# Ex. No.: 09

**Date:**

# AIM:

**TRIGGERS AND PACKAGES**

To implement PL/SQL triggers and packages for the university database system

# Table Structure (student):

CREATE TABLE student ( sid INT(10),

sname VARCHAR(20), dpmt VARCHAR(5), email VARCHAR(25),

contactno NUMERIC(10)

);

# Inserting Records:

INSERT INTO student VALUES

(101, 'Nisanth', 'AM', 'nisanthg.24mca@kongu.edu', 9345292781),

(102, 'Sivakumar', 'DBT', 'sivakumarp.24mca@kongu.edu', 8072363074),

(103, 'Sachin', 'C', 'sachins.24mca@kongu.edu', 8754681258),

(104, 'Siva', 'IOT', 'sivak.24mca@kongu.edu', 6380603146);

# Select \*from student;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SID** | **SNAME** | **DPMT** | **EMAIL** | **CONTACTNO** |
| 101 | Nisanth | AM | [nisanthg.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/nisanthg.24mca%40kongu.edu) | 9345292781 |
| 102 | Sivakumar | DBT | [sivakumarp.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivakumarp.24mca%40kongu.edu) | 8072363074 |
| 103 | Sachin | C | [sachins.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sachins.24mca%40kongu.edu) | 8754681258 |
| 104 | Siva | IOT | [sivak.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivak.24mca%40kongu.edu) | 6380603146 |

1. **Trigger t1: After Insert Trigger**

This trigger prints a message after inserting a new record into the student table. CREATE OR REPLACE TRIGGER t1

AFTER INSERT ON student BEGIN

DBMS\_OUTPUT.PUT\_LINE('Data inserted into student table.'); END;

/

# O/P

**Insert Records:**

INSERT INTO student VALUES (105, 'Anand', 'AI', 'anand.24mca@kongu.edu', 9876543210);

Select\*from student;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SID** | **SNAME** | **DPMT** | **EMAIL** | **CONTACTNO** |
| 101 | Nisanth | AM | [nisanthg.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/nisanthg.24mca%40kongu.edu) | 9345292781 |
| 102 | Sivakumar | DBT | [sivakumarp.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivakumarp.24mca%40kongu.edu) | 8072363074 |
| 103 | Sachin | C | [sachins.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sachins.24mca%40kongu.edu) | 8754681258 |
| 104 | Siva | IOT | [sivak.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivak.24mca%40kongu.edu) | 6380603146 |
| 105 | Anand | AI | [Anand.24mca@kongu.edu](file://localhost/C:/Users/NISANTH/Downloads/Anand.24mca%40kongu.edu) | 9876543210 |

# Trigger t2: After Insert or Update of sname

This trigger displays the old and new sname values after an update. CREATE OR REPLACE TRIGGER t2

AFTER INSERT OR UPDATE OF sname ON student FOR EACH ROW

BEGIN

IF INSERTING THEN

DBMS\_OUTPUT.PUT\_LINE('New student inserted: ' || :NEW.sname); ELSE

DBMS\_OUTPUT.PUT\_LINE('Student name updated');

DBMS\_OUTPUT.PUT\_LINE('New Name: ' || :NEW.sname || ' | Old Name: ' ||

:OLD.sname); END IF;

END;

/

# O/P

**Update Records:**

SQL> UPDATE student SET sname = 'Nisanth G' WHERE sid = 101; Student name updated

# New Name: Nisanth G | Old Name: Nisanth 1 row updated.

SQL> UPDATE student SET sname = 'Sivakumar P' WHERE sid = 102; Student name updated

# New Name: Sivakumar P | Old Name: Sivakumar

1 row updated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SID** | **SNAME** | **DPMT** | **EMAIL** | **CONTACTNO** |
| 101 | Nisanth G | AM | [nisanthg.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/nisanthg.24mca%40kongu.edu) | 9345292781 |
| 102 | Sivakumar P | DBT | [sivakumarp.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivakumarp.24mca%40kongu.edu) | 8072363074 |
| 103 | Sachin | C | [sachins.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sachins.24mca%40kongu.edu) | 8754681258 |
| 104 | Siva | IOT | [sivak.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivak.24mca%40kongu.edu) | 6380603146 |
| 105 | Anand | AI | [Anand.24mca@kongu.edu](file://localhost/C:/Users/NISANTH/Downloads/Anand.24mca%40kongu.edu) | 9876543210 |

# Trigger t3: Before Insert Trigger (Raise Error)

This trigger raises an error before inserting a new record. CREATE OR REPLACE TRIGGER t3

BEFORE INSERT ON student BEGIN

RAISE\_APPLICATION\_ERROR(-20001, 'Insert not allowed in student table.'); END;

# O/P

**Insert Record (Error):**

SQL> INSERT INTO student VALUES (106, 'Ravi', 'AI', 'ravi@kongu.edu', 9876543210); ERROR at line 1:

ORA-20001: Insert not allowed in student table. ORA-06512: at "SYSTEM.T3", line 2

ORA-04088: error during execution of trigger 'SYSTEM.T3'

# Trigger news: Before Update Trigger

This trigger stores the old values in a backup table stu\_new before updating the student table.

# Backup Table (stu\_new):

CREATE TABLE stu\_new ( sid INT(10),

sname VARCHAR(20), dpmt VARCHAR(5), email VARCHAR(25),

contactno NUMERIC(10)

);

# Trigger Code:

CREATE OR REPLACE TRIGGER news BEFORE UPDATE OF sname

ON student

FOR EACH ROW BEGIN

INSERT INTO stu\_new (sid, sname, dpmt, email, contactno)

VALUES (:OLD.sid, :OLD.sname, :OLD.dpmt, :OLD.email, :OLD.contactno); END;

/

# O/P

**Update Records:**

SQL> UPDATE student SET sname = 'Siva Kumar' WHERE sid = 104; 1 row updated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SID** | **SNAME** | **DPMT** | **EMAIL** | **CONTACTNO** |
| 101 | Nisanth | AM | [nisanthg.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/nisanthg.24mca%40kongu.edu) | 9345292781 |
| 102 | Sivakumar | DBT | [sivakumarp.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivakumarp.24mca%40kongu.edu) | 8072363074 |
| 103 | Sachin | C | [sachins.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sachins.24mca%40kongu.edu) | 8754681258 |
| 104 | SivaKumar | IOT | [sivak.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivak.24mca%40kongu.edu) | 6380603146 |
| 105 | Anand | AI | [Anand.24mca@kongu.edu](file://localhost/C:/Users/NISANTH/Downloads/Anand.24mca%40kongu.edu) | 9876543210 |

# Trigger t4: Before Delete Trigger

This trigger prevents deletion of records from the student table. CREATE OR REPLACE TRIGGER t4

BEFORE DELETE ON student BEGIN

RAISE\_APPLICATION\_ERROR(-20002, 'Deletion not allowed in student table.'); END;

/

# O/P

**Delete Record (Error):**

SQL> DELETE FROM student WHERE sid = 103;

ERROR at line 1:

ORA-20002: Deletion not allowed in student table. ORA-06512: at "SYSTEM.T4", line 2

ORA-04088: error during execution of trigger 'SYSTEM.T4'

# Dropping a Trigger:

DROP TRIGGER t4;

# PACKAGES

1. **Package Specification (p1):**

The package contains a procedure find\_contact to display the contact number of a student by sid.

SQL> CREATE OR REPLACE PACKAGE p1 AS

PROCEDURE find\_s(id student.sid%TYPE); END p1;

/**O/P**

Package created.

# Package Body:

SQL> CREATE OR REPLACE PACKAGE BODY p1 AS

PROCEDURE find\_s(id student.sid%TYPE) IS stu\_name student.sname%TYPE;

dept student.dpmt%TYPE; contact student.contactno%TYPE;

BEGIN

SELECT sname, dpmt, contactno INTO stu\_name, dept, contact FROM student

WHERE sid = id;

DBMS\_OUTPUT.PUT\_LINE('Student Name: ' || stu\_name); DBMS\_OUTPUT.PUT\_LINE('Department: ' || dept);

DBMS\_OUTPUT.PUT\_LINE('Contact No: ' || contact); END find\_s;

END p1;

# O/P

Package body created.

# Executing the Package:

**Retrieve Details for Student with SID = 101**

SQL> DECLARE

code student.sid%TYPE := 101; BEGIN

p1.find\_s(code); END;

# O/P

Student Name: **Nisanth** Department: **AM** Contact No: **9345292781**

PL/SQL procedure successfully completed.

|  |  |
| --- | --- |
| **COE (30)** |  |
| **RECORD (20)** |  |
| **VIVA (10)** |  |
| **TOTAL (60)** |  |

# Result:

Thus, the execution of PL/SQL **triggers** and **packages** for the university database system has been successfully implemented and verified with appropriate outputs.

# Ex. No.: 10

**Date:**

# EXCEPTION HANDLING

**AIM:**

To implement PL/SQL programs for performing exception handling in a university database system.

# Table Structure (student):

CREATE TABLE student ( sid INT(10),

sname VARCHAR(20), dpmt VARCHAR(5), email VARCHAR(25),

contactno NUMERIC(10)

);

# Inserting Records:

INSERT INTO student VALUES

(101, 'Nisanth', 'AM', 'nisanthg.24mca@kongu.edu', 9345292781),

(102, 'Sivakumar', 'DBT', 'sivakumarp.24mca@kongu.edu', 8072363074),

(103, 'Sachin', 'C', 'sachins.24mca@kongu.edu', 8754681258),

(104, 'Siva', 'IOT', 'sivak.24mca@kongu.edu', 6380603146);

# Select \*from student;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SID** | **SNAME** | **DPMT** | **EMAIL** | **CONTACTNO** |
| 101 | Nisanth | AM | [nisanthg.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/nisanthg.24mca%40kongu.edu) | 9345292781 |
| 102 | Sivakumar | DBT | [sivakumarp.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivakumarp.24mca%40kongu.edu) | 8072363074 |
| 103 | Sachin | C | [sachins.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sachins.24mca%40kongu.edu) | 8754681258 |
| 104 | Siva | IOT | [sivak.24mca@kongu.edu](https://d.docs.live.net/f3bf4eb6928400f1/Documents/sivak.24mca%40kongu.edu) | 6380603146 |

1. **Exception Handling for No Data Found**

DECLARE

sid\_to\_find INT := 105; -- Non-existent student ID student\_name VARCHAR2(20);

BEGIN

SELECT sname INTO student\_name FROM university\_student WHERE sid = sid\_to\_find;

DBMS\_OUTPUT.PUT\_LINE('Student Name: ' || student\_name); EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No student found with ID: ' || sid\_to\_find); END;

# Output:

No student found with ID: 105

PL/SQL procedure successfully completed.

# Dynamic Query to Fetch Student Name

DECLARE

sid\_to\_find INT := &sid\_to\_find; -- Accept input for SID student\_name VARCHAR2(20);

BEGIN

SELECT sname INTO student\_name FROM university\_student WHERE sid = sid\_to\_find;

DBMS\_OUTPUT.PUT\_LINE('Student Name: ' || student\_name); EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No student found with ID: ' || sid\_to\_find); END;

# Input and Output:

* 1. **Valid SID:**

Enter value for sid\_to\_find: 101 Student Name: Nisanth

PL/SQL procedure successfully completed.

# Invalid SID:

Enter value for sid\_to\_find: 110 No student found with ID:110

PL/SQL procedure successfully completed.

# Handling Too Many Rows Exception

DECLARE

sid INT;

student\_name university\_student.sname%TYPE; BEGIN

SELECT sid, sname INTO sid, student\_name FROM university\_student WHERE dpmT = 'CS';

DBMS\_OUTPUT.PUT\_LINE('Student: ' || sid || ' - ' || student\_name); EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Query returned more than one row.'); END;

# Output:

Error: Query returned more than one row. PL/SQL procedure successfully completed.

# Invalid Cursor Handling

DECLARE

student\_record university\_student%ROWTYPE;

CURSOR student\_cursor IS

SELECT \* FROM university\_student WHERE dpmT = 'XYZ'; -- Non-existent department

BEGIN

FETCH student\_cursor INTO student\_record; EXCEPTION

WHEN INVALID\_CURSOR THEN

DBMS\_OUTPUT.PUT\_LINE('Cursor is invalid or not opened.'); END;

# Output:

Cursor is invalid or not opened.

PL/SQL procedure successfully completed.

# Out of Range Exception Handling

DECLARE

student\_email VARCHAR2(50); invalid\_sid EXCEPTION;

sid\_to\_check INT := 5000; -- Out of range SID BEGIN

IF sid\_to\_check NOT BETWEEN 201 AND 204 THEN RAISE invalid\_sid;

ELSE

SELECT email INTO student\_email FROM university\_student WHERE sid = sid\_to\_check;

DBMS\_OUTPUT.PUT\_LINE('Student Email: ' || student\_email); END IF;

EXCEPTION

WHEN invalid\_sid THEN

DBMS\_OUTPUT.PUT\_LINE('Student ID ' || sid\_to\_check || ' is out of range.'); END;

# Output:

Student ID 5000 is out of range.

PL/SQL procedure successfully completed.

# Arithmetic Exception Handling

DECLARE

marks1 INT := &marks1; marks2 INT := &marks2; average FLOAT;

BEGIN

average := (marks1 + marks2) / 2; DBMS\_OUTPUT.PUT\_LINE('Average Marks: ' || average);

EXCEPTION

WHEN ZERO\_DIVIDE THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Division by zero encountered.'); END;

# Input and Output:

* + 1. **Valid Division:** Enter value for marks1: 85 Enter value for marks2: 90 Average Marks: 87.5

PL/SQL procedure successfully completed.

* + 1. **Valid Division:** Enter value for marks1: 50 Enter value for marks2: 100 Average Marks: 75

PL/SQL procedure successfully completed.

|  |  |
| --- | --- |
| **COE (30)** |  |
| **RECORD (20)** |  |
| **VIVA (10)** |  |
| **TOTAL (60)** |  |

# RESULT:

Thus, exception handling for a **university database system** has been successfully implemented.