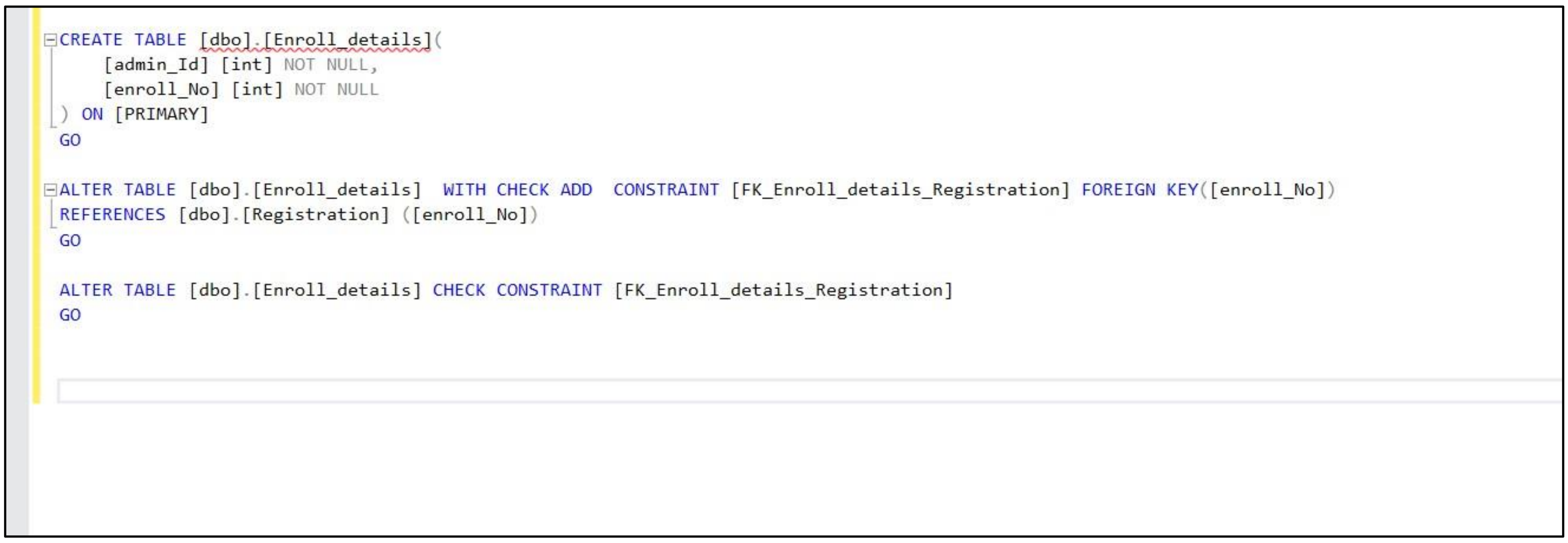
```
CREATE TABLE [dbo].[Department_Course](
    [Dep_No] [int] NOT NULL,
    [course_Id] [nvarchar](10) NOT NULL,
    [dep_name] [nvarchar](50) NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Department_Course]  WITH CHECK ADD  CONSTRAINT [FK_Department_Course_Course] FOREIGN KEY([course_Id])
REFERENCES [dbo].[Course] ([Course_Id])
GO

ALTER TABLE [dbo].[Department_Course] CHECK CONSTRAINT [FK_Department_Course_Course]
GO

ALTER TABLE [dbo].[Department_Course]  WITH CHECK ADD  CONSTRAINT [FK_Department_Course_Department_Faculty] FOREIGN KEY([Dep_No])
REFERENCES [dbo].[Department_Faculty] ([Dep_No])
GO

ALTER TABLE [dbo].[Department_Course] CHECK CONSTRAINT [FK_Department_Course_Department_Faculty]
GO
```



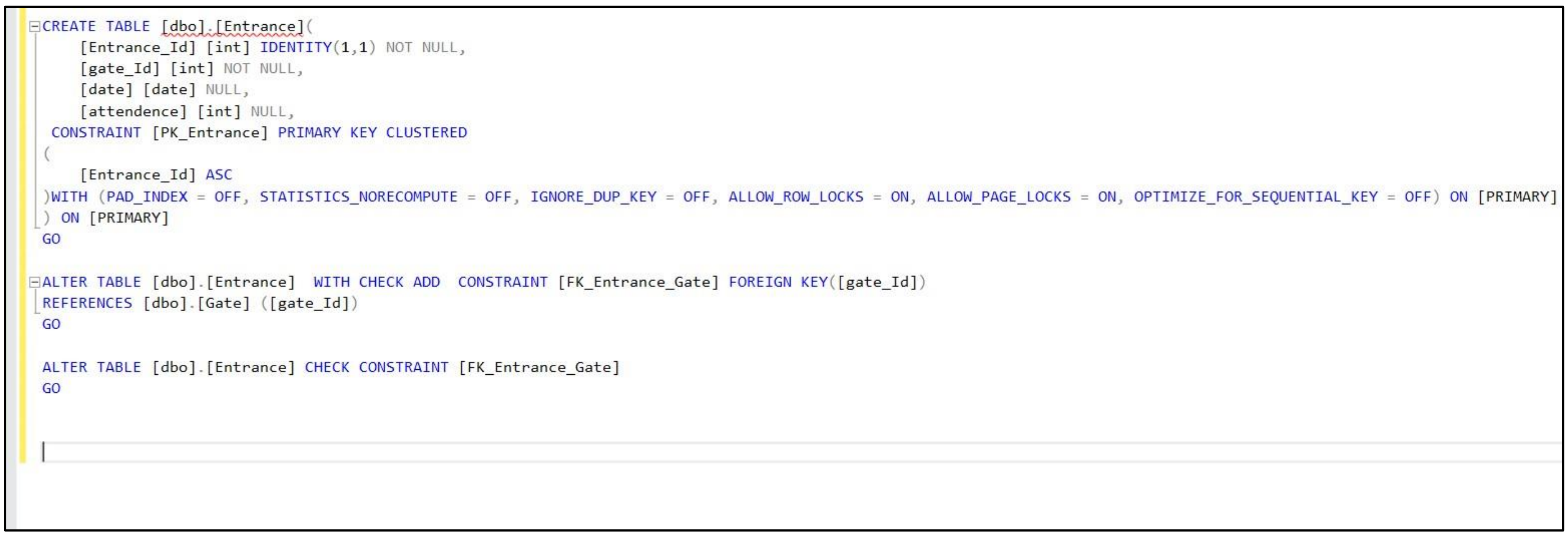
```
CREATE TABLE [dbo].[Department_Faculty](
    [Dep_No] [int] NOT NULL,
    [faculty_Id] [nvarchar](5) NOT NULL,
    [location] [nvarchar](50) NULL,
    [dean] [nvarchar](50) NULL,
 CONSTRAINT [PK_Department_Faculty_1] PRIMARY KEY CLUSTERED
```



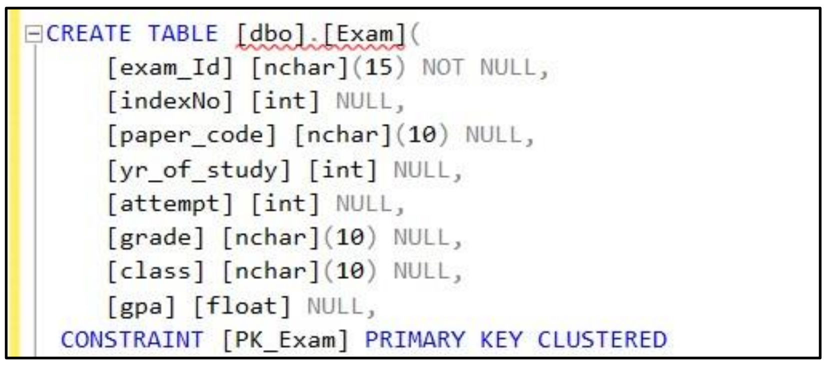
```
CREATE TABLE [dbo].[Dependent](
    [Std_Id] [int] NOT NULL,
    [dependent_name] [nvarchar](50) NOT NULL,
 CONSTRAINT [PK_Dependent] PRIMARY KEY CLUSTERED
(
    [Std_Id] ASC,
    [dependent_name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Dependent]  WITH CHECK ADD  CONSTRAINT [FK_Dependent_Student] FOREIGN KEY([Std_Id])
REFERENCES [dbo].[Student] ([Std_Id])
GO

ALTER TABLE [dbo].[Dependent] CHECK CONSTRAINT [FK_Dependent_Student]
GO
```



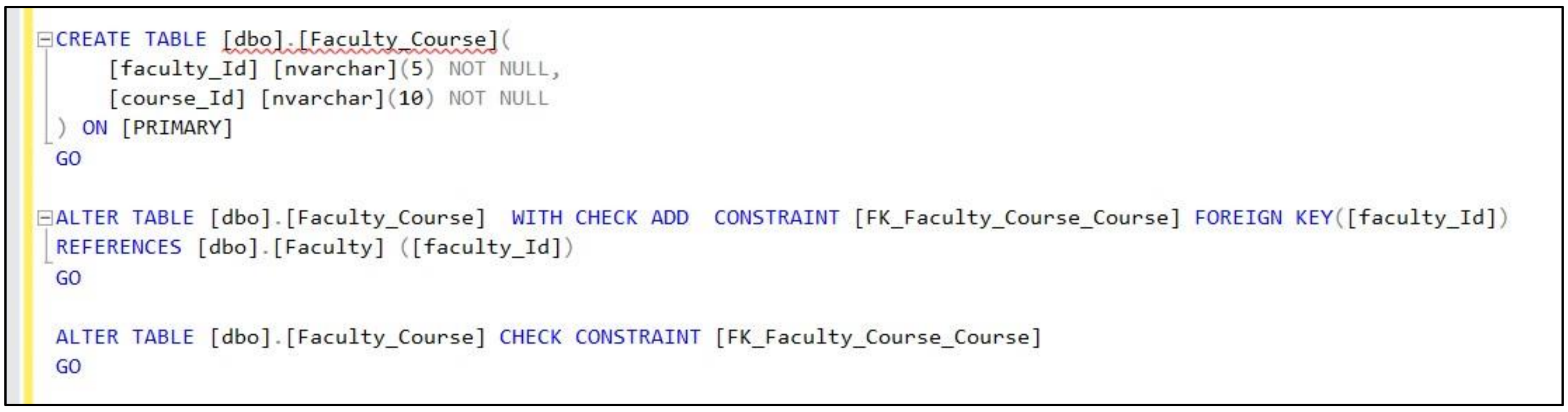
```
CREATE TABLE [dbo].[Enroll_details](
    [admin_Id] [int] NOT NULL,
    [enroll_No] [int] NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Enroll_details]  WITH CHECK ADD  CONSTRAINT [FK_Enroll_details_Registration] FOREIGN KEY([enroll_No])
REFERENCES [dbo].[Registration] ([enroll_No])
GO

ALTER TABLE [dbo].[Enroll_details] CHECK CONSTRAINT [FK_Enroll_details_Registration]
GO
```



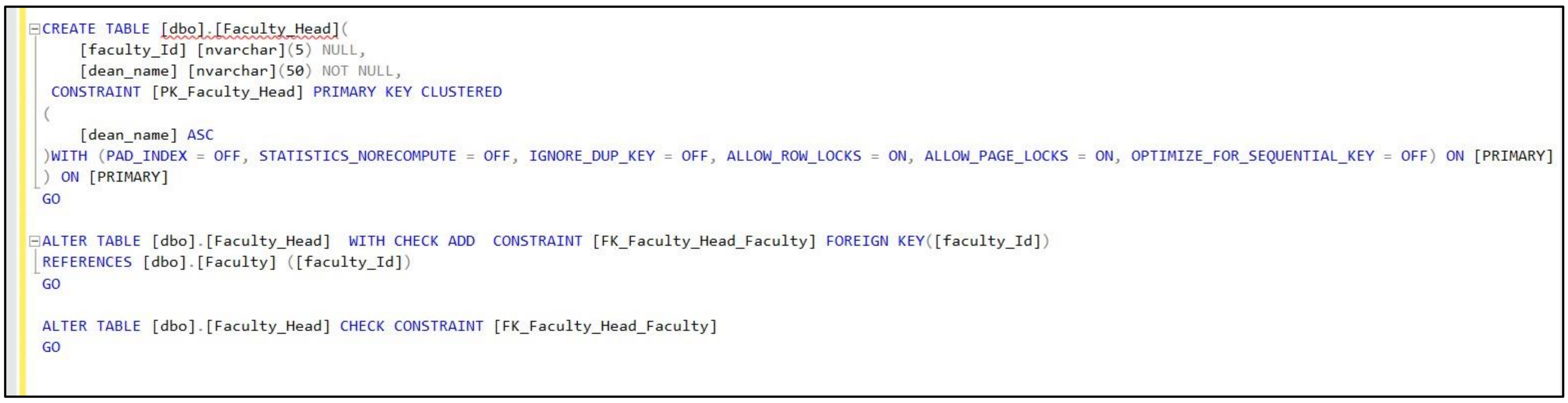
```sql
CREATE TABLE [dbo].[Entrance](
    [Entrance_Id] [int] IDENTITY(1,1) NOT NULL,
    [gate_Id] [int] NOT NULL,
    [date] [date] NULL,
    [attendence] [int] NULL,
 CONSTRAINT [PK_Entrance] PRIMARY KEY CLUSTERED
(
    [Entrance_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Entrance]  WITH CHECK ADD  CONSTRAINT [FK_Entrance_Gate] FOREIGN KEY([gate_Id])
REFERENCES [dbo].[Gate] ([gate_Id])
GO

ALTER TABLE [dbo].[Entrance] CHECK CONSTRAINT [FK_Entrance_Gate]
GO
```



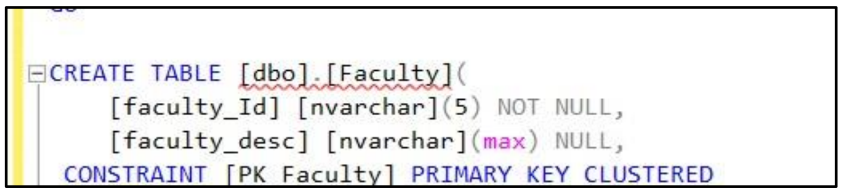
```sql
CREATE TABLE [dbo].[Exam](
    [exam_Id] [nchar](15) NOT NULL,
    [indexNo] [int] NULL,
    [paper_code] [nchar](10) NULL,
    [yr_of_study] [int] NULL,
    [attempt] [int] NULL,
    [grade] [nchar](10) NULL,
    [class] [nchar](10) NULL,
    [gpa] [float] NULL,
 CONSTRAINT [PK_Exam] PRIMARY KEY CLUSTERED
```



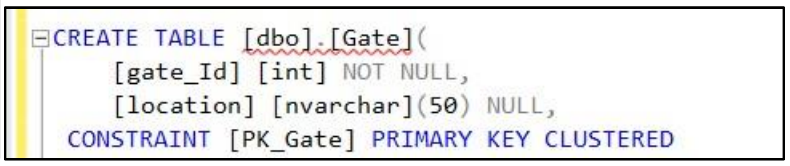
```sql
CREATE TABLE [dbo].[Faculty_Course](
    [faculty_Id] [nvarchar](5) NOT NULL,
    [course_Id] [nvarchar](10) NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Faculty_Course]  WITH CHECK ADD  CONSTRAINT [FK_Faculty_Course_Course] FOREIGN KEY([faculty_Id])
REFERENCES [dbo].[Faculty] ([faculty_Id])
GO

ALTER TABLE [dbo].[Faculty_Course] CHECK CONSTRAINT [FK_Faculty_Course_Course]
GO
```



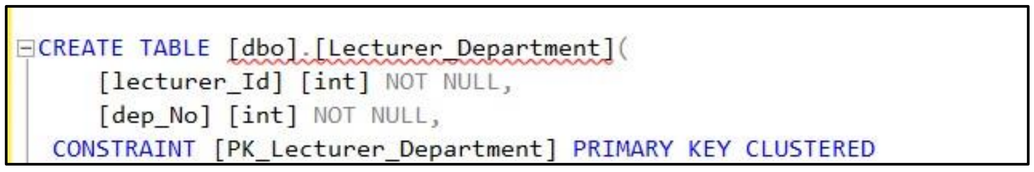
```sql
CREATE TABLE [dbo].[Faculty_Head](
    [faculty_Id] [nvarchar](5) NULL,
    [dean_name] [nvarchar](50) NOT NULL,
 CONSTRAINT [PK_Faculty_Head] PRIMARY KEY CLUSTERED
(
    [dean_name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Faculty_Head]  WITH CHECK ADD  CONSTRAINT [FK_Faculty_Head_Faculty] FOREIGN KEY([faculty_Id])
REFERENCES [dbo].[Faculty] ([faculty_Id])
GO

ALTER TABLE [dbo].[Faculty_Head] CHECK CONSTRAINT [FK_Faculty_Head_Faculty]
GO
```



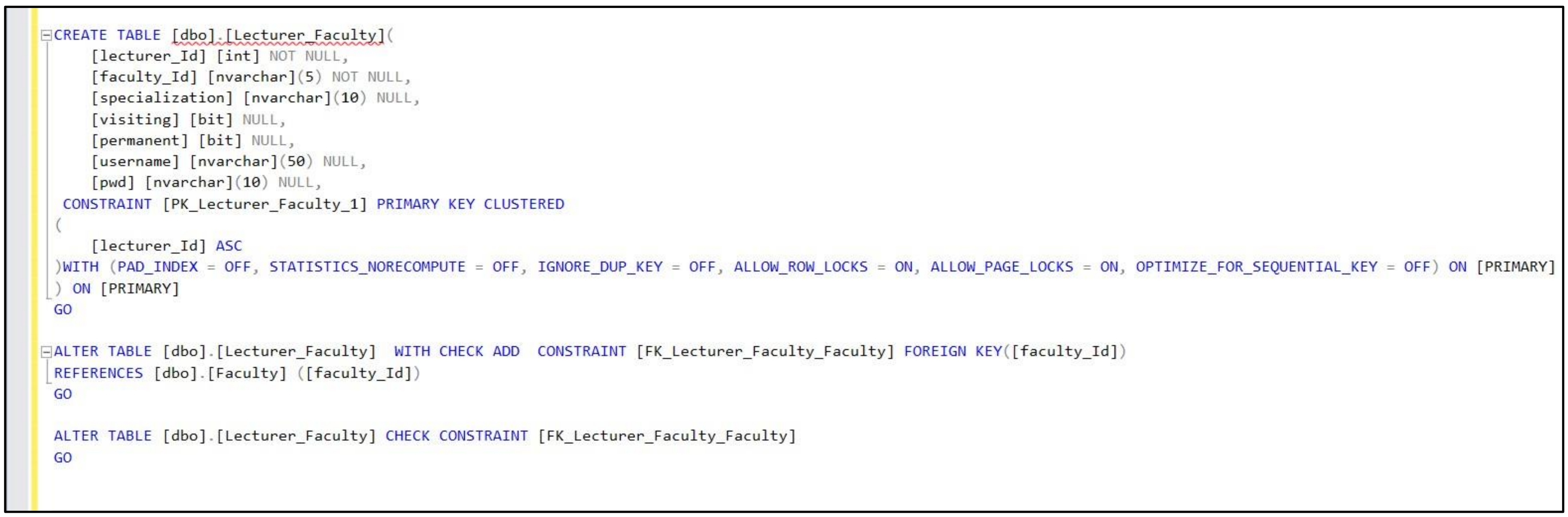
```sql
CREATE TABLE [dbo].[Faculty](
    [faculty_Id] [nvarchar](5) NOT NULL,
    [faculty_desc] [nvarchar](max) NULL,
 CONSTRAINT [PK_Faculty] PRIMARY KEY CLUSTERED
```



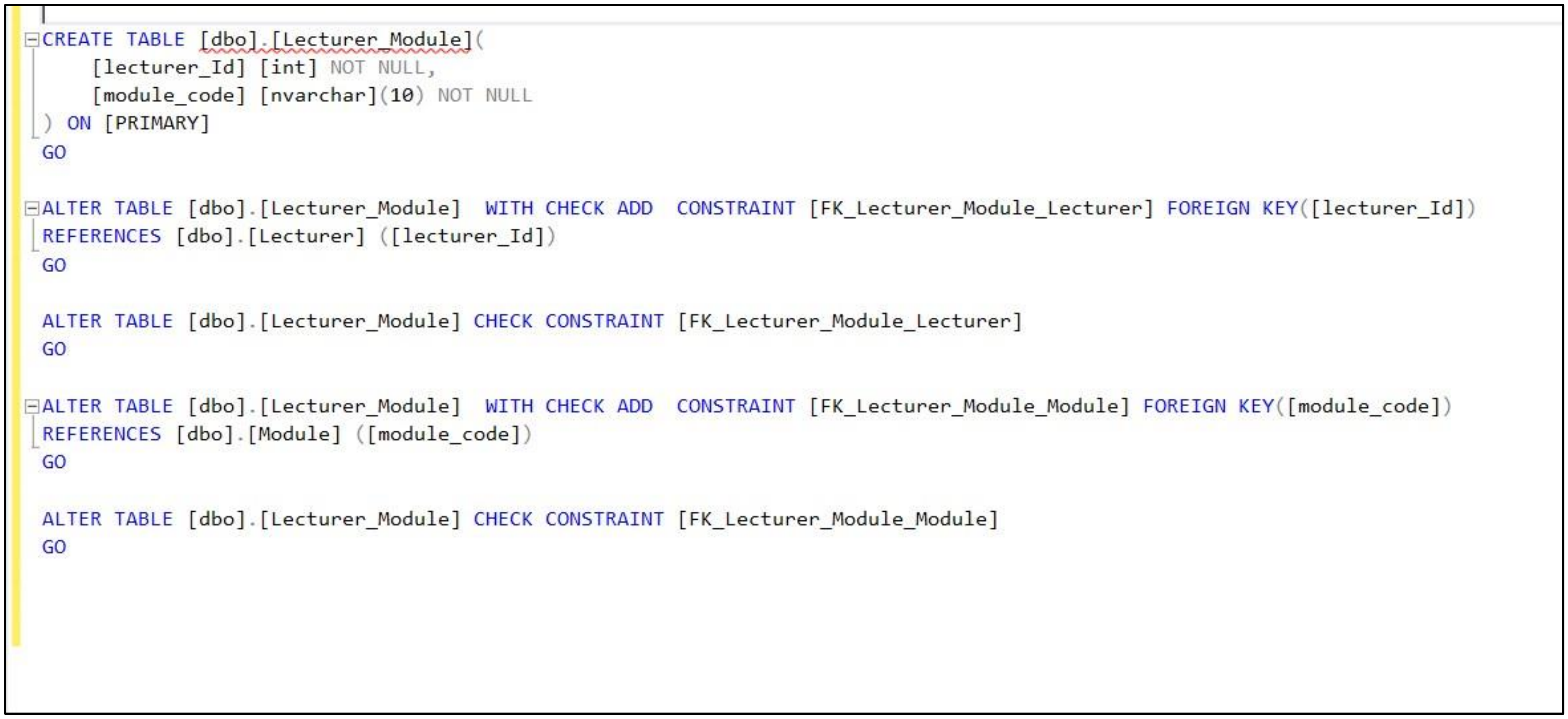
```sql
CREATE TABLE [dbo].[Gate](
    [gate_Id] [int] NOT NULL,
    [location] [nvarchar](50) NULL,
 CONSTRAINT [PK_Gate] PRIMARY KEY CLUSTERED
```



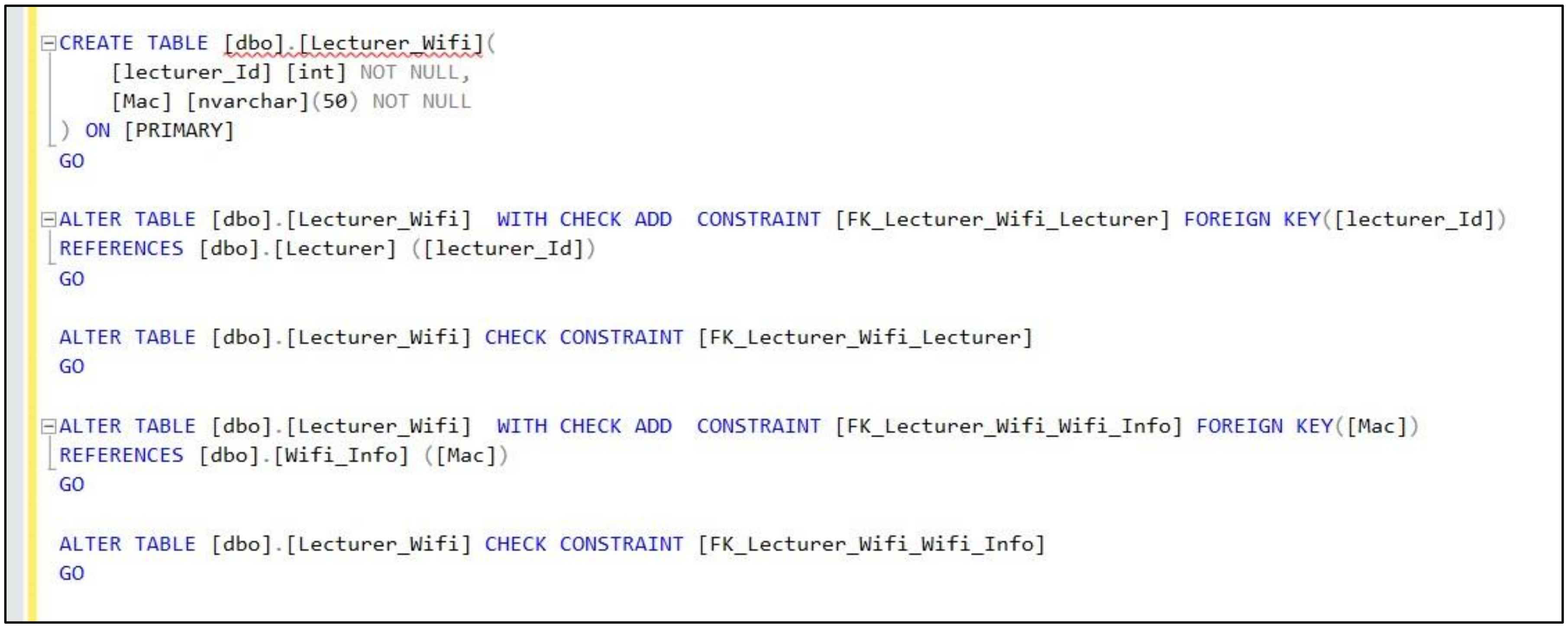
```sql
CREATE TABLE [dbo].[Lecturer_Department](
    [lecturer_Id] [int] NOT NULL,
    [dep_No] [int] NOT NULL,
 CONSTRAINT [PK_Lecturer_Department] PRIMARY KEY CLUSTERED
```



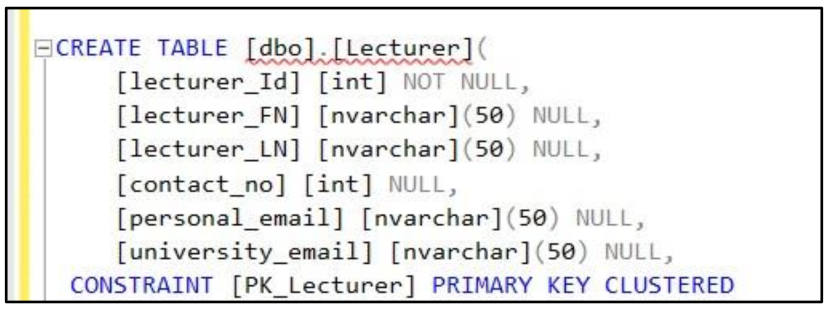
```sql
CREATE TABLE [dbo].[Lecturer_Faculty](
    [lecturer_Id] [int] NOT NULL,
    [faculty_Id] [nvarchar](5) NOT NULL,
    [specialization] [nvarchar](10) NULL,
    [visiting] [bit] NULL,
    [permanent] [bit] NULL,
    [username] [nvarchar](50) NULL,
    [pwd] [nvarchar](10) NULL,
 CONSTRAINT [PK_Lecturer_Faculty_1] PRIMARY KEY CLUSTERED
(
    [lecturer_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Lecturer_Faculty]  WITH CHECK ADD  CONSTRAINT [FK_Lecturer_Faculty_Faculty] FOREIGN KEY([faculty_Id])
REFERENCES [dbo].[Faculty] ([faculty_Id])
GO

ALTER TABLE [dbo].[Lecturer_Faculty] CHECK CONSTRAINT [FK_Lecturer_Faculty_Faculty]
GO
```



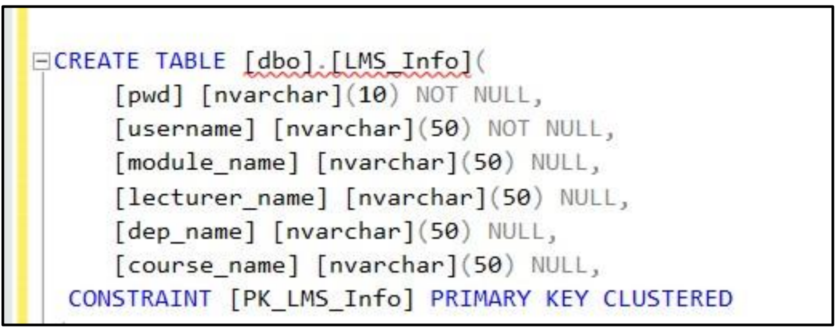
```sql
CREATE TABLE [dbo].[Lecturer_Module](
    [lecturer_Id] [int] NOT NULL,
    [module_code] [nvarchar](10) NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Lecturer_Module]  WITH CHECK ADD  CONSTRAINT [FK_Lecturer_Module_Lecturer] FOREIGN KEY([lecturer_Id])
REFERENCES [dbo].[Lecturer] ([lecturer_Id])
GO

ALTER TABLE [dbo].[Lecturer_Module] CHECK CONSTRAINT [FK_Lecturer_Module_Lecturer]
GO

ALTER TABLE [dbo].[Lecturer_Module]  WITH CHECK ADD  CONSTRAINT [FK_Lecturer_Module_Module] FOREIGN KEY([module_code])
REFERENCES [dbo].[Module] ([module_code])
GO

ALTER TABLE [dbo].[Lecturer_Module] CHECK CONSTRAINT [FK_Lecturer_Module_Module]
GO
```



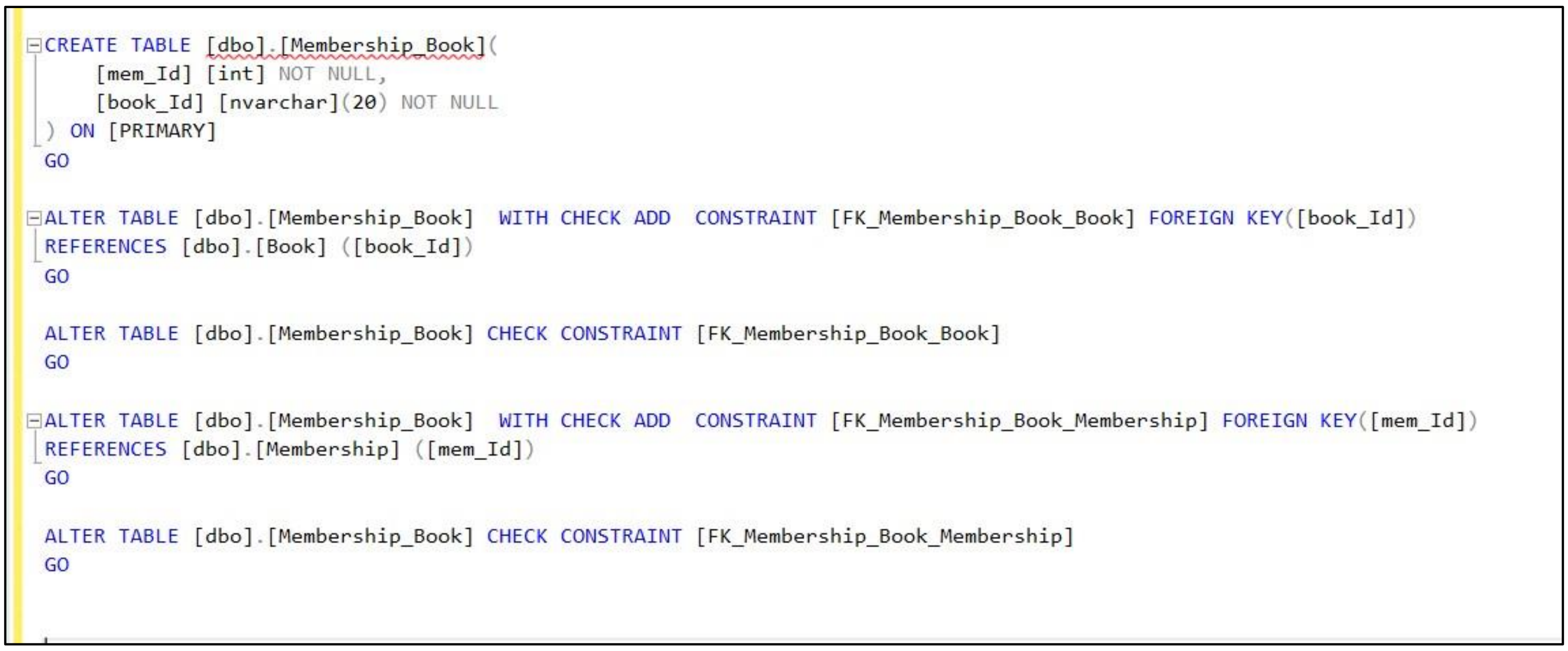
```sql
CREATE TABLE [dbo].[Lecturer_Wifi](
    [lecturer_Id] [int] NOT NULL,
    [Mac] [nvarchar](50) NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Lecturer_Wifi]  WITH CHECK ADD  CONSTRAINT [FK_Lecturer_Wifi_Lecturer] FOREIGN KEY([lecturer_Id])
REFERENCES [dbo].[Lecturer] ([lecturer_Id])
GO

ALTER TABLE [dbo].[Lecturer_Wifi] CHECK CONSTRAINT [FK_Lecturer_Wifi_Lecturer]
GO

ALTER TABLE [dbo].[Lecturer_Wifi]  WITH CHECK ADD  CONSTRAINT [FK_Lecturer_Wifi_Wifi_Info] FOREIGN KEY([Mac])
REFERENCES [dbo].[Wifi_Info] ([Mac])
GO

ALTER TABLE [dbo].[Lecturer_Wifi] CHECK CONSTRAINT [FK_Lecturer_Wifi_Wifi_Info]
GO
```



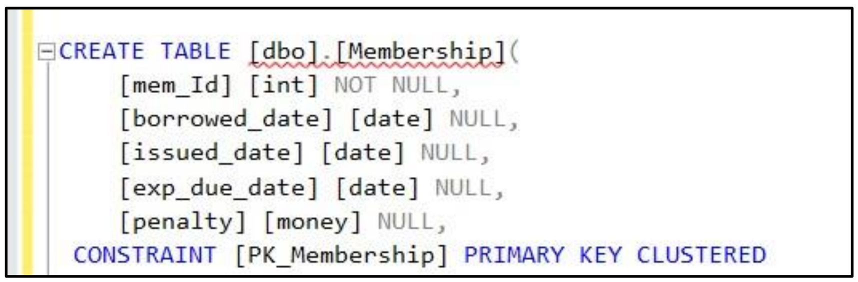
```sql
CREATE TABLE [dbo].[Lecturer](
    [lecturer_Id] [int] NOT NULL,
    [lecturer_FN] [nvarchar](50) NULL,
    [lecturer_LN] [nvarchar](50) NULL,
    [contact_no] [int] NULL,
    [personal_email] [nvarchar](50) NULL,
    [university_email] [nvarchar](50) NULL,
 CONSTRAINT [PK_Lecturer] PRIMARY KEY CLUSTERED
```



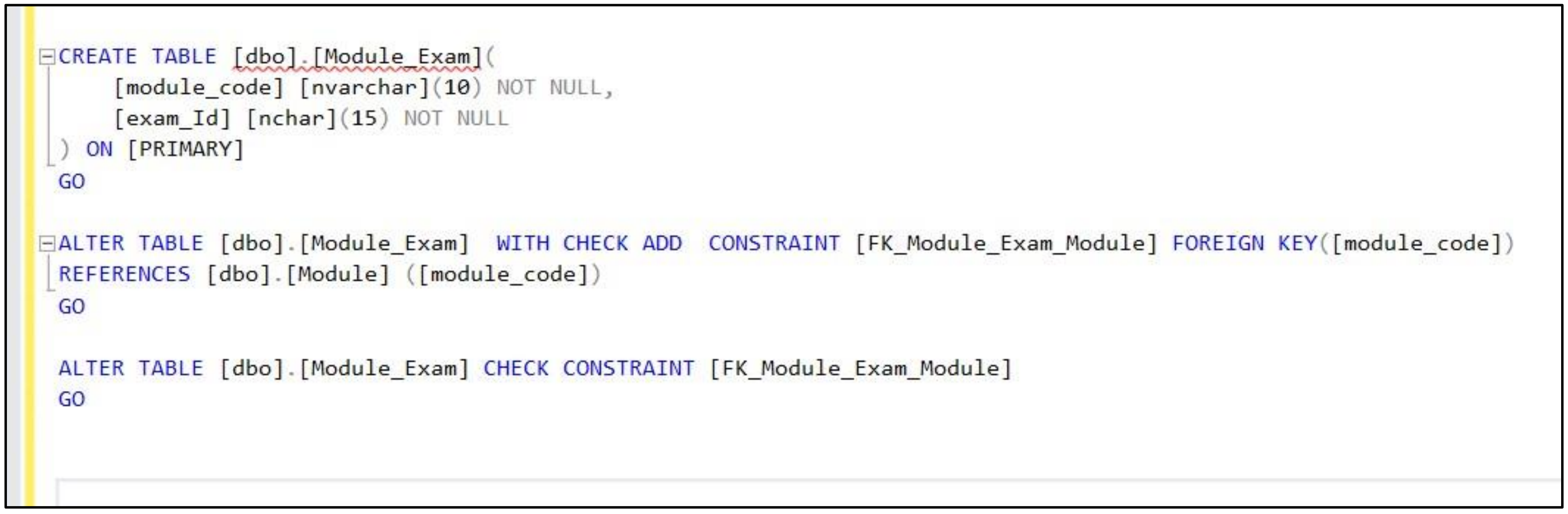
```sql
CREATE TABLE [dbo].[LMS_Info](
    [pwd] [nvarchar](10) NOT NULL,
    [username] [nvarchar](50) NOT NULL,
    [module_name] [nvarchar](50) NULL,
    [lecturer_name] [nvarchar](50) NULL,
    [dep_name] [nvarchar](50) NULL,
    [course_name] [nvarchar](50) NULL,
 CONSTRAINT [PK_LMS_Info] PRIMARY KEY CLUSTERED
```



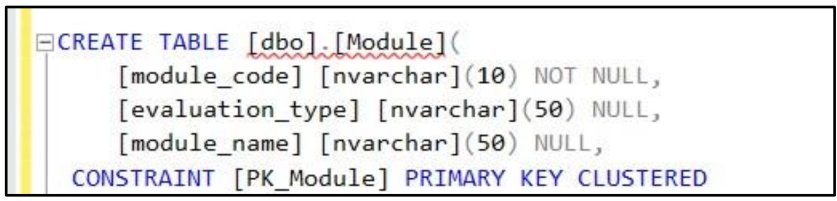
```
CREATE TABLE [dbo].[Membership_Book](
	[mem_Id] [int] NOT NULL,
	[book_Id] [nvarchar](20) NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Membership_Book]  WITH CHECK ADD  CONSTRAINT [FK_Membership_Book_Book] FOREIGN KEY([book_Id])
REFERENCES [dbo].[Book] ([book_Id])
GO

ALTER TABLE [dbo].[Membership_Book] CHECK CONSTRAINT [FK_Membership_Book_Book]
GO

ALTER TABLE [dbo].[Membership_Book]  WITH CHECK ADD  CONSTRAINT [FK_Membership_Book_Membership] FOREIGN KEY([mem_Id])
REFERENCES [dbo].[Membership] ([mem_Id])
GO

ALTER TABLE [dbo].[Membership_Book] CHECK CONSTRAINT [FK_Membership_Book_Membership]
GO
```



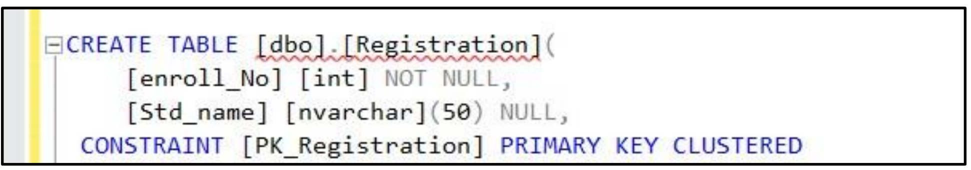
```
CREATE TABLE [dbo].[Membership](
	[mem_Id] [int] NOT NULL,
	[borrowed_date] [date] NULL,
	[issued_date] [date] NULL,
	[exp_due_date] [date] NULL,
	[penalty] [money] NULL,
 CONSTRAINT [PK_Membership] PRIMARY KEY CLUSTERED
```



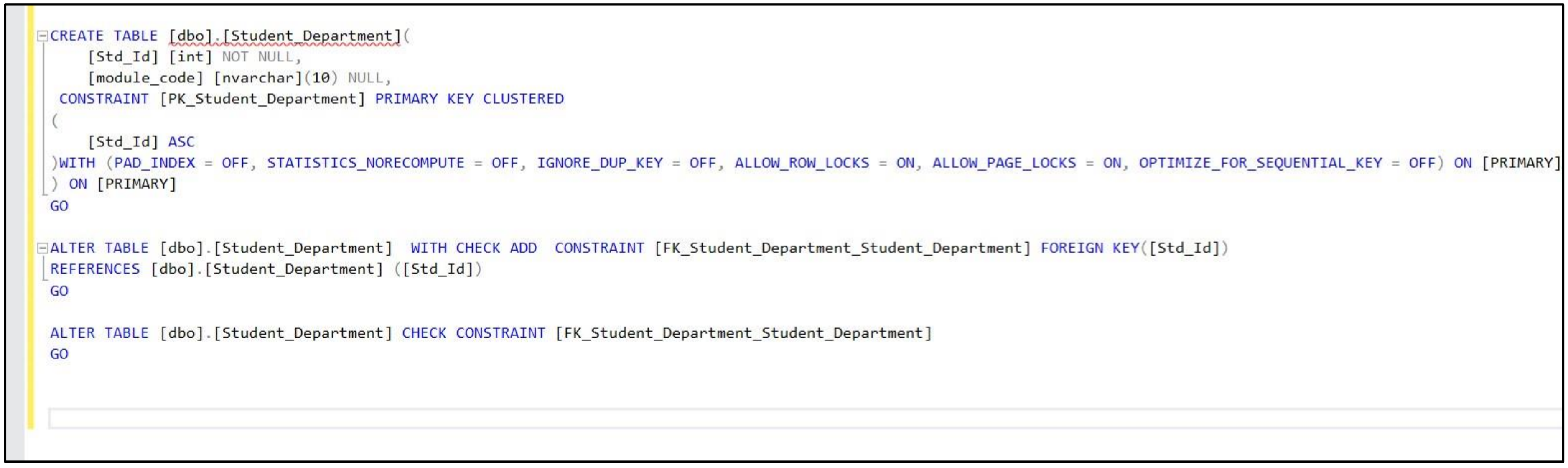
```
CREATE TABLE [dbo].[Module_Exam](
	[module_code] [nvarchar](10) NOT NULL,
	[exam_Id] [nchar](15) NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Module_Exam]  WITH CHECK ADD  CONSTRAINT [FK_Module_Exam_Module] FOREIGN KEY([module_code])
REFERENCES [dbo].[Module] ([module_code])
GO

ALTER TABLE [dbo].[Module_Exam] CHECK CONSTRAINT [FK_Module_Exam_Module]
GO
```



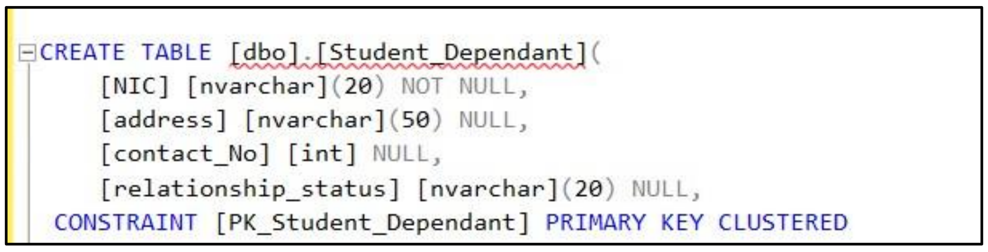
```
CREATE TABLE [dbo].[Module](
	[module_code] [nvarchar](10) NOT NULL,
	[evaluation_type] [nvarchar](50) NULL,
	[module_name] [nvarchar](50) NULL,
 CONSTRAINT [PK_Module] PRIMARY KEY CLUSTERED
```



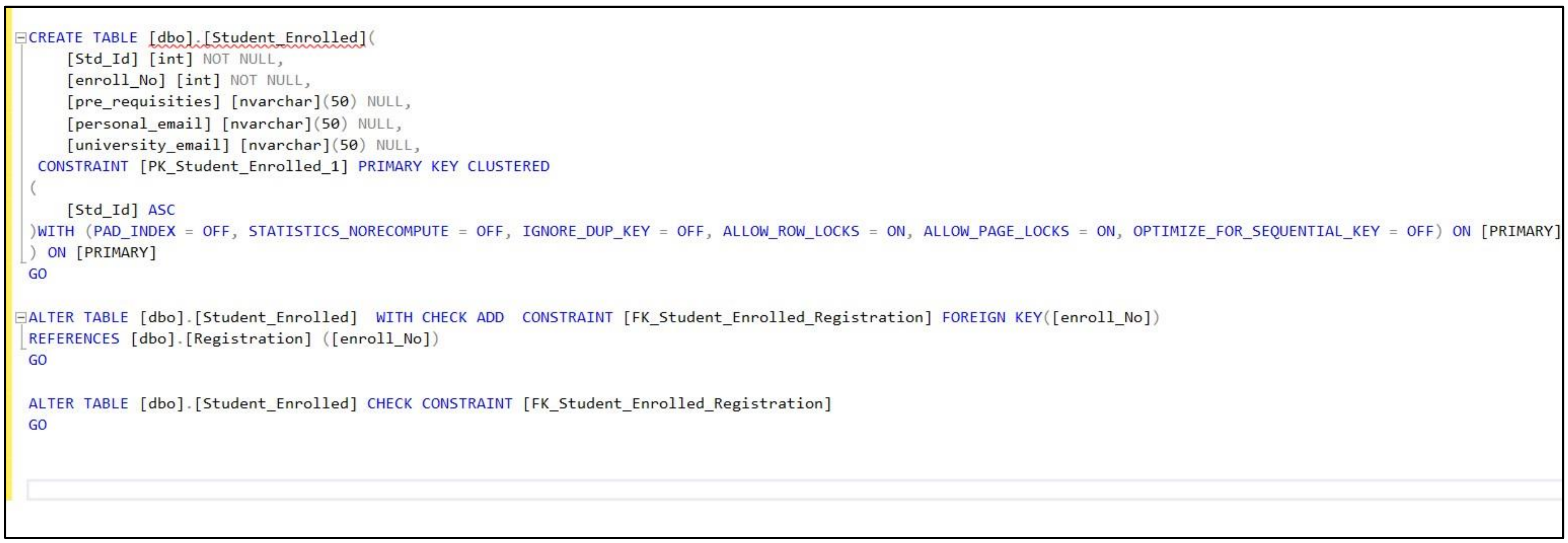
```
CREATE TABLE [dbo].[Registration](
	[enroll_No] [int] NOT NULL,
	[Std_name] [nvarchar](50) NULL,
 CONSTRAINT [PK_Registration] PRIMARY KEY CLUSTERED
```



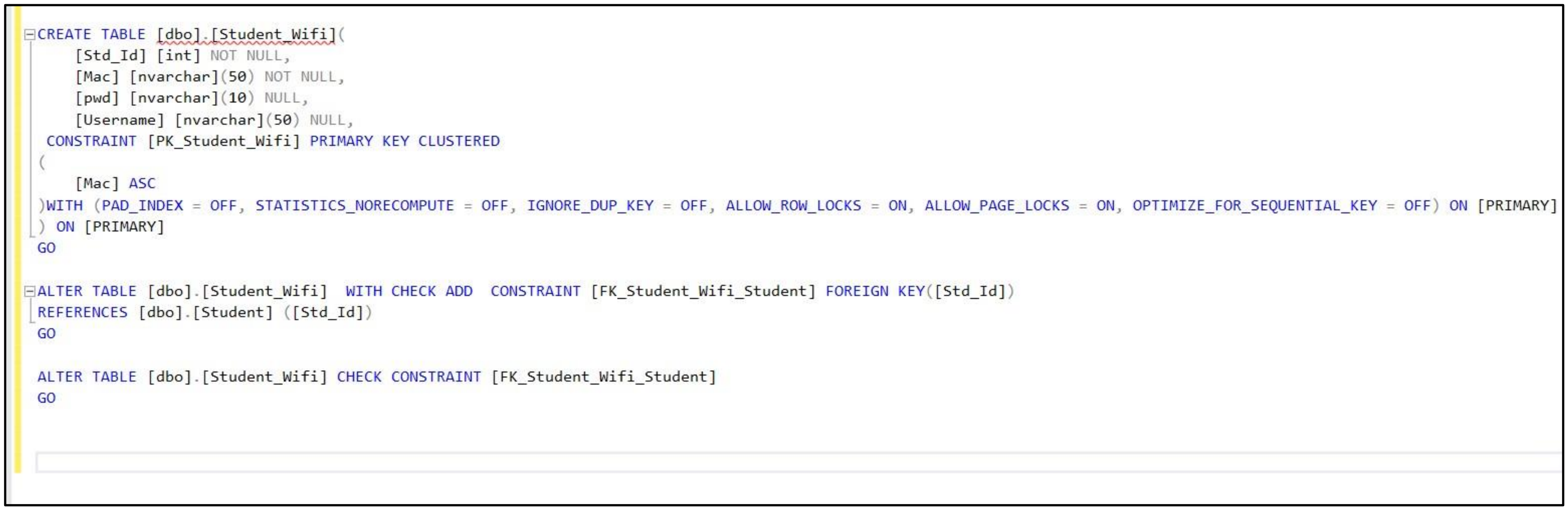
```
CREATE TABLE [dbo].[Student_Department](
	[Std_Id] [int] NOT NULL,
	[module_code] [nvarchar](10) NULL,
 CONSTRAINT [PK_Student_Department] PRIMARY KEY CLUSTERED
(
	[Std_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student_Department]  WITH CHECK ADD  CONSTRAINT [FK_Student_Department_Student_Department] FOREIGN KEY([Std_Id])
REFERENCES [dbo].[Student_Department] ([Std_Id])
GO

ALTER TABLE [dbo].[Student_Department] CHECK CONSTRAINT [FK_Student_Department_Student_Department]
GO
```



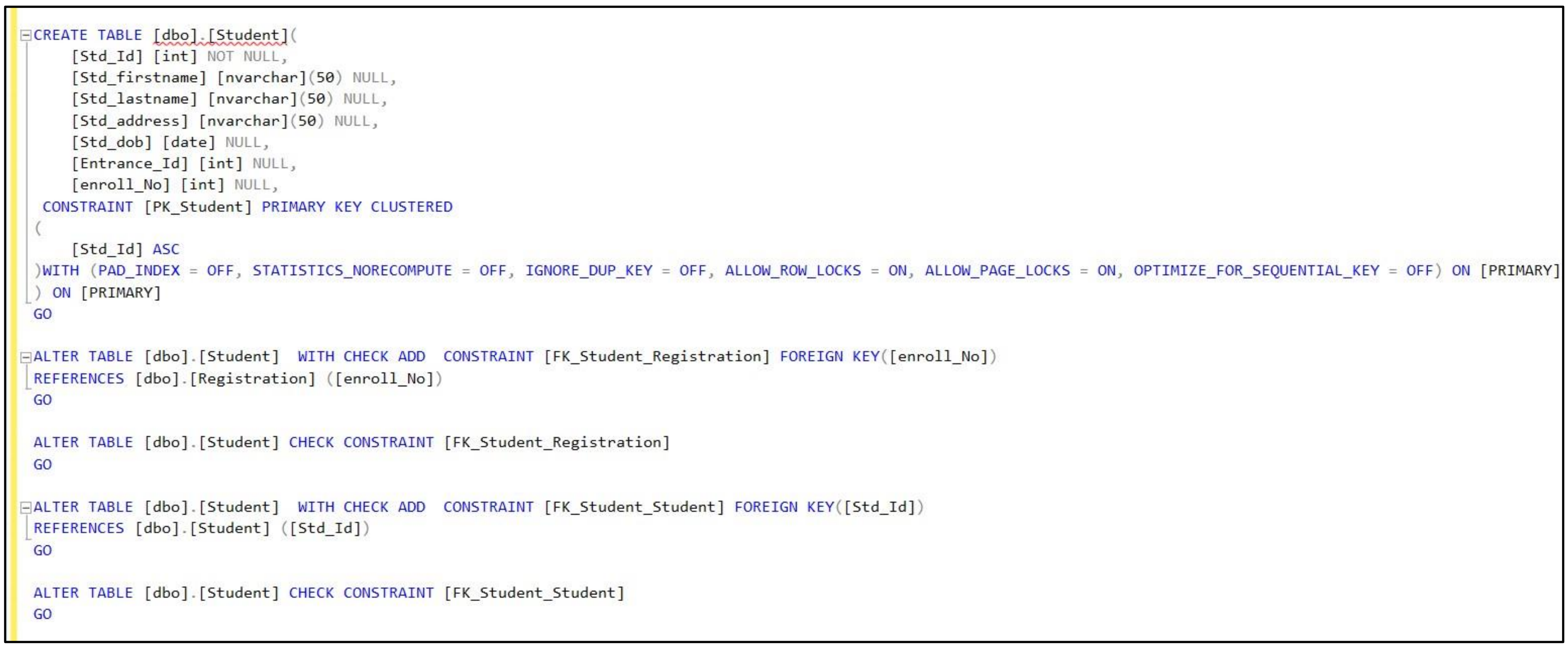
```
CREATE TABLE [dbo].[Student_Dependant](
	[NIC] [nvarchar](20) NOT NULL,
	[address] [nvarchar](50) NULL,
	[contact_No] [int] NULL,
	[relationship_status] [nvarchar](20) NULL,
 CONSTRAINT [PK_Student_Dependant] PRIMARY KEY CLUSTERED
```



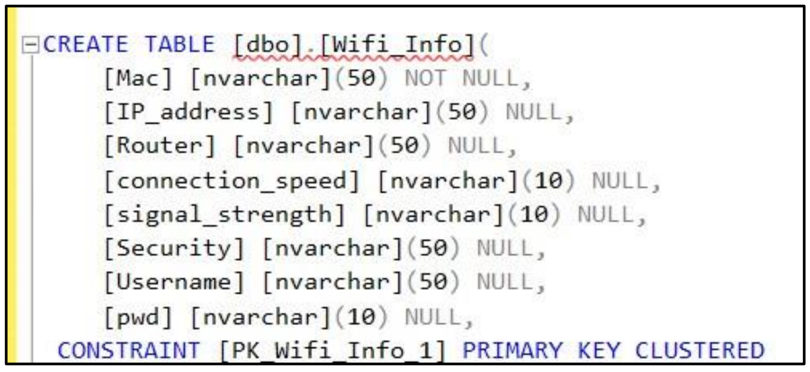
```sql
CREATE TABLE [dbo].[Student_Enrolled](
    [Std_Id] [int] NOT NULL,
    [enroll_No] [int] NOT NULL,
    [pre_requisities] [nvarchar](50) NULL,
    [personal_email] [nvarchar](50) NULL,
    [university_email] [nvarchar](50) NULL,
 CONSTRAINT [PK_Student_Enrolled_1] PRIMARY KEY CLUSTERED
(
    [Std_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student_Enrolled]  WITH CHECK ADD  CONSTRAINT [FK_Student_Enrolled_Registration] FOREIGN KEY([enroll_No])
REFERENCES [dbo].[Registration] ([enroll_No])
GO

ALTER TABLE [dbo].[Student_Enrolled] CHECK CONSTRAINT [FK_Student_Enrolled_Registration]
GO
```



```sql
CREATE TABLE [dbo].[Student_Wifi](
    [Std_Id] [int] NOT NULL,
    [Mac] [nvarchar](50) NOT NULL,
    [pwd] [nvarchar](10) NULL,
    [Username] [nvarchar](50) NULL,
 CONSTRAINT [PK_Student_Wifi] PRIMARY KEY CLUSTERED
(
    [Mac] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student_Wifi]  WITH CHECK ADD  CONSTRAINT [FK_Student_Wifi_Student] FOREIGN KEY([Std_Id])
REFERENCES [dbo].[Student] ([Std_Id])
GO

ALTER TABLE [dbo].[Student_Wifi] CHECK CONSTRAINT [FK_Student_Wifi_Student]
GO
```



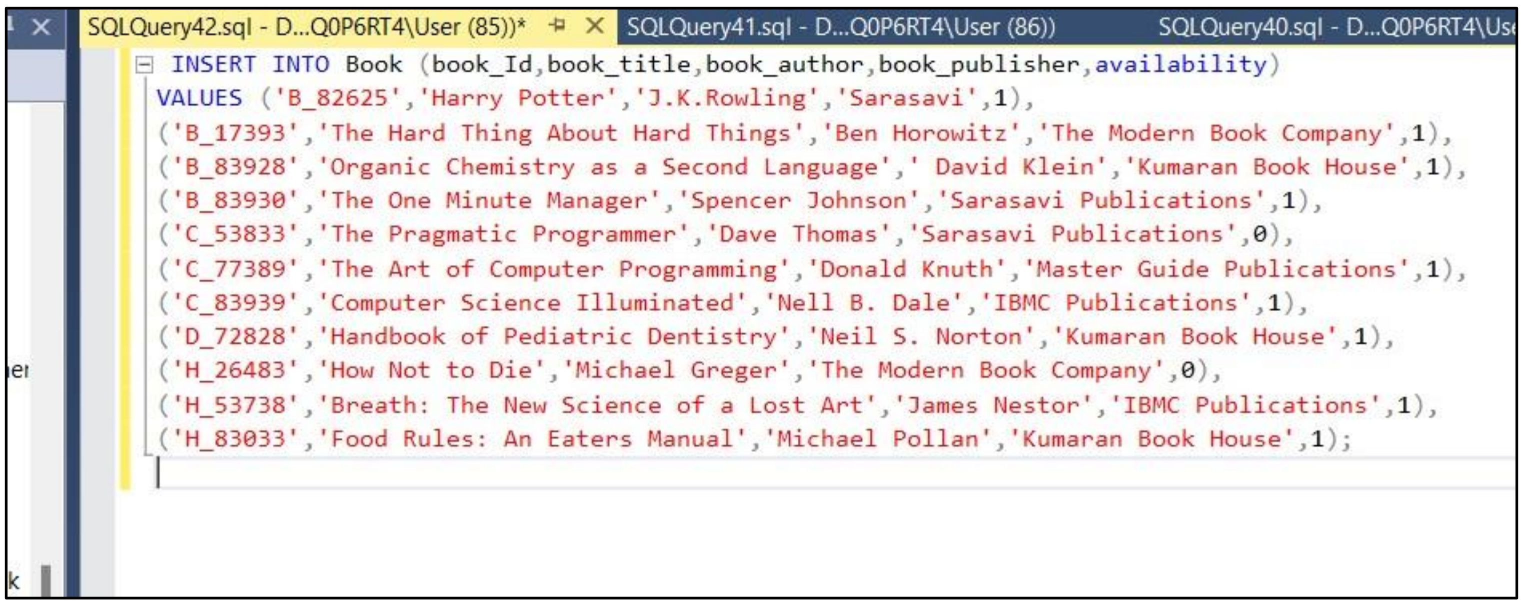
```sql
CREATE TABLE [dbo].[Student](
    [Std_Id] [int] NOT NULL,
    [Std_firstname] [nvarchar](50) NULL,
    [Std_lastname] [nvarchar](50) NULL,
    [Std_address] [nvarchar](50) NULL,
    [Std_dob] [date] NULL,
    [Entrance_Id] [int] NULL,
    [enroll_No] [int] NULL,
 CONSTRAINT [PK_Student] PRIMARY KEY CLUSTERED
(
    [Std_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student]  WITH CHECK ADD  CONSTRAINT [FK_Student_Registration] FOREIGN KEY([enroll_No])
REFERENCES [dbo].[Registration] ([enroll_No])
GO

ALTER TABLE [dbo].[Student] CHECK CONSTRAINT [FK_Student_Registration]
GO

ALTER TABLE [dbo].[Student]  WITH CHECK ADD  CONSTRAINT [FK_Student_Student] FOREIGN KEY([Std_Id])
REFERENCES [dbo].[Student] ([Std_Id])
GO

ALTER TABLE [dbo].[Student] CHECK CONSTRAINT [FK_Student_Student]
GO
```



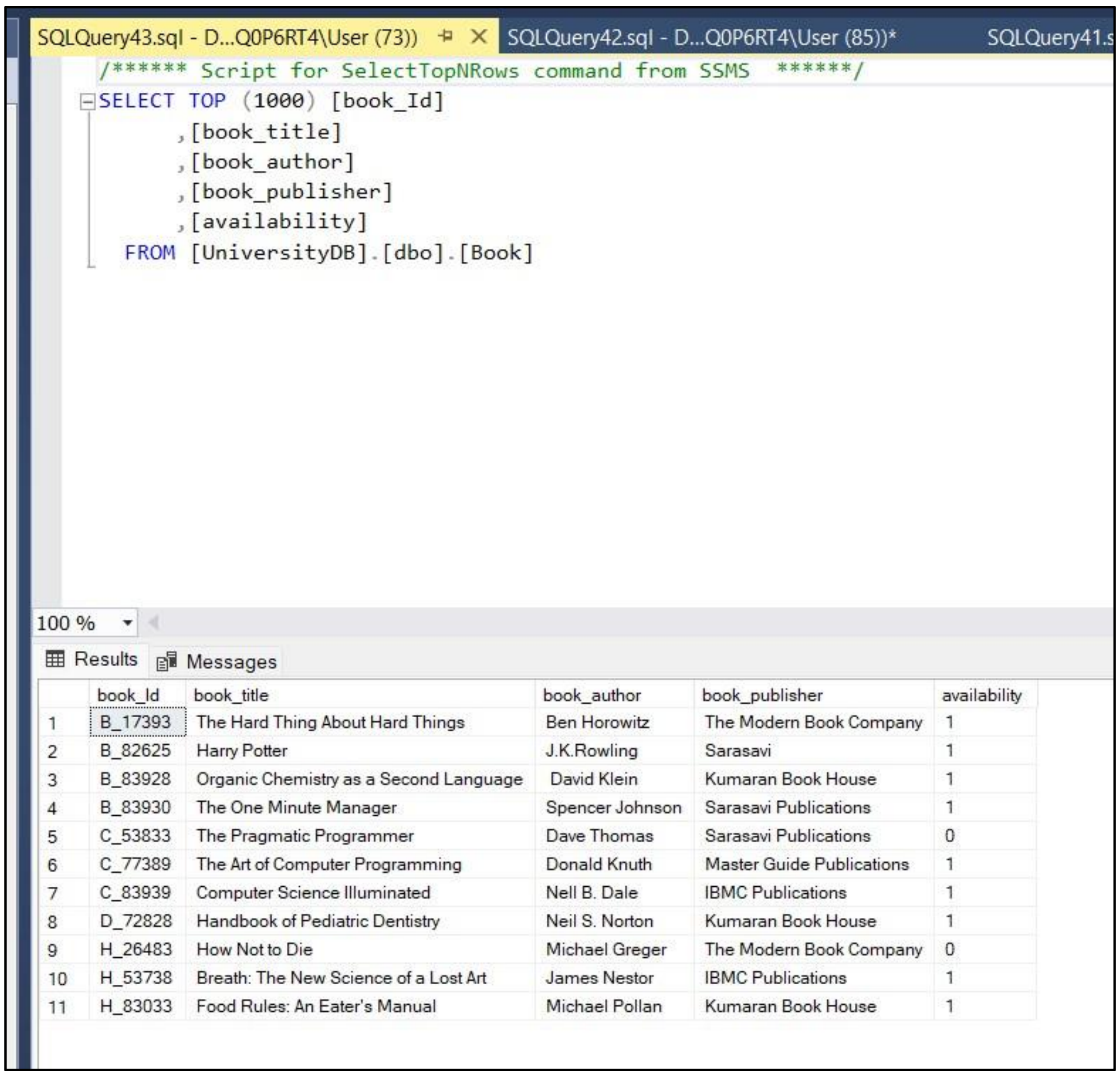
```sql
CREATE TABLE [dbo].[Wifi_Info](
    [Mac] [nvarchar](50) NOT NULL,
    [IP_address] [nvarchar](50) NULL,
    [Router] [nvarchar](50) NULL,
    [connection_speed] [nvarchar](10) NULL,
    [signal_strength] [nvarchar](10) NULL,
    [Security] [nvarchar](50) NULL,
    [Username] [nvarchar](50) NULL,
    [pwd] [nvarchar](10) NULL,
 CONSTRAINT [PK_Wifi_Info_1] PRIMARY KEY CLUSTERED
```

Insert Data Into Tables Queries

(Book Table)


SQLQuery42.sql - D...Q0P6RT4\User (85))*
SQLQuery41.sql - D...Q0P6RT4\User (86))
SQLQuery40.sql - D...Q0P6RT4\Us

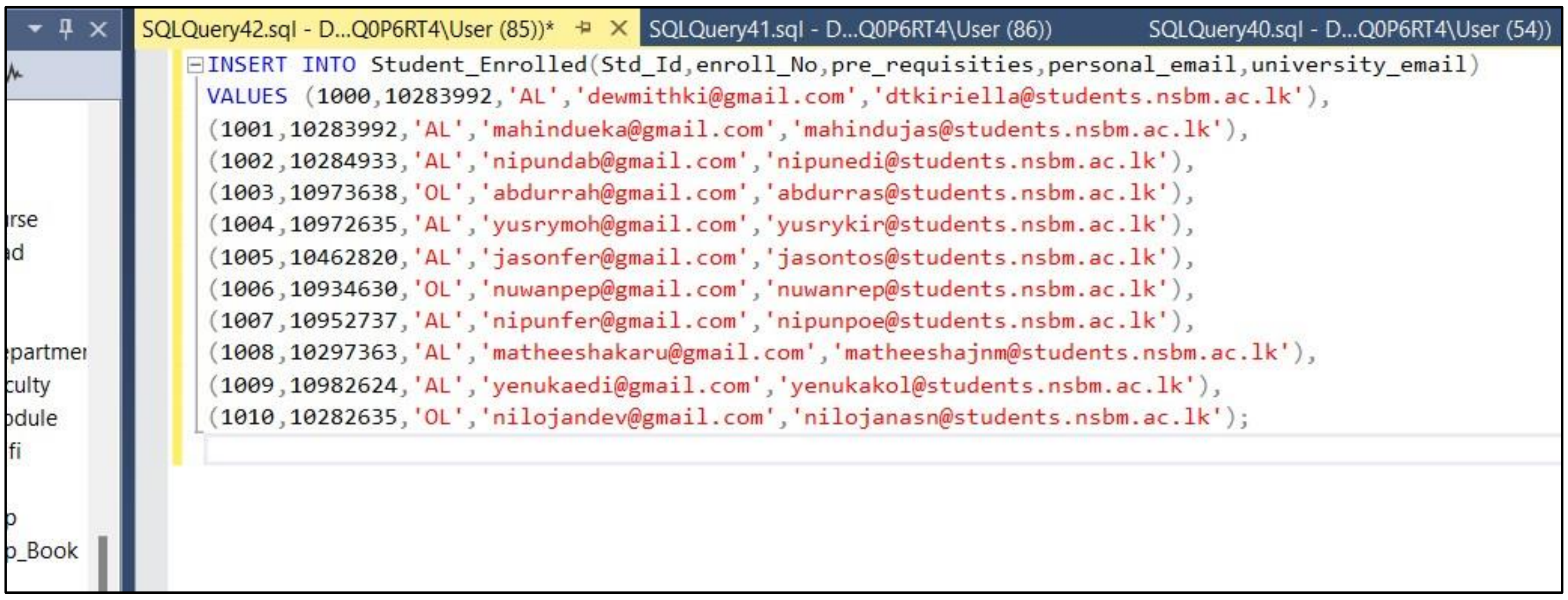
```sql
INSERT INTO Book (book_Id,book_title,book_author,book_publisher,availability)
VALUES ('B_82625','Harry Potter','J.K.Rowling','Sarasavi',1),
('B_17393','The Hard Thing About Hard Things','Ben Horowitz','The Modern Book Company',1),
('B_83928','Organic Chemistry as a Second Language',' David Klein','Kumaran Book House',1),
('B_83930','The One Minute Manager','Spencer Johnson','Sarasavi Publications',1),
('C_53833','The Pragmatic Programmer','Dave Thomas','Sarasavi Publications',0),
('C_77389','The Art of Computer Programming','Donald Knuth','Master Guide Publications',1),
('C_83939','Computer Science Illuminated','Nell B. Dale','IBMC Publications',1),
('D_72828','Handbook of Pediatric Dentistry','Neil S. Norton','Kumaran Book House',1),
('H_26483','How Not to Die','Michael Greger','The Modern Book Company',0),
('H_53738','Breath: The New Science of a Lost Art','James Nestor','IBMC Publications',1),
('H_83033','Food Rules: An Eaters Manual','Michael Pollan','Kumaran Book House',1);
```


SQLQuery43.sql - D...Q0P6RT4\User (73))
SQLQuery42.sql - D...Q0P6RT4\User (85))*
SQLQuery41.s
100 %
Results
Messages

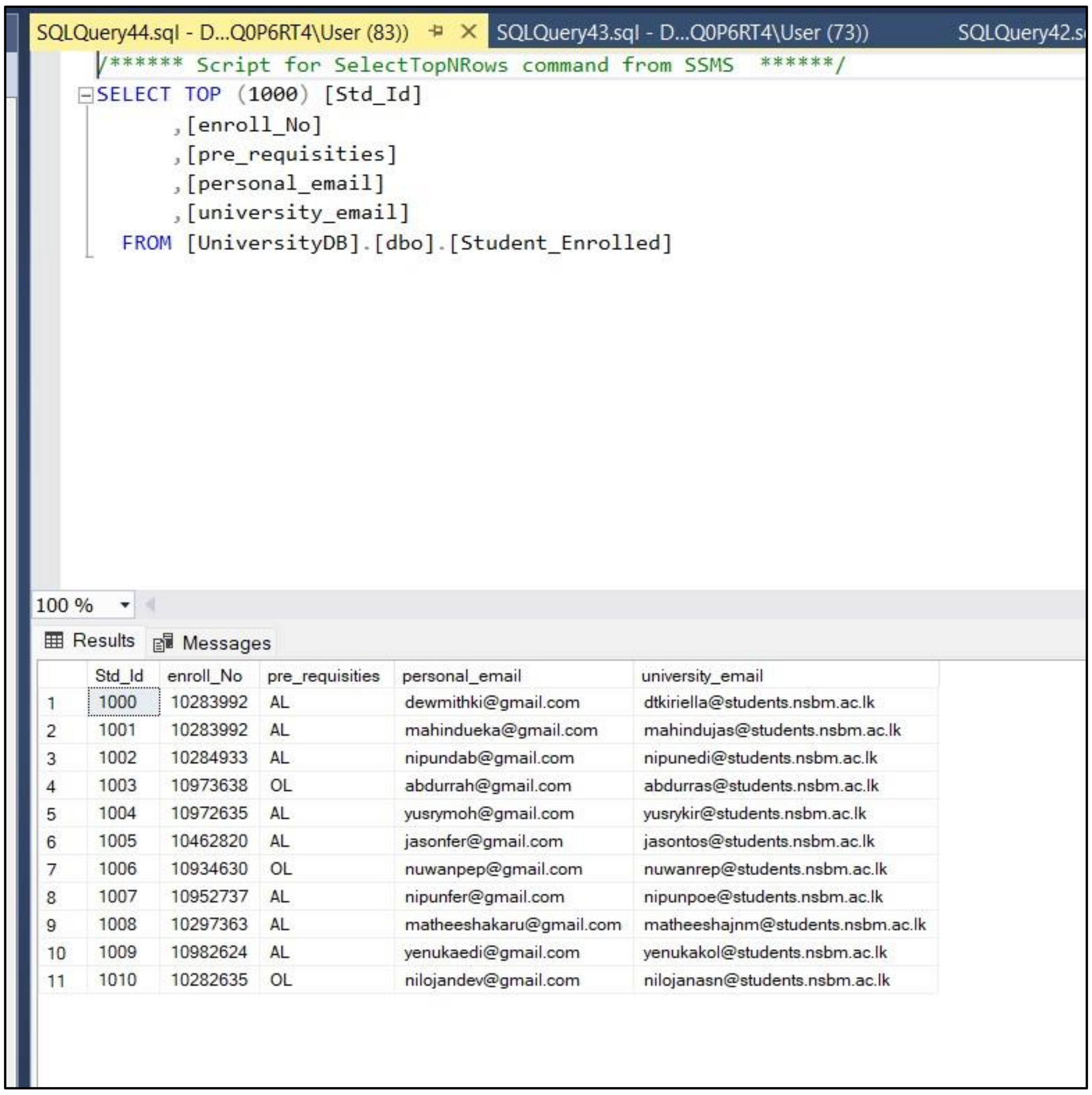
```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [book_Id]
      ,[book_title]
      ,[book_author]
      ,[book_publisher]
      ,[availability]
  FROM [UniversityDB].[dbo].[Book]
```

	book_Id	book_title	book_author	book_publisher	availability
1	B_17393	The Hard Thing About Hard Things	Ben Horowitz	The Modern Book Company	1
2	B_82625	Harry Potter	J.K.Rowling	Sarasavi	1
3	B_83928	Organic Chemistry as a Second Language	David Klein	Kumaran Book House	1
4	B_83930	The One Minute Manager	Spencer Johnson	Sarasavi Publications	1
5	C_53833	The Pragmatic Programmer	Dave Thomas	Sarasavi Publications	0
6	C_77389	The Art of Computer Programming	Donald Knuth	Master Guide Publications	1
7	C_83939	Computer Science Illuminated	Nell B. Dale	IBMC Publications	1
8	D_72828	Handbook of Pediatric Dentistry	Neil S. Norton	Kumaran Book House	1
9	H_26483	How Not to Die	Michael Greger	The Modern Book Company	0
10	H_53738	Breath: The New Science of a Lost Art	James Nestor	IBMC Publications	1
11	H_83033	Food Rules: An Eater's Manual	Michael Pollan	Kumaran Book House	1

(Student Enrolled Table)


SQLQuery42.sql - D...Q0P6RT4\User (85))*
SQLQuery41.sql - D...Q0P6RT4\User (86))
SQLQuery40.sql - D...Q0P6RT4\User (54))

```
INSERT INTO Student_Enrolled(Std_Id,enroll_No,pre_requisities,personal_email,university_email)
VALUES (1000,10283992,'AL','dewmithki@gmail.com','dtkiriella@students.nsbm.ac.lk'),
(1001,10283992,'AL','mahindueka@gmail.com','mahindujas@students.nsbm.ac.lk'),
(1002,10284933,'AL','nipundab@gmail.com','nipunedi@students.nsbm.ac.lk'),
(1003,10973638,'OL','abdurrah@gmail.com','abdurras@students.nsbm.ac.lk'),
(1004,10972635,'AL','yusrymoh@gmail.com','yusrykir@students.nsbm.ac.lk'),
(1005,10462820,'AL','jasonfer@gmail.com','jasontos@students.nsbm.ac.lk'),
(1006,10934630,'OL','nuwanpep@gmail.com','nuwanrep@students.nsbm.ac.lk'),
(1007,10952737,'AL','nipunfer@gmail.com','nipunpoe@students.nsbm.ac.lk'),
(1008,10297363,'AL','matheeshakaru@gmail.com','matheeshajnm@students.nsbm.ac.lk'),
(1009,10982624,'AL','yenukaedi@gmail.com','yenukakol@students.nsbm.ac.lk'),
(1010,10282635,'OL','nilojandev@gmail.com','nilojanasn@students.nsbm.ac.lk');
```

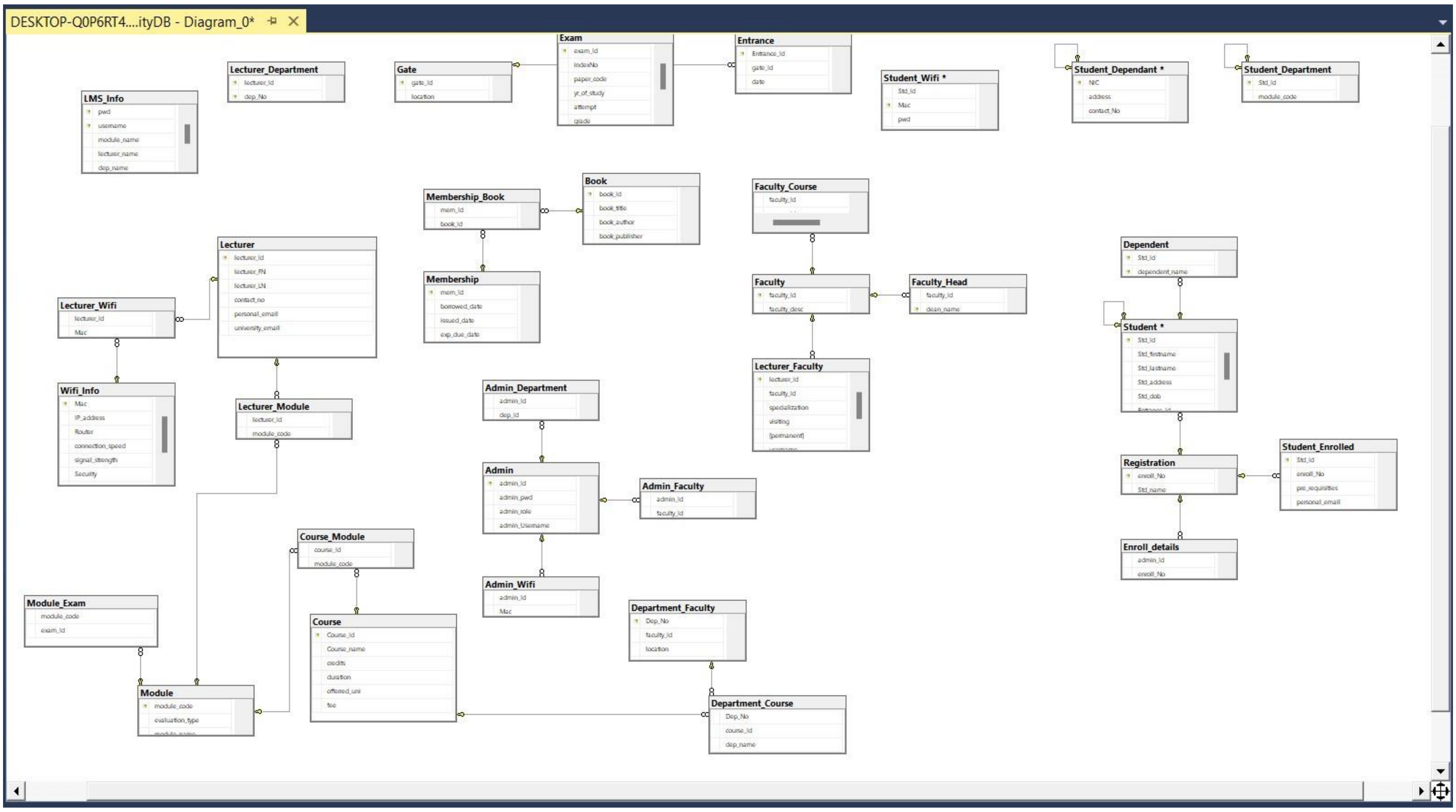

SQLQuery44.sql - D...Q0P6RT4\User (83))
SQLQuery43.sql - D...Q0P6RT4\User (73))
SQLQuery42.s
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Std_Id]
      ,[enroll_No]
      ,[pre_requisities]
      ,[personal_email]
      ,[university_email]
  FROM [UniversityDB].[dbo].[Student_Enrolled]
```

	Std_Id	enroll_No	pre_requisities	personal_email	university_email
1	1000	10283992	AL	dewmithki@gmail.com	dtkiriella@students.nsbm.ac.lk
2	1001	10283992	AL	mahindueka@gmail.com	mahindujas@students.nsbm.ac.lk
3	1002	10284933	AL	nipundab@gmail.com	nipunedi@students.nsbm.ac.lk
4	1003	10973638	OL	abdurrah@gmail.com	abdurras@students.nsbm.ac.lk
5	1004	10972635	AL	yusrymoh@gmail.com	yusrykir@students.nsbm.ac.lk
6	1005	10462820	AL	jasonfer@gmail.com	jasontos@students.nsbm.ac.lk
7	1006	10934630	OL	nuwanpep@gmail.com	nuwanrep@students.nsbm.ac.lk
8	1007	10952737	AL	nipunfer@gmail.com	nipunpoe@students.nsbm.ac.lk
9	1008	10297363	AL	matheeshakaru@gmail.com	matheeshajnm@students.nsbm.ac.lk
10	1009	10982624	AL	yenukaedi@gmail.com	yenukakol@students.nsbm.ac.lk
11	1010	10282635	OL	nilojandev@gmail.com	nilojanasn@students.nsbm.ac.lk

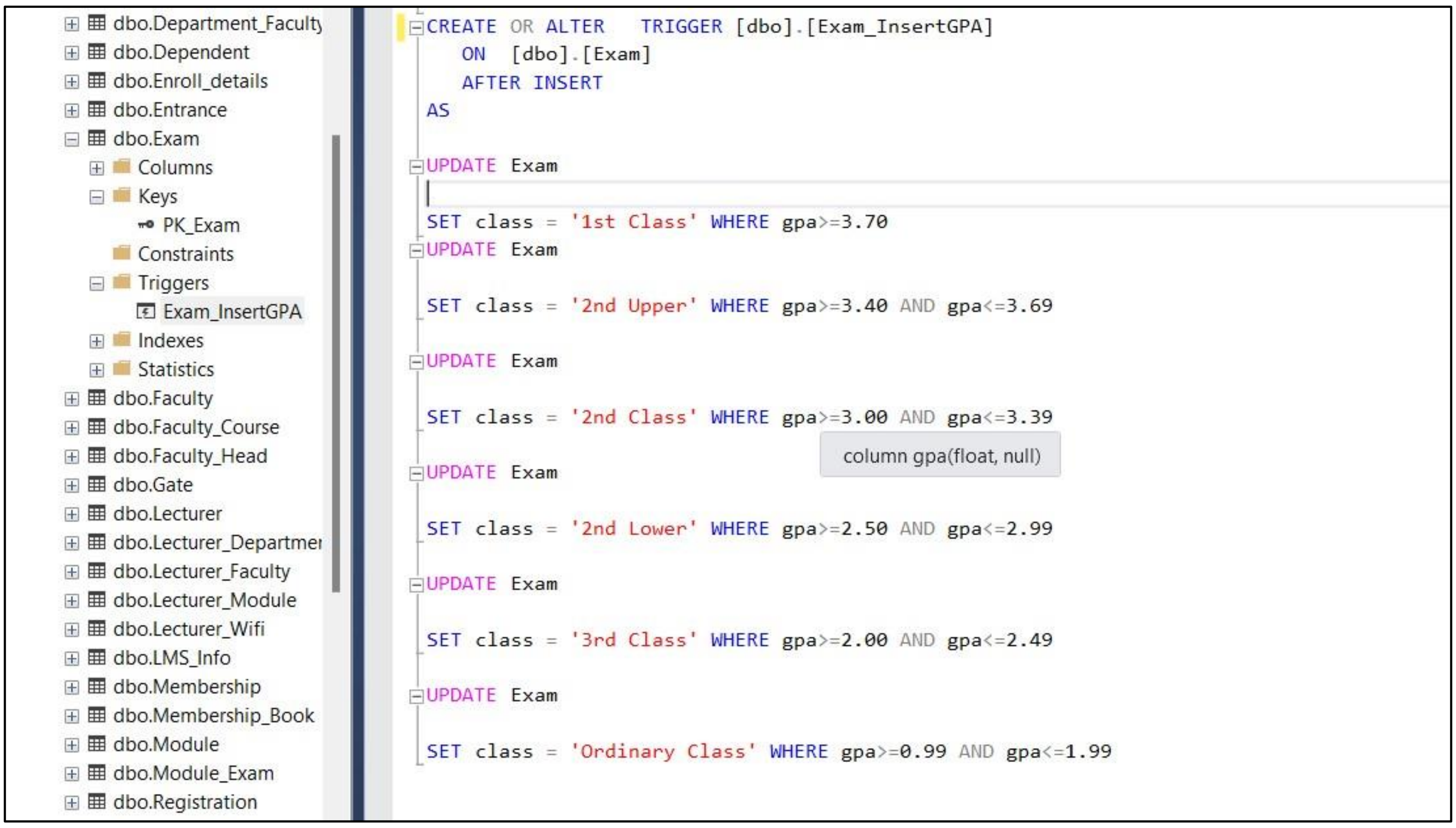
We have used the same method to insert data into other tables as well. We didn’t attach the Screenshots of all the table Data insertion queries as it is the same process. However the tables with data can be seen in our database file (.bak)

DataBase Diagram

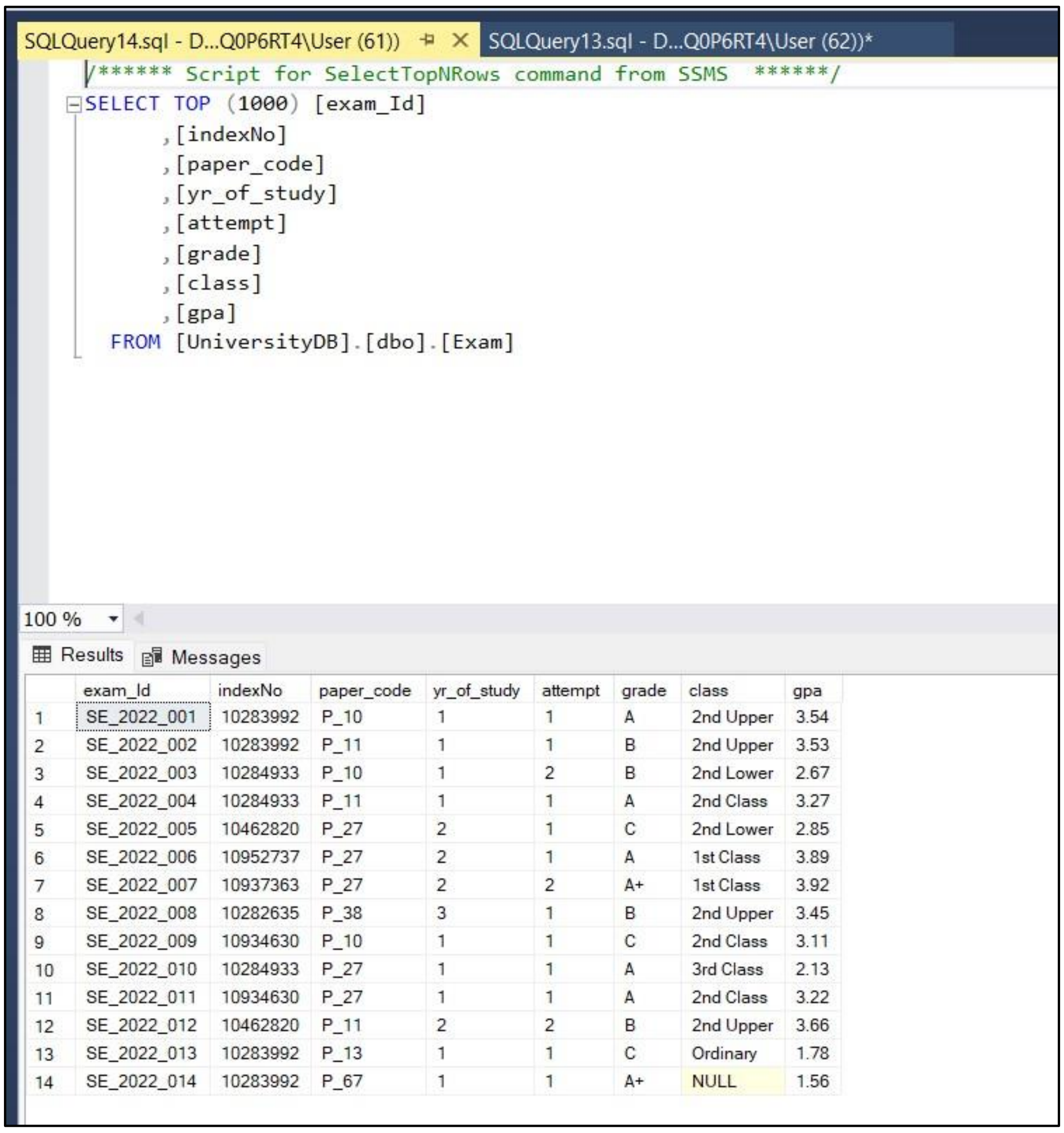


SQL Triggers

Trigger 1



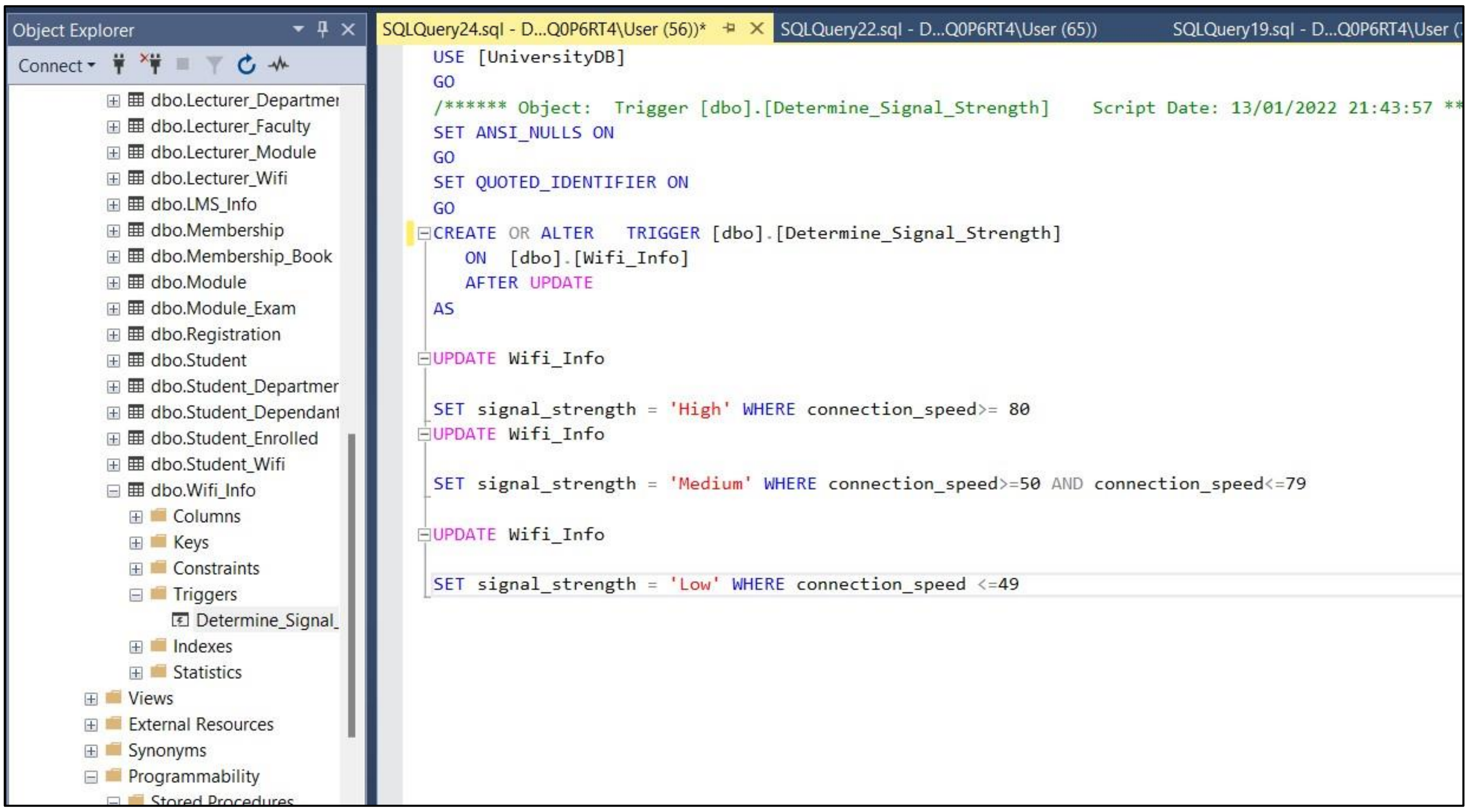
```
CREATE OR ALTER   TRIGGER [dbo].[Exam_InsertGPA]
   ON  [dbo].[Exam]
   AFTER INSERT
AS

UPDATE Exam

SET class = '1st Class' WHERE gpa>=3.70
UPDATE Exam

SET class = '2nd Upper' WHERE gpa>=3.40 AND gpa<=3.69

UPDATE Exam

SET class = '2nd Class' WHERE gpa>=3.00 AND gpa<=3.39

UPDATE Exam

SET class = '2nd Lower' WHERE gpa>=2.50 AND gpa<=2.99

UPDATE Exam

SET class = '3rd Class' WHERE gpa>=2.00 AND gpa<=2.49

UPDATE Exam

SET class = 'Ordinary Class' WHERE gpa>=0.99 AND gpa<=1.99
```


SQLQuery14.sql - D...Q0P6RT4\User (61))
SQLQuery13.sql - D...Q0P6RT4\User (62))*
100 %
Results
Messages

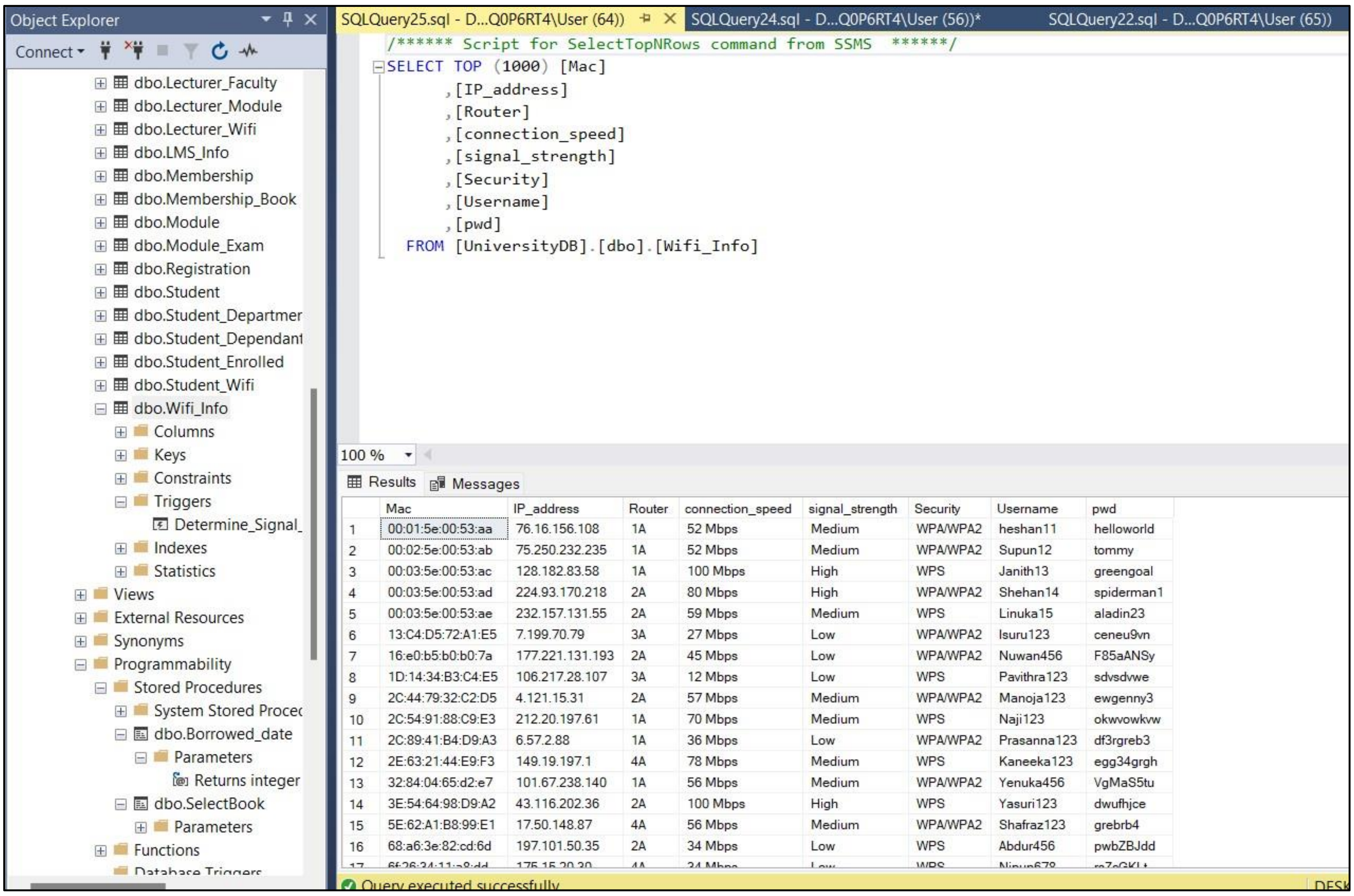
```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [exam_Id]
      ,[indexNo]
      ,[paper_code]
      ,[yr_of_study]
      ,[attempt]
      ,[grade]
      ,[class]
      ,[gpa]
  FROM [UniversityDB].[dbo].[Exam]
```

	exam_Id	indexNo	paper_code	yr_of_study	attempt	grade	class	gpa
1	SE_2022_001	10283992	P_10	1	1	A	2nd Upper	3.54
2	SE_2022_002	10283992	P_11	1	1	B	2nd Upper	3.53
3	SE_2022_003	10284933	P_10	1	2	B	2nd Lower	2.67
4	SE_2022_004	10284933	P_11	1	1	A	2nd Class	3.27
5	SE_2022_005	10462820	P_27	2	1	C	2nd Lower	2.85
6	SE_2022_006	10952737	P_27	2	1	A	1st Class	3.89
7	SE_2022_007	10937363	P_27	2	2	A+	1st Class	3.92
8	SE_2022_008	10282635	P_38	3	1	B	2nd Upper	3.45
9	SE_2022_009	10934630	P_10	1	1	C	2nd Class	3.11
10	SE_2022_010	10284933	P_27	1	1	A	3rd Class	2.13
11	SE_2022_011	10934630	P_27	1	1	A	2nd Class	3.22
12	SE_2022_012	10462820	P_11	2	2	B	2nd Upper	3.66
13	SE_2022_013	10283992	P_13	1	1	C	Ordinary	1.78
14	SE_2022_014	10283992	P_67	1	1	A+	NULL	1.56

Trigger 2



```
USE [UniversityDB]
GO
/****** Object:  Trigger [dbo].[Determine_Signal_Strength]    Script Date: 13/01/2022 21:43:57 **
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE OR ALTER   TRIGGER [dbo].[Determine_Signal_Strength]
   ON  [dbo].[Wifi_Info]
   AFTER UPDATE
AS

UPDATE Wifi_Info

SET signal_strength = 'High' WHERE connection_speed>= 80
UPDATE Wifi_Info

SET signal_strength = 'Medium' WHERE connection_speed>=50 AND connection_speed<=79

UPDATE Wifi_Info

SET signal_strength = 'Low' WHERE connection_speed <=49
```



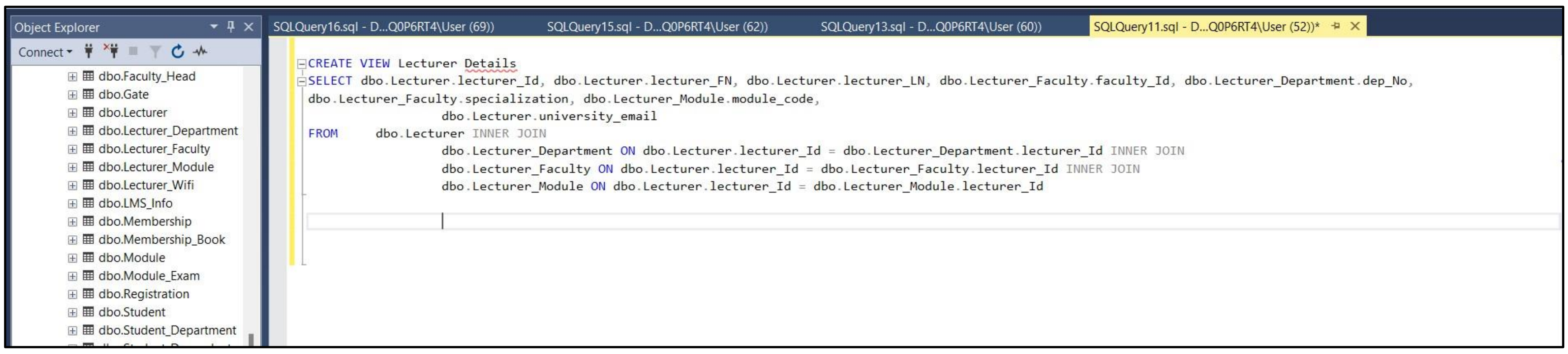
```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Mac]
      ,[IP_address]
      ,[Router]
      ,[connection_speed]
      ,[signal_strength]
      ,[Security]
      ,[Username]
      ,[pwd]
  FROM [UniversityDB].[dbo].[Wifi_Info]
```

	Mac	IP_address	Router	connection_speed	signal_strength	Security	Username	pwd
1	00:01:5e:00:53:aa	76.16.156.108	1A	52 Mbps	Medium	WPA/WPA2	heshan11	helloworld
2	00:02:5e:00:53:ab	75.250.232.235	1A	52 Mbps	Medium	WPA/WPA2	Supun12	tommy
3	00:03:5e:00:53:ac	128.182.83.58	1A	100 Mbps	High	WPS	Janith13	greengoal
4	00:03:5e:00:53:ad	224.93.170.218	2A	80 Mbps	High	WPA/WPA2	Shehan14	spiderman1
5	00:03:5e:00:53:ae	232.157.131.55	2A	59 Mbps	Medium	WPS	Linuka15	aladin23
6	13:C4:D5:72:A1:E5	7.199.70.79	3A	27 Mbps	Low	WPA/WPA2	Isuru123	ceneu9vn
7	16:e0:b5:b0:b0:7a	177.221.131.193	2A	45 Mbps	Low	WPA/WPA2	Nuwan456	F85aANSy
8	1D:14:34:B3:C4:E5	106.217.28.107	3A	12 Mbps	Low	WPS	Pavithra123	sdvsdvwe
9	2C:44:79:32:C2:D5	4.121.15.31	2A	57 Mbps	Medium	WPA/WPA2	Manoja123	ewgenny3
10	2C:54:91:88:C9:E3	212.20.197.61	1A	70 Mbps	Medium	WPS	Naji123	okwvowkvw
11	2C:89:41:B4:D9:A3	6.57.2.88	1A	36 Mbps	Low	WPA/WPA2	Prasanna123	df3rgreb3
12	2E:63:21:44:E9:F3	149.19.197.1	4A	78 Mbps	Medium	WPS	Kaneeka123	egg34grgh
13	32:84:04:65:d2:e7	101.67.238.140	1A	56 Mbps	Medium	WPA/WPA2	Yenuka456	VgMaS5tu
14	3E:54:64:98:D9:A2	43.116.202.36	2A	100 Mbps	High	WPS	Yasuri123	dwufhjce
15	5E:62:A1:B8:99:E1	17.50.148.87	4A	56 Mbps	Medium	WPA/WPA2	Shafraz123	grebrb4
16	68:a6:3e:82:cd:6d	197.101.50.35	2A	34 Mbps	Low	WPS	Abdur456	pwbZBJdd
17	6f:26:34:11:a8:dd	175.15.20.30	4A	34 Mbps	Low	WPS	Ninun678	[illegible]

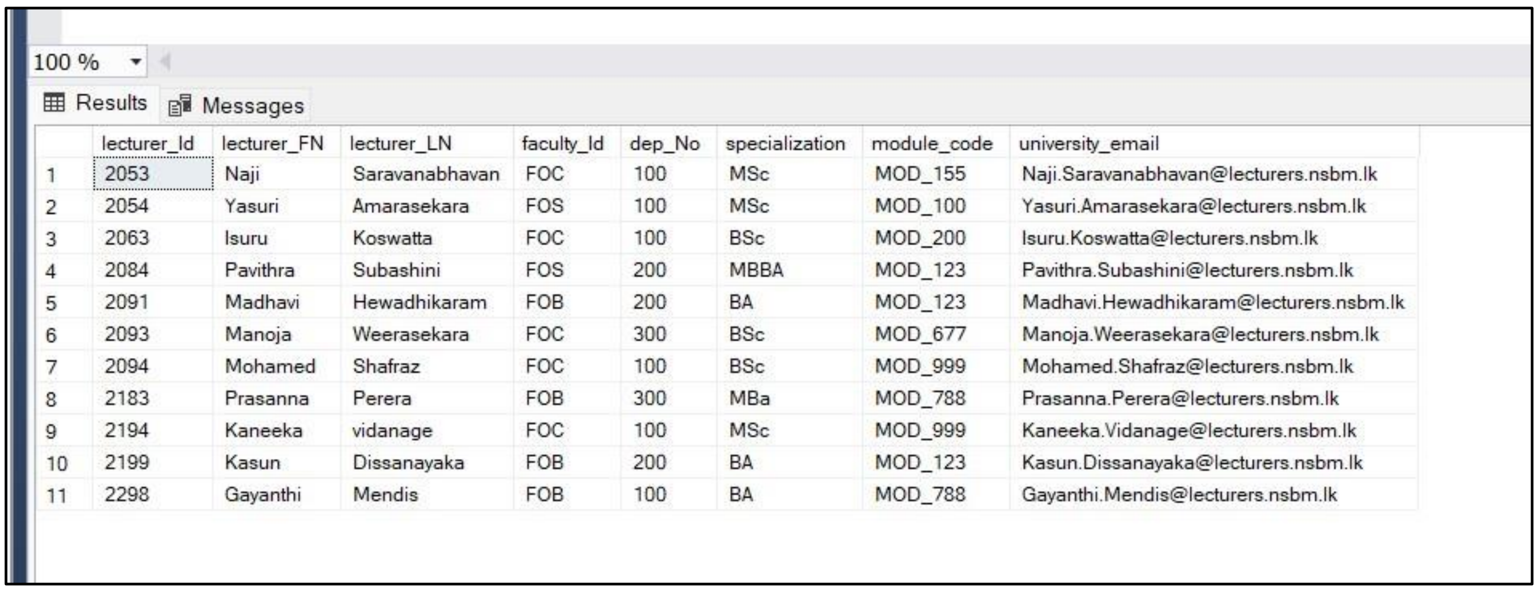
Query executed successfully.

SQL Views

View 1

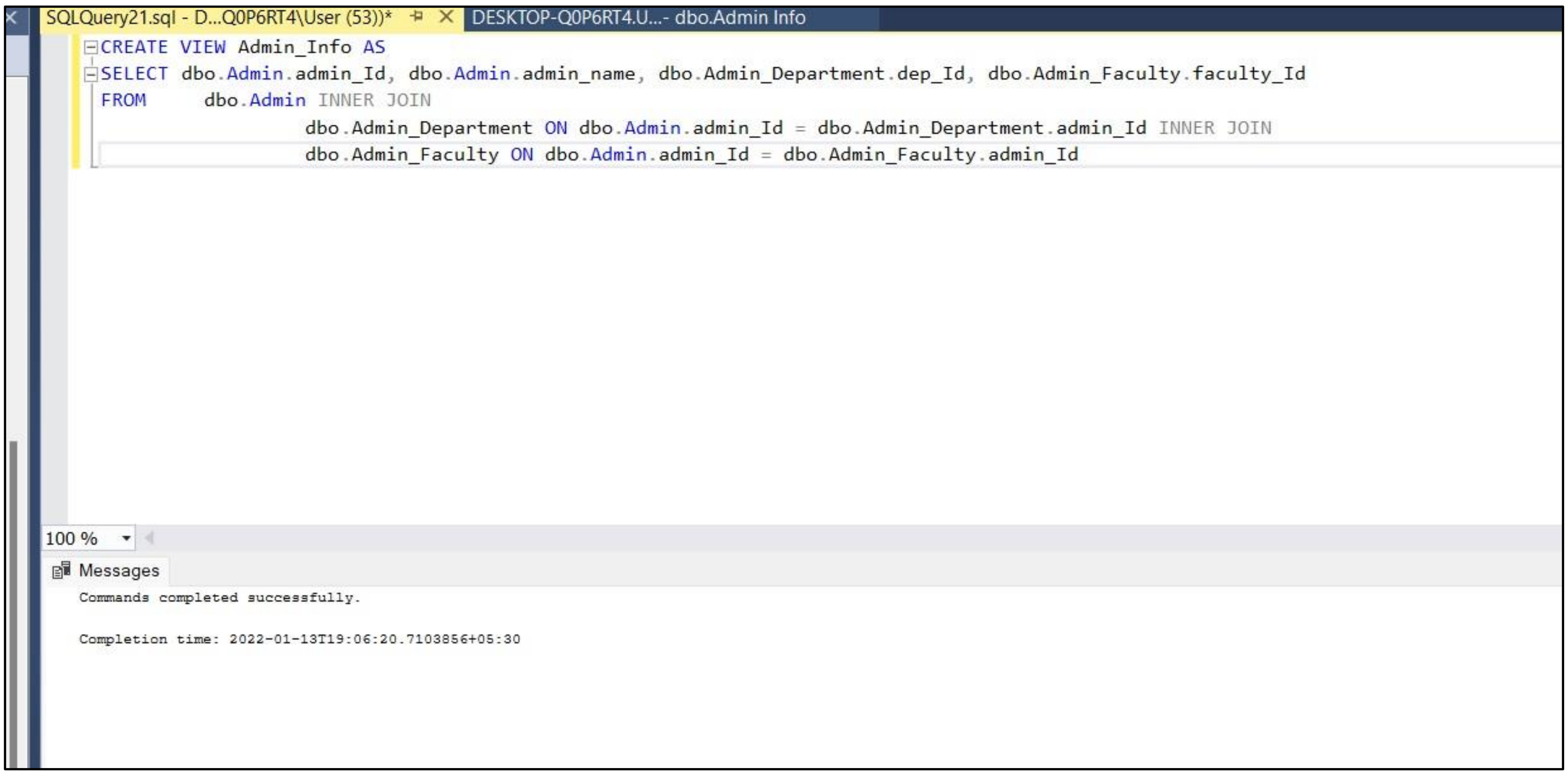


```sql
CREATE VIEW Lecturer Details
SELECT dbo.Lecturer.lecturer_Id, dbo.Lecturer.lecturer_FN, dbo.Lecturer.lecturer_LN, dbo.Lecturer_Faculty.faculty_Id, dbo.Lecturer_Department.dep_No,
dbo.Lecturer_Faculty.specialization, dbo.Lecturer_Module.module_code,
                  dbo.Lecturer.university_email
FROM      dbo.Lecturer INNER JOIN
                  dbo.Lecturer_Department ON dbo.Lecturer.lecturer_Id = dbo.Lecturer_Department.lecturer_Id INNER JOIN
                  dbo.Lecturer_Faculty ON dbo.Lecturer.lecturer_Id = dbo.Lecturer_Faculty.lecturer_Id INNER JOIN
                  dbo.Lecturer_Module ON dbo.Lecturer.lecturer_Id = dbo.Lecturer_Module.lecturer_Id
```



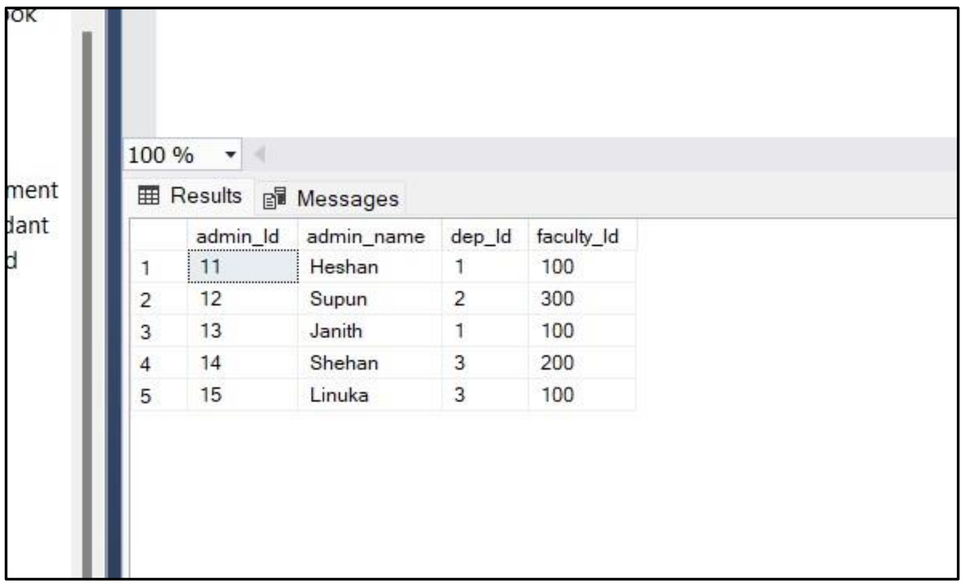
	lecturer_Id	lecturer_FN	lecturer_LN	faculty_Id	dep_No	specialization	module_code	university_email
1	2053	Naji	Saravanabhavan	FOC	100	MSc	MOD_155	Naji.Saravanabhavan@lecturers.nsbm.lk
2	2054	Yasuri	Amarasekara	FOS	100	MSc	MOD_100	Yasuri.Amarasekara@lecturers.nsbm.lk
3	2063	Isuru	Koswatta	FOC	100	BSc	MOD_200	Isuru.Koswatta@lecturers.nsbm.lk
4	2084	Pavithra	Subashini	FOS	200	MBBA	MOD_123	Pavithra.Subashini@lecturers.nsbm.lk
5	2091	Madhavi	Hewadhikaram	FOB	200	BA	MOD_123	Madhavi.Hewadhikaram@lecturers.nsbm.lk
6	2093	Manoja	Weerasekara	FOC	300	BSc	MOD_677	Manoja.Weerasekara@lecturers.nsbm.lk
7	2094	Mohamed	Shafraz	FOC	100	BSc	MOD_999	Mohamed.Shafraz@lecturers.nsbm.lk
8	2183	Prasanna	Perera	FOB	300	MBa	MOD_788	Prasanna.Perera@lecturers.nsbm.lk
9	2194	Kaneeka	vidanage	FOC	100	MSc	MOD_999	Kaneeka.Vidanage@lecturers.nsbm.lk
10	2199	Kasun	Dissanayaka	FOB	200	BA	MOD_123	Kasun.Dissanayaka@lecturers.nsbm.lk
11	2298	Gayanthi	Mendis	FOB	100	BA	MOD_788	Gayanthi.Mendis@lecturers.nsbm.lk

View 2


SQLQuery21.sql - D...Q0P6RT4\User (53))*
DESKTOP-Q0P6RT4.U...- dbo.Admin Info
100 %
Messages

```
CREATE VIEW Admin_Info AS
SELECT dbo.Admin.admin_Id, dbo.Admin.admin_name, dbo.Admin_Department.dep_Id, dbo.Admin_Faculty.faculty_Id
FROM      dbo.Admin INNER JOIN
                  dbo.Admin_Department ON dbo.Admin.admin_Id = dbo.Admin_Department.admin_Id INNER JOIN
                  dbo.Admin_Faculty ON dbo.Admin.admin_Id = dbo.Admin_Faculty.admin_Id
```

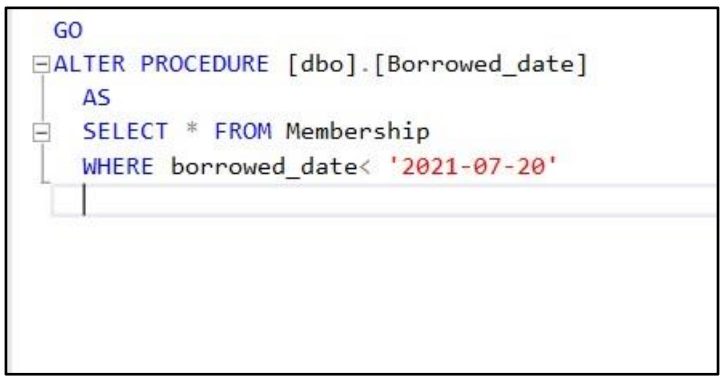
```
Commands completed successfully.

Completion time: 2022-01-13T19:06:20.7103856+05:30
```



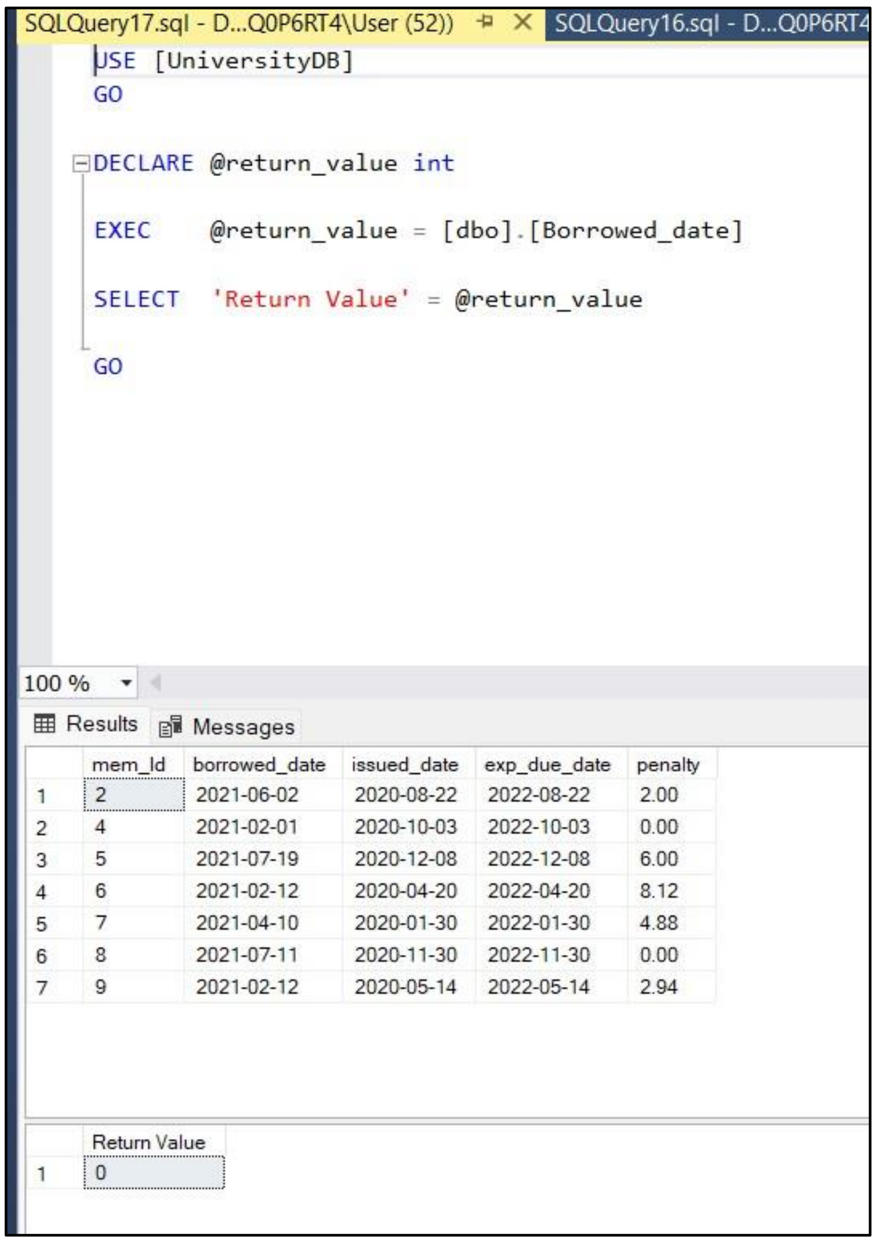
	admin_Id	admin_name	dep_Id	faculty_Id
1	11	Heshan	1	100
2	12	Supun	2	300
3	13	Janith	1	100
4	14	Shehan	3	200
5	15	Linuka	3	100

Stored Procedures

Stored Procedure 1



```
GO
ALTER PROCEDURE [dbo].[Borrowed_date]
  AS
  SELECT * FROM Membership
  WHERE borrowed_date< '2021-07-20'
```

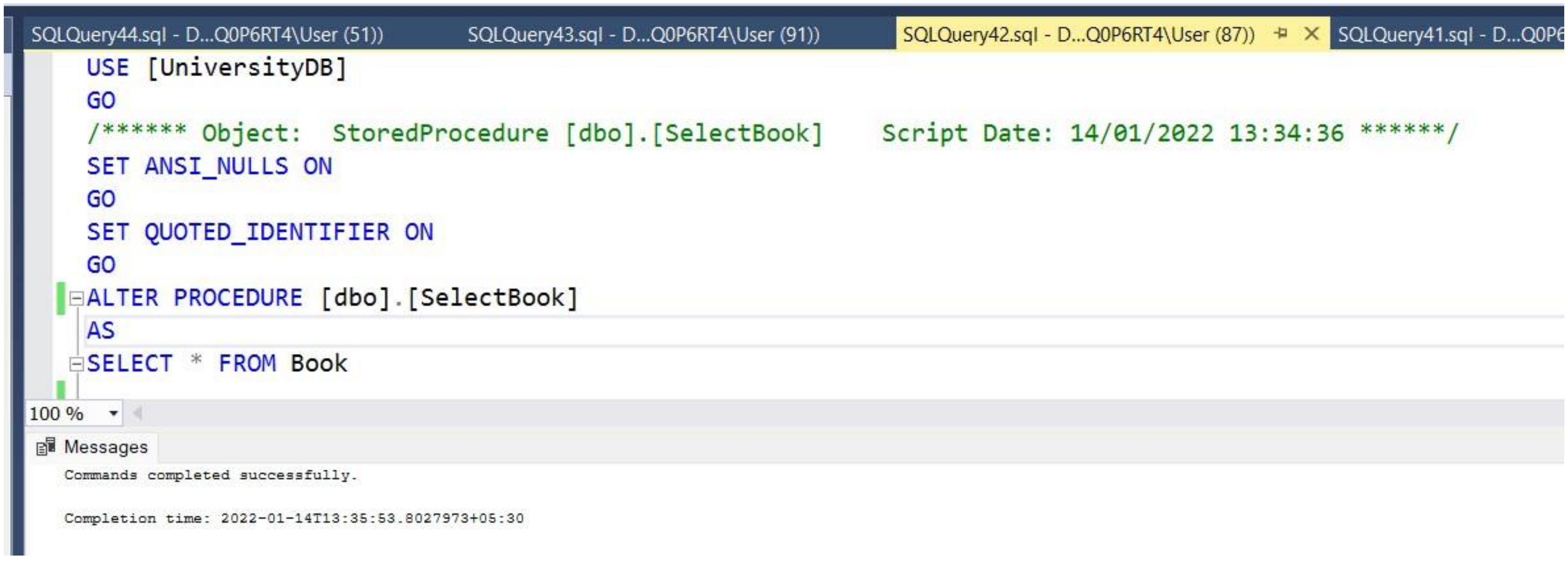

SQLQuery17.sql - D...Q0P6RT4\User (52))
SQLQuery16.sql - D...Q0P6RT4
100 %
Results
Messages

```
USE [UniversityDB]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[Borrowed_date]

SELECT  'Return Value' = @return_value

GO
```

	mem_Id	borrowed_date	issued_date	exp_due_date	penalty
1	2	2021-06-02	2020-08-22	2022-08-22	2.00
2	4	2021-02-01	2020-10-03	2022-10-03	0.00
3	5	2021-07-19	2020-12-08	2022-12-08	6.00
4	6	2021-02-12	2020-04-20	2022-04-20	8.12
5	7	2021-04-10	2020-01-30	2022-01-30	4.88
6	8	2021-07-11	2020-11-30	2022-11-30	0.00
7	9	2021-02-12	2020-05-14	2022-05-14	2.94

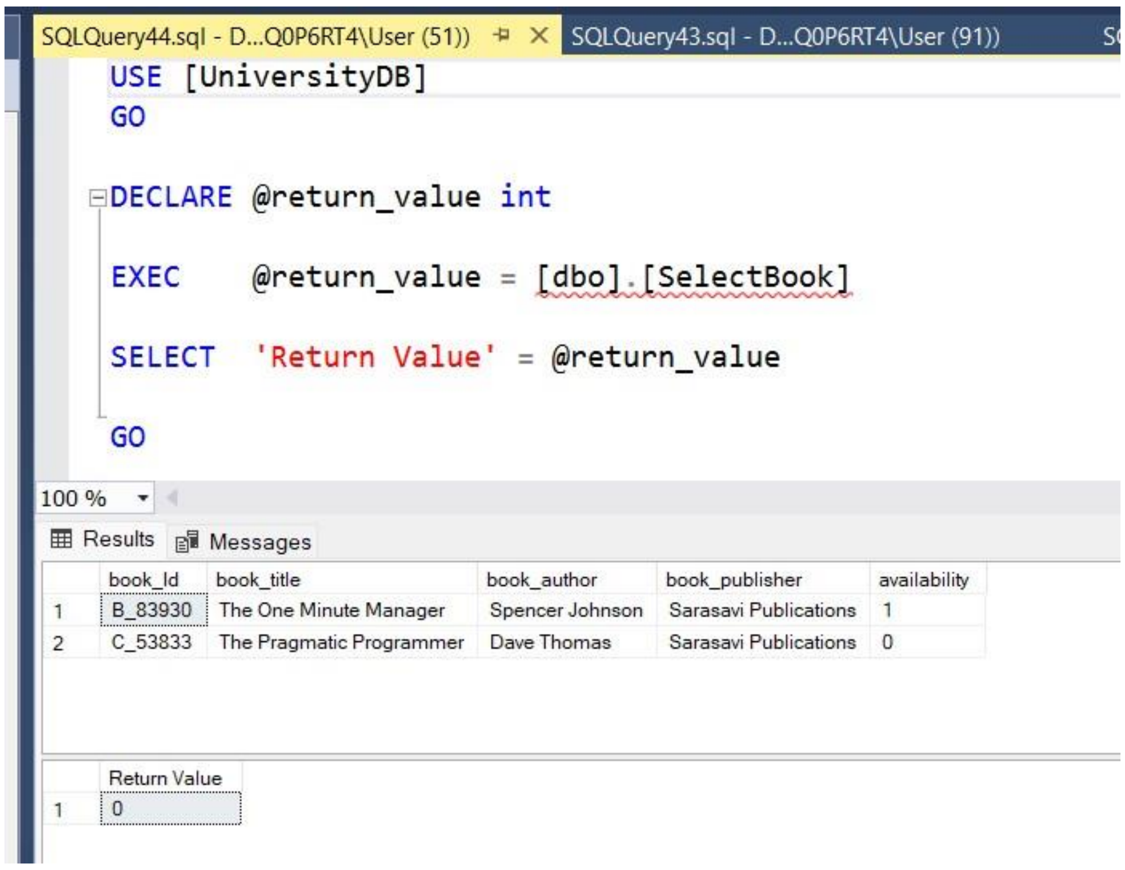
	Return Value
1	0

Stored Procedure 2


SQLQuery44.sql - D...Q0P6RT4\User (51))
SQLQuery43.sql - D...Q0P6RT4\User (91))
SQLQuery42.sql - D...Q0P6RT4\User (87))
SQLQuery41.sql - D...Q0P6
100 %
Messages

```
USE [UniversityDB]
GO
/****** Object:  StoredProcedure [dbo].[SelectBook]    Script Date: 14/01/2022 13:34:36 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[SelectBook]
AS
SELECT * FROM Book
```

```
Commands completed successfully.

Completion time: 2022-01-14T13:35:53.8027973+05:30
```


SQLQuery44.sql - D...Q0P6RT4\User (51))
SQLQuery43.sql - D...Q0P6RT4\User (91))
S
100 %
Results
Messages

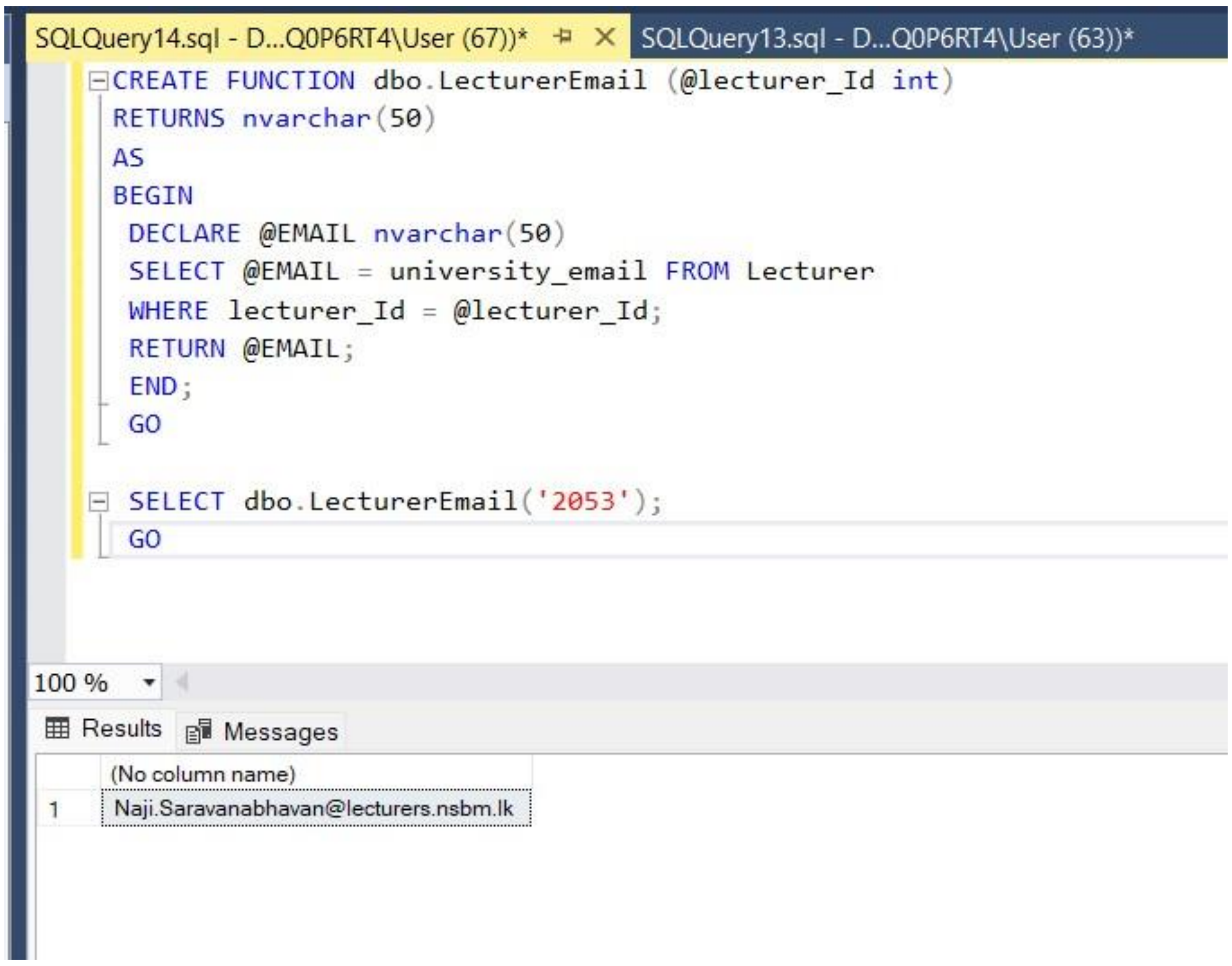
```
USE [UniversityDB]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[SelectBook]

SELECT  'Return Value' = @return_value

GO
```

	book_Id	book_title	book_author	book_publisher	availability
1	B_83930	The One Minute Manager	Spencer Johnson	Sarasavi Publications	1
2	C_53833	The Pragmatic Programmer	Dave Thomas	Sarasavi Publications	0

	Return Value
1	0

Functions

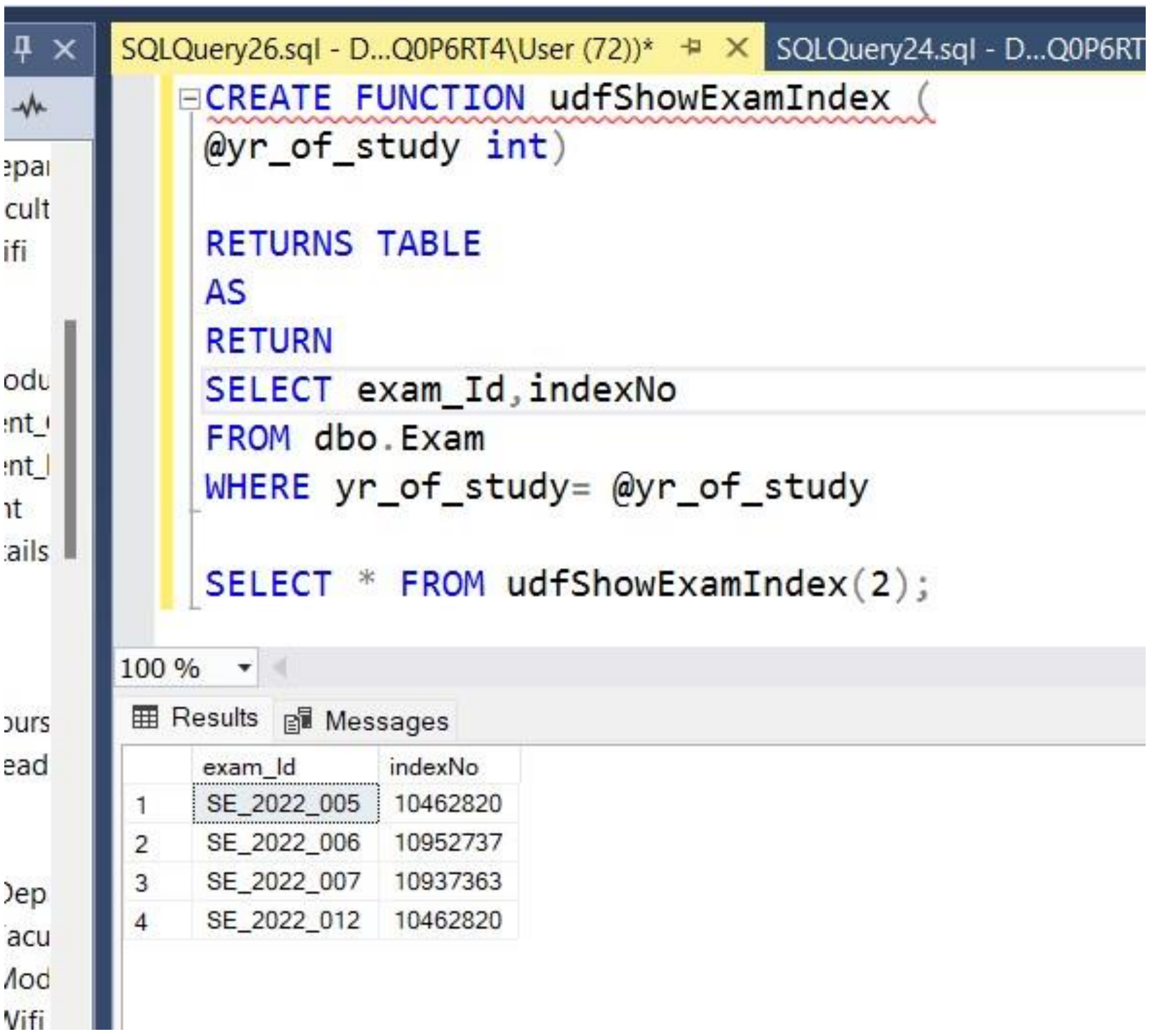
Function 1


SQLQuery14.sql - D...Q0P6RT4\User (67))*
SQLQuery13.sql - D...Q0P6RT4\User (63))*
100 %
Results
Messages

```sql
CREATE FUNCTION dbo.LecturerEmail (@lecturer_Id int)
RETURNS nvarchar(50)
AS
BEGIN
 DECLARE @EMAIL nvarchar(50)
 SELECT @EMAIL = university_email FROM Lecturer
 WHERE lecturer_Id = @lecturer_Id;
 RETURN @EMAIL;
 END;
 GO

 SELECT dbo.LecturerEmail('2053');
 GO
```

	(No column name)
1	Naji.Saravanabhavan@lecturers.nsbm.lk

Function 2


SQLQuery26.sql - D...Q0P6RT4\User (72))*
SQLQuery24.sql - D...Q0P6RT
100 %
Results
Messages

```sql
CREATE FUNCTION udfShowExamIndex (
@yr_of_study int)

RETURNS TABLE
AS
RETURN
SELECT exam_Id,indexNo
FROM dbo.Exam
WHERE yr_of_study= @yr_of_study

SELECT * FROM udfShowExamIndex(2);
```

	exam_Id	indexNo
1	SE_2022_005	10462820
2	SE_2022_006	10952737
3	SE_2022_007	10937363
4	SE_2022_012	10462820

Critical Evaluation & Future Implementation

Triggers

Automate the change in class in the exam table - when the gpa is updated, automatically the relevant class will also be updated according to the inserted query.

Automate the change in Signal strength in the Wifi_info Table – When the connection speed field data is updated the Relevant Signal Strength will also be updated according to the inserted query.

Views

Display Lecturer Details – All the relevant information can be obtained under a single table which determines for the specified retrieval.

Display Admin Details – All the relevant information can be obtained under a single table which determines for the specified retrieval.

Procedures

Procedure to display Membership details – All the information about the members will be displayed, where the members who have borrowed books after the specified date in the Procedure.

Procedure to display Book details – All the details of the books will be displayed once the relevant book publishers name is specified in the procedure.

User Defined Functions

Display information based on the User – Once the relevant Lecturer Id is inserted the specific university email address of the lecturer will be displayed.

- Once the relevant yr_of_study is inserted the specific exam_Id and IndexNo of all students of that Yr_of_study will be displayed.