CTU 2024

Software Development

SUBJECT NAME: Business Programming Semester 2

SUBJECT CODE: PRG522

Edward Nhlapo

Student Number – **20220865**

20220865@ctucareer.co.za

28th April 2024:

Questions:

CTU requires a comprehensive report that combines student information and their enrolled courses. Create a query that retrieves each student's full name, email address, and the names of the courses they are enrolled in. The report should be sorted by student name and include only students who are currently enrolled.

Implementation Instructions:

Database Name: CTU_DB

Tables to Create:

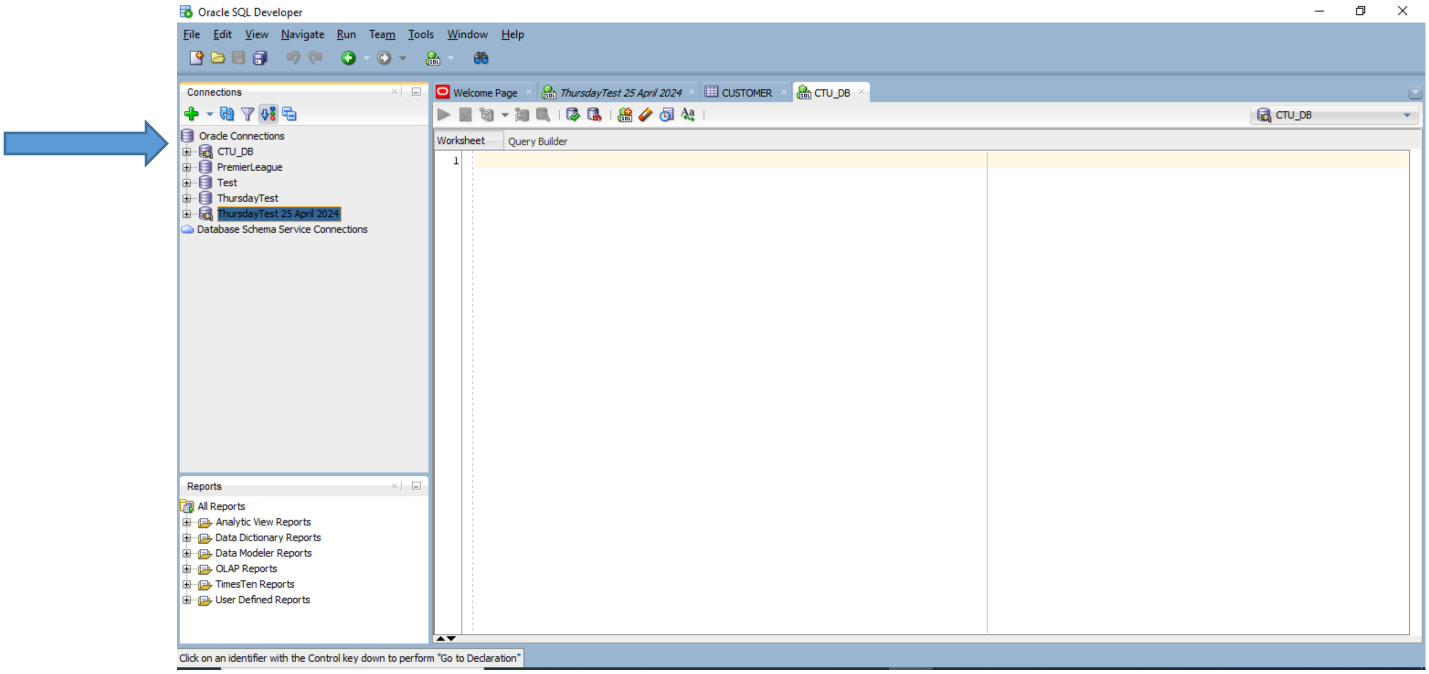
A. Students:

•Columns: student_id (Primary Key), first_name, last_name, age, email_address, enrolled_flag

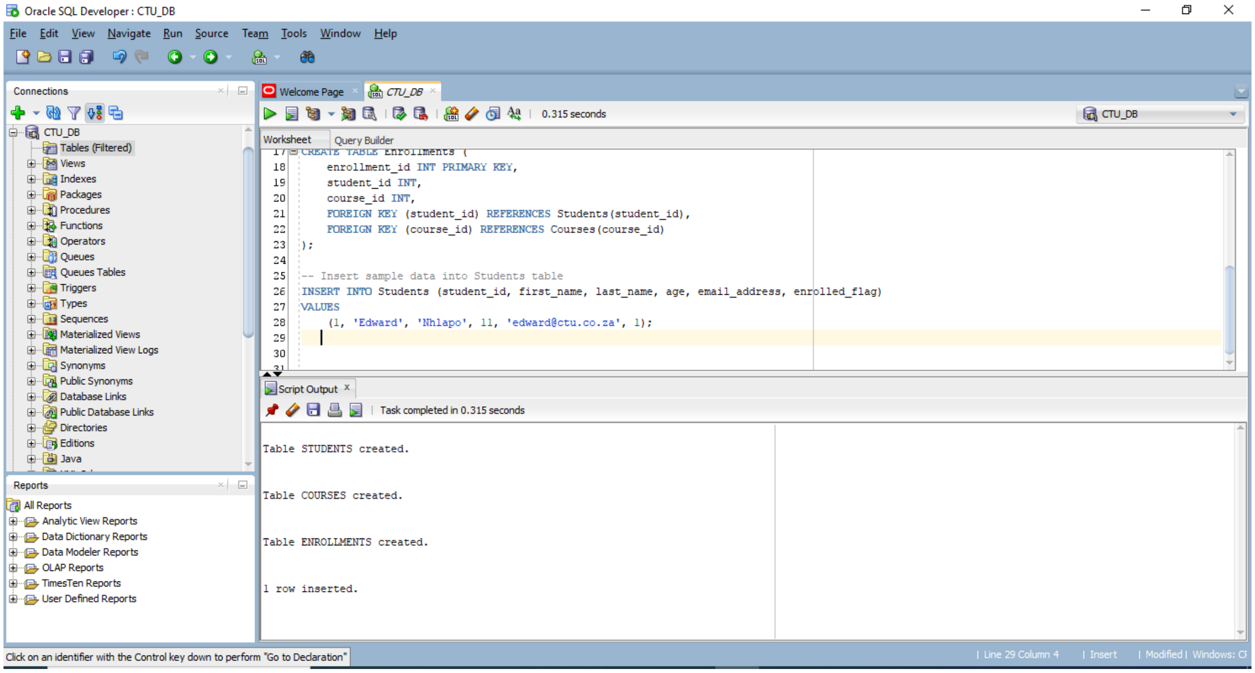
•Example Dummy Data: (1, 'John', 'Doe', 22, 'john@example.com', 1)

•Load 10 Ficticious records

Database connection created



Click on an identifier with the Control key down to perform "Go to Declaration"

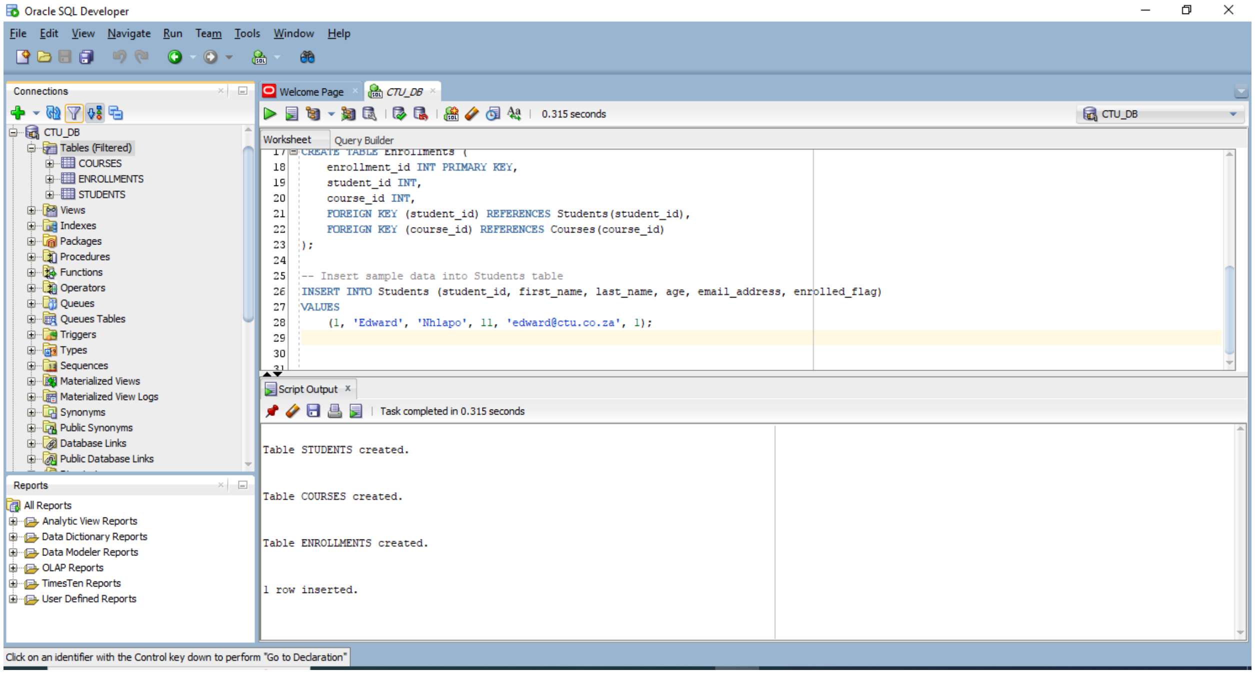


```
CREATE TABLE Enrollments (
    enrollment_id INT PRIMARY KEY,
    student_id INT,
    course_id INT,
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);

-- Insert sample data into Students table
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (1, 'Edward', 'Nhlapo', 11, 'edward@ctu.co.za', 1);
```

Task completed in 0.315 seconds

```
Table STUDENTS created.


Table COURSES created.


Table ENROLLMENTS created.


1 row inserted.
```

Click on an identifier with the Control key down to perform "Go to Declaration"



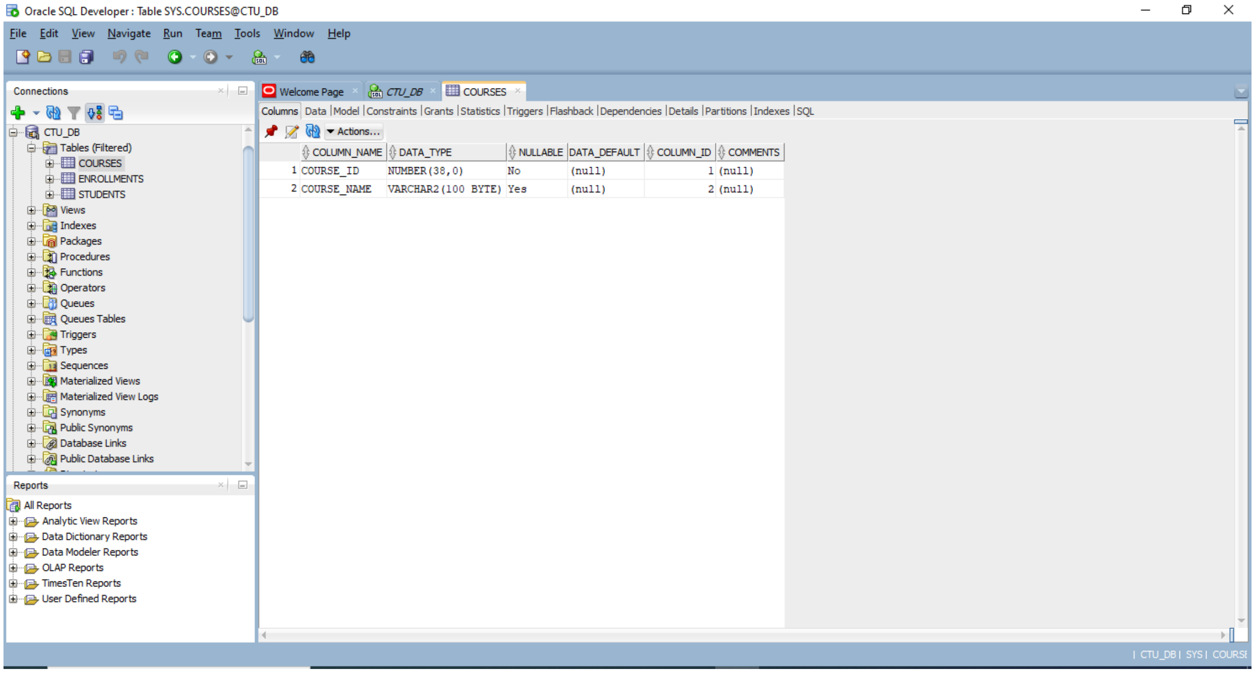
```
CREATE TABLE Enrollments (
    enrollment_id INT PRIMARY KEY,
    student_id INT,
    course_id INT,
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)
);

-- Insert sample data into Students table
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (1, 'Edward', 'Nhlapo', 11, 'edward@ctu.co.za', 1);
```

Task completed in 0.315 seconds

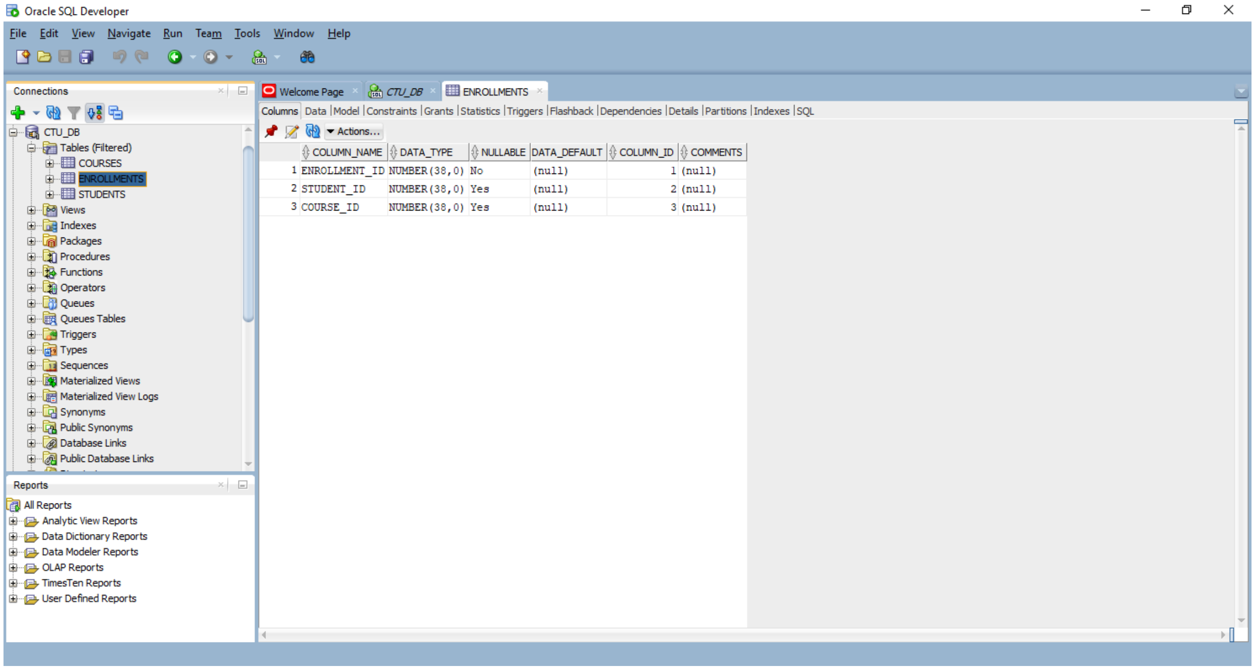
```
Table STUDENTS created.


Table COURSES created.


Table ENROLLMENTS created.


1 row inserted.
```

Click on an identifier with the Control key down to perform "Go to Declaration"

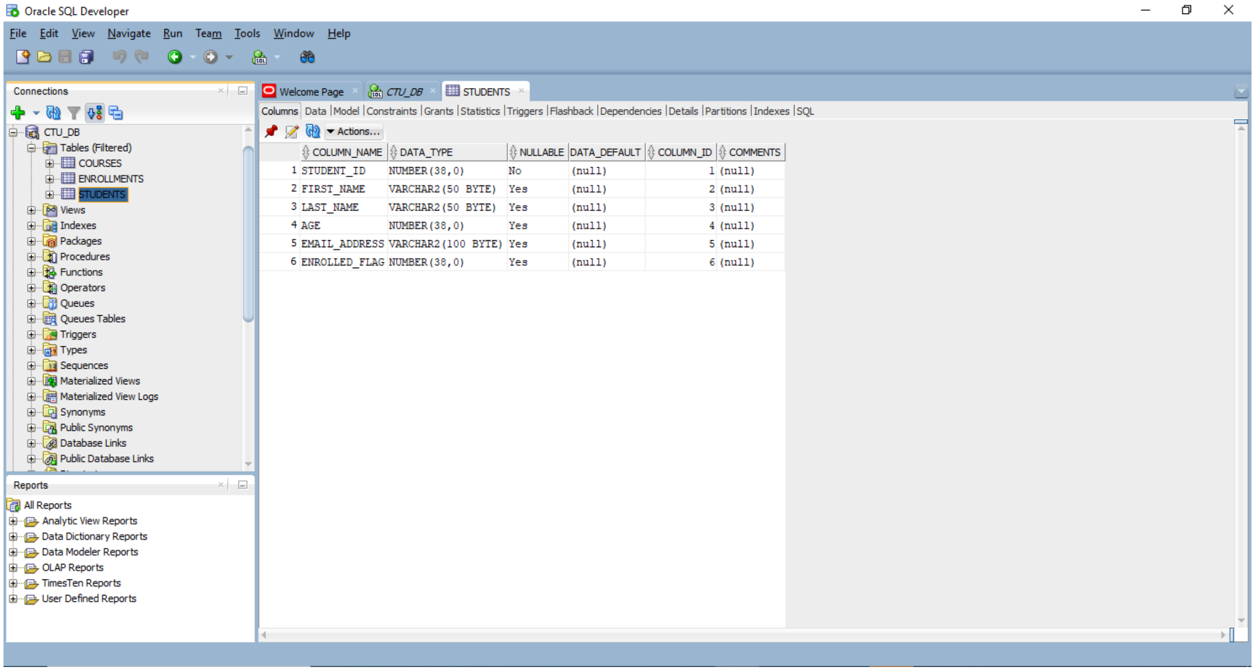
```
-- Create Students table
CREATE TABLE Students (
    student_id INT PRIMARY KEY,
    first_name VARCHAR2(50),
    last_name VARCHAR2(50),
    age INT,
    email_address VARCHAR2(100),
    enrolled_flag INT
);
-- Create Courses table
CREATE TABLE Courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR2(100)
);

-- Create Enrollments table
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)
);

-- Insert sample data into Students table
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (1, 'Edward', 'Nhlapo', 11, 'edward@ctu.co.za', 1);
```



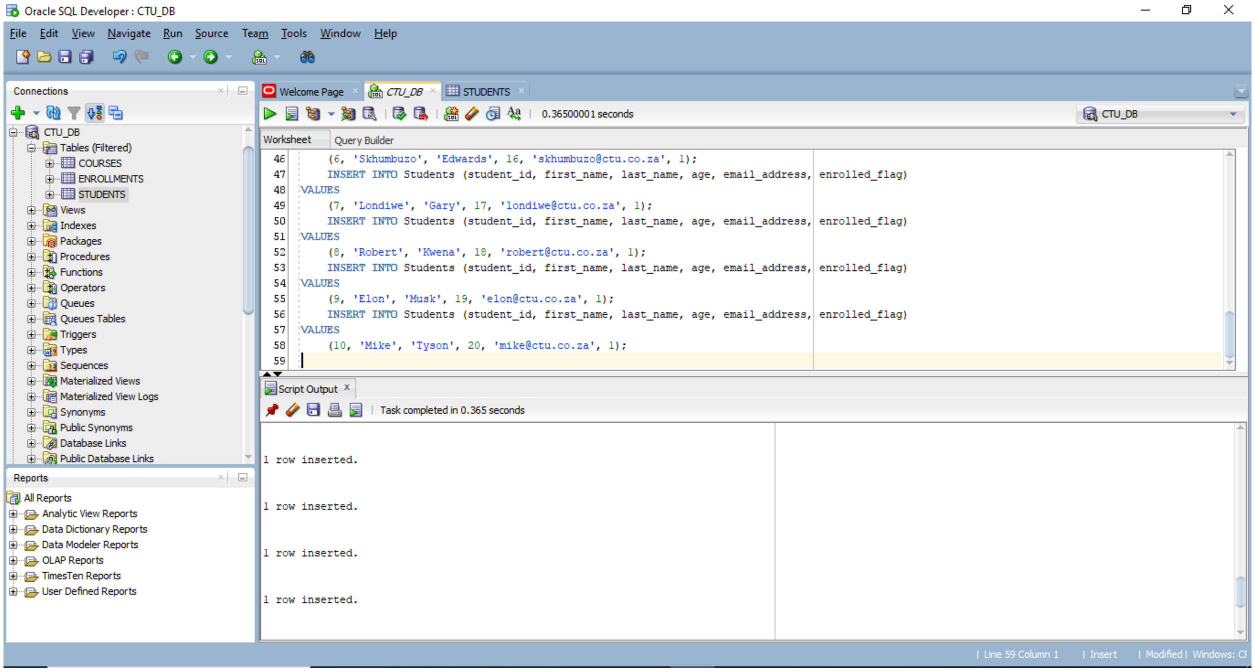
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COURSE_ID	NUMBER(38,0)	No	(null)	1	(null)
2	COURSE_NAME	VARCHAR2(100 BYTE)	Yes	(null)	2	(null)



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ENROLLMENT_ID	NUMBER(38,0)	No	(null)	1	(null)
2	STUDENT_ID	NUMBER(38,0)	Yes	(null)	2	(null)
3	COURSE_ID	NUMBER(38,0)	Yes	(null)	3	(null)



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	STUDENT_ID	NUMBER(38,0)	No	(null)	1	(null)
2	FIRST_NAME	VARCHAR2(50 BYTE)	Yes	(null)	2	(null)
3	LAST_NAME	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)
4	AGE	NUMBER(38,0)	Yes	(null)	4	(null)
5	EMAIL_ADDRESS	VARCHAR2(100 BYTE)	Yes	(null)	5	(null)
6	ENROLLED_FLAG	NUMBER(38,0)	Yes	(null)	6	(null)



```
    (6, 'Skhumbuzo', 'Edwards', 16, 'skhumbuzo@ctu.co.za', 1);
    INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (7, 'Londiwe', 'Gary', 17, 'londiwe@ctu.co.za', 1);
    INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (8, 'Robert', 'Kwena', 18, 'robert@ctu.co.za', 1);
    INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (9, 'Elon', 'Musk', 19, 'elon@ctu.co.za', 1);
    INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (10, 'Mike', 'Tyson', 20, 'mike@ctu.co.za', 1);
```

Task completed in 0.365 seconds

```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

CODE

-- Insert sample data into Students table

```
INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (1, 'Edward', 'Nhlapo', 11, 'edward@ctu.co.za', 1);

INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (2, 'Faith', 'Jackson', 12, 'faith@ctu.co.za', 1);

 INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (3, 'Gugu', 'Dlamini', 13, 'gugu@ctu.co.za', 1);

INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (4, 'Sibusiso', 'Timm', 14, 'sibusiso@ctu.co.za', 1);
    INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (5, 'Thabang', 'Junior', 15, 'thabang@ctu.co.za', 1);
    INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (6, 'Skhumbuzo', 'Edwards', 16, 'skhumbuzo@ctu.co.za', 1);
    INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (7, 'Londiwe', 'Gary', 17, 'londiwe@ctu.co.za', 1);
    INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
    (8, 'Robert', 'Kwena', 18, 'robert@ctu.co.za', 1);
    INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)
VALUES
```

(9, 'Elon', 'Musk', 19, 'elon@ctu.co.za', 1);

INSERT INTO Students (student_id, first_name, last_name, age, email_address, enrolled_flag)

VALUES

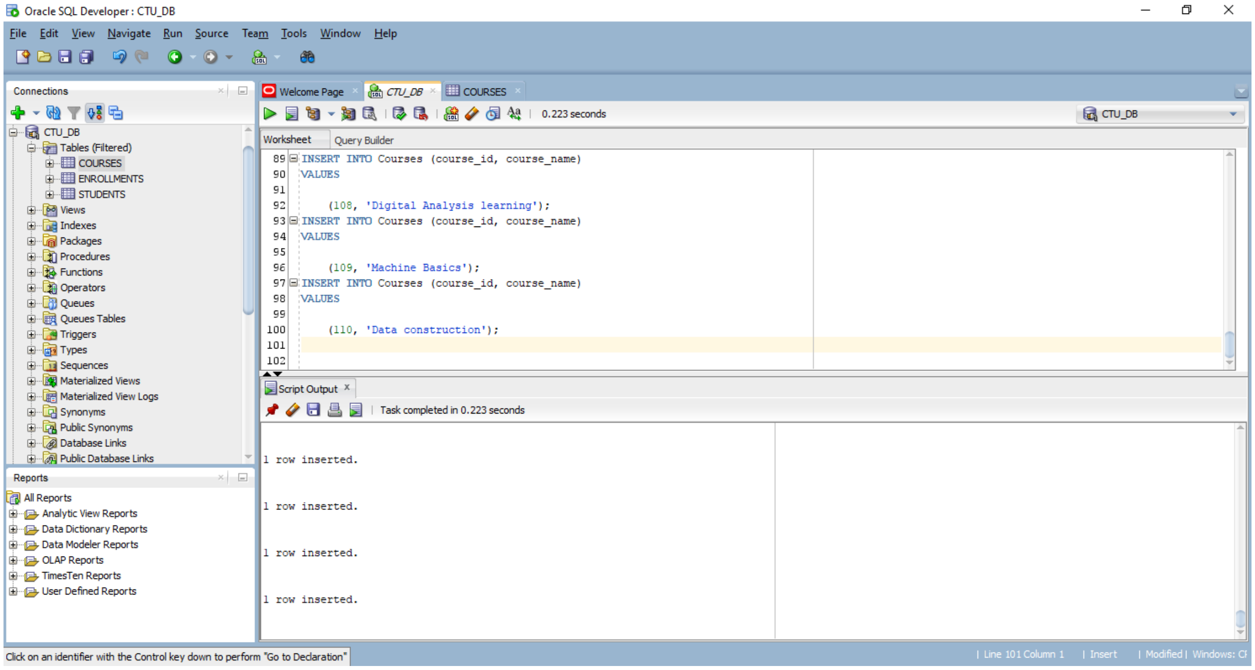
(10, 'Mike', 'Tyson', 20, 'mike@ctu.co.za', 1);

B Courses:

•Columns: course_id (Primary Key), course_name

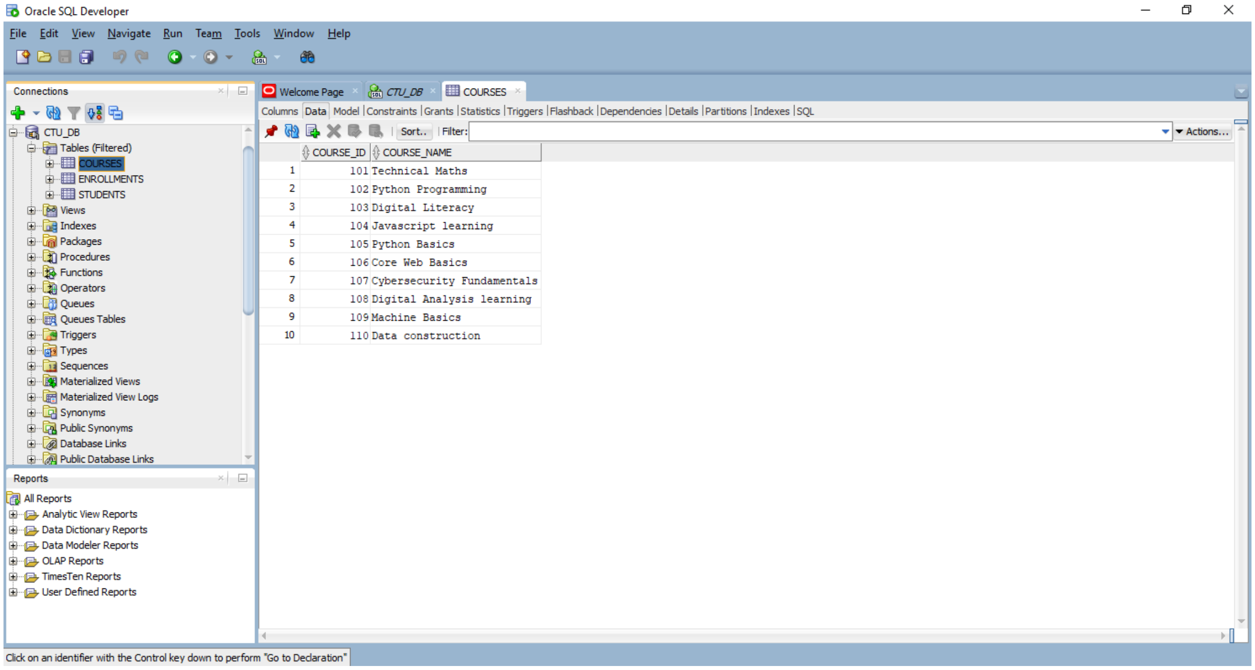
•Example Dummy Data: (101, 'Database Fundamentals'), (102, 'SQL Mastery'), (103, 'Data AnalysisTechniques')

•Load 10 Ficticious records



```
INSERT INTO Courses (course_id, course_name)
VALUES

    (108, 'Digital Analysis learning');
INSERT INTO Courses (course_id, course_name)
VALUES

    (109, 'Machine Basics');
INSERT INTO Courses (course_id, course_name)
VALUES

    (110, 'Data construction');
```

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```



	COURSE_ID	COURSE_NAME
1	101	Technical Maths
2	102	Python Programming
3	103	Digital Literacy
4	104	Javascript learning
5	105	Python Basics
6	106	Core Web Basics
7	107	Cybersecurity Fundamentals
8	108	Digital Analysis learning
9	109	Machine Basics
10	110	Data construction

CODE

```
-- Insert sample data into Courses table

INSERT INTO Courses (course_id, course_name)

VALUES

   (101, 'Technical Maths');

INSERT INTO Courses (course_id, course_name)

VALUES



   (102, 'Python Programming');

INSERT INTO Courses (course_id, course_name)

VALUES



   (103, 'Digital Literacy');

INSERT INTO Courses (course_id, course_name)
```

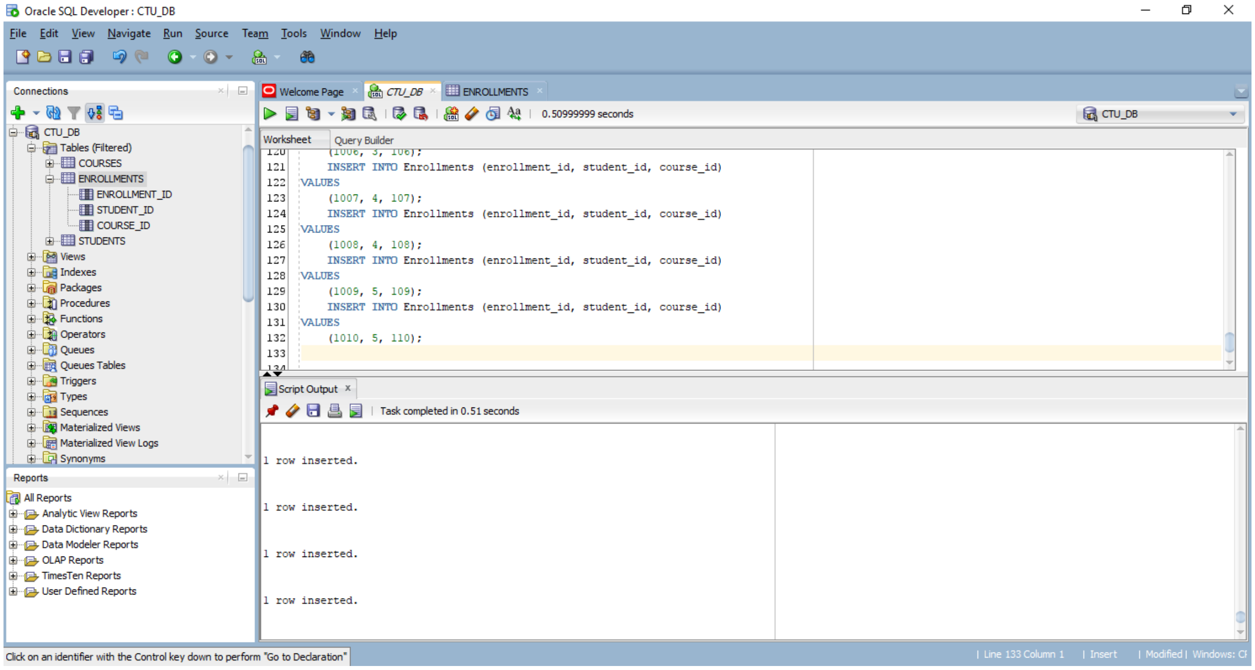
```
VALUES

    (104, 'Javascript learning');
INSERT INTO Courses (course_id, course_name)
VALUES

    (105, 'Python Basics');
INSERT INTO Courses (course_id, course_name)
VALUES

    (106, 'Core Web Basics');
INSERT INTO Courses (course_id, course_name)
VALUES

    (107, 'Cybersecurity Fundamentals');
INSERT INTO Courses (course_id, course_name)
VALUES

    (108, 'Digital Analysis learning');
INSERT INTO Courses (course_id, course_name)
VALUES

    (109, 'Machine Basics');
INSERT INTO Courses (course_id, course_name)
VALUES

    (110, 'Data construction');
```

C Enrollments:

•Columns: enrollment_id (Primary Key), student_id (Foreign Key referencing Students), course_id(Foreign Key referencing Courses)

•Example Dummy Data: (1001, 1, 101), (1002, 1, 102)

•Load 10 Ficticious records



0.50999999 seconds

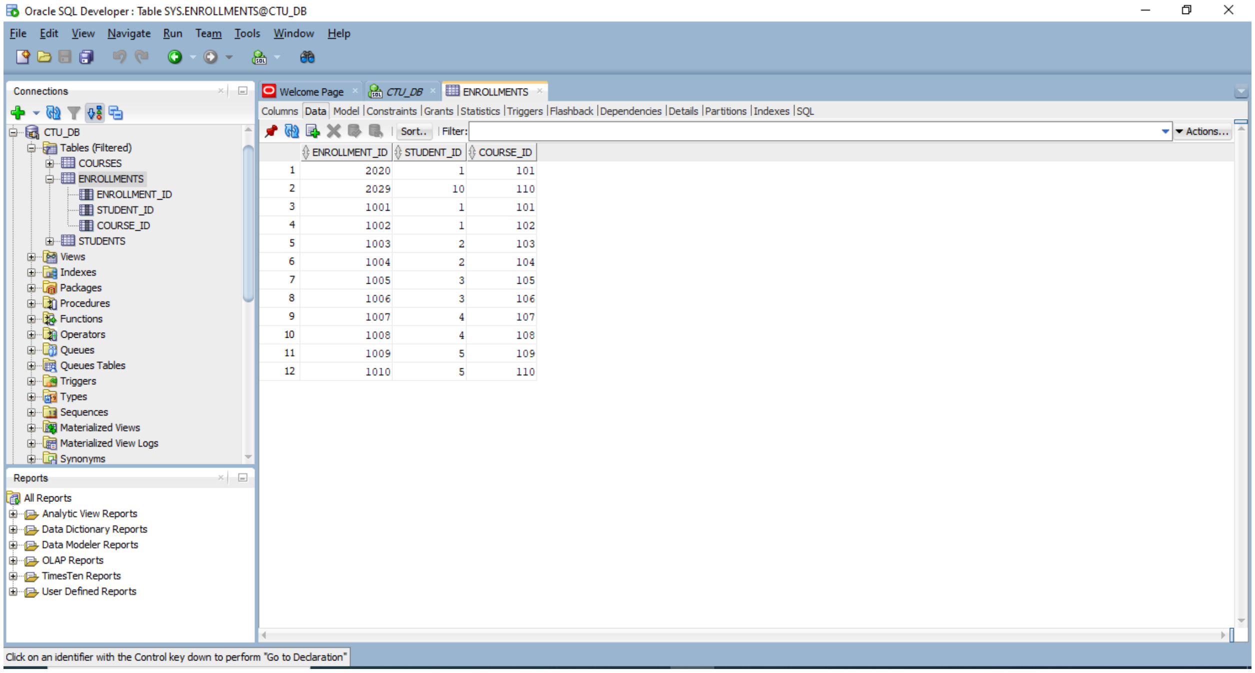
```
    (1006, 3, 106);
    INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
    (1007, 4, 107);
    INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
    (1008, 4, 108);
    INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
    (1009, 5, 109);
    INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
    (1010, 5, 110);
```

Task completed in 0.51 seconds

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

```
-- Insert sample data into 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, 1, 102);

   INSERT INTO Enrollments (enrollment_id, student_id, course_id)

VALUES
```

```
  (1003, 2, 103);
  INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
  (1004, 2, 104);
  INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
  (1005, 3, 105);
  INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
  (1006, 3, 106);
  INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
  (1007, 4, 107);
  INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
  (1008, 4, 108);
  INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
  (1009, 5, 109);
  INSERT INTO Enrollments (enrollment_id, student_id, course_id)
VALUES
  (1010, 5, 110);
```



	ENROLLMENT_ID	STUDENT_ID	COURSE_ID
1	2020	1	101
2	2029	10	110
3	1001	1	101
4	1002	1	102
5	1003	2	103
6	1004	2	104
7	1005	3	105
8	1006	3	106
9	1007	4	107
10	1008	4	108
11	1009	5	109
12	1010	5	110

Click on an identifier with the Control key down to perform "Go to Declaration"

Scoring and Evaluation:

Students will be evaluated based on their ability to:

•Develop and execute SQL queries that address the requirements for each learning unit.

•Create appropriate functions, expressions, and aggregations to customize and manipulate data.

•Design a normalized database schema that represents the required entities and relationships.

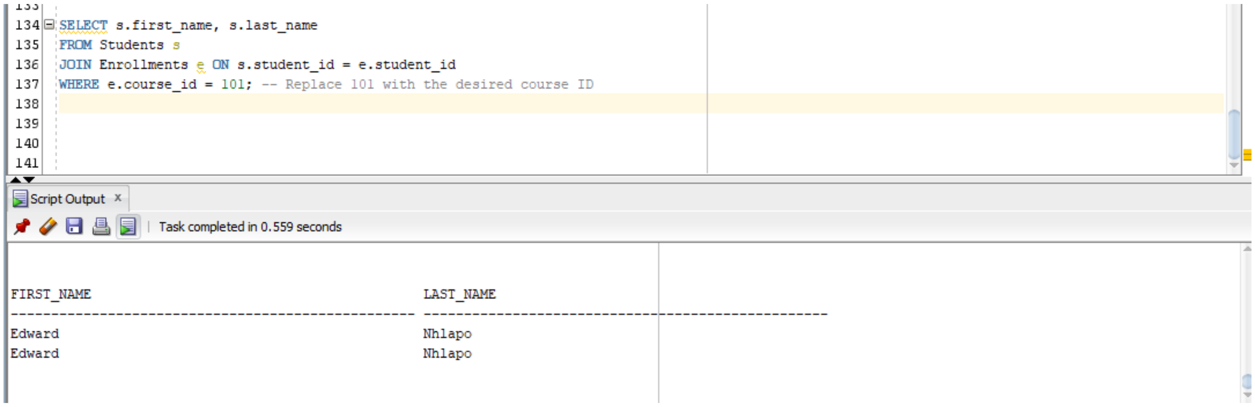
•Successfully load the dummy data into the created tables.

SELECT s.first_name, s.last_name

FROM Students s

JOIN Enrollments e ON s.student_id = e.student_id

WHERE e.course_id = 101; -- Replace 101 with the desired course ID



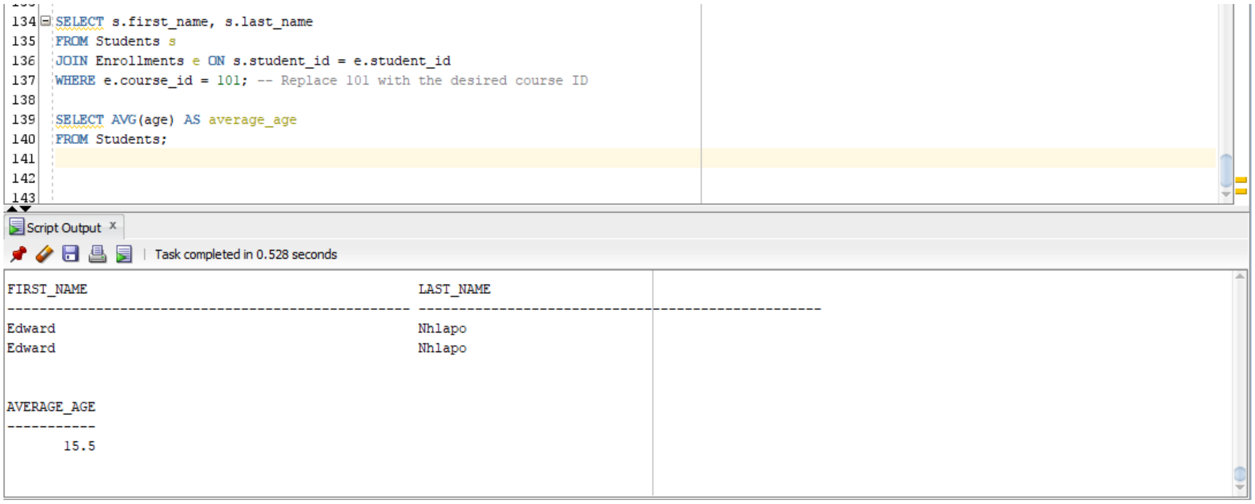
```
SELECT s.first_name, s.last_name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.course_id = 101; -- Replace 101 with the desired course ID
```

Task completed in 0.559 seconds

```
FIRST_NAME                                         LAST_NAME
-------------------------------------------------- --------------------------------------------------
Edward                                             Nhlapo
Edward                                             Nhlapo
```

SELECT AVG(age) AS average_age

FROM Students;



```
SELECT s.first_name, s.last_name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.course_id = 101; -- Replace 101 with the desired course ID

SELECT AVG(age) AS average_age
FROM Students;
```

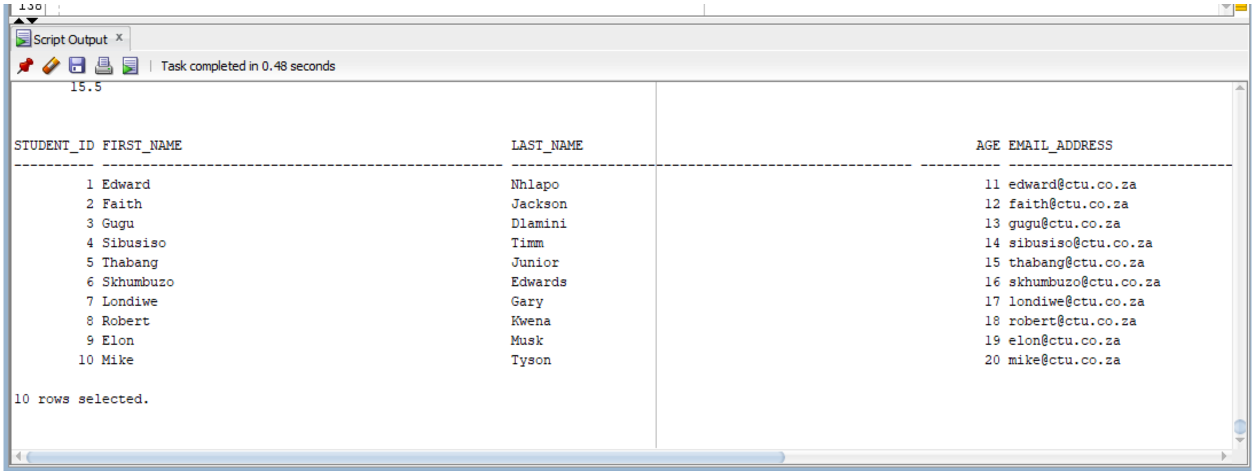
Task completed in 0.528 seconds

```
FIRST_NAME                                         LAST_NAME
-------------------------------------------------- --------------------------------------------------
Edward                                             Nhlapo
Edward                                             Nhlapo


AVERAGE_AGE
-----------
       15.5
```

SELECT *

FROM Students

WHERE enrolled_flag = 1;



Task completed in 0.48 seconds

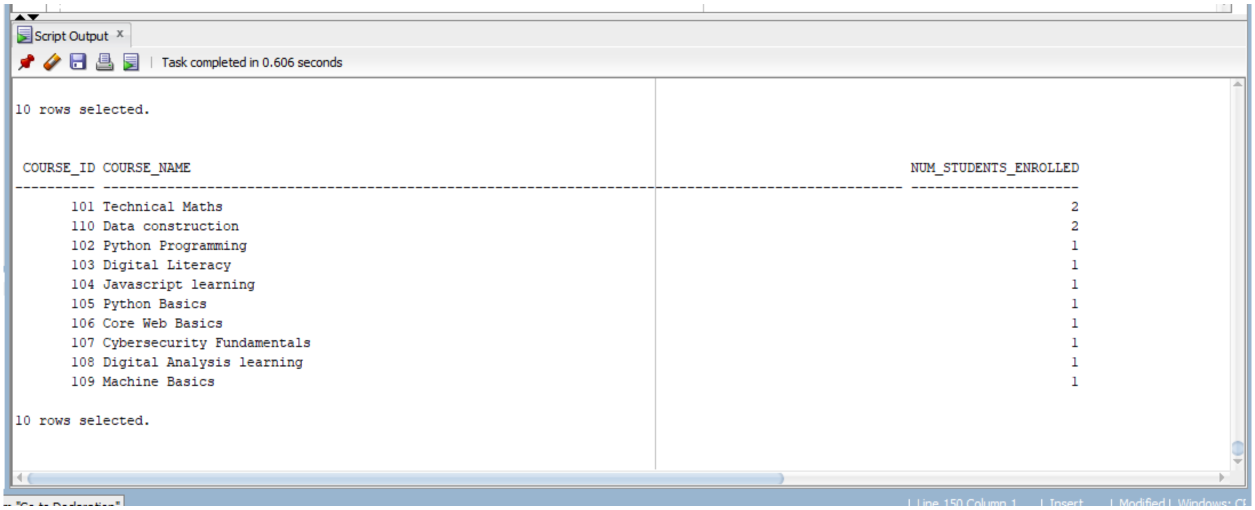
```
       15.5


STUDENT_ID FIRST_NAME                                         LAST_NAME                                                 AGE EMAIL_ADDRESS
---------- -------------------------------------------------- -------------------------------------------------- ---------- ------------------------------
         1 Edward                                             Nhlapo                                                     11 edward@ctu.co.za
         2 Faith                                              Jackson                                                    12 faith@ctu.co.za
         3 Gugu                                               Dlamini                                                    13 gugu@ctu.co.za
         4 Sibusiso                                           Timm                                                       14 sibusiso@ctu.co.za
         5 Thabang                                            Junior                                                     15 thabang@ctu.co.za
         6 Skhumbuzo                                          Edwards                                                    16 skhumbuzo@ctu.co.za
         7 Londiwe                                            Gary                                                       17 londiwe@ctu.co.za
         8 Robert                                             Kwena                                                      18 robert@ctu.co.za
         9 Elon                                               Musk                                                       19 elon@ctu.co.za
        10 Mike                                               Tyson                                                      20 mike@ctu.co.za

10 rows selected.
```

SELECT c.course_id, c.course_name, COUNT(e.student_id) AS num_students_enrolled

FROM Courses c

LEFT JOIN Enrollments e ON c.course_id = e.course_id

GROUP BY c.course_id, c.course_name;



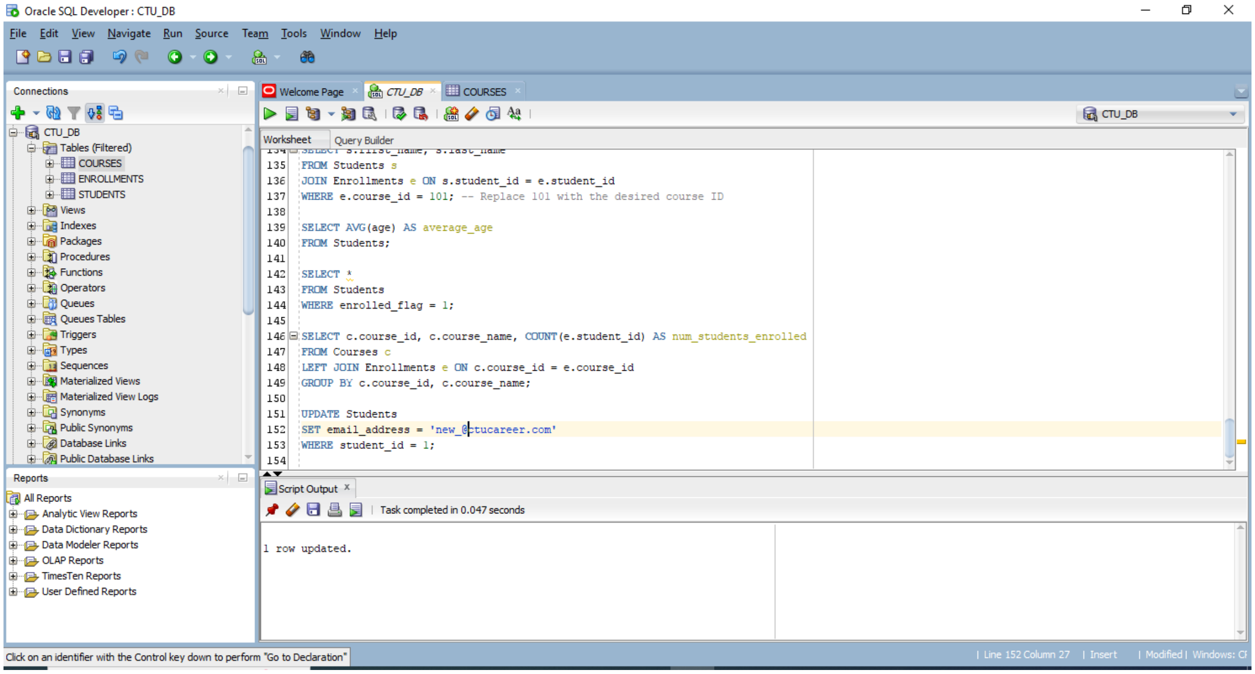
Task completed in 0.606 seconds

```
10 rows selected.


 COURSE_ID COURSE_NAME                                                                                          NUM_STUDENTS_ENROLLED
---------- ---------------------------------------------------------------------------------------------------- ---------------------
       101 Technical Maths                                                                                                          2
       110 Data construction                                                                                                        2
       102 Python Programming                                                                                                       1
       103 Digital Literacy                                                                                                         1
       104 Javascript learning                                                                                                      1
       105 Python Basics                                                                                                            1
       106 Core Web Basics                                                                                                          1
       107 Cybersecurity Fundamentals                                                                                               1
       108 Digital Analysis learning                                                                                                1
       109 Machine Basics                                                                                                           1

10 rows selected.
```

UPDATE Students

SET email_address = **'new_@ctucareer.com'**

WHERE student_id = 1;

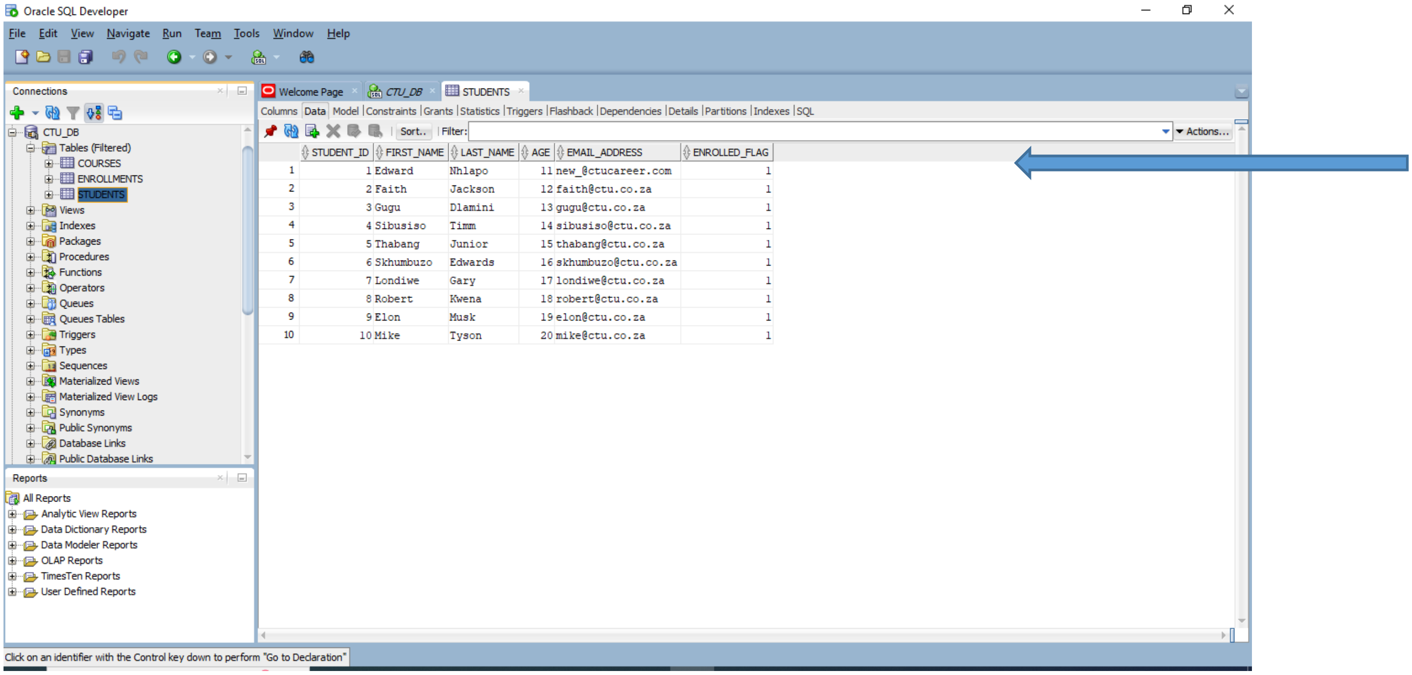


```
SELECT s.first_name, s.last_name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.course_id = 101; -- Replace 101 with the desired course ID

SELECT AVG(age) AS average_age
FROM Students;

SELECT *
FROM Students
WHERE enrolled_flag = 1;

SELECT c.course_id, c.course_name, COUNT(e.student_id) AS num_students_enrolled
FROM Courses c
LEFT JOIN Enrollments e ON c.course_id = e.course_id
GROUP BY c.course_id, c.course_name;

UPDATE Students
SET email_address = 'new_@ctucareer.com'
WHERE student_id = 1;
```

Task completed in 0.047 seconds

```
1 row updated.
```

Email updated

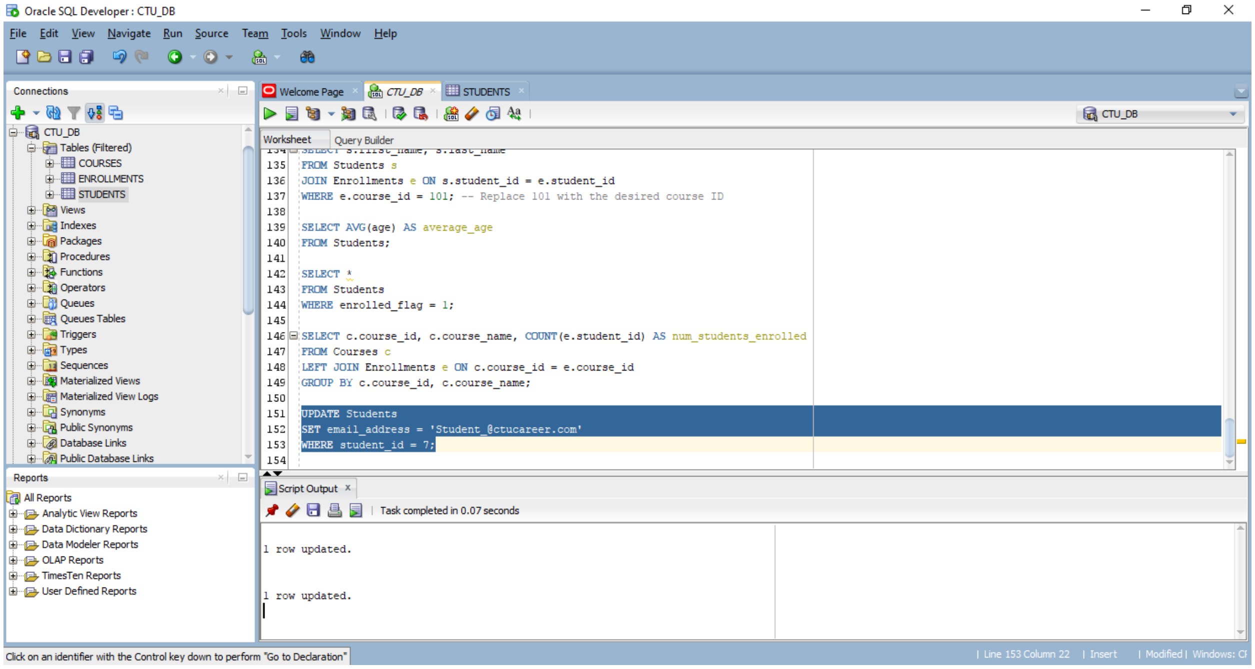


	STUDENT_ID	FIRST_NAME	LAST_NAME	AGE	EMAIL_ADDRESS	ENROLLED_FLAG
1	1	Edward	Nhlapo	11	new_@ctucareer.com	1
2	2	Faith	Jackson	12	faith@ctu.co.za	1
3	3	Gugu	Dlamini	13	gugu@ctu.co.za	1
4	4	Sibusiso	Timm	14	sibusiso@ctu.co.za	1
5	5	Thabang	Junior	15	thabang@ctu.co.za	1
6	6	Skhumbuzo	Edwards	16	skhumbuzo@ctu.co.za	1
7	7	Londiwe	Gary	17	londiwe@ctu.co.za	1
8	8	Robert	Kwena	18	robert@ctu.co.za	1
9	9	Elon	Musk	19	elon@ctu.co.za	1
10	10	Mike	Tyson	20	mike@ctu.co.za	1

UPDATE Students

SET email_address = **'Student_@ctucareer.com'**

WHERE student_id = 7;

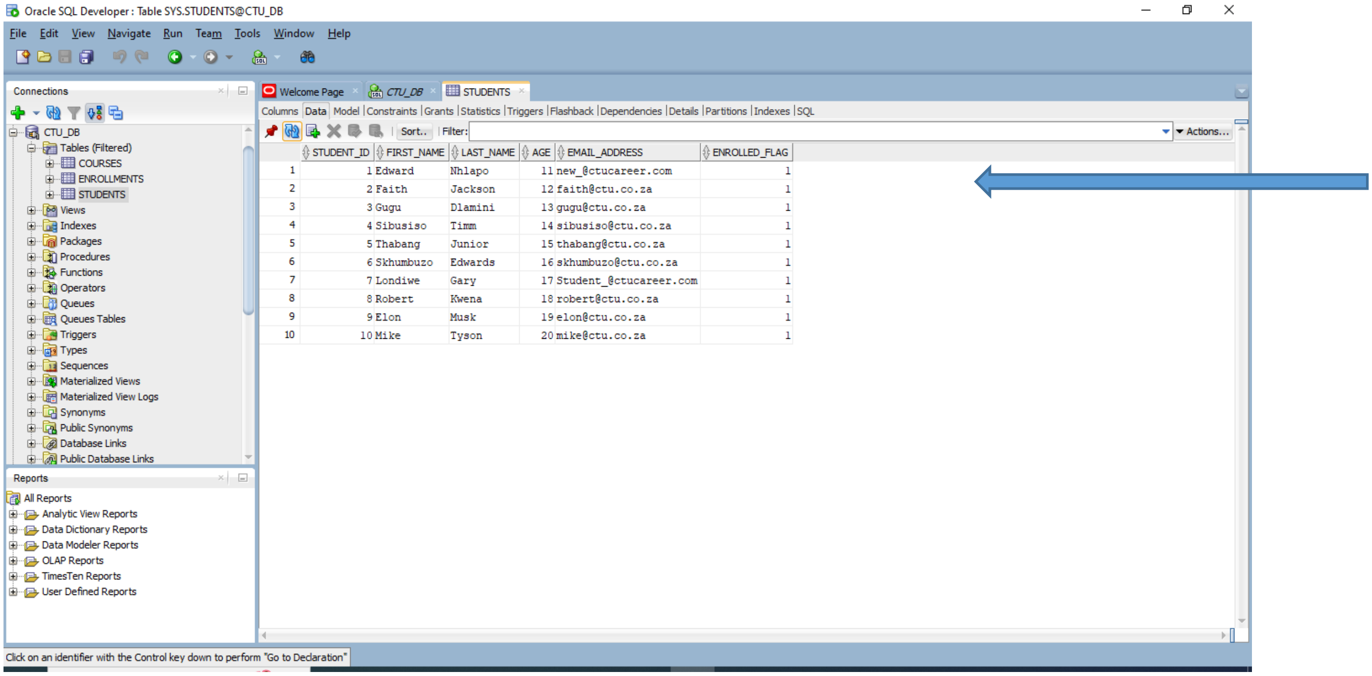


```
SELECT s.first_name, s.last_name
FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.course_id = 101; -- Replace 101 with the desired course ID

SELECT AVG(age) AS average_age
FROM Students;

SELECT *
FROM Students
WHERE enrolled_flag = 1;

SELECT c.course_id, c.course_name, COUNT(e.student_id) AS num_students_enrolled
FROM Courses c
LEFT JOIN Enrollments e ON c.course_id = e.course_id
GROUP BY c.course_id, c.course_name;

UPDATE Students
SET email_address = 'Student_@ctucareer.com'
WHERE student_id = 7;
```

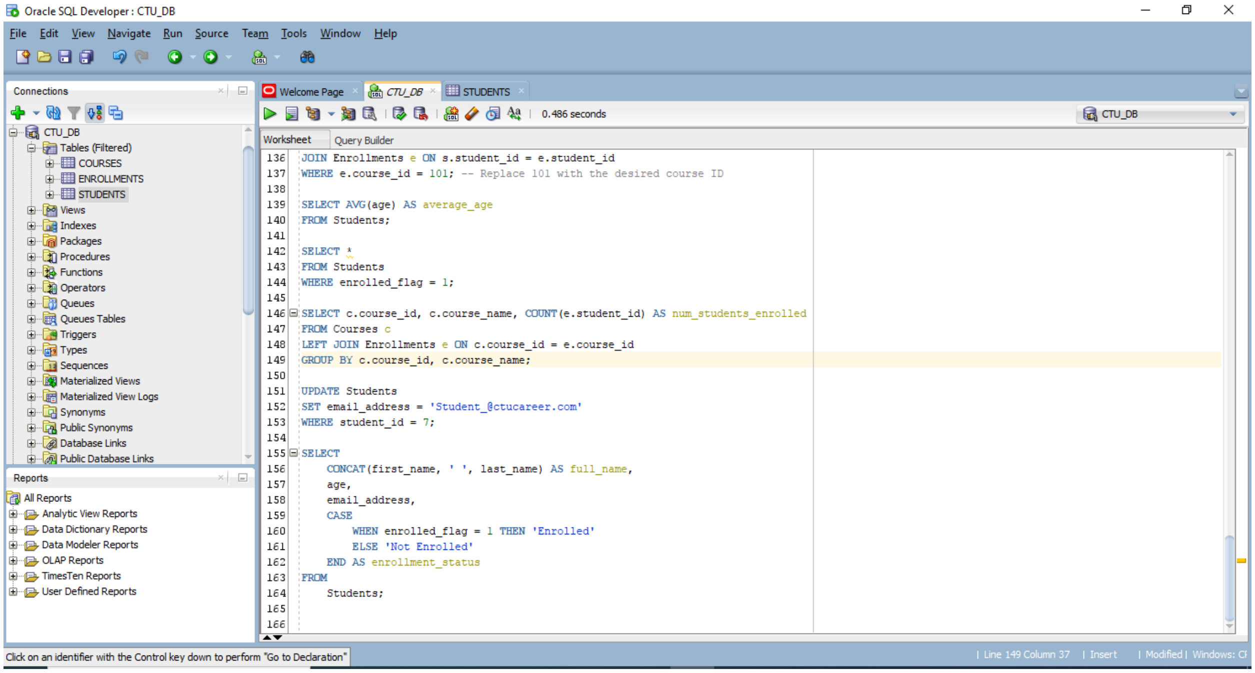
Task completed in 0.07 seconds

```
1 row updated.


1 row updated.
```

Email Updated again



	STUDENT_ID	FIRST_NAME	LAST_NAME	AGE	EMAIL_ADDRESS	ENROLLED_FLAG
1	1	Edward	Nhlapo	11	new_@ctucareer.com	1
2	2	Faith	Jackson	12	faith@ctu.co.za	1
3	3	Gugu	Dlamini	13	gugu@ctu.co.za	1
4	4	Sibusiso	Timm	14	sibusiso@ctu.co.za	1
5	5	Thabang	Junior	15	thabang@ctu.co.za	1
6	6	Skhumbuzo	Edwards	16	skhumbuzo@ctu.co.za	1
7	7	Londiwe	Gary	17	Student_@ctucareer.com	1
8	8	Robert	Kwena	18	robert@ctu.co.za	1
9	9	Elon	Musk	19	elon@ctu.co.za	1
10	10	Mike	Tyson	20	mike@ctu.co.za	1



```
JOIN Enrollments e ON s.student_id = e.student_id
WHERE e.course_id = 101; -- Replace 101 with the desired course ID

SELECT AVG(age) AS average_age
FROM Students;

SELECT *
FROM Students
WHERE enrolled_flag = 1;

SELECT c.course_id, c.course_name, COUNT(e.student_id) AS num_students_enrolled
FROM Courses c
LEFT JOIN Enrollments e ON c.course_id = e.course_id
GROUP BY c.course_id, c.course_name;

UPDATE Students
SET email_address = 'Student_@ctucareer.com'
WHERE student_id = 7;

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

Click on an identifier with the Control key down to perform "Go to Declaration"