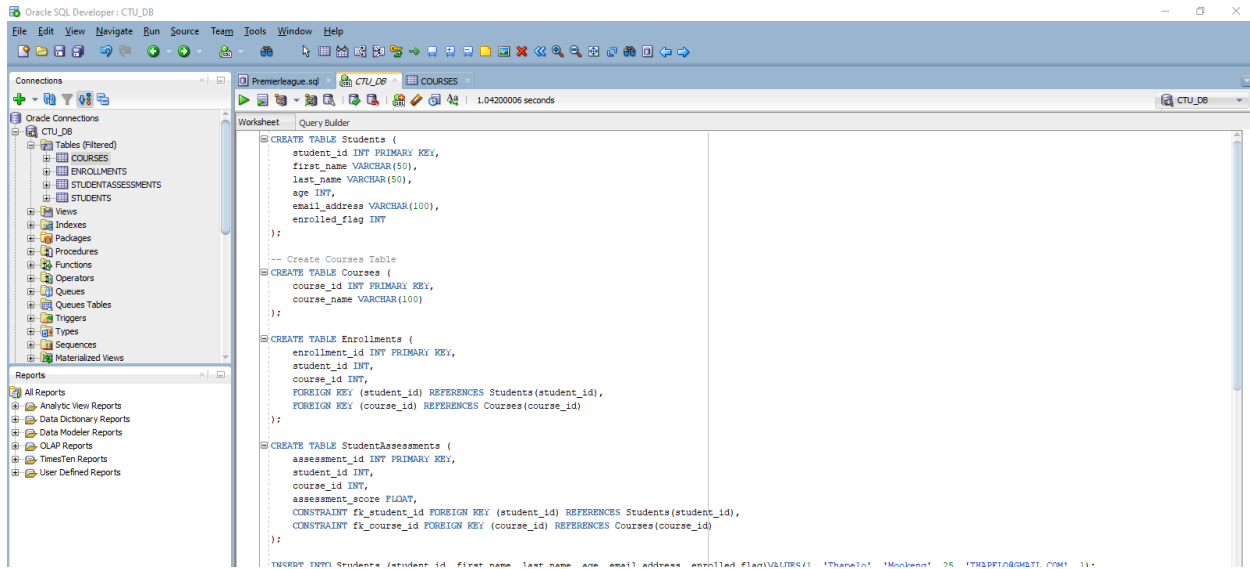


**Information Technology FET
Certificate: Information
Technology: Database
Development Software
Development Advanced Java
Semester 2 PRG522
Thapelo Mbookeng**

Tables Created for Students, Course, Enrollments and StudentAssessments



Code for inserting data on Student table

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(1, 'Thapelo', 'Mookeng', 25, 'THAPELO@GMAIL.COM', 1);
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(2, 'Sipho', 'Magwaza', 22, 'SIPHO@YAHOO.COM', 1);
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(3, 'Mike', 'Tyson', 28, 'mike@webmail.com', 1);
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(4, 'Lerato', 'Mojela', 24, 'lerato@ctu.com', 1);
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(5, 'Tumi', 'Lego', 30, 'tumi@gmail.com', 1);
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(6, 'Faith', 'Ntuli', 26, 'FAITH@YAHOO.COM', 1);
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(7, 'Renny', 'Martin', 23, 'renny@multichoice.com', 1);
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(8, 'John', 'Doe', 22, 'john@WEBMAIL.com', 1);
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(9, 'Jane', 'Smith', 25, 'jane@Yahoo.com', 1);
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)VALUES(10, 'David', 'Brown', 30, 'david@gmail.com', 0);
```

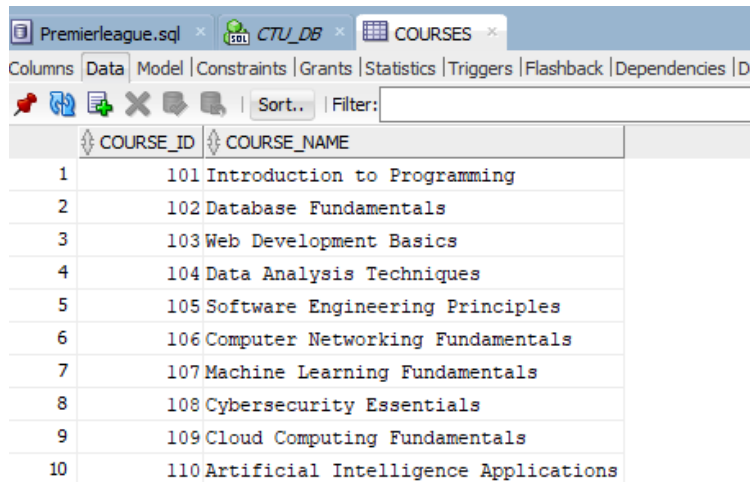
Output

Premierleague.sql x CTU_DB x STUDENTS x						
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL						
Sort.. Filter:						
	STUDENT_ID	FIRST_NAME	LAST_NAME	AGE	EMAIL_ADDRESS	ENROLLED_FLAG
1	1	Thapelo	Mookeng	25	THAPELO@GMAIL.COM	1
2	2	Sipho	Magwaza	22	SIPHO@YAHOO.COM	1
3	3	Mike	Tyson	28	mike@webmail.com	1
4	4	Lerato	Mojela	24	lerato@ctu.com	1
5	5	Tumi	Lego	30	tumi@gmail.com	1
6	6	Faith	Ntuli	26	FAITH@YAHOO.COM	1
7	7	Renny	Martin	23	renny@multichoice.com	1
8	8	John	Doe	22	john@WEBMAIL.com	1
9	9	Jane	Smith	25	jane@Yahoo.com	1
10	10	David	Brown	30	david@gmail.com	0

Code for inserting data on Courses table

```
INSERT INTO Courses (course_id, course_name)VALUES(101, 'Introduction to Programming');
INSERT INTO Courses (course_id, course_name)VALUES (102, 'Database Fundamentals');
INSERT INTO Courses (course_id, course_name)VALUES(103, 'Web Development Basics');
INSERT INTO Courses (course_id, course_name)VALUES(104, 'Data Analysis Techniques');
INSERT INTO Courses (course_id, course_name)VALUES(105, 'Software Engineering Principles');
INSERT INTO Courses (course_id, course_name)VALUES(106, 'Computer Networking Fundamentals');
INSERT INTO Courses (course_id, course_name)VALUES(107, 'Machine Learning Fundamentals');
INSERT INTO Courses (course_id, course_name)VALUES(108, 'Cybersecurity Essentials');
INSERT INTO Courses (course_id, course_name)VALUES(109, 'Cloud Computing Fundamentals');
INSERT INTO Courses (course_id, course_name)VALUES(110, 'Artificial Intelligence Applications');
```

Output

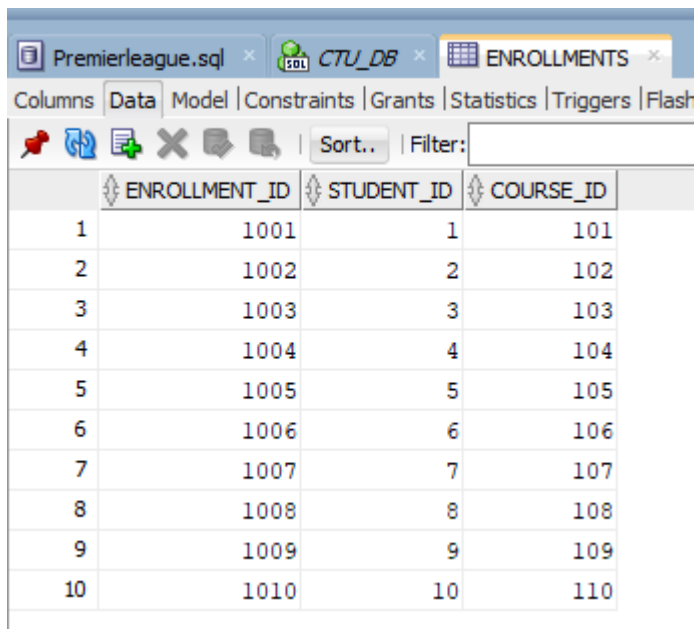


	COURSE_ID	COURSE_NAME
1	101	Introduction to Programming
2	102	Database Fundamentals
3	103	Web Development Basics
4	104	Data Analysis Techniques
5	105	Software Engineering Principles
6	106	Computer Networking Fundamentals
7	107	Machine Learning Fundamentals
8	108	Cybersecurity Essentials
9	109	Cloud Computing Fundamentals
10	110	Artificial Intelligence Applications

Code for inserting data on Enrollments table

```
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1001, 1, 101);
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1002, 2, 102);
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1003, 3, 103);
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1004, 4, 104);
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1005, 5, 105);
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1006, 6, 106);
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1007, 7, 107);
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1008, 8, 108);
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1009, 9, 109);
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1010, 10, 110);
```

OutPut



The screenshot shows a database management interface with three tabs: 'Premierleague.sql', 'CTU_DB', and 'ENROLLMENTS'. The 'ENROLLMENTS' tab is active, displaying a table with four columns: 'ENROLLMENT_ID', 'STUDENT_ID', and 'COURSE_ID'. The table contains 10 rows of data, numbered 1 to 10. The interface includes a toolbar with various icons and a 'Filter:' input field.

	ENROLLMENT_ID	STUDENT_ID	COURSE_ID
1	1001	1	101
2	1002	2	102
3	1003	3	103
4	1004	4	104
5	1005	5	105
6	1006	6	106
7	1007	7	107
8	1008	8	108
9	1009	9	109
10	1010	10	110

Code for inserting data on Student Assessments table

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(1, 1, 101, 85.5);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(2, 2, 102, 78.9);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(3, 3, 103, 92.3);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(4, 4, 104, 70.2);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(5, 5, 105, 88.7);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(6, 6, 106, 81.4);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(7, 7, 107, 95.0);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(8, 8, 108, 73.6);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(9, 9, 109, 86.8);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(10, 10, 110, 79.1);
```

OutPut

Premierleague.sqlCTU_DBSTUDENTASSESSMENTS					
ColumnsDataModelConstraintsGrantsStatisticsTriggersFlashbackDependenciesDetailsPartitionsIndexesSQL					
Sort..Filter:					
	ASSESSMENT_ID	STUDENT...	COURSE_ID	ASSESSMENT_SCORE	
1	1	1	101	85.5	
2	2	2	102	78.9	
3	3	3	103	92.3	
4	4	4	104	70.2	
5	5	5	105	88.7	
6	6	6	106	81.4	
7	7	7	107	95	
8	8	8	108	73.6	
9	9	9	109	86.8	
10	10	10	110	79.1	

Code & output for Generating customized student reports:

Premierleague.sqlCTU_DBSTUDENTASSESSMENTS

0.87900001 seconds

WorksheetQuery Builder

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(9, 9, 109, 86.8);
INSERT INTO StudentAssessments (assessment_id, student_id, course_id, assessment_score)VALUES(10, 10, 110, 79.1);

SELECT
    first_name || ' ' || last_name AS full_name,
    age,
    LOWER(email_address) AS email_address,
    CASE
        WHEN enrolled_flag = 1 THEN 'Enrolled'
        ELSE 'Not Enrolled'
    END AS enrollment_status
FROM
    Students;
```

Script Output

Task completed in 0.879 seconds

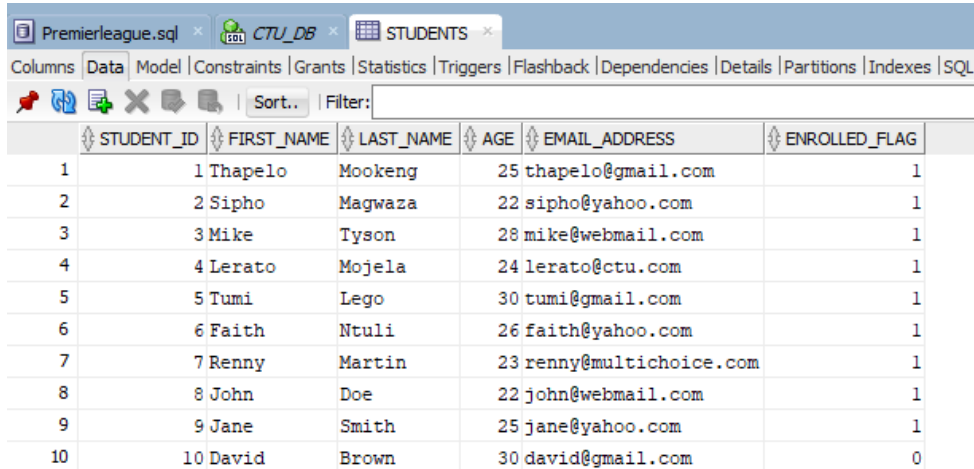
Error report -
ORA-00001: unique constraint (SYS.SYS_C008350) violated

FULL_NAME	AGE	EMAIL_ADDRESS
Thapelo Mookeng	25	thapelo@gmail.com
Sipho Magwaza	22	sipho@yahoo.com
Mike Tyson	28	mike@webmail.com
Lerato Mojela	24	lerato@ctu.com
Tumi Lego	30	tumi@gmail.com
Faith Ntuli	26	faith@yahoo.com
Renny Martin	23	renny@multichoice.com
John Doe	22	john@webmail.com
Jane Smith	25	jane@yahoo.com
David Brown	30	david@gmail.com

10 rows selected.

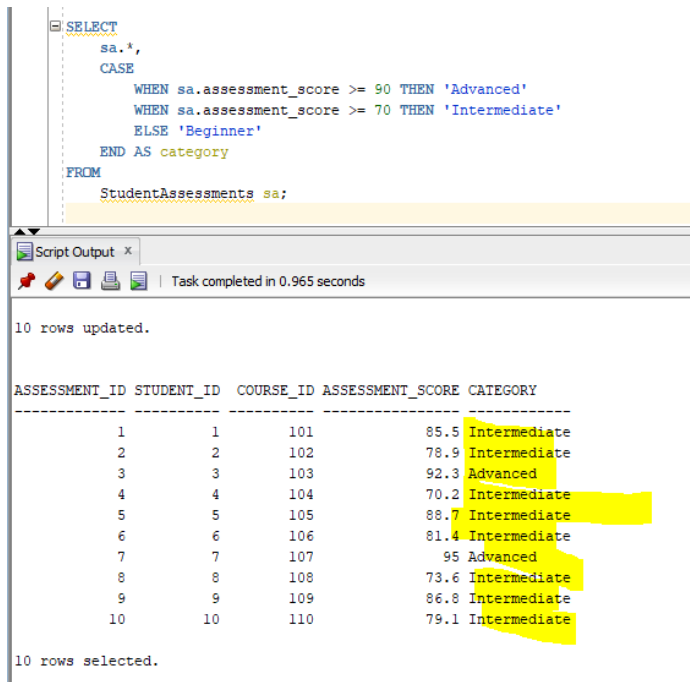
Updated email to be on lowercase code: All email address are on LowerCase

```
UPDATE Students
SET email_address = LOWER(email_address);
```



STUDENT_ID	FIRST_NAME	LAST_NAME	AGE	EMAIL_ADDRESS	ENROLLED_FLAG
1	Thapelo	Mookeng	25	thapelo@gmail.com	1
2	Sipho	Magwaza	22	sipho@yahoo.com	1
3	Mike	Tyson	28	mike@webmail.com	1
4	Lerato	Mojela	24	lerato@ctu.com	1
5	Tumi	Lego	30	tumi@gmail.com	1
6	Faith	Ntuli	26	faith@yahoo.com	1
7	Renny	Martin	23	renny@multichoice.com	1
8	John	Doe	22	john@webmail.com	1
9	Jane	Smith	25	jane@yahoo.com	1
10	David	Brown	30	david@gmail.com	0

Classifying students into different categories based on assessment scores (using a conditional expression): Code & OutPut



```
SELECT
  sa.*,
  CASE
    WHEN sa.assessment_score >= 90 THEN 'Advanced'
    WHEN sa.assessment_score >= 70 THEN 'Intermediate'
    ELSE 'Beginner'
  END AS category
FROM
  StudentAssessments sa;
```

10 rows updated.

ASSESSMENT_ID	STUDENT_ID	COURSE_ID	ASSESSMENT_SCORE	CATEGORY
1	1	101	85.5	Intermediate
2	2	102	78.9	Intermediate
3	3	103	92.3	Advanced
4	4	104	70.2	Intermediate
5	5	105	88.7	Intermediate
6	6	106	81.4	Intermediate
7	7	107	95	Advanced
8	8	108	73.6	Intermediate
9	9	109	86.8	Intermediate
10	10	110	79.1	Intermediate

10 rows selected.

Generating reports on course performance: Code & output

Premierleague.sql CTU_DB STUDENTS 1.28499997 seconds

Worksheet Query Builder

```
SELECT
  c.course_name,
  COUNT(e.student_id) AS num_students_enrolled,
  AVG(s.assessment_score) AS avg_assessment_score
FROM
  Courses c
JOIN
  Enrollments e ON c.course_id = e.course_id
JOIN
  StudentAssessments s ON e.student_id = s.student_id
GROUP BY
  c.course_name;
```

Script Output x Task completed in 1.285 seconds

10 rows selected.

COURSE_NAME	NUM_STUDENTS_ENROLLED	AVG_ASSESSMENT_SCORE
Introduction to Programming	1	85.5
Database Fundamentals	1	79.9
Web Development Basics	1	92.3
Data Analysis Techniques	1	70.2
Software Engineering Principles	1	88.7
Computer Networking Fundamentals	1	81.4
Machine Learning Fundamentals	1	95
Cybersecurity Essentials	1	73.6
Cloud Computing Fundamentals	1	86.8
Artificial Intelligence Applications	1	79.1

10 rows selected.

Retrieving each student's full name, email address, and the names of the courses they are enrolled in:

CODE & OUTPUT

<pre>SELECT s.first_name ' ' s.last_name AS full_name, s.email_address, c.course_name FROM Students s JOIN Enrollments e ON s.student_id = e.student_id JOIN Courses c ON e.course_id = c.course_id WHERE s.enrolled_flag = 1 ORDER BY full_name;</pre>	
Script Output x Task completed in 1.19 seconds	
FULL_NAME	EMAIL_ADDRESS
Faith Ntuli	faith@yahoo.com
Jane Smith	jane@yahoo.com
John Doe	john@webmail.com
Lerato Mojela	lerato@ctu.com
Mike Tyson	mike@webmail.com
Renny Martin	renny@multichoice.com
Sipho Magwaza	sipho@yahoo.com
Thapelo Mookeng	thapelo@gmail.com
Tumi Lego	tumi@gmail.com

Raw Code:

```
CREATE TABLE Students (  
    student_id INT PRIMARY KEY,  
    first_name VARCHAR(50),  
    last_name VARCHAR(50),  
    age INT,  
    email_address VARCHAR(100),  
    enrolled_flag INT  
);
```

-- Create Courses Table

```
CREATE TABLE Courses (  
    course_id INT PRIMARY KEY,  
    course_name VARCHAR(100)  
);
```

```
CREATE TABLE Enrollments (  
    enrollment_id INT PRIMARY KEY,  
    student_id INT,  
    course_id INT,  
    FOREIGN KEY (student_id) REFERENCES Students(student_id),  
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)  
);
```

```
CREATE TABLE StudentAssessments (  
    assessment_id INT PRIMARY KEY,  
    student_id INT,  
    course_id INT,  
    assessment_score FLOAT,
```

```
CONSTRAINT fk_student_id FOREIGN KEY (student_id) REFERENCES Students(student_id),  
CONSTRAINT fk_course_id FOREIGN KEY (course_id) REFERENCES Courses(course_id)  
);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(1, 'Thapelo', 'Mookeng', 25, 'THAPELO@GMAIL.COM', 1);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(2, 'Sipho', 'Magwaza', 22, 'SIPHO@YAHOO.COM', 1);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(3, 'Mike', 'Tyson', 28, 'mike@webmail.com', 1);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(4, 'Lerato', 'Mojela', 24, 'lerato@ctu.com', 1);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(5, 'Tumi', 'Lego', 30, 'tumi@gmail.com', 1);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(6, 'Faith', 'Ntuli', 26, 'FAITH@YAHOO.COM', 1);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(7, 'Renny', 'Martin', 23, 'renny@multichoice.com', 1);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(8, 'John', 'Doe', 22, 'john@WEBMAIL.com', 1);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(9, 'Jane', 'Smith', 25, 'jane@Yahoo.com', 1);
```

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address,  
enrolled_flag)VALUES(10, 'David', 'Brown', 30, 'david@gmail.com', 0);
```

```
INSERT INTO Courses (course_id, course_name)VALUES(101, 'Introduction to Programming');
```

```
INSERT INTO Courses (course_id, course_name)VALUES (102, 'Database Fundamentals');
```

```
INSERT INTO Courses (course_id, course_name)VALUES(103, 'Web Development Basics');
```

```
INSERT INTO Courses (course_id, course_name)VALUES(104, 'Data Analysis Techniques');
```

```
INSERT INTO Courses (course_id, course_name)VALUES(105, 'Software Engineering Principles');
```

```
INSERT INTO Courses (course_id, course_name)VALUES(106, 'Computer Networking Fundamentals');
```

```
INSERT INTO Courses (course_id, course_name)VALUES(107, 'Machine Learning Fundamentals');
```

```
INSERT INTO Courses (course_id, course_name)VALUES(108, 'Cybersecurity Essentials');  
INSERT INTO Courses (course_id, course_name)VALUES(109, 'Cloud Computing Fundamentals');  
INSERT INTO Courses (course_id, course_name)VALUES(110, 'Artificial Intelligence Applications');
```

```
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1001, 1, 101);  
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1002, 2, 102);  
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1003, 3, 103);  
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1004, 4, 104);  
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1005, 5, 105);  
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1006, 6, 106);  
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1007, 7, 107);  
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1008, 8, 108);  
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1009, 9, 109);  
INSERT INTO Enrollments (enrollment_id, student_id, course_id)VALUES(1010, 10, 110);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,  
assessment_score)VALUES(1, 1, 101, 85.5);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,  
assessment_score)VALUES(2, 2, 102, 78.9);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,  
assessment_score)VALUES(3, 3, 103, 92.3);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,  
assessment_score)VALUES(4, 4, 104, 70.2);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,  
assessment_score)VALUES(5, 5, 105, 88.7);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,  
assessment_score)VALUES(6, 6, 106, 81.4);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,  
assessment_score)VALUES(7, 7, 107, 95.0);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,  
assessment_score)VALUES(8, 8, 108, 73.6);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,
assessment_score)VALUES(9, 9, 109, 86.8);
```

```
INSERT INTO StudentAssessments (assessment_id, student_id, course_id,
assessment_score)VALUES(10, 10, 110, 79.1);
```

```
SELECT
```

```
    first_name || ' ' || last_name AS full_name,
```

```
    age,
```

```
    LOWER(email_address) AS email_address,
```

```
    CASE
```

```
        WHEN enrolled_flag = 1 THEN 'Enrolled'
```

```
        ELSE 'Not Enrolled'
```

```
    END AS enrollment_status
```

```
FROM
```

```
    Students;
```

```
UPDATE Students
```

```
SET email_address = LOWER(email_address);
```

```
SELECT
```

```
    sa.*,
```

```
    CASE
```

```
        WHEN sa.assessment_score >= 90 THEN 'Advanced'
```

```
        WHEN sa.assessment_score >= 70 THEN 'Intermediate'
```

```
        ELSE 'Beginner'
```

```
    END AS category
```

```
FROM
```

```
    StudentAssessments sa;
```

```
SELECT
    c.course_name,
    COUNT(e.student_id) AS num_students_enrolled,
    AVG(s.assessment_score) AS avg_assessment_score
FROM
    Courses c
JOIN
    Enrollments e ON c.course_id = e.course_id
JOIN
    StudentAssessments s ON e.student_id = s.student_id
GROUP BY
    c.course_name;
```

```
SELECT
    CONCAT(s.first_name, ' ', s.last_name) AS full_name,
    s.email_address,
    c.course_name
FROM
    Students s
JOIN
    Enrollments e ON s.student_id = e.student_id
JOIN
    Courses c ON e.course_id = c.course_id
WHERE
    s.enrolled_flag = 1
ORDER BY
    full_name;
```

```
SELECT
```

```
s.first_name || ' ' || s.last_name AS full_name,  
s.email_address,  
c.course_name  
FROM  
    Students s  
JOIN  
    Enrollments e ON s.student_id = e.student_id  
JOIN  
    Courses c ON e.course_id = c.course_id  
WHERE  
    s.enrolled_flag = 1  
ORDER BY  
    full_name;
```