**Batch T4**

**Practical No. 2**

**Title of Assignment :**

**MYSQL / PSM Review, Object Relational Databases**

**Student Name: Parshwa Herwade**

**Student PRN: 22510064**

I. MySQL / PSM Review :

a) Create a table called test\_table with 2 columns RecordNumber

(type : Number(3)) and CurrentDate (type : Date)). Write a

procedure in PSM which will insert 50 records into test\_table. Insert

the current date value into the table.

b) Create a products table products(ProductID number(4),category

char(3),detail varchar2(30),price number(10,2),stock number(5)).

Insert the sample data.

Write PSM procedure with two arguments X & Y which will increase

price by X% for all products in category Y. X and Y will be given by

user.

II. Object Relational Databases:

a) Create Object Table containing field “name” of size 50

characters and member function “countNoOfWords” which

returns the no. of words in “name” field.

Demonstrate the working by entering different data.

b) Create an address type with the following attributes : address,

city, state & pincode. Include the following methods

i. to extract the addresses based on given keyword.

j. to return the no. of words in each given field (method

should accept the name of attribute/field)

c) Create a user defined data type course\_Type with 2 attributes

course\_id, description :

i. Create an object table based on the type created.

j. Insert rows into the table

Demonstrate the working with different data sets

**Title**

**Working with MySQL, PSM, and Object Relational Databases**

**Objective / Aim**

* To practice and demonstrate the creation and manipulation of relational databases using SQL and PSM.
* To explore object-relational database concepts, including user-defined types, object tables, and methods.

**Introduction**

This assignment aims to familiarize students with advanced database concepts, including procedural SQL programming (PSM) and object-relational database design. It covers:

1. Creating and populating relational tables using SQL procedures.
2. Implementing object-oriented features such as user-defined types, methods, and object tables in SQL.
3. Demonstrating the functionality through practical examples and observations.

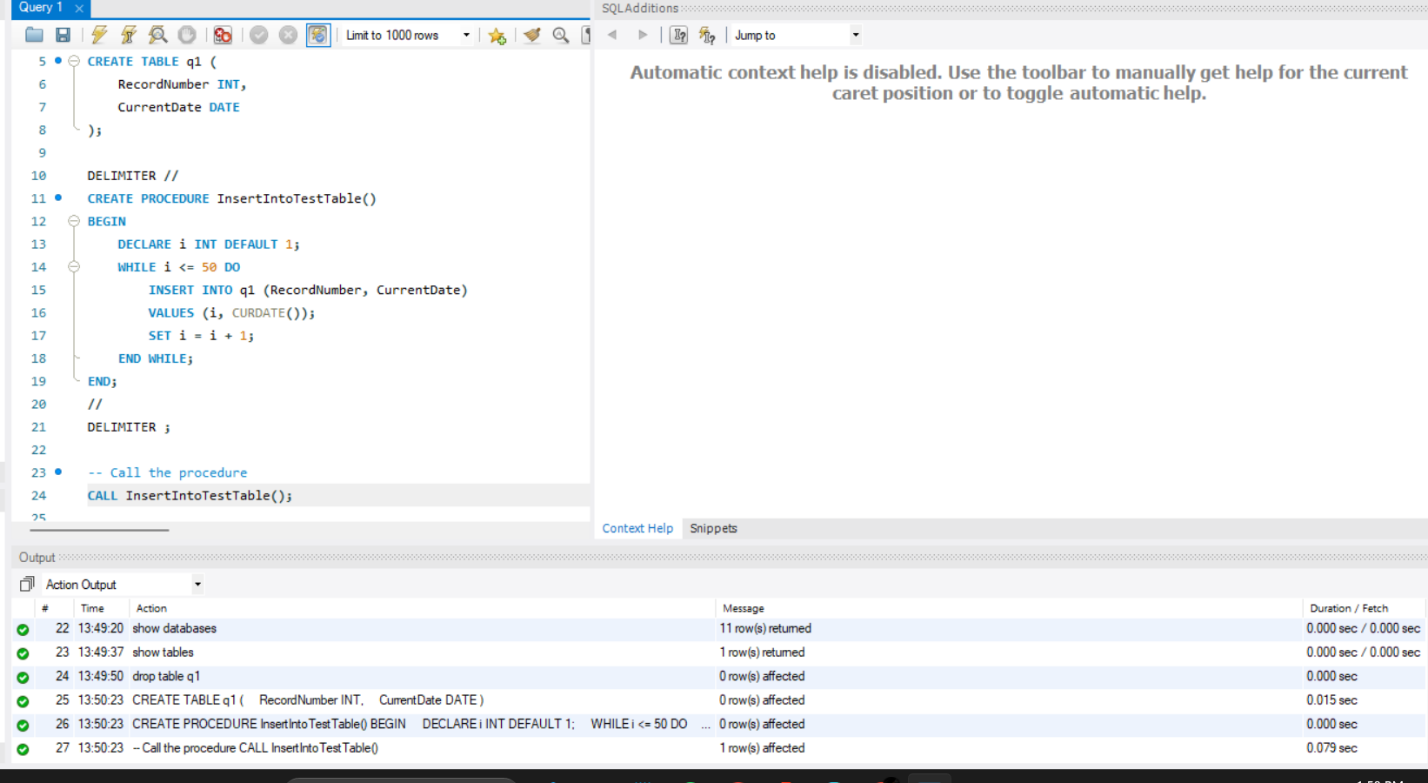
**Theory / Algorithms**

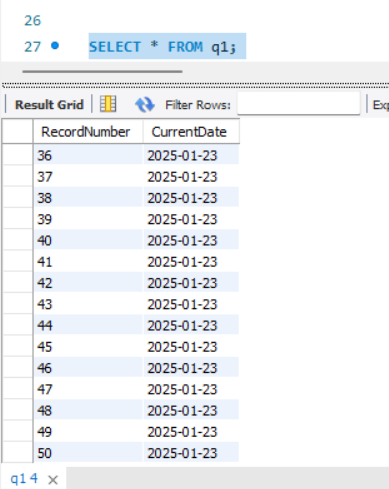
**Part I: MySQL / PSM Review**

1. **Table Creation and Record Insertion**
   * Create a table with two columns: RecordNumber and CurrentDate.
   * Use a PSM procedure with a loop to insert 50 records into the table.

**Algorithm:**

* + Initialize a counter i to 1.
  + Use a loop to insert records, incrementing the counter each iteration.
  + Insert the current date using CURDATE().



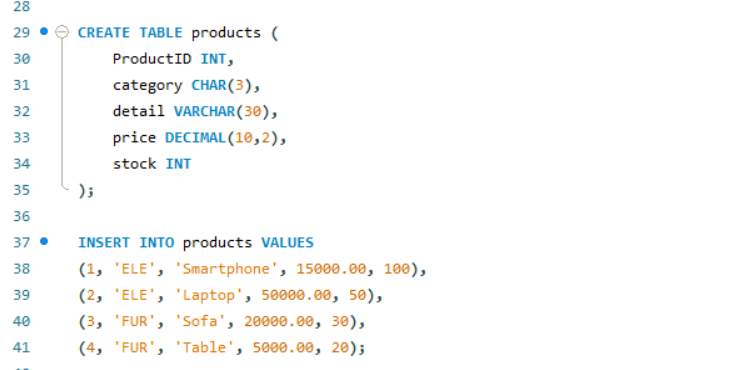


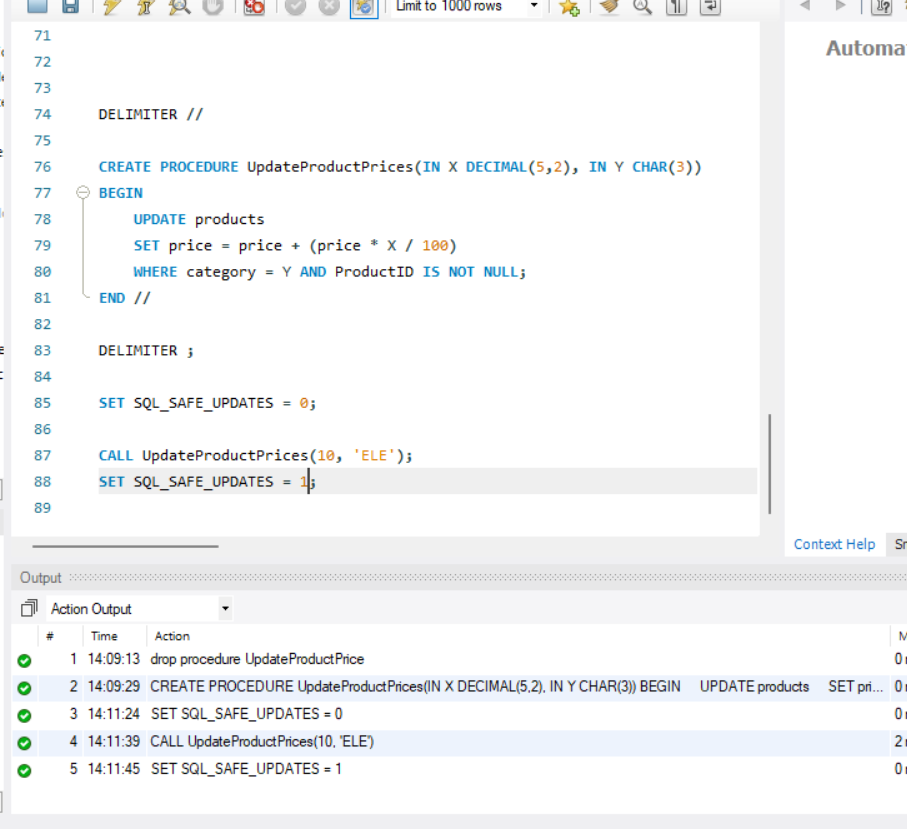
**Above will give us the entries for upto 50 as per the question and the code for it.**

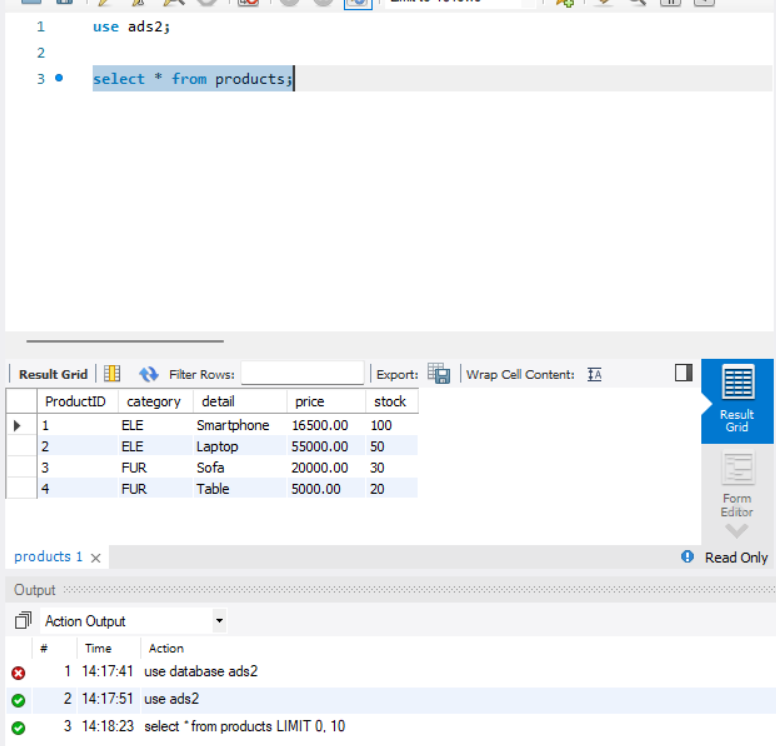
1. **Product Price Update by Category**
   * Create a products table to store product details.
   * Use a procedure that takes two arguments: X (percentage increase) and Y (category).
   * Update the price of all products in the specified category by X%.

**Algorithm:**

* + Use an UPDATE statement with a WHERE clause filtering by category.
  + Calculate the new price using the formula: price + (price \* X / 100).



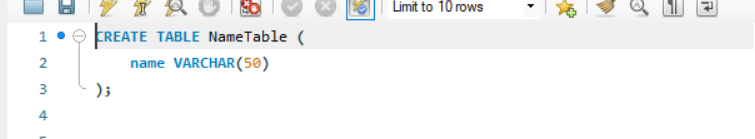


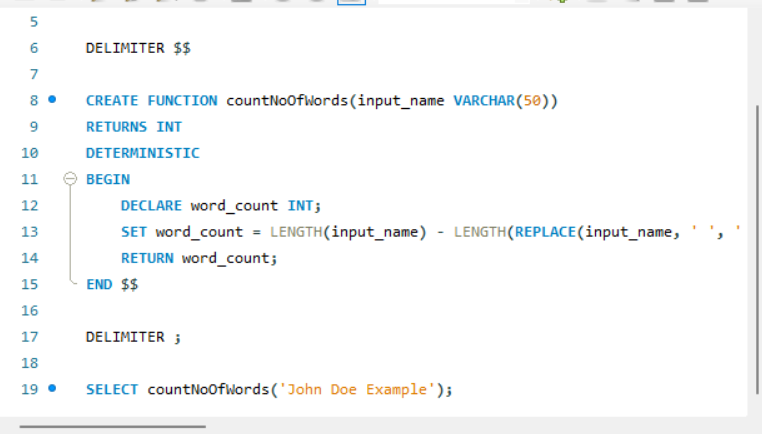


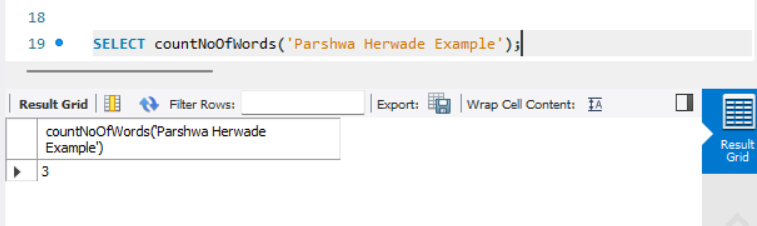
**Here we can verify that the prices of the products have increased by the usage of the formula and it can be checked by the select \* command.**

**Part II: Object Relational Databases**

1. **Object Table with Method for Word Count**
   * Define a user-defined type with a field name and a method countNoOfWords.
   * Method calculates the number of words in the name field using string functions.







**Different words can be used here to know about the no. of words in the string as u can see above with different examples.**

Object Table Creation:

* CREATE TABLE NameTable (name VARCHAR(50));
  + This line creates a table named "NameTable" with a single column "name" of type VARCHAR(50), which can store up to 50 characters.

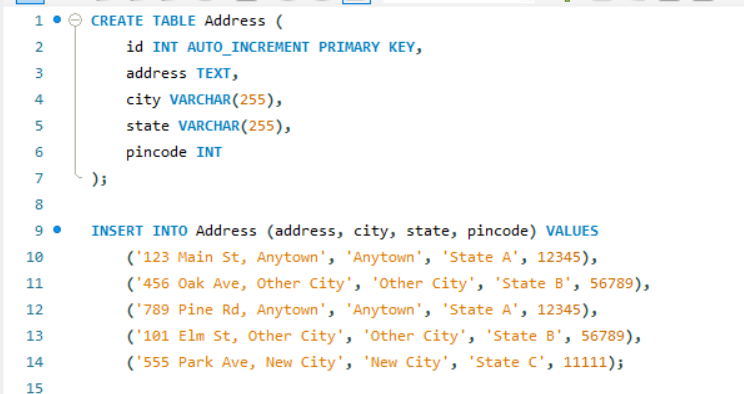
User-Defined Function Creation:

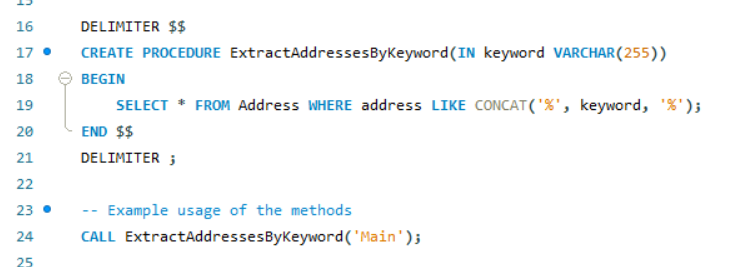
* CREATE FUNCTION countNoOfWords (input\_name VARCHAR(50)) RETURNS INT DETERMINISTIC BEGIN ... END $$
  + This defines a function named "countNoOfWords" that takes a single argument "input\_name" of type VARCHAR(50).
  + The function returns an integer value.
  + DETERMINISTIC keyword indicates that the function always produces the same output for the same input.
  + Inside the function:
    - DECLARE word\_count INT; declares a variable "word\_count" to store the number of words.
    - SET word\_count = LENGTH(input\_name) - LENGTH(REPLACE(input\_name, ' ', '')); calculates the number of words by subtracting the length of the input string without spaces from the original length.
    - RETURN word\_count; returns the calculated number of words.

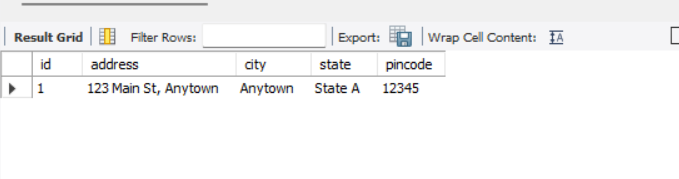
Function Demonstration:

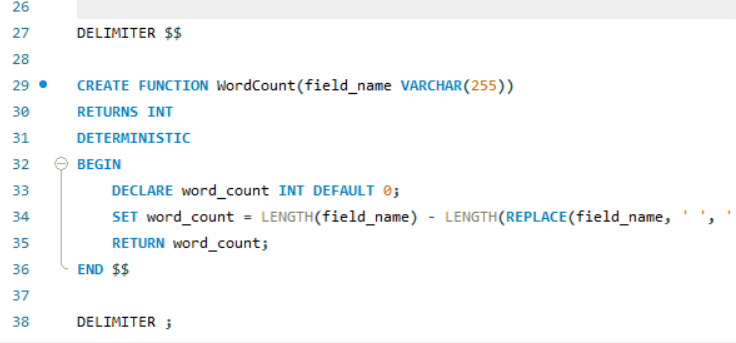
* SELECT countNoOfWords('Parshwa Herwade Example');
  + This query calls the "countNoOfWords" function with the input string "Parshwa Herwade Example".
  + The function will count the number of spaces in the string and return the count, which represents the number of words.

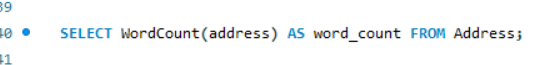
1. **Address Type with Methods**
   * Define a user-defined type with attributes: address, city, state, and pincode.
   * Methods:
     + extractAddress(keyword) to filter addresses containing a keyword.
     + wordCount(fieldName) to count words in the specified field.

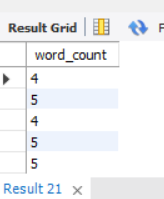


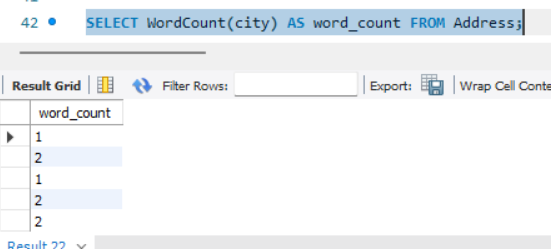






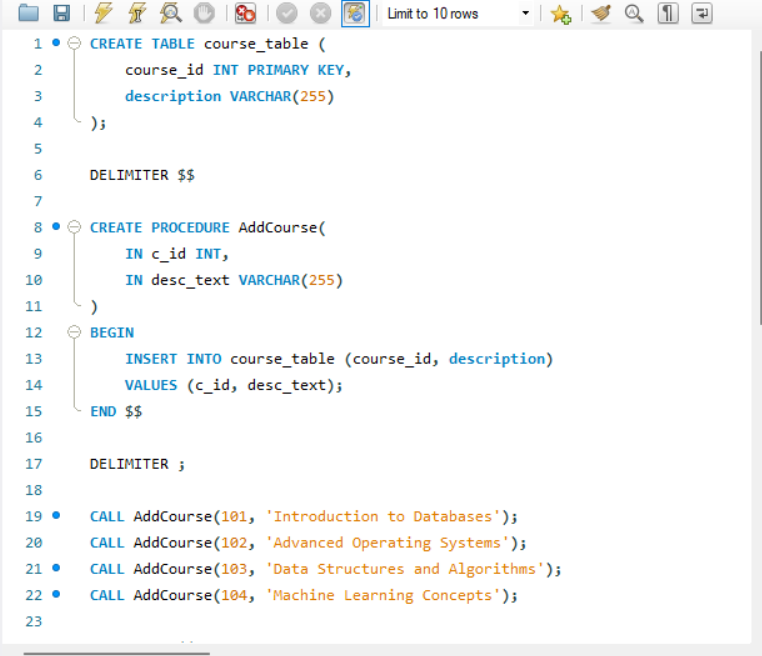


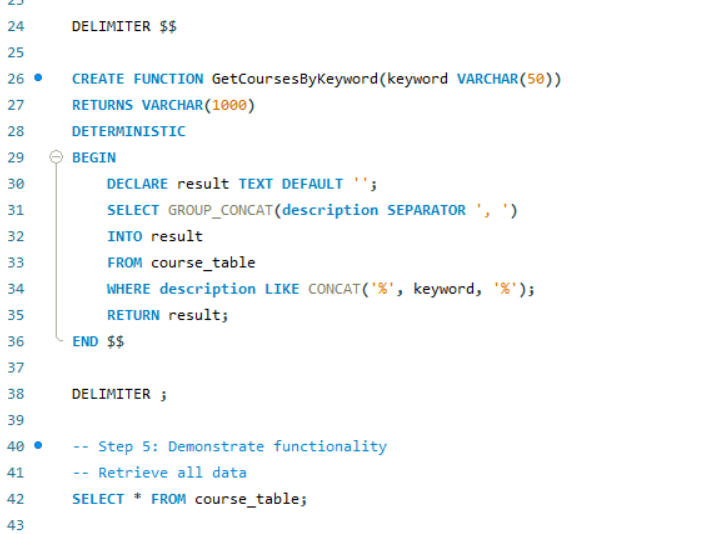


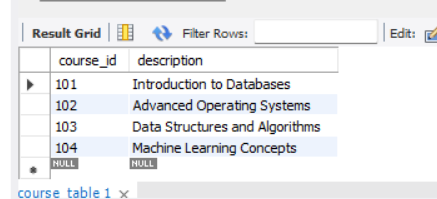


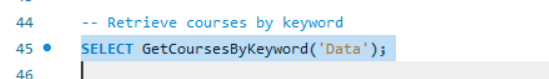
**In this code , we can see that we extract a key word from the names of the different addresses and showcase it in the output  
Also, the word count can be displayed by verifying by typing select command as per above.**

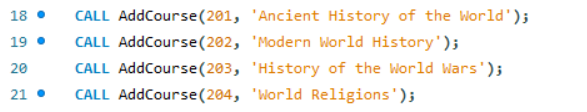
1. **User-Defined Data Type for Courses**
   * Create a user-defined type course\_Type with attributes course\_id and description.
   * Use this type to create an object table and insert rows.

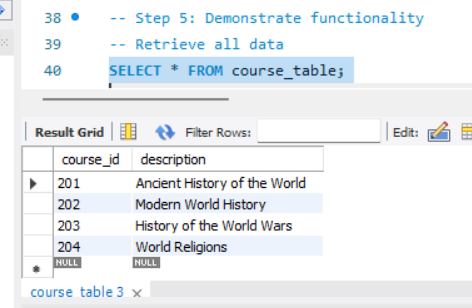


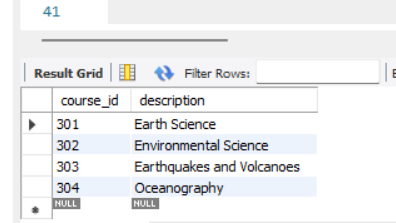
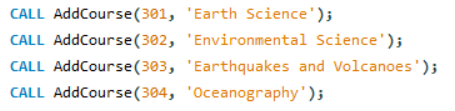
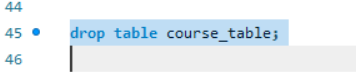
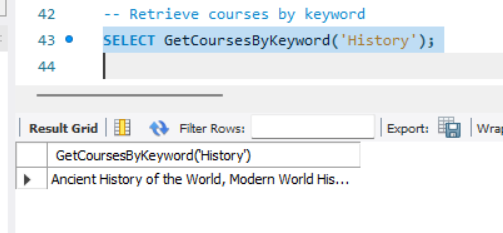


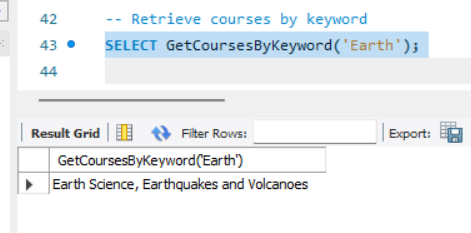


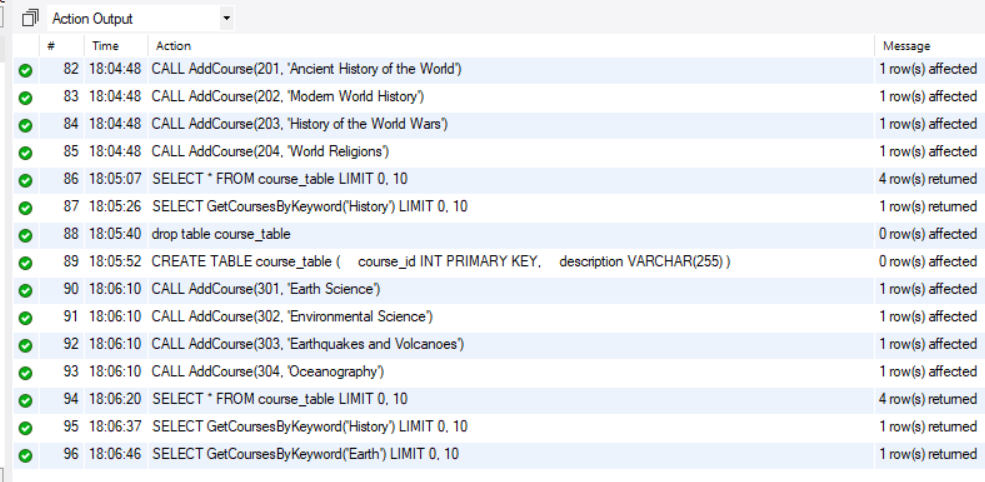












**Here as you can see a table is created and then different datasets are tried and called upon and it returns the word that is common in the given names of the courses.  
  
Using diff. names by replacing or dropping data content we have created the outputs.**

**Create a User-Defined Data Type:**

* CREATE TYPE course\_Type AS OBJECT (course\_id INT, description VARCHAR(255));
  + This defines a user-defined data type named "course\_Type" as an object with two attributes:
    - course\_id of type INT
    - description of type VARCHAR(255)

**Create an Object Table:**

* CREATE TABLE course\_ObjectTable OF course\_Type;
  + This creates a table named "course\_ObjectTable" based on the previously defined "course\_Type".
  + This table will store objects of the "course\_Type".

**Insert Rows into the Object Table:**

* INSERT INTO course\_ObjectTable VALUES (course\_Type(101, 'Introduction to Databases'));
  + This inserts a row into the "course\_ObjectTable" with the specified values for course\_id and description within the "course\_Type" constructor.
  + Similar statements are used to insert other rows with different course information.

**Select Data from the Object Table:**

* SELECT \* FROM course\_ObjectTable;
  + This retrieves all rows from the "course\_ObjectTable", displaying the course\_id and description for each course object.

**Documentation: Functional Block Diagram/DFD**

**1. Functional Block Diagram for PSM Procedures:**

|  |
| --- |
| START |
| Create Table |
| Define Procedure |
| Execute Procedure:  Loop through rows  Insert records  End Procedure |

**2. Functional Block Diagram for Object Relational Database:**

**1. Functional Block Diagram for PSM Procedures:**

|  |
| --- |
| Define User-Defined Type |
| Create Object Table |
| Insert Records |
| Execute Methods Calculate word count Extract address  End |

**Procedure**

1. **Create Relational Tables:**
   * Write SQL commands to define tables (test\_table, products).
   * Populate the products table with sample data.
2. **Write PSM Procedures:**
   * Create a procedure to insert records into test\_table.
   * Create another procedure to update product prices by a percentage.
3. **Object Relational Concepts:**
   * Define user-defined types for objects (ObjectTableType, AddressType, CourseType).
   * Create object tables and insert sample data.
   * Demonstrate method functionality using queries.
4. **Observe Results:**
   * Use SELECT queries to verify the changes.
   * Capture relevant screenshots of table data and method outputs.

**Actual Experiments/Simulation, Results / Observations**

* **Screenshot 1:** Result of inserting 50 records into test\_table.
* **Screenshot 2:** Updated prices in the products table after calling the UpdateProductPrice procedure.
* **Screenshot 3:** Query result for word count using countNoOfWords method in object\_table.
* **Screenshot 4:** Query result for address extraction using extractAddress method.
* **Screenshot 5:** Query result for rows in the course\_object\_table.

**Conclusion**

* Successfully created and manipulated relational tables using MySQL and PSM.
* Demonstrated object-relational database concepts by defining user-defined types, methods, and object tables.
* Validated the functionality of PSM procedures and object methods with practical examples.