Kwantlen Polytechnic University

PROJECT ACADEMIC SYSTEM

INFO2312– S10

Database Systems

Team ACAD

Submitted to: Prof. Caesar Lopez Castellanos

Date of Submission: March 14, 2025

PRODUCT DESCRIPTION

Project:	**Database Design & Implementation Experimental Case**
Date:	April 17, 2025
Product Title:	**Academic System**
Authors:	Sukhdeep Singh Gurnoor Kaur Jashanpreet Kaur Kartik Palta
Client:	INFO 2312 – KPU
Document Ref:	Product description
Version No:	1.5

1. PRODUCT DESCRIPTION HISTORY

1.1. Revision History

Revision date	**Previous revision date**	**Summary of Changes**	**Changes marked**
2025-03-15	-	Initial ER diagram and schema drafted	Yes
2025-03-20	2025-03-15	Updated relationships and added new tables	Yes
2025-03-21	2025-03-20	Finalized ER-Diagram for the project	Yes
2025-03-28	2025-03-21	Database Planning & Updating ER	Yes
2025-04-01	2025-03-28	Testing Queries in the Database	Yes
2025-04-10	2025-04-01	Revisiting entire project and updating some attributes and	No
2025-04-17	2025-04-10	Final proofread and cleanup	No

1.2. Distribution

This document has been distributed to:

Name	**Title**	**Date of Issue**	**Version**
INFO 2312 INSTRUCTOR	INFO2312 PROFESSOR	April 17, 2025	1.5

Table of Contents

page

2. OVERVIEW

2.1. Identifier

INFO2312-SectionS10-Team07

2.2. Title

Academic System for Jaegar State University

2.3. Purpose

The purpose of this product is to provide a comprehensive database system designed to manage academic data within a university setting. This Product Description aims to:

- Detail the structure, purpose, and function of the database.
- Define end users, such as academic staff and administration.
- Establish the required quality standards: data integrity, relational efficiency, and usability.
- Describe functions such as course enrolment, assessment tracking, staff and student records, and graduation status.
- Identify required activities for creating, testing, and using the product.
- Define roles/skills needed for implementation: SQL proficiency, ER modelling, and relational design.
- Specify the DBMS used: **MySQL Workbench**.

2.4. Composition

This academic database system for fictional Jaegar State University, includes ten main entities: Student, Department, Staff, Courses, Classes, Enrollment, Assessment, Graduation, ClassFeedback, and Class-Enrollment relationships. The student table stores information about each student and connects to several parts of the system. Students belong to departments, enroll in classes, complete assessments, give feedback to instructors, and can apply for graduation. The Department table holds details about each academic department and is linked to both students and staff.

Staff are the instructors and are responsible for teaching Classes, which are specific versions of Courses offered at certain times and days. Students enroll in these classes each semester through the Enrollment table. They also complete Assessments like projects or assignments for each class, which are tracked separately. The ClassFeedback table lets students give ratings and comments about their instructors. The Graduation table records which students have completed their programs and whether they received honors.

All these entities are connected to support important academic functions like scheduling, performance tracking, feedback collection, and graduation processing. The system is designed to keep everything organized and running smoothly for both students and staff.

Table 1. Definition of Table: Department

Attribute	Datatype	Primary Key		Foreign Key		Referential Integrity				Additional Comments
		Yes	No	Yes	No	Parent		On Update	On Delete	
						Table	Attribute			
depCode	VARCHAR	✓			✓					Primary Key
officeLocation	VARCHAR		✓		✓					
depName	VARCHAR		✓		✓					

Table 2. Definition of Table: Courses

Attribute	Datatype	Primary Key		Foreign Key		Referential Integrity				Additional Comment
		Yes	No	Yes	No	Parent		On Update	On Delete	
						Table	Attribute			
courseID	VARCHAR	✓			✓					
courseName	VARCHAR		✓		✓					
courseAbbreviation	CHAR		✓		✓					
depCode	VARCHAR		✓	✓		Department	depCode	CASCADE	CASCADE	

Table 3. Definition of Table: Classes

Attribute	Datatype	Primary Key		Foreign Key		Referential Integrity				Additional Comments
		Yes	No	Yes	No	Parent		On Update	On Delete	
						Table	Attribute			
classID	VARCHAR	✓			✓					
CRN	VARCHAR		✓		✓					
courseID	VARCHAR		✓	✓		Course counseID			CASCADE	
sectionNo	VARCHAR		✓		✓					
classDay	VARCHAR		✓		✓					
timing	TIME		✓		✓					
staffID	VARCHAR		✓	✓		Staff	staffID	CASCADE	CASCADE	

Table 4. Definition of Table: Student

Attribute	Datatype	Primary Key		Foreign Key		Referential Integrity				Additional Comments
		Yes	No	Yes	No	Parent		On Update	On Delete	
						Table	Attribute			
studentID	CHAR	✓			✓					
fName	VARCHAR		✓		✓					
lName	CHAR		✓		✓					
email	VARCHAR		✓		✓					
gpa	CHAR		✓		✓					
major	VARCHAR		✓		✓					
totalCredit	SMALLINT		✓		✓					
depCode	VARCHAR		✓	✓		Department	depCode	CASCADE	CASCADE	SET NULL
DOB	DATE		✓		✓					

Table 5. Definition of Table: Enrolment

Attribute	Datatype	Primary Key		Foreign Key		Referential Integrity				Additional Comments
		Yes	No	Yes	No	Parent		On Update	On Delete	
						Table	Attribute			
enrollID	CHAR	✓			✓					
enrolldate	DATE		✓		✓					
enrollTime	DATE		✓		✓					Current timestamp
enrollSemester	VARCHAR		✓		✓					
StudentID	CHAR		✓	✓		Student	studentID	CASACDE	CASCADE	Links enrolment to student

Table 6. Definition of Table: Graduation

Attribute	Datatype	Primary Key		Foreign Key		Referential Integrity				Additional Comments
		Yes	No	Yes	No	Parent		On Update	On Delete	
						Table	Attribute			
graduationID	CHAR	✓			✓					
gradeDate	DATE		✓		✓					
honors	CHAR		✓		✓					Y or N
studentID	CHAR		✓	✓		Student	studentID	CASCADE	CASCADE	

Table 7. Definition of Table: ClassFeedback

Attribute	Datatype	Primary Key		Foreign Key		Referential Integrity				Additional Comments
		Yes	No	Yes	No	Parent		On Update	On Delete	
						Table	Attribute			
feedbackID	VARCHAR	✓			✓					
staffID	VARCHAR		✓	✓		Staff	staffID	CASCADE	CASCADE	
studentID	CHAR		✓	✓		Student	studentID	CASCADE	CASCADE	
rating	VARCHAR		✓		✓					Optional letter grade
content	VARCHAR		✓		✓					Feedback text
timeStamp	DATETIME		✓		✓					

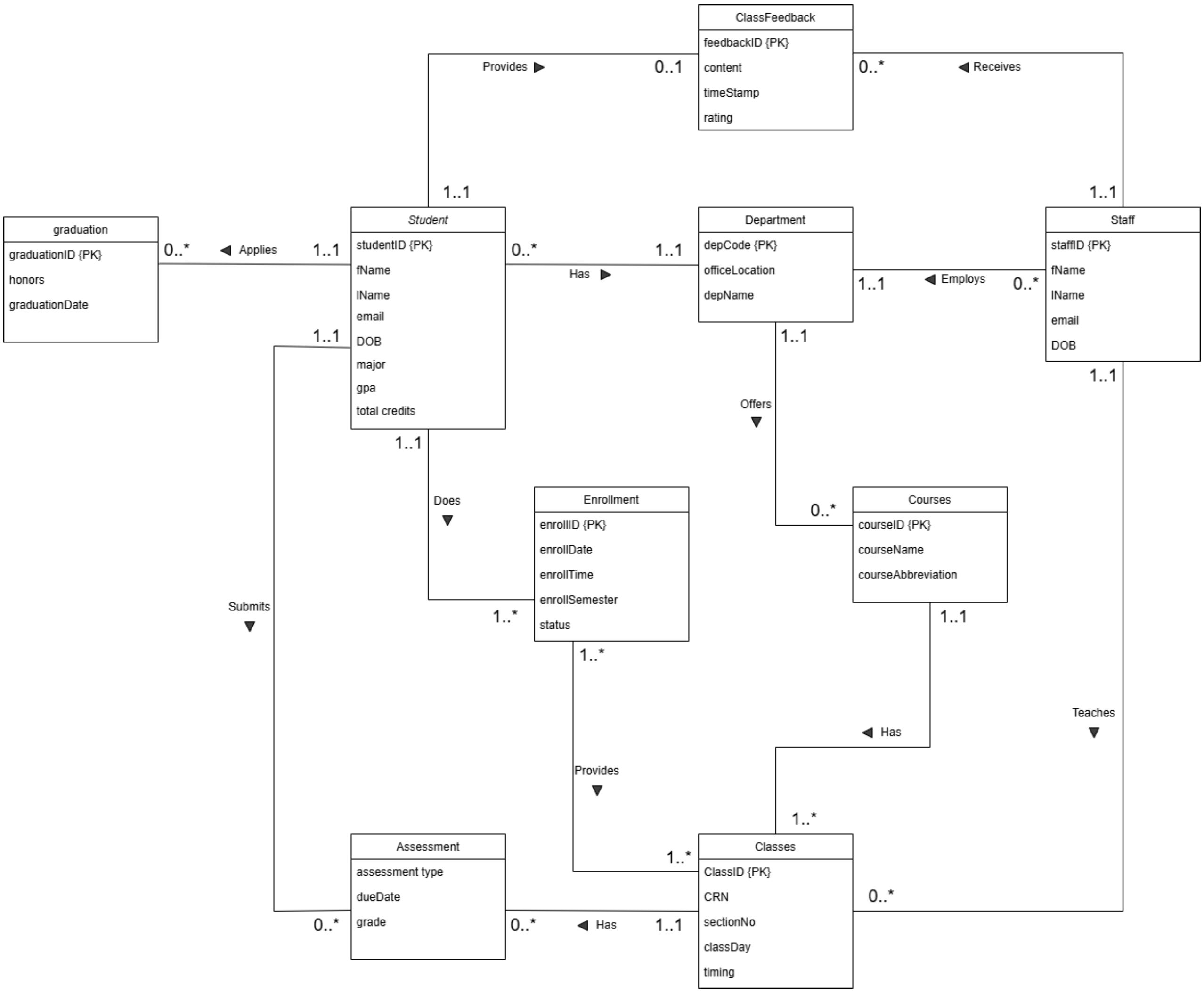
Table 8. Definition of Table: Assessment

Attribute	Datatype	Primary Key		Foreign Key		Referential Integrity				Additional Comments
		Yes	No	Yes	No	Parent		On Update	On Delete	
						Table	Attribute			
assessmentType	VARCHAR	✓			✓					Part of composite PK
classID	VARCHAR	✓		✓		Class	classID	CASCADE	CASCADE	Part of composite PK + FK
studentID	CHAR	✓		✓		Student	studentID	CASCADE	CASCADE	Part of composite PK + FK
dueDate	DATE		✓		✓					
grade	DECIMAL		✓		✓					

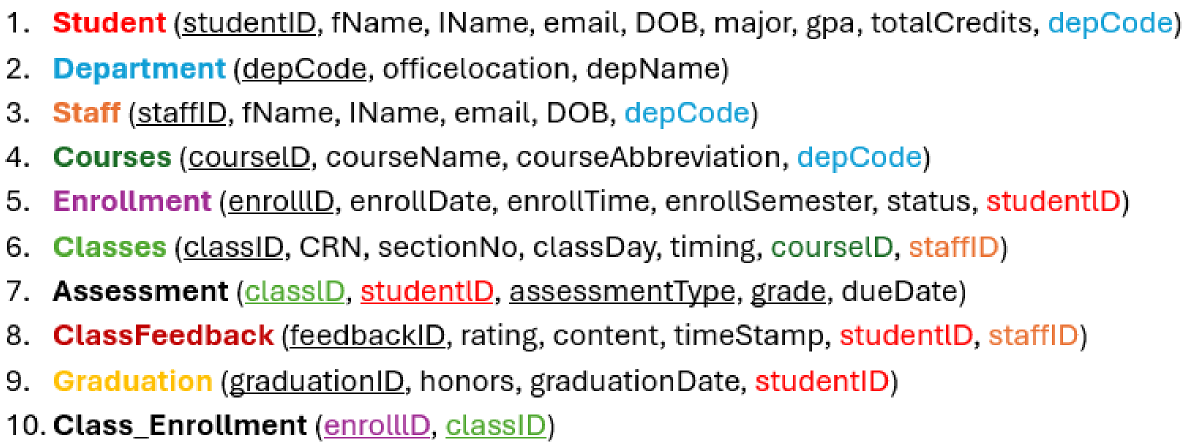
Table 9. Definition of Table: Staff

Attribute	Datatype	Primary Key		Foreign Key		Referential Integrity				Additional Comments
		Yes	No	Yes	No	Parent		On Update	On Delete	
						Table	Attribute			
staffID	VARCHAR	✓			✓					
fName	VARCHAR		✓		✓					
lName	VARCHAR		✓		✓					
email	VARCHAR		✓		✓					
DOB	DATE		✓		✓					
depCode	VARCHAR		✓	✓		Department	depCode	CASCADE	CASCADE	

ER DIAGRAM


ClassFeedback
feedbackID {PK}
content
timeStamp
rating
Provides ▶
0..1
0..*
◀ Receives
1..1
1..1
graduation
graduationID {PK}
honors
graduationDate
0..*
◀ Applies
1..1
Student
studentID {PK}
fName
lName
email
DOB
major
gpa
total credits
0..*
Has ▶
1..1
Department
depCode {PK}
officeLocation
depName
1..1
◀ Employs
0..*
Staff
staffID {PK}
fName
lName
email
DOB
1..1
1..1
1..1
Offers ▼
1..1
Does ▼
Enrollment
enrollID {PK}
enrollDate
enrollTime
enrollSemester
status
1..*
0..*
Courses
courseID {PK}
courseName
courseAbbreviation
1..1
Submits ▼
1..*
Provides ▼
◀ Has
Teaches ▼
1..*
1..*
Assessment
assessment type
dueDate
grade
0..*
0..*
◀ Has
1..1
Classes
ClassID {PK}
CRN
sectionNo
classDay
timing
0..*

RELATIONAL SCHEMA



1. **Student** (studentID, fName, lName, email, DOB, major, gpa, totalCredits, depCode)
2. **Department** (depCode, officelocation, depName)
3. **Staff** (staffID, fName, lName, email, DOB, depCode)
4. **Courses** (courseID, courseName, courseAbbreviation, depCode)
5. **Enrollment** (enrollID, enrollDate, enrollTime, enrollSemester, status, studentID)
6. **Classes** (classID, CRN, sectionNo, classDay, timing, courseID, staffID)
7. **Assessment** (classID, studentID, assessmentType, grade, dueDate)
8. **ClassFeedback** (feedbackID, rating, content, timeStamp, studentID, staffID)
9. **Graduation** (graduationID, honors, graduationDate, studentID)
10. **Class_Enrollment** (enrollID, classID)

BUSINESS RULES

Student

1. Mandatory for students to belong to a department
2. A student must belong to only one department.
3. Mandatory for students to be enrolled in a class
4. Students can enroll in many classes
5. The student might not submit any assessment
6. A student may submit multiple assessments
7. Optional for the student to give course feedback
8. A student can provide only one feedback related to one course

Staff

9. A staff member must be employed with only one department
10. Mandatory for staff to be employed by a department
11. Staff may not receive any feedback for the course they teach
12. Staff can get multiple feedback for the course they teach
13. A staff member can teach many classes.
14. Staff might not teach any class

CourseFeedback

15. One course feedback must be done by one student
16. Course feedback belongs to a single student
17. Course feedback belongs to the staff member
18. Course feedback belongs to only one staff member

Department

19. A department can offer many courses.
20. A department may not offer a course
21. A department employs many staff members.
22. A department may not have any staff member attached to it
23. A Department may not have a student
24. A Department can have many students

Courses

25. A course can be offered by a department.
26. A course should be linked to only one department
27. A course can have many class sections.
28. A course should have at least one class

Classes

29. A class must belong to only a course.
30. A class must belong to only one course
31. Mandatory for the class to have a student
32. A class can have multiple students
33. A Class should be taught by a staff member
34. A class should be taught by one and only one staff member
35. A class may not have any assessment
36. A class can have multiple assessments

Assessment

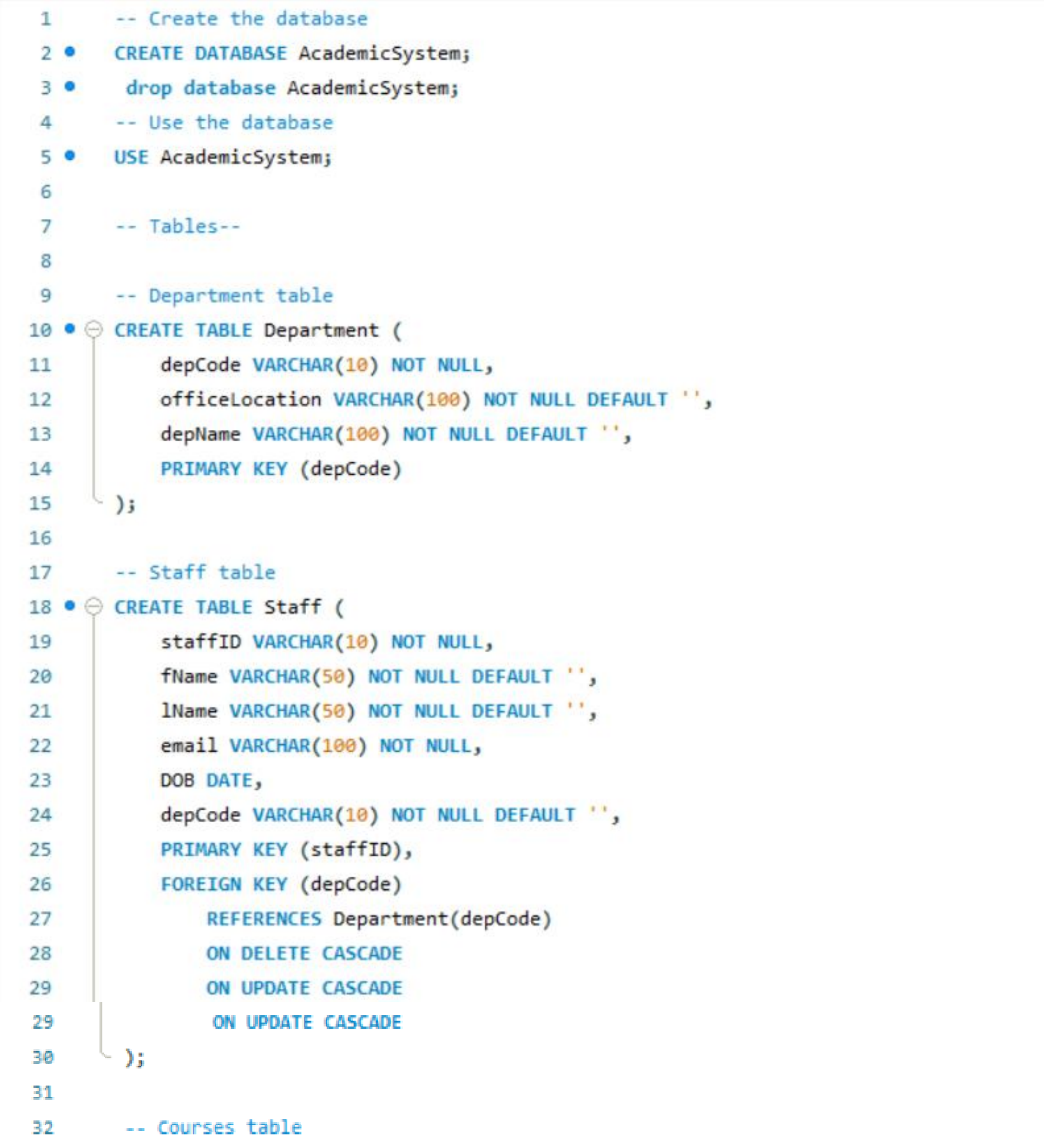
37. An assessment belongs to a student.
38. An assessment belongs to only one student
39. An assessment belongs to a class.
40. An assessment belongs to only one class.

Enrollment

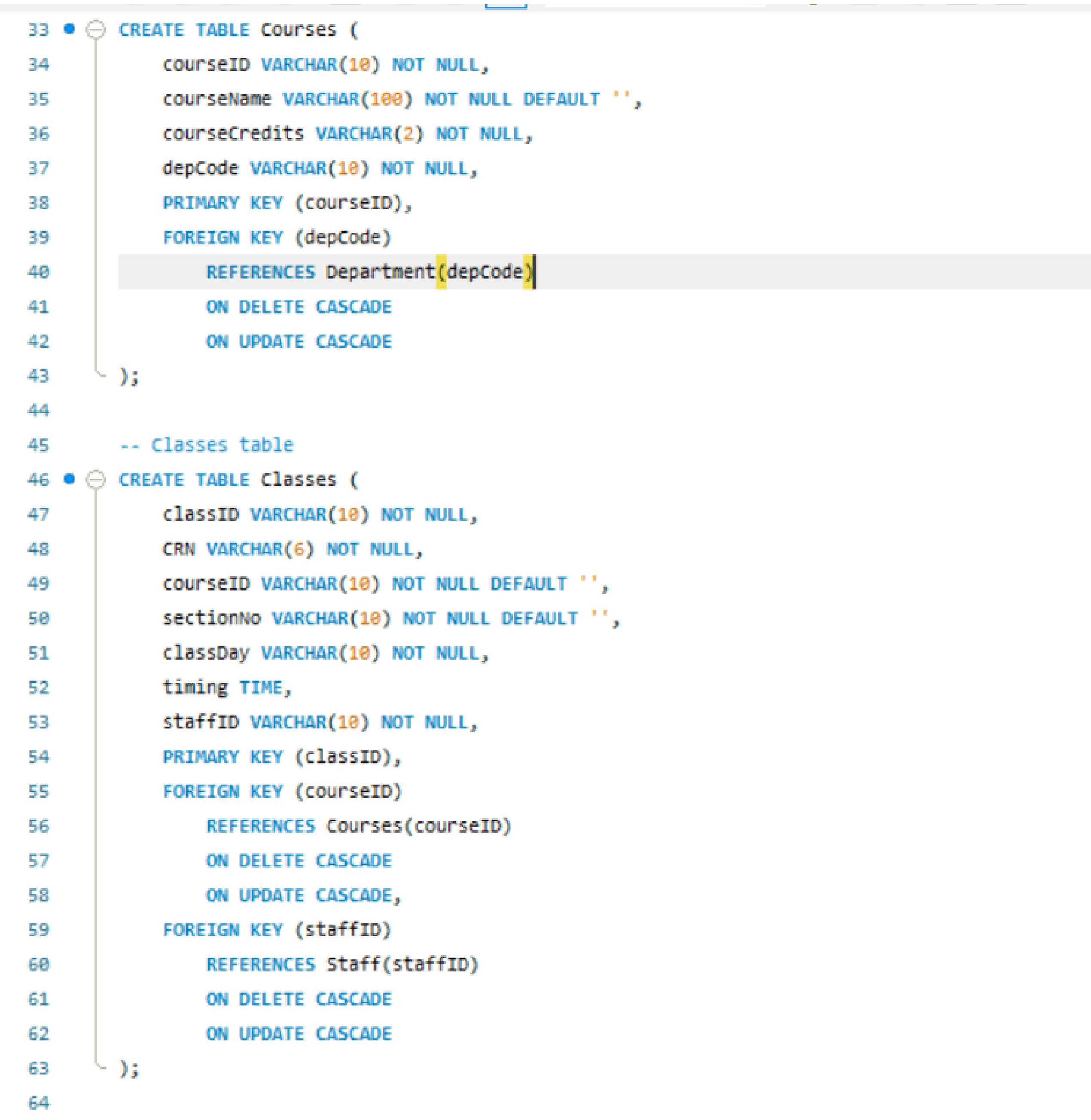
41. Enrollment should belong to a student
42. It should belong to one and only one student
43. Enrollment should have at least one class
44. It can have multiple classes

DATABASE OF THE ACADEMIC SYSTEM

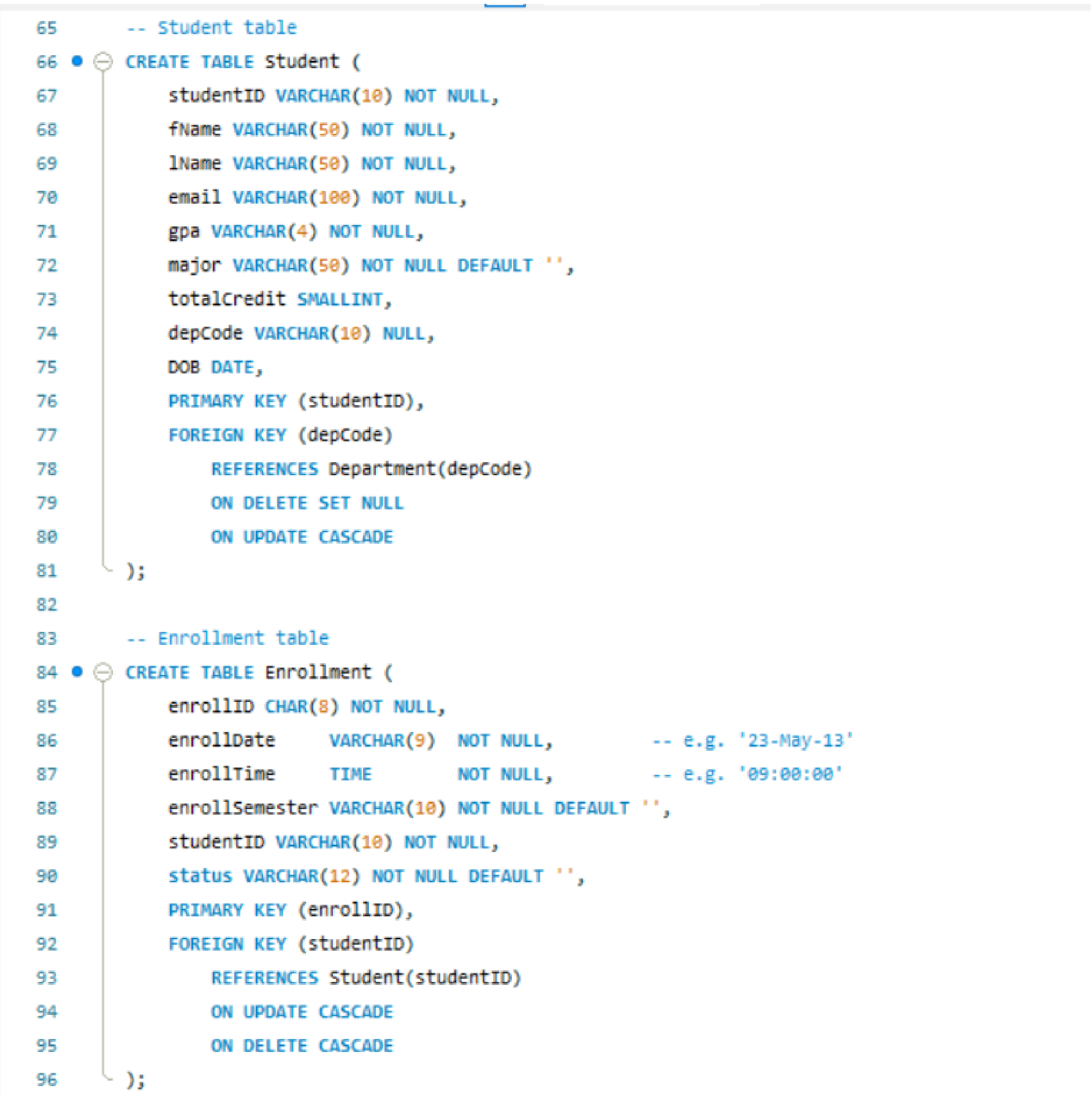
Here, it the sample database for the Jaegar State University. The SQL file is also attached with this project.



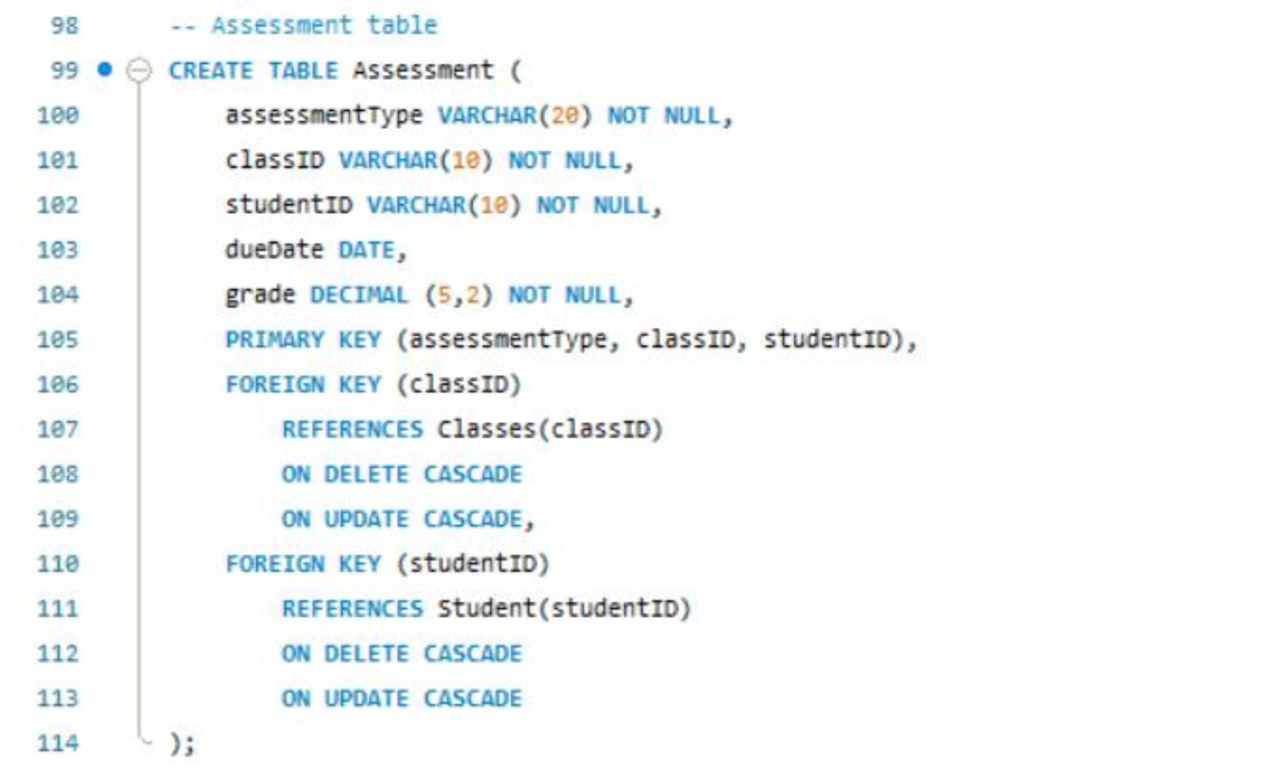
```sql
-- Create the database
CREATE DATABASE AcademicSystem;
 drop database AcademicSystem;
-- Use the database
USE AcademicSystem;

-- Tables--

-- Department table
CREATE TABLE Department (
    depCode VARCHAR(10) NOT NULL,
    officeLocation VARCHAR(100) NOT NULL DEFAULT '',
    depName VARCHAR(100) NOT NULL DEFAULT '',
    PRIMARY KEY (depCode)
);

-- Staff table
CREATE TABLE Staff (
    staffID VARCHAR(10) NOT NULL,
    fName VARCHAR(50) NOT NULL DEFAULT '',
    lName VARCHAR(50) NOT NULL DEFAULT '',
    email VARCHAR(100) NOT NULL,
    DOB DATE,
    depCode VARCHAR(10) NOT NULL DEFAULT '',
    PRIMARY KEY (staffID),
    FOREIGN KEY (depCode)
        REFERENCES Department(depCode)
        ON DELETE CASCADE
        ON UPDATE CASCADE
        ON UPDATE CASCADE
);

-- Courses table
```



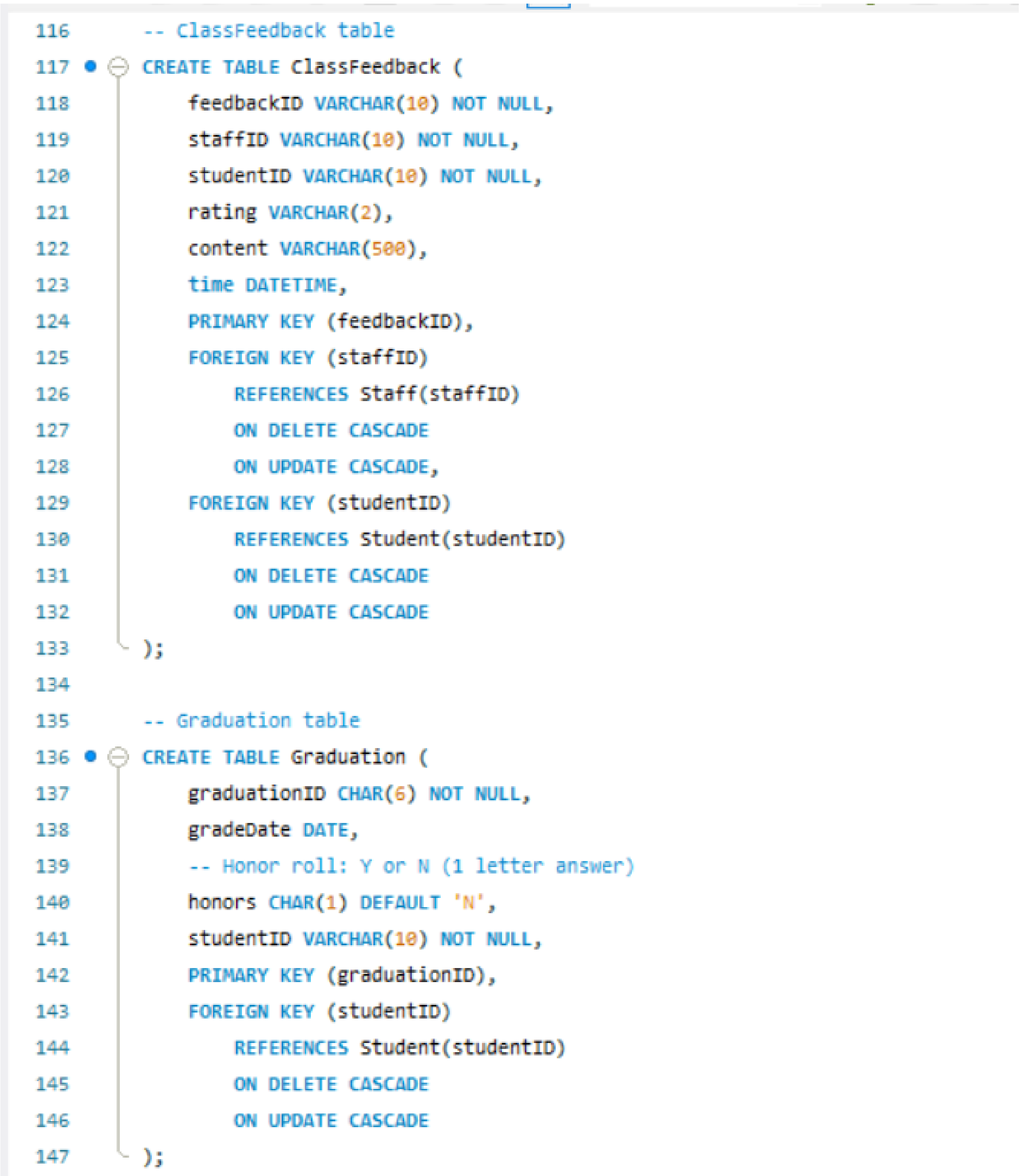
```sql
CREATE TABLE Courses (
    courseID VARCHAR(10) NOT NULL,
    courseName VARCHAR(100) NOT NULL DEFAULT '',
    courseCredits VARCHAR(2) NOT NULL,
    depCode VARCHAR(10) NOT NULL,
    PRIMARY KEY (courseID),
    FOREIGN KEY (depCode)
        REFERENCES Department(depCode)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- Classes table
CREATE TABLE Classes (
    classID VARCHAR(10) NOT NULL,
    CRN VARCHAR(6) NOT NULL,
    courseID VARCHAR(10) NOT NULL DEFAULT '',
    sectionNo VARCHAR(10) NOT NULL DEFAULT '',
    classDay VARCHAR(10) NOT NULL,
    timing TIME,
    staffID VARCHAR(10) NOT NULL,
    PRIMARY KEY (classID),
    FOREIGN KEY (courseID)
        REFERENCES Courses(courseID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (staffID)
        REFERENCES Staff(staffID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```



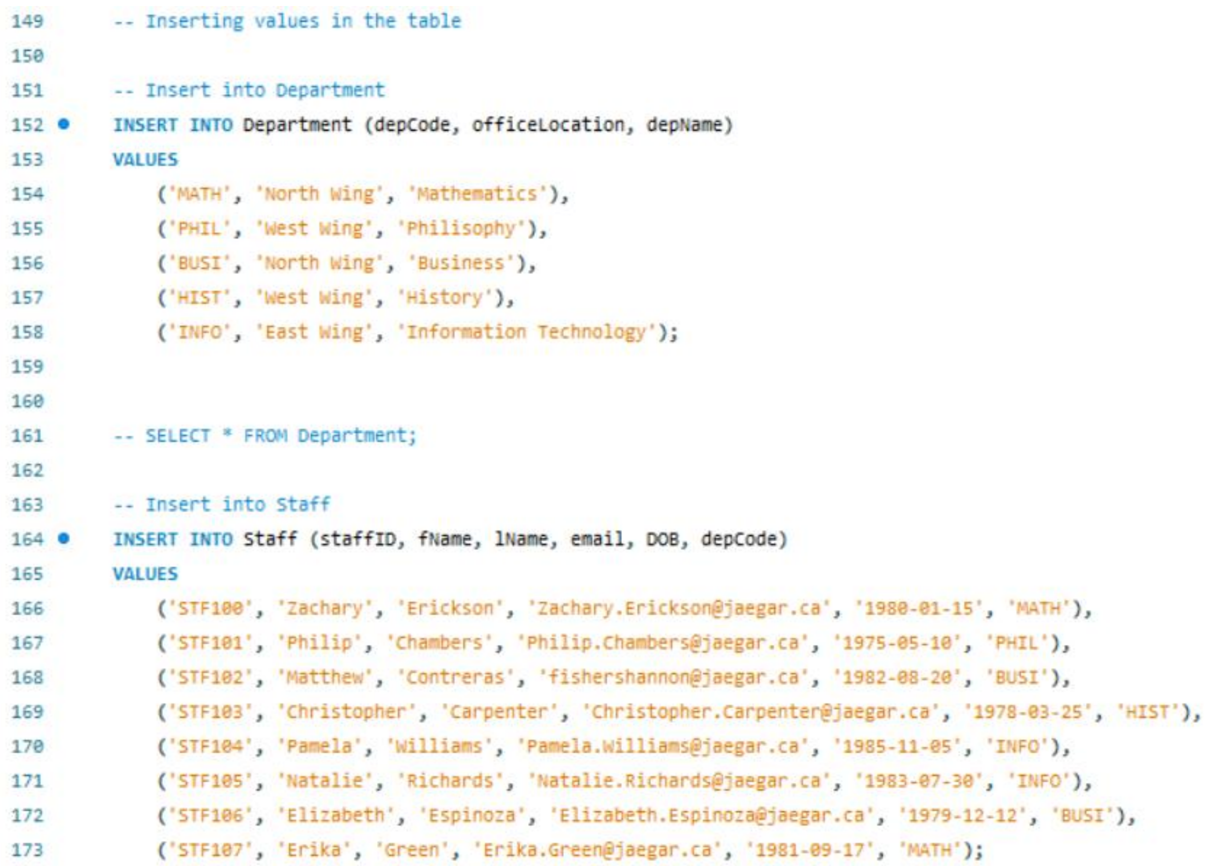
```sql
-- Student table
CREATE TABLE Student (
    studentID VARCHAR(10) NOT NULL,
    fName VARCHAR(50) NOT NULL,
    lName VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL,
    gpa VARCHAR(4) NOT NULL,
    major VARCHAR(50) NOT NULL DEFAULT '',
    totalCredit SMALLINT,
    depCode VARCHAR(10) NULL,
    DOB DATE,
    PRIMARY KEY (studentID),
    FOREIGN KEY (depCode)
        REFERENCES Department(depCode)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);

-- Enrollment table
CREATE TABLE Enrollment (
    enrollID CHAR(8) NOT NULL,
    enrollDate     VARCHAR(9)  NOT NULL,         -- e.g. '23-May-13'
    enrollTime     TIME        NOT NULL,         -- e.g. '09:00:00'
    enrollSemester VARCHAR(10) NOT NULL DEFAULT '',
    studentID VARCHAR(10) NOT NULL,
    status VARCHAR(12) NOT NULL DEFAULT '',
    PRIMARY KEY (enrollID),
    FOREIGN KEY (studentID)
        REFERENCES Student(studentID)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
```



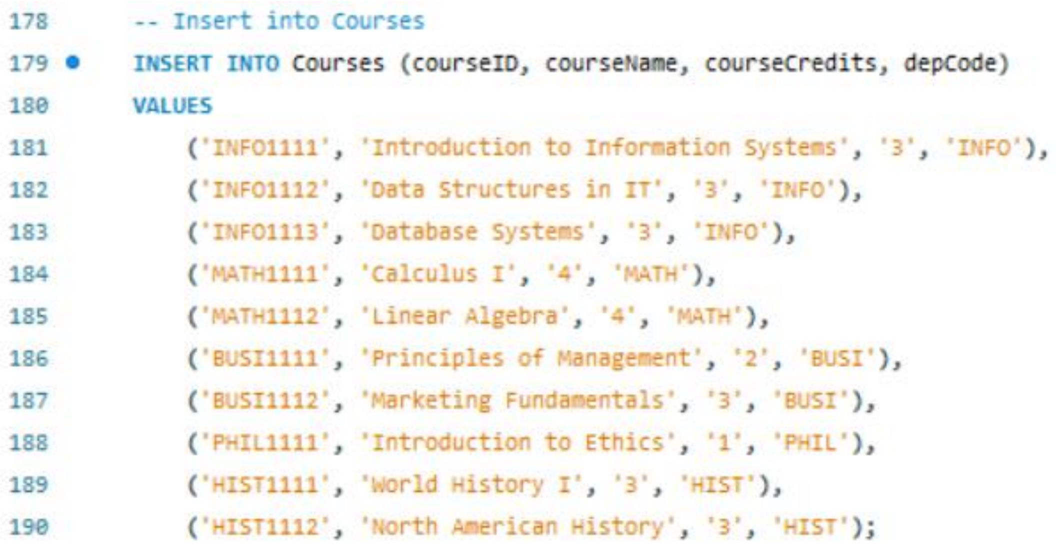
```sql
-- Assessment table
CREATE TABLE Assessment (
    assessmentType VARCHAR(20) NOT NULL,
    classID VARCHAR(10) NOT NULL,
    studentID VARCHAR(10) NOT NULL,
    dueDate DATE,
    grade DECIMAL (5,2) NOT NULL,
    PRIMARY KEY (assessmentType, classID, studentID),
    FOREIGN KEY (classID)
        REFERENCES Classes(classID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (studentID)
        REFERENCES Student(studentID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```



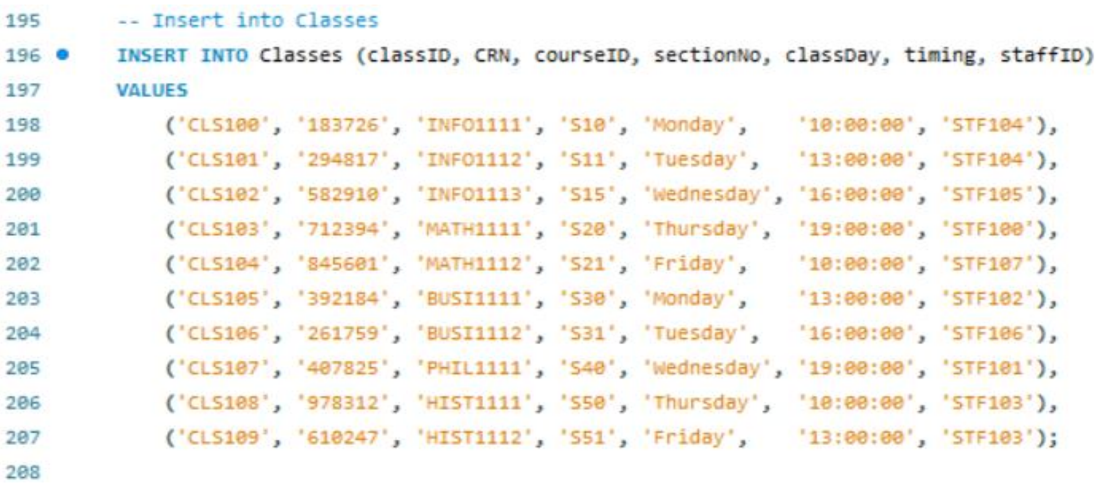
```sql
-- ClassFeedback table
CREATE TABLE ClassFeedback (
    feedbackID VARCHAR(10) NOT NULL,
    staffID VARCHAR(10) NOT NULL,
    studentID VARCHAR(10) NOT NULL,
    rating VARCHAR(2),
    content VARCHAR(500),
    time DATETIME,
    PRIMARY KEY (feedbackID),
    FOREIGN KEY (staffID)
        REFERENCES Staff(staffID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (studentID)
        REFERENCES Student(studentID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

-- Graduation table
CREATE TABLE Graduation (
    graduationID CHAR(6) NOT NULL,
    gradeDate DATE,
    -- Honor roll: Y or N (1 letter answer)
    honors CHAR(1) DEFAULT 'N',
    studentID VARCHAR(10) NOT NULL,
    PRIMARY KEY (graduationID),
    FOREIGN KEY (studentID)
        REFERENCES Student(studentID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```



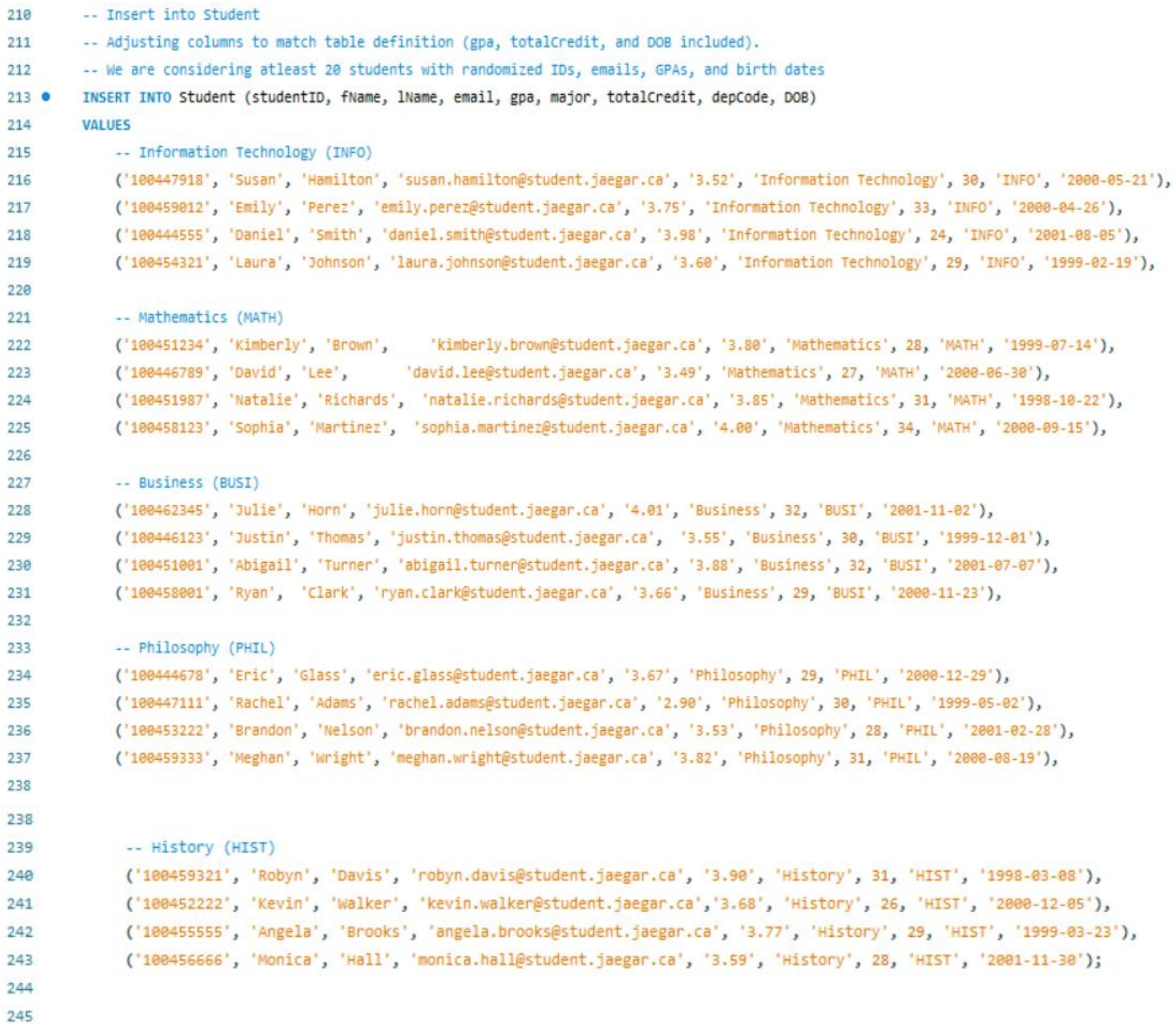
```sql
-- Inserting values in the table

-- Insert into Department
INSERT INTO Department (depCode, officeLocation, depName)
VALUES
    ('MATH', 'North Wing', 'Mathematics'),
    ('PHIL', 'West Wing', 'Philisophy'),
    ('BUSI', 'North Wing', 'Business'),
    ('HIST', 'West Wing', 'History'),
    ('INFO', 'East Wing', 'Information Technology');


-- SELECT * FROM Department;

-- Insert into Staff
INSERT INTO Staff (staffID, fName, lName, email, DOB, depCode)
VALUES
    ('STF100', 'Zachary', 'Erickson', 'Zachary.Erickson@jaegar.ca', '1980-01-15', 'MATH'),
    ('STF101', 'Philip', 'Chambers', 'Philip.Chambers@jaegar.ca', '1975-05-10', 'PHIL'),
    ('STF102', 'Matthew', 'Contreras', 'fishershannon@jaegar.ca', '1982-08-20', 'BUSI'),
    ('STF103', 'Christopher', 'Carpenter', 'Christopher.Carpenter@jaegar.ca', '1978-03-25', 'HIST'),
    ('STF104', 'Pamela', 'Williams', 'Pamela.Williams@jaegar.ca', '1985-11-05', 'INFO'),
    ('STF105', 'Natalie', 'Richards', 'Natalie.Richards@jaegar.ca', '1983-07-30', 'INFO'),
    ('STF106', 'Elizabeth', 'Espinoza', 'Elizabeth.Espinoza@jaegar.ca', '1979-12-12', 'BUSI'),
    ('STF107', 'Erika', 'Green', 'Erika.Green@jaegar.ca', '1981-09-17', 'MATH');
```



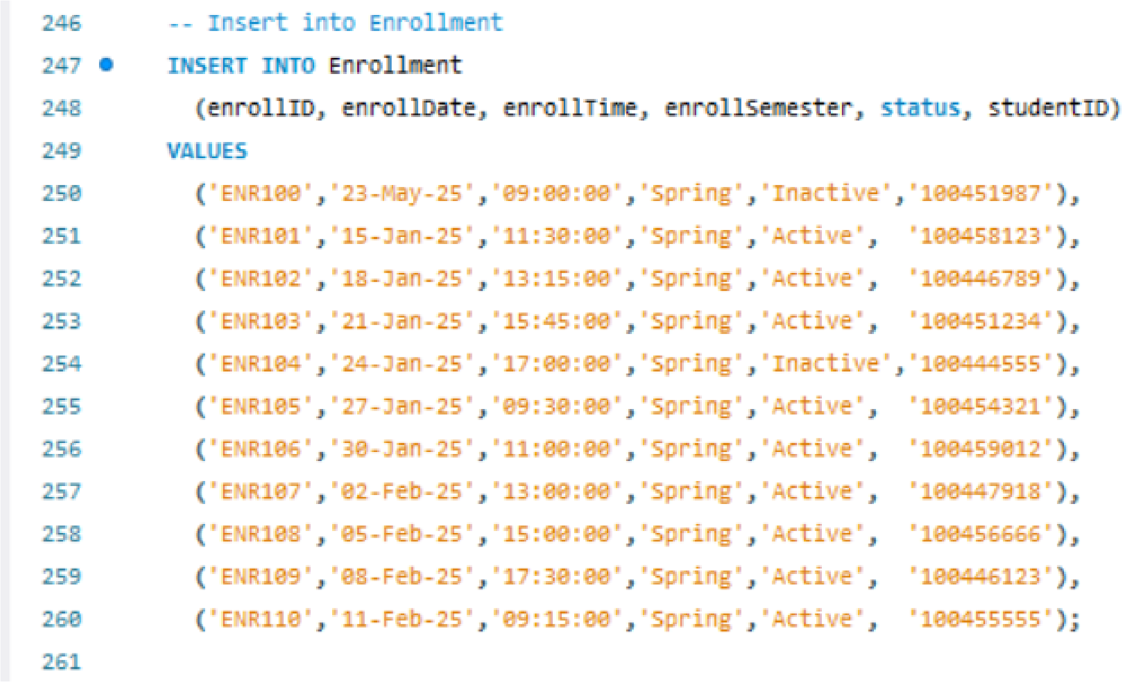
```sql
-- Insert into Courses
INSERT INTO Courses (courseID, courseName, courseCredits, depCode)
VALUES
    ('INFO1111', 'Introduction to Information Systems', '3', 'INFO'),
    ('INFO1112', 'Data Structures in IT', '3', 'INFO'),
    ('INFO1113', 'Database Systems', '3', 'INFO'),
    ('MATH1111', 'Calculus I', '4', 'MATH'),
    ('MATH1112', 'Linear Algebra', '4', 'MATH'),
    ('BUSI1111', 'Principles of Management', '2', 'BUSI'),
    ('BUSI1112', 'Marketing Fundamentals', '3', 'BUSI'),
    ('PHIL1111', 'Introduction to Ethics', '1', 'PHIL'),
    ('HIST1111', 'World History I', '3', 'HIST'),
    ('HIST1112', 'North American History', '3', 'HIST');
```



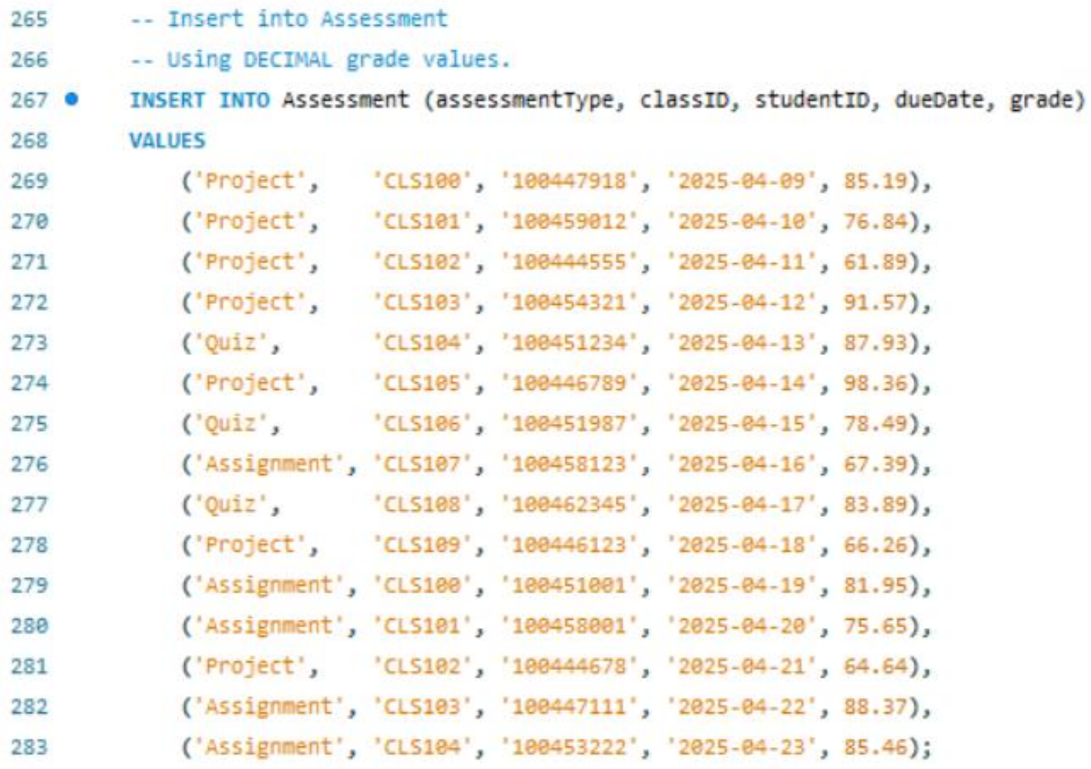
```sql
-- Insert into Classes
INSERT INTO Classes (classID, CRN, courseID, sectionNo, classDay, timing, staffID)
VALUES
    ('CLS100', '183726', 'INFO1111', 'S10', 'Monday',    '10:00:00', 'STF104'),
    ('CLS101', '294817', 'INFO1112', 'S11', 'Tuesday',   '13:00:00', 'STF104'),
    ('CLS102', '582910', 'INFO1113', 'S15', 'Wednesday', '16:00:00', 'STF105'),
    ('CLS103', '712394', 'MATH1111', 'S20', 'Thursday',  '19:00:00', 'STF100'),
    ('CLS104', '845601', 'MATH1112', 'S21', 'Friday',    '10:00:00', 'STF107'),
    ('CLS105', '392184', 'BUSI1111', 'S30', 'Monday',    '13:00:00', 'STF102'),
    ('CLS106', '261759', 'BUSI1112', 'S31', 'Tuesday',   '16:00:00', 'STF106'),
    ('CLS107', '407825', 'PHIL1111', 'S40', 'Wednesday', '19:00:00', 'STF101'),
    ('CLS108', '978312', 'HIST1111', 'S50', 'Thursday',  '10:00:00', 'STF103'),
    ('CLS109', '610247', 'HIST1112', 'S51', 'Friday',    '13:00:00', 'STF103');

```



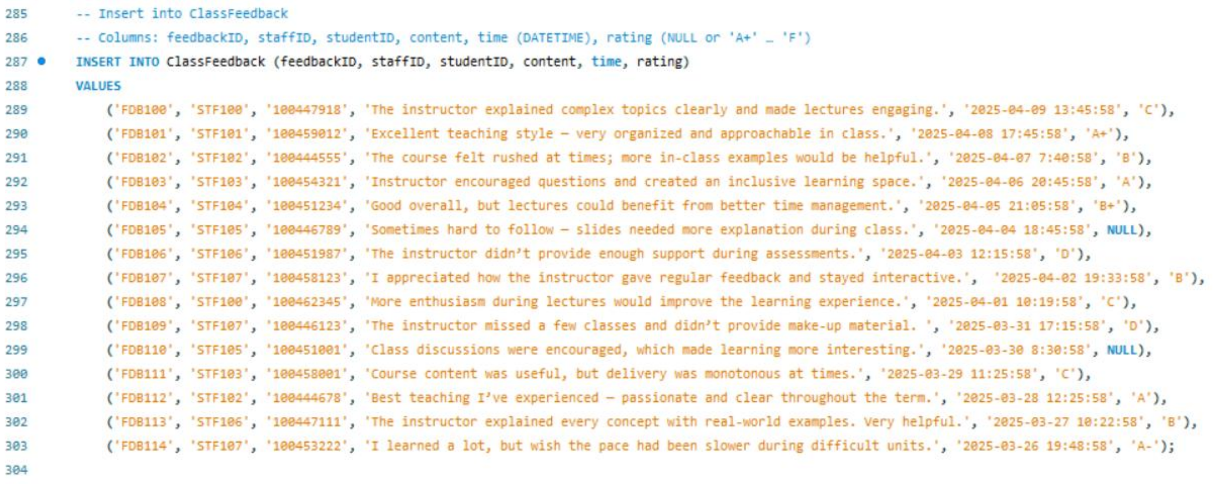
```sql
-- Insert into Student
-- Adjusting columns to match table definition (gpa, totalCredit, and DOB included).
-- We are considering atleast 20 students with randomized IDs, emails, GPAs, and birth dates
INSERT INTO Student (studentID, fName, lName, email, gpa, major, totalCredit, depCode, DOB)
VALUES
    -- Information Technology (INFO)
    ('100447918', 'Susan', 'Hamilton', 'susan.hamilton@student.jaegar.ca', '3.52', 'Information Technology', 30, 'INFO', '2000-05-21'),
    ('100459012', 'Emily', 'Perez', 'emily.perez@student.jaegar.ca', '3.75', 'Information Technology', 33, 'INFO', '2000-04-26'),
    ('100444555', 'Daniel', 'Smith', 'daniel.smith@student.jaegar.ca', '3.98', 'Information Technology', 24, 'INFO', '2001-08-05'),
    ('100454321', 'Laura', 'Johnson', 'laura.johnson@student.jaegar.ca', '3.60', 'Information Technology', 29, 'INFO', '1999-02-19'),

    -- Mathematics (MATH)
    ('100451234', 'Kimberly', 'Brown',     'kimberly.brown@student.jaegar.ca', '3.80', 'Mathematics', 28, 'MATH', '1999-07-14'),
    ('100446789', 'David', 'Lee',       'david.lee@student.jaegar.ca', '3.49', 'Mathematics', 27, 'MATH', '2000-06-30'),
    ('100451987', 'Natalie', 'Richards',  'natalie.richards@student.jaegar.ca', '3.85', 'Mathematics', 31, 'MATH', '1998-10-22'),
    ('100458123', 'Sophia', 'Martinez',  'sophia.martinez@student.jaegar.ca', '4.00', 'Mathematics', 34, 'MATH', '2000-09-15'),

    -- Business (BUSI)
    ('100462345', 'Julie', 'Horn', 'julie.horn@student.jaegar.ca', '4.01', 'Business', 32, 'BUSI', '2001-11-02'),
    ('100446123', 'Justin', 'Thomas', 'justin.thomas@student.jaegar.ca',  '3.55', 'Business', 30, 'BUSI', '1999-12-01'),
    ('100451001', 'Abigail', 'Turner', 'abigail.turner@student.jaegar.ca', '3.88', 'Business', 32, 'BUSI', '2001-07-07'),
    ('100458001', 'Ryan',  'Clark', 'ryan.clark@student.jaegar.ca', '3.66', 'Business', 29, 'BUSI', '2000-11-23'),

    -- Philosophy (PHIL)
    ('100444678', 'Eric', 'Glass', 'eric.glass@student.jaegar.ca', '3.67', 'Philosophy', 29, 'PHIL', '2000-12-29'),
    ('100447111', 'Rachel', 'Adams', 'rachel.adams@student.jaegar.ca', '2.90', 'Philosophy', 30, 'PHIL', '1999-05-02'),
    ('100453222', 'Brandon', 'Nelson', 'brandon.nelson@student.jaegar.ca', '3.53', 'Philosophy', 28, 'PHIL', '2001-02-28'),
    ('100459333', 'Meghan', 'Wright', 'meghan.wright@student.jaegar.ca', '3.82', 'Philosophy', 31, 'PHIL', '2000-08-19'),


     -- History (HIST)
     ('100459321', 'Robyn', 'Davis', 'robyn.davis@student.jaegar.ca', '3.90', 'History', 31, 'HIST', '1998-03-08'),
     ('100452222', 'Kevin', 'Walker', 'kevin.walker@student.jaegar.ca','3.68', 'History', 26, 'HIST', '2000-12-05'),
     ('100455555', 'Angela', 'Brooks', 'angela.brooks@student.jaegar.ca', '3.77', 'History', 29, 'HIST', '1999-03-23'),
     ('100456666', 'Monica', 'Hall', 'monica.hall@student.jaegar.ca', '3.59', 'History', 28, 'HIST', '2001-11-30');


```



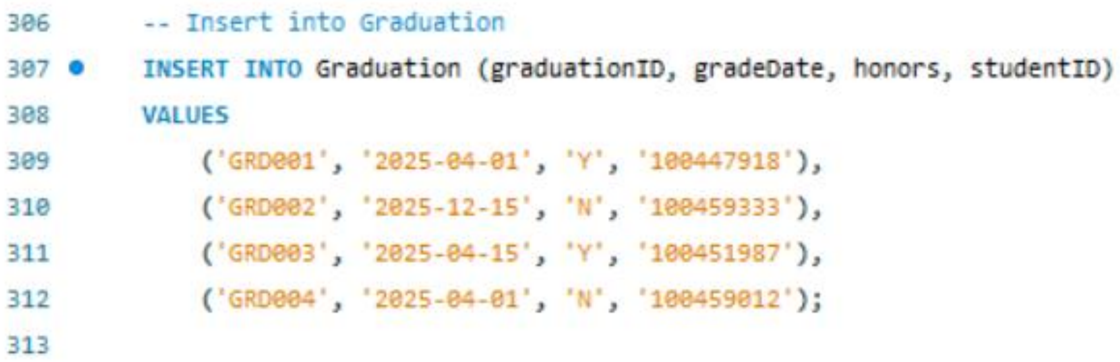
```sql
-- Insert into Enrollment
INSERT INTO Enrollment
  (enrollID, enrollDate, enrollTime, enrollSemester, status, studentID)
VALUES
  ('ENR100','23-May-25','09:00:00','Spring','Inactive','100451987'),
  ('ENR101','15-Jan-25','11:30:00','Spring','Active',  '100458123'),
  ('ENR102','18-Jan-25','13:15:00','Spring','Active',  '100446789'),
  ('ENR103','21-Jan-25','15:45:00','Spring','Active',  '100451234'),
  ('ENR104','24-Jan-25','17:00:00','Spring','Inactive','100444555'),
  ('ENR105','27-Jan-25','09:30:00','Spring','Active',  '100454321'),
  ('ENR106','30-Jan-25','11:00:00','Spring','Active',  '100459012'),
  ('ENR107','02-Feb-25','13:00:00','Spring','Active',  '100447918'),
  ('ENR108','05-Feb-25','15:00:00','Spring','Active',  '100456666'),
  ('ENR109','08-Feb-25','17:30:00','Spring','Active',  '100446123'),
  ('ENR110','11-Feb-25','09:15:00','Spring','Active',  '100455555');
```



```sql
-- Insert into Assessment
-- Using DECIMAL grade values.
INSERT INTO Assessment (assessmentType, classID, studentID, dueDate, grade)
VALUES
    ('Project',    'CLS100', '100447918', '2025-04-09', 85.19),
    ('Project',    'CLS101', '100459012', '2025-04-10', 76.84),
    ('Project',    'CLS102', '100444555', '2025-04-11', 61.89),
    ('Project',    'CLS103', '100454321', '2025-04-12', 91.57),
    ('Quiz',       'CLS104', '100451234', '2025-04-13', 87.93),
    ('Project',    'CLS105', '100446789', '2025-04-14', 98.36),
    ('Quiz',       'CLS106', '100451987', '2025-04-15', 78.49),
    ('Assignment', 'CLS107', '100458123', '2025-04-16', 67.39),
    ('Quiz',       'CLS108', '100462345', '2025-04-17', 83.89),
    ('Project',    'CLS109', '100446123', '2025-04-18', 66.26),
    ('Assignment', 'CLS100', '100451001', '2025-04-19', 81.95),
    ('Assignment', 'CLS101', '100458001', '2025-04-20', 75.65),
    ('Project',    'CLS102', '100444678', '2025-04-21', 64.64),
    ('Assignment', 'CLS103', '100447111', '2025-04-22', 88.37),
    ('Assignment', 'CLS104', '100453222', '2025-04-23', 85.46);
```

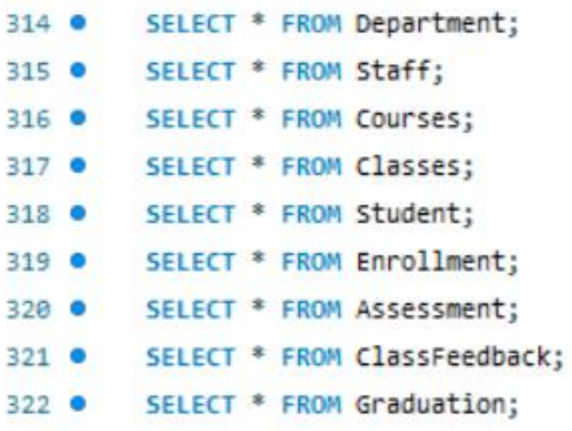


```sql
-- Insert into ClassFeedback
-- Columns: feedbackID, staffID, studentID, content, time (DATETIME), rating (NULL or 'A+' … 'F')
INSERT INTO ClassFeedback (feedbackID, staffID, studentID, content, time, rating)
VALUES
    ('FDB100', 'STF100', '100447918', 'The instructor explained complex topics clearly and made lectures engaging.', '2025-04-09 13:45:58', 'C'),
    ('FDB101', 'STF101', '100459012', 'Excellent teaching style – very organized and approachable in class.', '2025-04-08 17:45:58', 'A+'),
    ('FDB102', 'STF102', '100444555', 'The course felt rushed at times; more in-class examples would be helpful.', '2025-04-07 7:40:58', 'B'),
    ('FDB103', 'STF103', '100454321', 'Instructor encouraged questions and created an inclusive learning space.', '2025-04-06 20:45:58', 'A'),
    ('FDB104', 'STF104', '100451234', 'Good overall, but lectures could benefit from better time management.', '2025-04-05 21:05:58', 'B+'),
    ('FDB105', 'STF105', '100446789', 'Sometimes hard to follow – slides needed more explanation during class.', '2025-04-04 18:45:58', NULL),
    ('FDB106', 'STF106', '100451987', 'The instructor didn’t provide enough support during assessments.', '2025-04-03 12:15:58', 'D'),
    ('FDB107', 'STF107', '100458123', 'I appreciated how the instructor gave regular feedback and stayed interactive.',  '2025-04-02 19:33:58', 'B'),
    ('FDB108', 'STF100', '100462345', 'More enthusiasm during lectures would improve the learning experience.', '2025-04-01 10:19:58', 'C'),
    ('FDB109', 'STF107', '100446123', 'The instructor missed a few classes and didn’t provide make-up material. ', '2025-03-31 17:15:58', 'D'),
    ('FDB110', 'STF105', '100451001', 'Class discussions were encouraged, which made learning more interesting.', '2025-03-30 8:30:58', NULL),
    ('FDB111', 'STF103', '100458001', 'Course content was useful, but delivery was monotonous at times.', '2025-03-29 11:25:58', 'C'),
    ('FDB112', 'STF102', '100444678', 'Best teaching I’ve experienced – passionate and clear throughout the term.', '2025-03-28 12:25:58', 'A'),
    ('FDB113', 'STF106', '100447111', 'The instructor explained every concept with real-world examples. Very helpful.', '2025-03-27 10:22:58', 'B'),
    ('FDB114', 'STF107', '100453222', 'I learned a lot, but wish the pace had been slower during difficult units.', '2025-03-26 19:48:58', 'A-');
```



```sql
-- Insert into Graduation
INSERT INTO Graduation (graduationID, gradeDate, honors, studentID)
VALUES
    ('GRD001', '2025-04-01', 'Y', '100447918'),
    ('GRD002', '2025-12-15', 'N', '100459333'),
    ('GRD003', '2025-04-15', 'Y', '100451987'),
    ('GRD004', '2025-04-01', 'N', '100459012');
```

We can run these below mentioned commands in order to show these tables:-

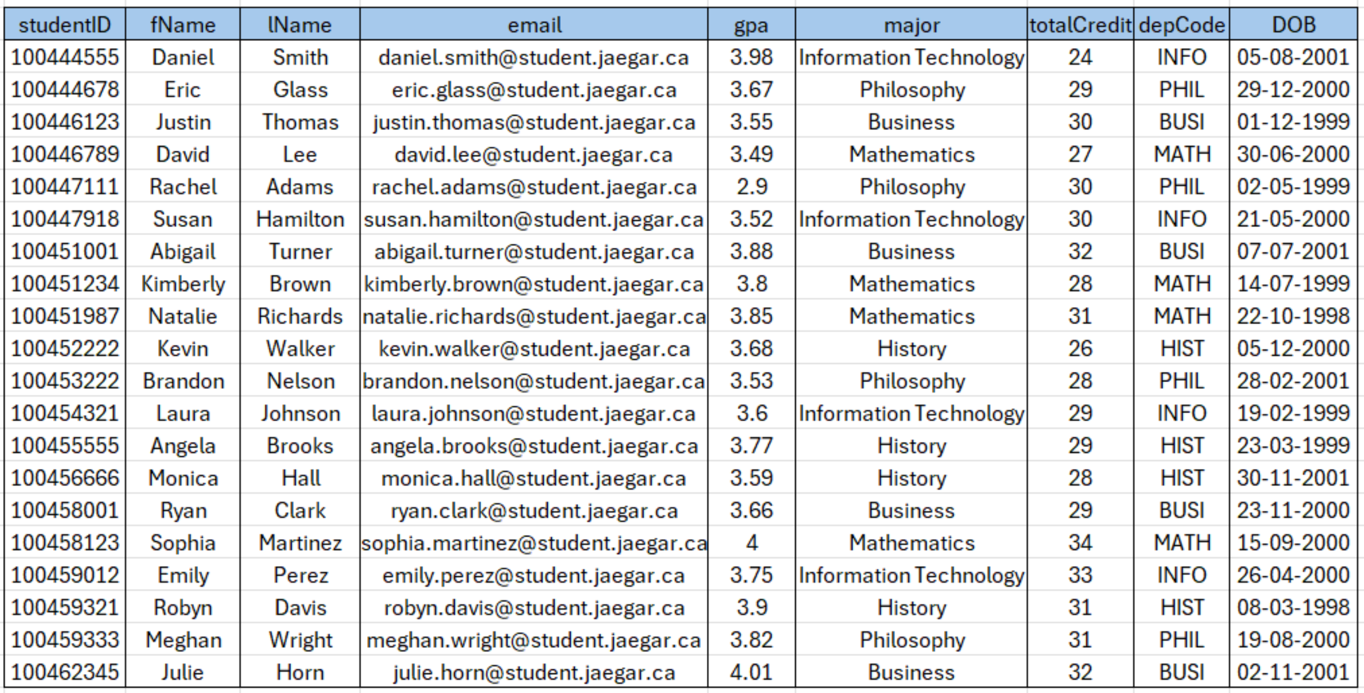


```sql
SELECT * FROM Department;
SELECT * FROM Staff;
SELECT * FROM Courses;
SELECT * FROM Classes;
SELECT * FROM Student;
SELECT * FROM Enrollment;
SELECT * FROM Assessment;
SELECT * FROM ClassFeedback;
SELECT * FROM Graduation;
```

These commands would return the below mentioned tables as the outputs.

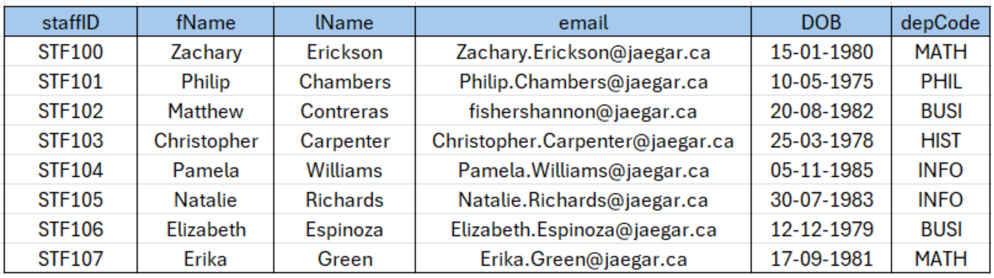
INSTANCES OF THE ACADEMIC SYSTEM

Student Entity



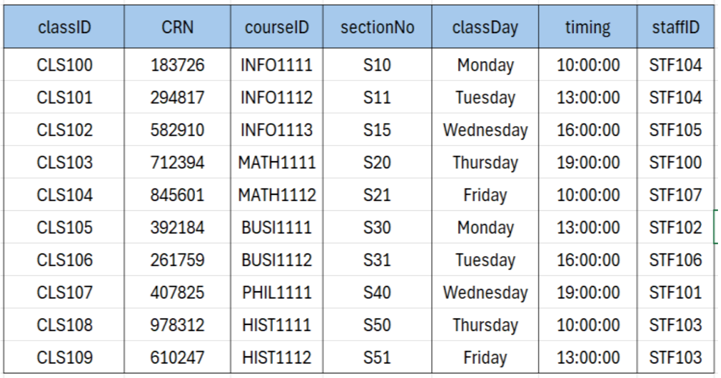
studentID	fName	lName	email	gpa	major	totalCredit	depCode	DOB
100444555	Daniel	Smith	daniel.smith@student.jaegar.ca	3.98	Information Technology	24	INFO	05-08-2001
100444678	Eric	Glass	eric.glass@student.jaegar.ca	3.67	Philosophy	29	PHIL	29-12-2000
100446123	Justin	Thomas	justin.thomas@student.jaegar.ca	3.55	Business	30	BUSI	01-12-1999
100446789	David	Lee	david.lee@student.jaegar.ca	3.49	Mathematics	27	MATH	30-06-2000
100447111	Rachel	Adams	rachel.adams@student.jaegar.ca	2.9	Philosophy	30	PHIL	02-05-1999
100447918	Susan	Hamilton	susan.hamilton@student.jaegar.ca	3.52	Information Technology	30	INFO	21-05-2000
100451001	Abigail	Turner	abigail.turner@student.jaegar.ca	3.88	Business	32	BUSI	07-07-2001
100451234	Kimberly	Brown	kimberly.brown@student.jaegar.ca	3.8	Mathematics	28	MATH	14-07-1999
100451987	Natalie	Richards	natalie.richards@student.jaegar.ca	3.85	Mathematics	31	MATH	22-10-1998
100452222	Kevin	Walker	kevin.walker@student.jaegar.ca	3.68	History	26	HIST	05-12-2000
100453222	Brandon	Nelson	brandon.nelson@student.jaegar.ca	3.53	Philosophy	28	PHIL	28-02-2001
100454321	Laura	Johnson	laura.johnson@student.jaegar.ca	3.6	Information Technology	29	INFO	19-02-1999
100455555	Angela	Brooks	angela.brooks@student.jaegar.ca	3.77	History	29	HIST	23-03-1999
100456666	Monica	Hall	monica.hall@student.jaegar.ca	3.59	History	28	HIST	30-11-2001
100458001	Ryan	Clark	ryan.clark@student.jaegar.ca	3.66	Business	29	BUSI	23-11-2000
100458123	Sophia	Martinez	sophia.martinez@student.jaegar.ca	4	Mathematics	34	MATH	15-09-2000
100459012	Emily	Perez	emily.perez@student.jaegar.ca	3.75	Information Technology	33	INFO	26-04-2000
100459321	Robyn	Davis	robyn.davis@student.jaegar.ca	3.9	History	31	HIST	08-03-1998
100459333	Meghan	Wright	meghan.wright@student.jaegar.ca	3.82	Philosophy	31	PHIL	19-08-2000
100462345	Julie	Horn	julie.horn@student.jaegar.ca	4.01	Business	32	BUSI	02-11-2001

Staff Entity



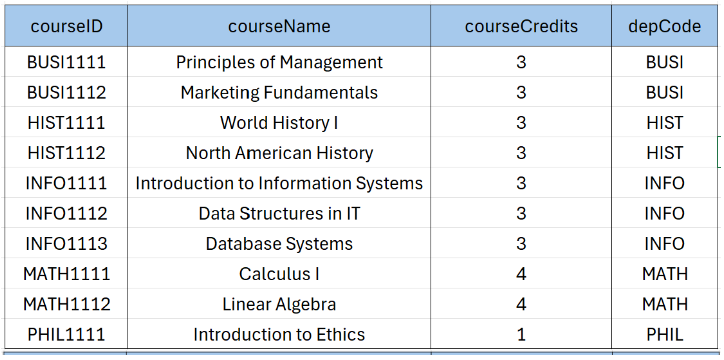
staffID	fName	lName	email	DOB	depCode
STF100	Zachary	Erickson	Zachary.Erickson@jaegar.ca	15-01-1980	MATH
STF101	Philip	Chambers	Philip.Chambers@jaegar.ca	10-05-1975	PHIL
STF102	Matthew	Contreras	fishershannon@jaegar.ca	20-08-1982	BUSI
STF103	Christopher	Carpenter	Christopher.Carpenter@jaegar.ca	25-03-1978	HIST
STF104	Pamela	Williams	Pamela.Williams@jaegar.ca	05-11-1985	INFO
STF105	Natalie	Richards	Natalie.Richards@jaegar.ca	30-07-1983	INFO
STF106	Elizabeth	Espinoza	Elizabeth.Espinoza@jaegar.ca	12-12-1979	BUSI
STF107	Erika	Green	Erika.Green@jaegar.ca	17-09-1981	MATH

Classes Entity



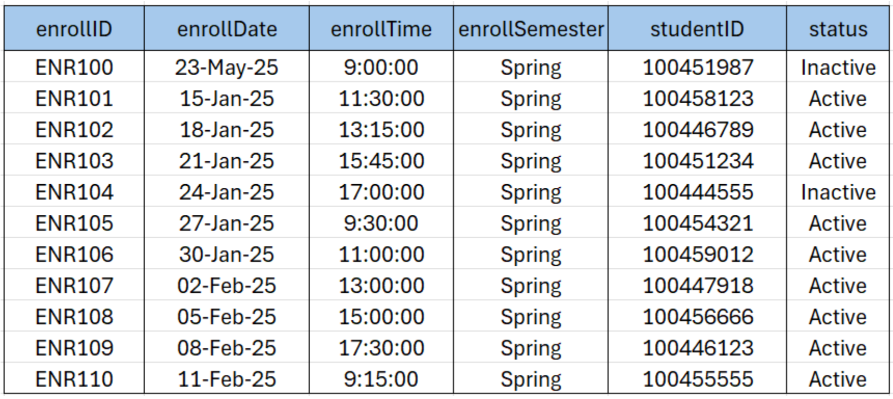
classID	CRN	courseID	sectionNo	classDay	timing	staffID
CLS100	183726	INFO1111	S10	Monday	10:00:00	STF104
CLS101	294817	INFO1112	S11	Tuesday	13:00:00	STF104
CLS102	582910	INFO1113	S15	Wednesday	16:00:00	STF105
CLS103	712394	MATH1111	S20	Thursday	19:00:00	STF100
CLS104	845601	MATH1112	S21	Friday	10:00:00	STF107
CLS105	392184	BUSI1111	S30	Monday	13:00:00	STF102
CLS106	261759	BUSI1112	S31	Tuesday	16:00:00	STF106
CLS107	407825	PHIL1111	S40	Wednesday	19:00:00	STF101
CLS108	978312	HIST1111	S50	Thursday	10:00:00	STF103
CLS109	610247	HIST1112	S51	Friday	13:00:00	STF103

Courses Entity



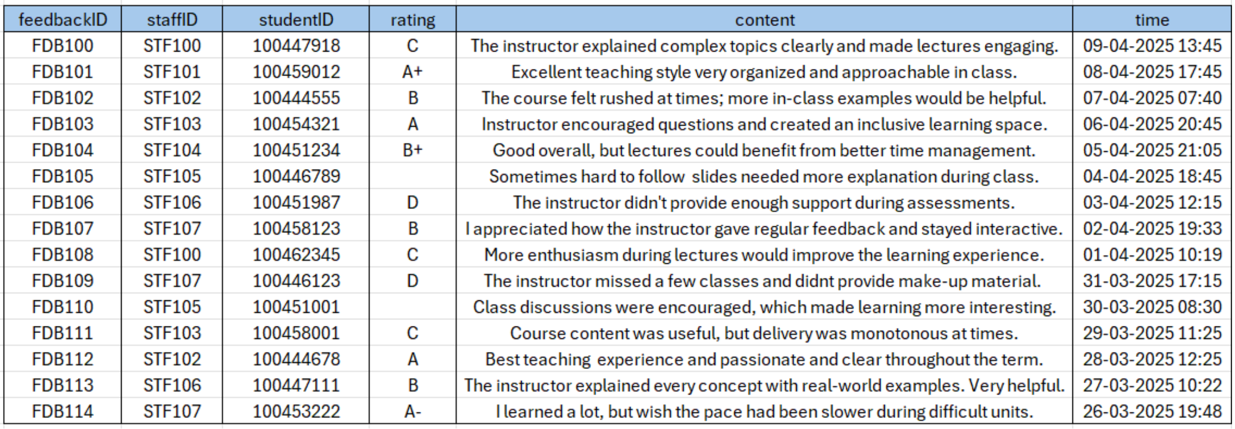
courseID	courseName	courseCredits	depCode
BUSI1111	Principles of Management	3	BUSI
BUSI1112	Marketing Fundamentals	3	BUSI
HIST1111	World History I	3	HIST
HIST1112	North American History	3	HIST
INFO1111	Introduction to Information Systems	3	INFO
INFO1112	Data Structures in IT	3	INFO
INFO1113	Database Systems	3	INFO
MATH1111	Calculus I	4	MATH
MATH1112	Linear Algebra	4	MATH
PHIL1111	Introduction to Ethics	1	PHIL

Enrollment Entity



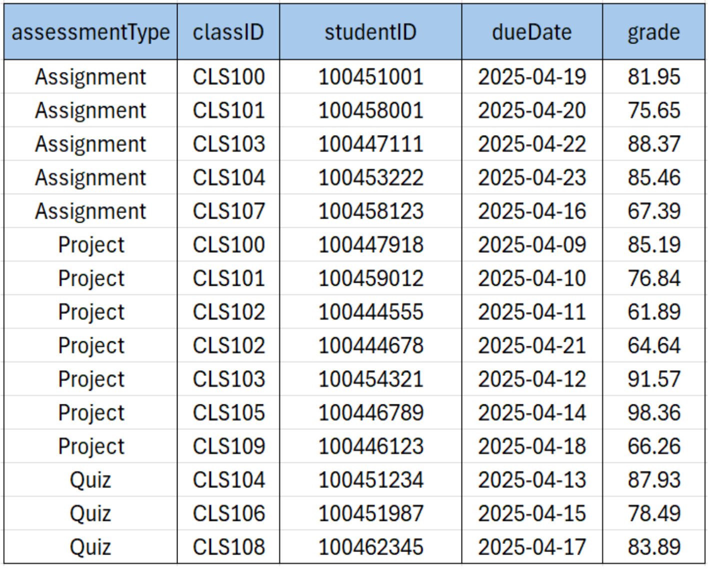
enrollID	enrollDate	enrollTime	enrollSemester	studentID	status
ENR100	23-May-25	9:00:00	Spring	100451987	Inactive
ENR101	15-Jan-25	11:30:00	Spring	100458123	Active
ENR102	18-Jan-25	13:15:00	Spring	100446789	Active
ENR103	21-Jan-25	15:45:00	Spring	100451234	Active
ENR104	24-Jan-25	17:00:00	Spring	100444555	Inactive
ENR105	27-Jan-25	9:30:00	Spring	100454321	Active
ENR106	30-Jan-25	11:00:00	Spring	100459012	Active
ENR107	02-Feb-25	13:00:00	Spring	100447918	Active
ENR108	05-Feb-25	15:00:00	Spring	100456666	Active
ENR109	08-Feb-25	17:30:00	Spring	100446123	Active
ENR110	11-Feb-25	9:15:00	Spring	100455555	Active

Class Feedback Entity



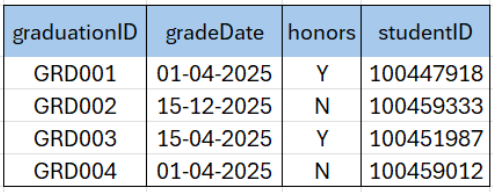
feedbackID	staffID	studentID	rating	content	time
FDB100	STF100	100447918	C	The instructor explained complex topics clearly and made lectures engaging.	09-04-2025 13:45
FDB101	STF101	100459012	A+	Excellent teaching style very organized and approachable in class.	08-04-2025 17:45
FDB102	STF102	100444555	B	The course felt rushed at times; more in-class examples would be helpful.	07-04-2025 07:40
FDB103	STF103	100454321	A	Instructor encouraged questions and created an inclusive learning space.	06-04-2025 20:45
FDB104	STF104	100451234	B+	Good overall, but lectures could benefit from better time management.	05-04-2025 21:05
FDB105	STF105	100446789		Sometimes hard to follow slides needed more explanation during class.	04-04-2025 18:45
FDB106	STF106	100451987	D	The instructor didn't provide enough support during assessments.	03-04-2025 12:15
FDB107	STF107	100458123	B	I appreciated how the instructor gave regular feedback and stayed interactive.	02-04-2025 19:33
FDB108	STF100	100462345	C	More enthusiasm during lectures would improve the learning experience.	01-04-2025 10:19
FDB109	STF107	100446123	D	The instructor missed a few classes and didnt provide make-up material.	31-03-2025 17:15
FDB110	STF105	100451001		Class discussions were encouraged, which made learning more interesting.	30-03-2025 08:30
FDB111	STF103	100458001	C	Course content was useful, but delivery was monotonous at times.	29-03-2025 11:25
FDB112	STF102	100444678	A	Best teaching experience and passionate and clear throughout the term.	28-03-2025 12:25
FDB113	STF106	100447111	B	The instructor explained every concept with real-world examples. Very helpful.	27-03-2025 10:22
FDB114	STF107	100453222	A-	I learned a lot, but wish the pace had been slower during difficult units.	26-03-2025 19:48

Assesement Entity



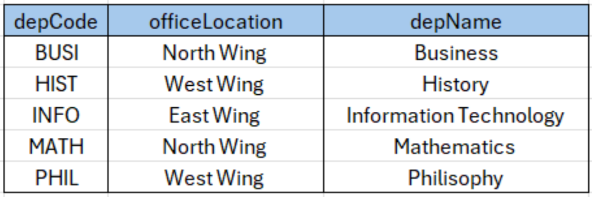
assessmentType	classID	studentID	dueDate	grade
Assignment	CLS100	100451001	2025-04-19	81.95
Assignment	CLS101	100458001	2025-04-20	75.65
Assignment	CLS103	100447111	2025-04-22	88.37
Assignment	CLS104	100453222	2025-04-23	85.46
Assignment	CLS107	100458123	2025-04-16	67.39
Project	CLS100	100447918	2025-04-09	85.19
Project	CLS101	100459012	2025-04-10	76.84
Project	CLS102	100444555	2025-04-11	61.89
Project	CLS102	100444678	2025-04-21	64.64
Project	CLS103	100454321	2025-04-12	91.57
Project	CLS105	100446789	2025-04-14	98.36
Project	CLS109	100446123	2025-04-18	66.26
Quiz	CLS104	100451234	2025-04-13	87.93
Quiz	CLS106	100451987	2025-04-15	78.49
Quiz	CLS108	100462345	2025-04-17	83.89

Graduation Entity



graduationID	gradeDate	honors	studentID
GRD001	01-04-2025	Y	100447918
GRD002	15-12-2025	N	100459333
GRD003	15-04-2025	Y	100451987
GRD004	01-04-2025	N	100459012

Department Entity



depCode	officeLocation	depName
BUSI	North Wing	Business
HIST	West Wing	History
INFO	East Wing	Information Technology
MATH	North Wing	Mathematics
PHIL	West Wing	Philisophy

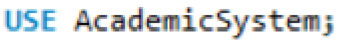
3. USING THE PRODUCT

3.1. Special Considerations

Throughout this project, we developed and tested several SQL queries to retrieve meaningful and practical data from the AcademicSystem relational database. These queries reflect real-world scenarios, such as department planning, instructor workload analysis, class scheduling, and performance evaluation. They also demonstrate the robustness of our database structure in supporting fast, reliable, and insightful reporting for academic operations.

3.2. Instructions for use

Before executing any query, ensure that the following command is run to activate the correct database:

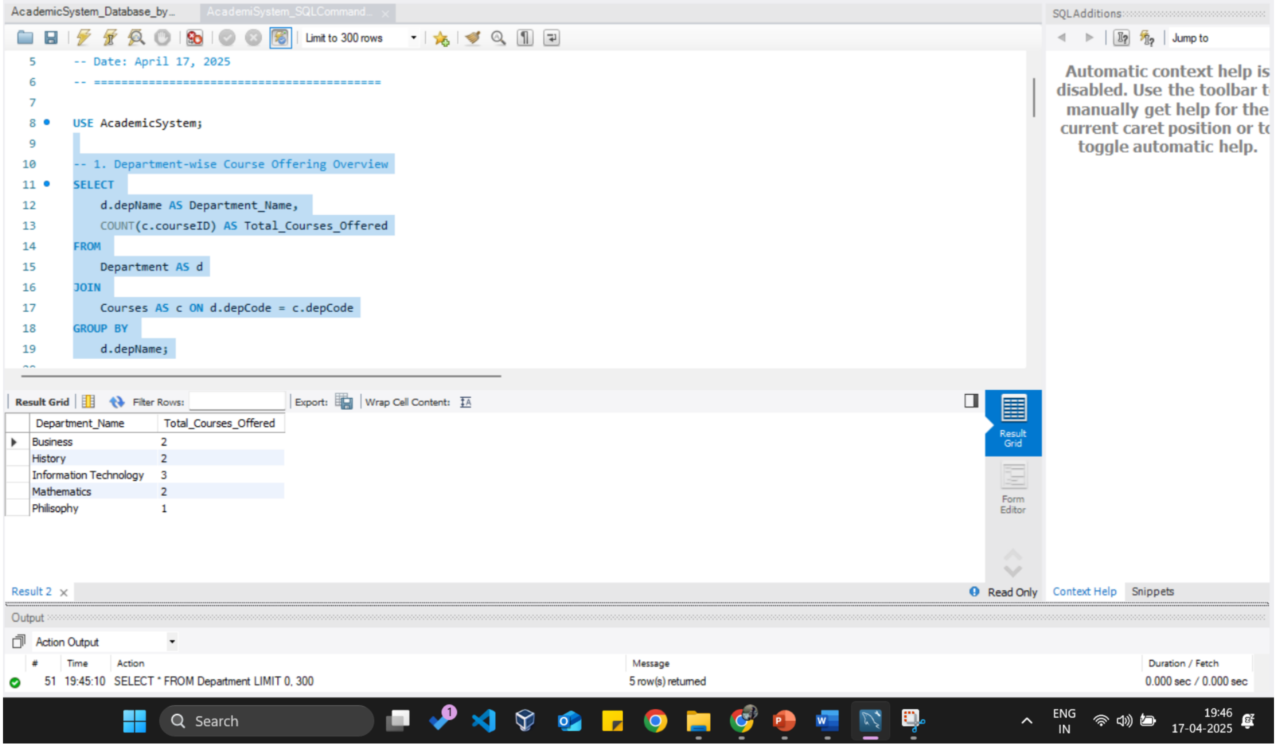


```
USE AcademicSystem;
```

This command sets the **AcademicSystem** as the active schema in your SQL environment. To run any query, simply select it and click the **execute/run button** in your SQL editor or benchmarking tool.

To run the SQL commands, select the command and click on the ⚡ button.

Query 1 – Department-wise Course Offering



```
-- Date: April 17, 2025
-- ==========================================

USE AcademicSystem;

-- 1. Department-wise Course Offering Overview
SELECT
    d.depName AS Department_Name,
    COUNT(c.courseID) AS Total_Courses_Offered
FROM
    Department AS d
JOIN
    Courses AS c ON d.depCode = c.depCode
GROUP BY
    d.depName;
```

Department_Name	Total_Courses_Offered
Business	2
History	2
Information Technology	3
Mathematics	2
Philisophy	1

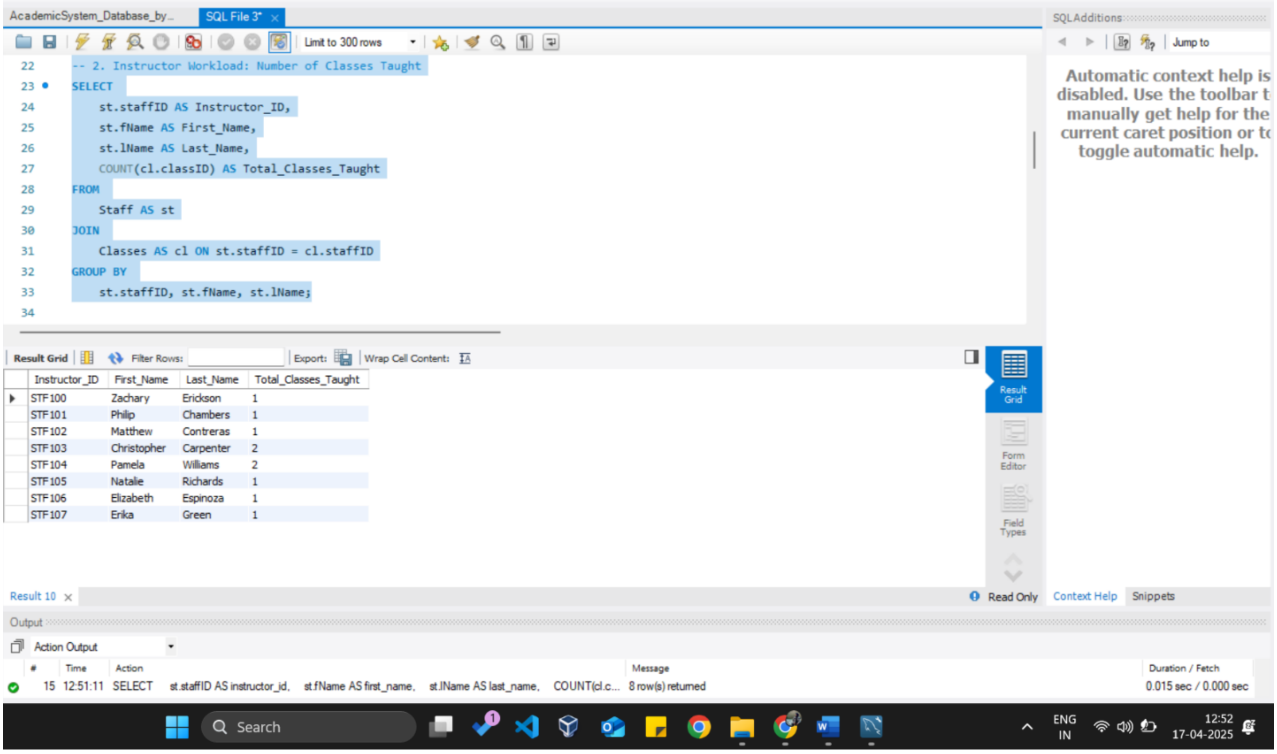
```
#   Time      Action                                   Message               Duration / Fetch
51  19:45:10  SELECT * FROM Department LIMIT 0, 300    5 row(s) returned     0.000 sec / 0.000 sec
```

This query displays the number of courses offered by each department.
It performs an **INNER JOIN** between the Department and Courses tables using the depCode foreign key.

Using the COUNT() function, it calculates how many courses exist under each department.

The result is grouped by depName and given clear aliases for a clean and readable summary — ideal for curriculum planning and resource allocation.

Query 2 – Instructor Workload (Classes Taught)



```
-- 2. Instructor Workload: Number of Classes Taught
SELECT
    st.staffID AS Instructor_ID,
    st.fName AS First_Name,
    st.lName AS Last_Name,
    COUNT(cl.classID) AS Total_Classes_Taught
FROM
    Staff AS st
JOIN
    Classes AS cl ON st.staffID = cl.staffID
GROUP BY
    st.staffID, st.fName, st.lName;
```

Instructor_ID	First_Name	Last_Name	Total_Classes_Taught
STF100	Zachary	Erickson	1
STF101	Philip	Chambers	1
STF102	Matthew	Contreras	1
STF103	Christopher	Carpenter	2
STF104	Pamela	Williams	2
STF105	Natalie	Richards	1
STF106	Elizabeth	Espinoza	1
STF107	Erika	Green	1

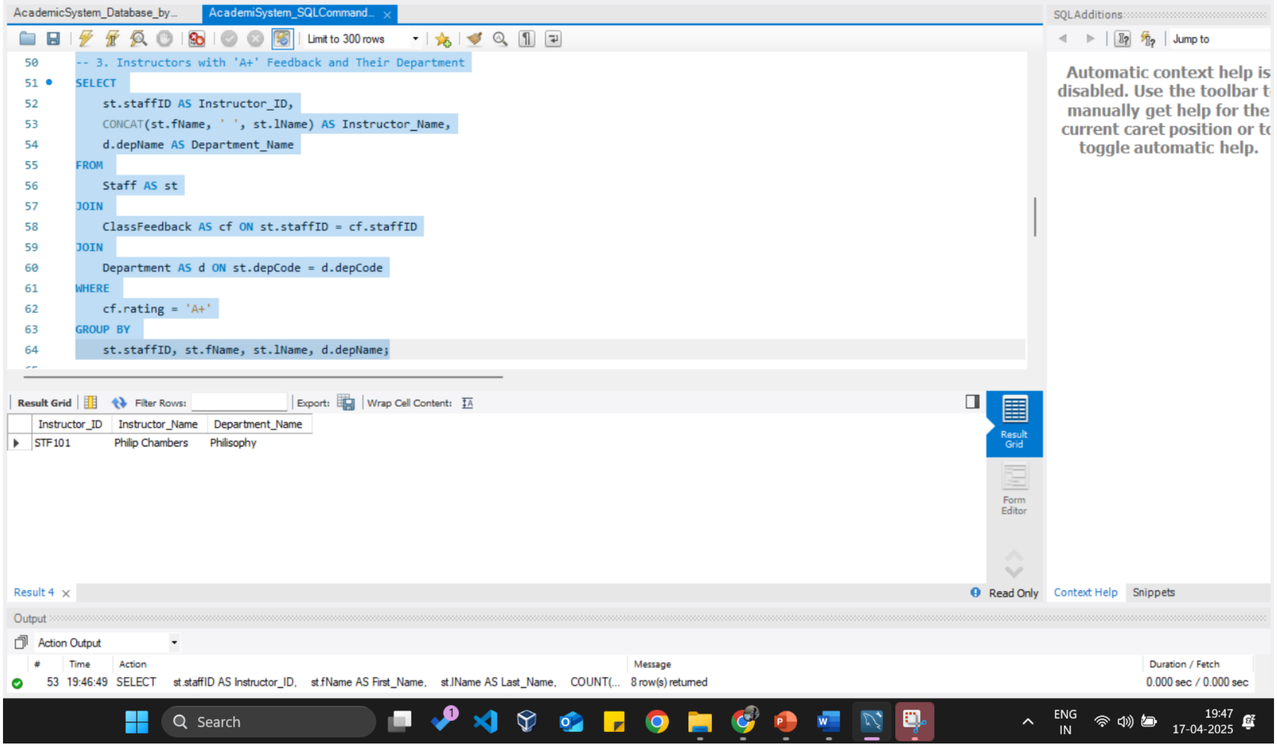
```
#   Time      Action                                                                       Message              Duration / Fetch
15  12:51:11  SELECT  st.staffID AS instructor_id,  st.fName AS first_name,  st.lName AS last_name,  COUNT(cl.c...  8 row(s) returned  0.015 sec / 0.000 sec
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or tc toggle automatic help.

This query provides insight into how many classes each instructor is currently teaching.
It joins the Staff and Classes tables on the staffID field and uses the COUNT() function to tally the number of classes linked to each instructor.

The results are grouped by staff ID and name, giving a complete picture of instructor workload — useful for department heads to balance teaching responsibilities.

Query 3 – Instructors with ‘A+’ Feedback



```sql
-- 3. Instructors with 'A+' Feedback and Their Department
SELECT
    st.staffID AS Instructor_ID,
    CONCAT(st.fName, ' ', st.lName) AS Instructor_Name,
    d.depName AS Department_Name
FROM
    Staff AS st
JOIN
    ClassFeedback AS cf ON st.staffID = cf.staffID
JOIN
    Department AS d ON st.depCode = d.depCode
WHERE
    cf.rating = 'A+'
GROUP BY
    st.staffID, st.fName, st.lName, d.depName;
```

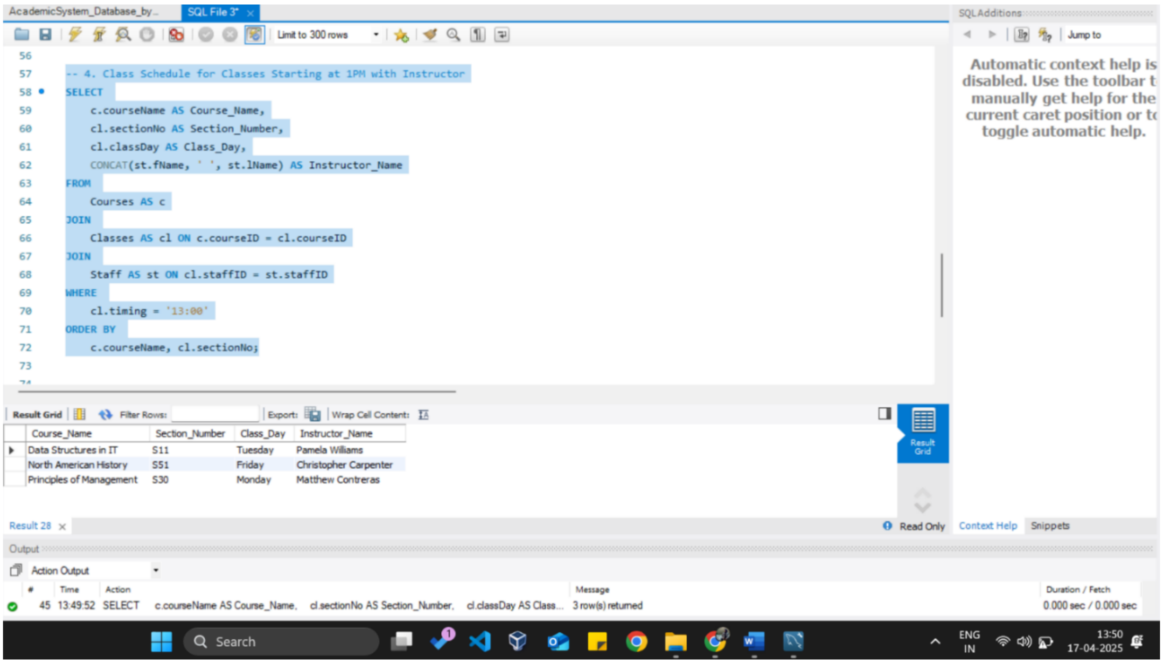
Instructor_ID	Instructor_Name	Department_Name
STF101	Philip Chambers	Philisophy

```
#   Time      Action                                                                   Message               Duration / Fetch
53  19:46:49  SELECT  st.staffID AS Instructor_ID,  st.fName AS First_Name,  st.lName AS Last_Name,  COUNT(...  8 row(s) returned  0.000 sec / 0.000 sec
```

This query identifies instructors who received an 'A+' rating from students in their class feedback. It joins three tables: Staff, ClassFeedback, and Department, using staffID and depCode as joining keys.

The CONCAT() function is used to combine the instructor's first and last name into a single column. The WHERE clause filters only those feedback entries where the rating is 'A+'. This query is ideal for performance recognition and departmental analysis of teaching excellence.

Query 4 – Class Schedule at 1 PM



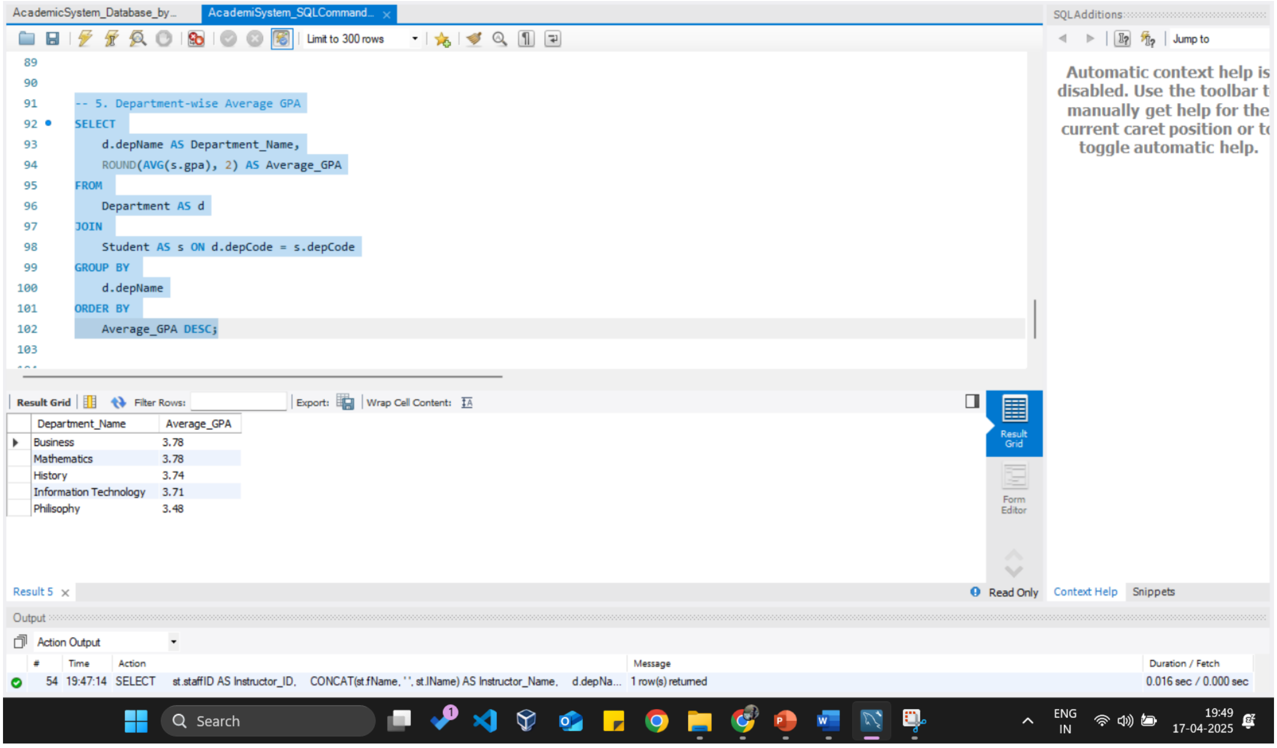
```sql
-- 4. Class Schedule for Classes Starting at 1PM with Instructor
SELECT
    c.courseName AS Course_Name,
    cl.sectionNo AS Section_Number,
    cl.classDay AS Class_Day,
    CONCAT(st.fName, ' ', st.lName) AS Instructor_Name
FROM
    Courses AS c
JOIN
    Classes AS cl ON c.courseID = cl.courseID
JOIN
    Staff AS st ON cl.staffID = st.staffID
WHERE
    cl.timing = '13:00'
ORDER BY
    c.courseName, cl.sectionNo;
```

Course_Name	Section_Number	Class_Day	Instructor_Name
Data Structures in IT	S11	Tuesday	Pamela Williams
North American History	S51	Friday	Christopher Carpenter
Principles of Management	S30	Monday	Matthew Contreras

```
#   Time      Action                                                                          Message               Duration / Fetch
45  13:49:52  SELECT  c.courseName AS Course_Name,  cl.sectionNo AS Section_Number,  cl.classDay AS Class...  3 row(s) returned  0.000 sec / 0.000 sec
```

This query fetches all classes that start at exactly 1:00 PM, along with their course details and assigned instructors. It joins the Courses, Classes, and Staff tables using courseID and staffID. The WHERE clause restricts results to those classes with a timing of '13:00'. Data is displayed in an organized format showing course name, section, class day, and instructor name — and sorted by course and section.
This is especially useful for academic advisors, students, and timetabling staff.

Query 5 – Department-wise Average GPA



```sql
-- 5. Department-wise Average GPA
SELECT
    d.depName AS Department_Name,
    ROUND(AVG(s.gpa), 2) AS Average_GPA
FROM
    Department AS d
JOIN
    Student AS s ON d.depCode = s.depCode
GROUP BY
    d.depName
ORDER BY
    Average_GPA DESC;
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

Department_Name	Average_GPA
Business	3.78
Mathematics	3.78
History	3.74
Information Technology	3.71
Philisophy	3.48

```
#   Time      Action                                                                                   Message            Duration / Fetch
54  19:47:14  SELECT  st.staffID AS Instructor_ID,  CONCAT(st.fName, ' ', st.lName) AS Instructor_Name,  d.depNa...  1 row(s) returned  0.016 sec / 0.000 sec
```

This query calculates the average GPA of students in each department. It joins the Department and Student tables using the shared depCode FK. The AVG() function computes the mean GPA, and ROUND(..., 2) is applied to limit the result to two decimal places for clarity.
Further, Grouped by department name and ordered from highest to lowest GPA, the result offers valuable insight into academic performance across departments.