

Question 01)

Steps for the populate the table:

1. Create the tables of courses and modules.
2. Write the PL/SQL procedure to insert records into both tables with validation.
3. Populate the tables using data from the appendix.

1. Create the tables of courses and modules

-- Create the course and modules tables

```
CREATE TABLE course (  
  course_id VARCHAR2(13) NOT NULL PRIMARY KEY,  
  title VARCHAR2(200),  
  description VARCHAR2(2000),  
  course_leader VARCHAR2(200),  
  date_modified DATE,  
  number_of_credits NUMBER  
);
```

```
CREATE TABLE modules (  
  module_id VARCHAR2(100) NOT NULL PRIMARY KEY,  
  course_id VARCHAR2(13) REFERENCES course (course_id)  
);
```

SQL Worksheet

```
2 v CREATE TABLE course (  
3   course_id VARCHAR2(13) NOT NULL PRIMARY KEY,  
4   title VARCHAR2(200),  
5   description VARCHAR2(2000),  
6   course_leader VARCHAR2(200),  
7   date_modified DATE,  
8   number_of_credits NUMBER  
9 );  
10  
11 -- Create the modules table  
12 v CREATE TABLE modules (  
13   module_id VARCHAR2(100) NOT NULL PRIMARY KEY,  
14   course_id VARCHAR2(13) REFERENCES course (course_id)  
15 );  
16
```

Table created.

Table created.

2. Write the PL/SQL procedure to insert records into both tables with validation.

```
CREATE OR REPLACE PROCEDURE add_course_and_modules (  
    p_course_id    IN VARCHAR2,  
    p_title        IN VARCHAR2,  
    p_description   IN VARCHAR2,  
    p_course_leader IN VARCHAR2,  
    p_date_modified IN DATE,  
    p_number_of_credits IN NUMBER,  
    p_module_ids    IN SYS.ODCIVARCHAR2LIST -- Using Oracle's collection type  
) IS  
    -- Exception to handle invalid input  
    invalid_input EXCEPTION;  
BEGIN  
    -- Validate input for course_id, title, course_leader, and number_of_credits  
    IF p_course_id IS NULL OR p_title IS NULL OR p_course_leader IS NULL OR p_number_of_credits IS  
    NULL THEN  
        RAISE invalid_input;  
    END IF;  
  
    IF p_number_of_credits <= 0 THEN  
        RAISE invalid_input;  
    END IF;  
  
    -- Insert data into the course table  
    INSERT INTO course (  
        course_id, title, description, course_leader, date_modified, number_of_credits  
    ) VALUES (  
        p_course_id, p_title, p_description, p_course_leader, p_date_modified, p_number_of_credits  
    );  
  
    -- Insert corresponding modules into the modules table  
    FOR i IN 1..p_module_ids.COUNT LOOP  
        INSERT INTO modules (module_id, course_id)  
        VALUES (p_module_ids(i), p_course_id);  
    END LOOP;  
  
    -- Commit the transaction  
    COMMIT;  
  
EXCEPTION  
    WHEN invalid_input THEN  
        -- Rollback in case of invalid input  
        ROLLBACK;
```

```

    DBMS_OUTPUT.PUT_LINE('Error: Invalid input values.');
```



WHEN OTHERS THEN

```

    -- Handle other exceptions
    ROLLBACK;
    DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

Explanation:

- **Input Validation:** The procedure verifies that if `course_id`, `title`, `course_leader`, and `number_of_credits` are provided and if `number_of_credits` is a positive number.
- **Inserting Course:** This procedure inserts a new course record into the course table.
- **Inserting Modules:** A collection (Oracle's `SYS.ODCIVARCHAR2LIST`) is used to pass multiple `module_id` values, and each module is inserted into the modules table.
- **Exception Handling:** The transaction is rolled back if any errors are found, including invalid inputs.



Live SQL

SQL Worksheet

```

43      INSERT INTO modules (module_id, course_id)
44      VALUES (p_module_ids(i), p_course_id);
45  END LOOP;
46
47  COMMIT;
48
49  EXCEPTION
50  WHEN invalid_input THEN
51      ROLLBACK;
52      DBMS_OUTPUT.PUT_LINE('Error: Invalid input values.');
```

```


53  WHEN OTHERS THEN
54      ROLLBACK;
55      DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
56  END;
57  /
```

Table created.

Table created.

Procedure created.

ADD_COURSE_AND_MODULES



Procedure
 Status: Valid
 Created 19 minutes ago

COURSE




Table
 Status: Valid
 Created 18 minutes ago

MODULES




Table
 Status: Valid
 Created 18 minutes ago

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

Schema \

COURSE

Show AllColumnsConstraintsRelated ConstraintsTriggersIndexesCode

Syntax HelpActionsView All Objects

Columns

Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	COURSE_ID	VARCHAR2	13			No	Byte	
2	TITLE	VARCHAR2	200			Yes	Byte	
3	DESCRIPTION	VARCHAR2	2000			Yes	Byte	
4	COURSE_LEADER	VARCHAR2	200			Yes	Byte	
5	DATE_MODIFIED	DATE	7			Yes		
6	NUMBER_OF_CREDITS	NUMBER	22			Yes		

Course columns

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

Schema \

MODULES

Show All

Columns

Constraints

Related Constraints

Triggers

Indexes

Code

Syntax Help

Actions

View All Objects

Columns

Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	MODULE_ID	VARCHAR2	100			No	Byte	
2	COURSE_ID	VARCHAR2	13			Yes	Byte	

Modules columns

3. Populate the Tables using data from the appendix

1. Inserting Course C001 (Computing)

DECLARE

v_module_ids SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('IS1S464', 'IS3S662');

BEGIN

add_course_and_modules(

p_course_id => 'C001',

p_title => 'Computing',

p_description => 'Focuses on developing skills that employers demand for roles in the IT industry.',

p_course_leader => 'Richard Jones',

p_date_modified => TO_DATE('20-APR-2024', 'DD-MON-YYYY'),

p_number_of_credits => 120,

p_module_ids => v_module_ids

);

END;

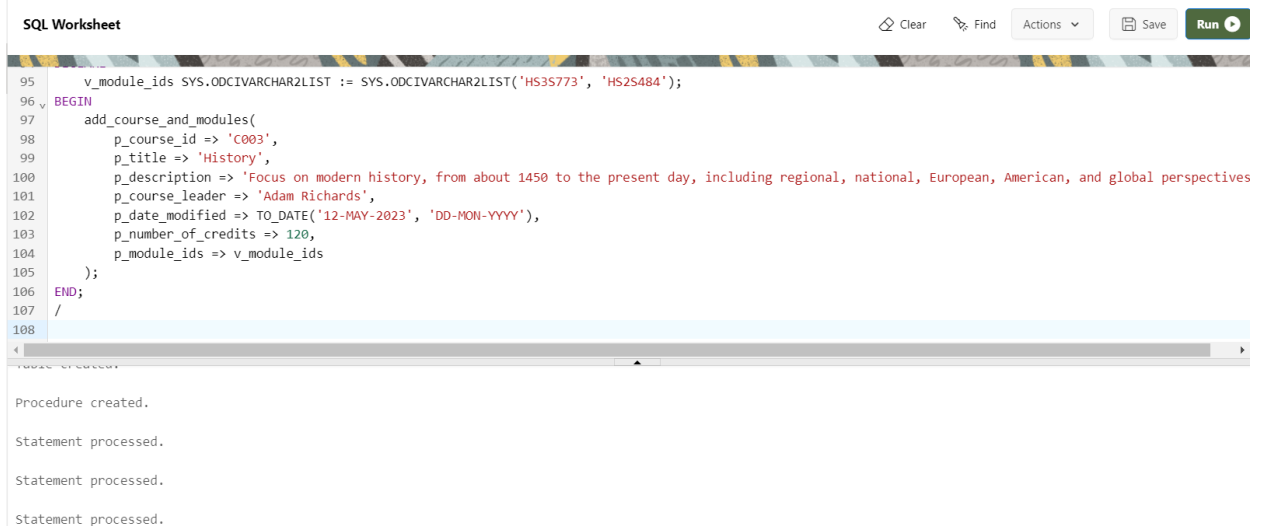
/

2. Inserting Course C002 (Business)

```
DECLARE
  v_module_ids SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('BS1S737', 'BS3S374');
BEGIN
  add_course_and_modules(
    p_course_id => 'C002',
    p_title => 'Business',
    p_description => 'Focuses on management and looks at the different functions of business and how
they inter-relate.',
    p_course_leader => 'Thomas Page',
    p_date_modified => TO_DATE('03-MAR-2021', 'DD-MON-YYYY'),
    p_number_of_credits => 120,
    p_module_ids => v_module_ids
  );
END;
/
```

3. Inserting Course C003 (History)

```
DECLARE
  v_module_ids SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('HS3S773', 'HS2S484');
BEGIN
  add_course_and_modules(
    p_course_id => 'C003',
    p_title => 'History',
    p_description => 'Focus on modern history, from about 1450 to the present day, including regional,
national, European, American and global perspectives.',
    p_course_leader => 'Adam Richards',
    p_date_modified => TO_DATE('12-MAY-2023', 'DD-MON-YYYY'),
    p_number_of_credits => 120,
    p_module_ids => v_module_ids
  );
END;
/
```



SQL Worksheet

Clear Find Actions Save Run

```
95      v_module_ids SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('HS3S773', 'HS2S484');
96  BEGIN
97      add_course_and_modules(
98          p_course_id => 'C003',
99          p_title => 'History',
100         p_description => 'Focus on modern history, from about 1450 to the present day, including regional, national, European, American, and global perspectives
101         p_course_leader => 'Adam Richards',
102         p_date_modified => TO_DATE('12-MAY-2023', 'DD-MON-YYYY'),
103         p_number_of_credits => 120,
104         p_module_ids => v_module_ids
105     );
106 END;
107 /
108
```

Procedure created.

Statement processed.

Statement processed.

Statement processed.

To verify that the records were inserted correctly, run the following SQL queries:

-- Check the course table

SELECT * FROM course;

SQL Worksheet

ClearFindActionsSaveRun

1 -- Check the course table

2 SELECT * FROM course;

3

COURSE_ID	TITLE	DESCRIPTION	COURSE_LEADER	DATE_MODIFIED	NUMBER_OF_CREDITS
C001	Computing	Focuses on developing skills that employers demand for roles in the IT industry.	Richard Jones	20-APR-24	120
C002	Business	Focuses on management and looks at the different functions of business and how they inter-relate.	Thomas Page	03-MAR-21	120
C003	History	Focus on modern history, from about 1450 to the present day, including regional, national, European, American, and global perspectives.	Adam Richards	12-MAY-23	120

Download CSV

3 rows selected.

-- Check the modules table

SELECT * FROM modules;

SQL Worksheet

ClearFindActionsSaveRun

1 -- Check the modules table

2 SELECT * FROM modules;

3

MODULE_ID	COURSE_ID
IS1S464	C001
IS3S662	C001
BS1S737	C002
BS3S374	C002
HS3S773	C003
HS2S484	C003

Download CSV

6 rows selected.

Whole Script

-- 1. Create the course and modules tables

```
CREATE TABLE course (  
  course_id VARCHAR2(13) NOT NULL PRIMARY KEY,  
  title VARCHAR2(200),  
  description VARCHAR2(2000),  
  course_leader VARCHAR2(200),  
  date_modified DATE,  
  number_of_credits NUMBER  
);  
  
CREATE TABLE modules (  
  module_id VARCHAR2(100) NOT NULL PRIMARY KEY,  
  course_id VARCHAR2(13) REFERENCES course (course_id)  
);
```

-- 2. Create the procedure to insert a new course and modules

```
CREATE OR REPLACE PROCEDURE add_course_and_modules (  
  p_course_id    IN VARCHAR2,  
  p_title        IN VARCHAR2,  
  p_description   IN VARCHAR2,  
  p_course_leader IN VARCHAR2,  
  p_date_modified IN DATE,  
  p_number_of_credits IN NUMBER,  
  p_module_ids   IN SYS.ODCIVARCHAR2LIST  
) IS  
  invalid_input EXCEPTION;  
BEGIN  
  IF p_course_id IS NULL OR p_title IS NULL OR p_course_leader IS NULL OR p_number_of_credits IS  
  NULL THEN  
    RAISE invalid_input;  
  END IF;  
  
  IF p_number_of_credits <= 0 THEN  
    RAISE invalid_input;  
  END IF;  
  
  INSERT INTO course (  
    course_id, title, description, course_leader, date_modified, number_of_credits  
  ) VALUES (  
    p_course_id, p_title, p_description, p_course_leader, p_date_modified, p_number_of_credits  
  );  
  
  FOR i IN 1..p_module_ids.COUNT LOOP
```

```

        INSERT INTO modules (module_id, course_id)
        VALUES (p_module_ids(i), p_course_id);
    END LOOP;

    COMMIT;

EXCEPTION
    WHEN invalid_input THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error: Invalid input values. ');
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/

```

-- 3. Insert records into the course and modules tables using the procedure

-- Example for course C001

```

DECLARE
    v_module_ids SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('IS1S464', 'IS3S662');
BEGIN
    add_course_and_modules(
        p_course_id => 'C001',
        p_title => 'Computing',
        p_description => 'Focuses on developing skills that employers demand for roles in the IT industry.',
        p_course_leader => 'Richard Jones',
        p_date_modified => TO_DATE('20-APR-2024', 'DD-MON-YYYY'),
        p_number_of_credits => 120,
        p_module_ids => v_module_ids
    );
END;
/

```

-- Example for course C002

```

DECLARE
    v_module_ids SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('BS1S737', 'BS3S374');
BEGIN
    add_course_and_modules(
        p_course_id => 'C002',
        p_title => 'Business',
        p_description => 'Focuses on management and looks at the different functions of business and how they inter-relate.',
        p_course_leader => 'Thomas Page',

```



```
    p_date_modified => TO_DATE('03-MAR-2021', 'DD-MON-YYYY'),
    p_number_of_credits => 120,
    p_module_ids => v_module_ids
);
END;
/
```

-- Example for course C003

```
DECLARE
    v_module_ids SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('HS3S773', 'HS2S484');
BEGIN
    add_course_and_modules(
        p_course_id => 'C003',
        p_title => 'History',
        p_description => 'Focus on modern history, from about 1450 to the present day, including regional,
national, European, American, and global perspectives.',
        p_course_leader => 'Adam Richards',
        p_date_modified => TO_DATE('12-MAY-2023', 'DD-MON-YYYY'),
        p_number_of_credits => 120,
        p_module_ids => v_module_ids
    );
END;
/
```