|  |  |  |
| --- | --- | --- |
| **Ex.No.: 13** | | **WORKING WITH TRIGGER TRIGGER** |
| **Date:** | **29.10.2024** |

Program 1

Write a code in PL/SQL to develop a trigger that enforces referential integrity by preventing the deletion of a parent record if child records exist.



CREATE OR REPLACE TRIGGER prevent\_parent\_deletion BEFORE DELETE ON parent\_table

FOR EACH ROW DECLARE

child\_count NUMBER; BEGIN

SELECT COUNT(\*) INTO child\_count FROM child\_table

WHERE parent\_id = :OLD.parent\_id;

IF child\_count > 0 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Cannot delete parent record as child records exist.');

END IF;

END;

Testing of Trigger

DELETE FROM parent\_table WHERE parent\_id = 1;

Write a code in PL/SQL to create a trigger that checks for duplicate values in a specific column and raises an exception if found.

CREATE OR REPLACE TRIGGER check\_duplicate\_value BEFORE INSERT OR UPDATE ON table\_name

FOR EACH ROW DECLARE

v\_count NUMBER; BEGIN

-- Check if the new value already exists in the table SELECT COUNT(\*) INTO v\_count

FROM table\_name

WHERE specific\_column = :NEW.specific\_column;

-- If a duplicate is found, raise an error IF v\_count > 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Duplicate value detected in specific column.'); END IF;

END;

/

# Output:

ORA-20002: Duplicate value detected in specific column.

Write a code in PL/SQL to create a trigger that restricts the insertion of new rows if the total of a column's values exceeds a certain threshold.

CREATE OR REPLACE TRIGGER restrict\_insertion BEFORE INSERT ON table\_name

FOR EACH ROW DECLARE

v\_total NUMBER;

v\_threshold CONSTANT NUMBER := 10000; -- Set your threshold here BEGIN

-- Calculate the total sum of the column values

SELECT SUM(column\_name) INTO v\_total FROM table\_name;

-- Prevent insertion if the threshold is exceeded

IF v\_total + :NEW.column\_name > v\_threshold THEN RAISE\_APPLICATION\_ERROR(-20003, 'Cannot insert, total column value

exceeds threshold.');

END IF;

END;

/

# Output:

ORA-20003: Cannot insert, total column value exceeds threshold.

Write a code in PL/SQL to design a trigger that captures changes made to specific columns and logs them in an audit table.

CREATE OR REPLACE TRIGGER log\_column\_changes AFTER UPDATE ON table\_name

FOR EACH ROW BEGIN

-- Check if specific columns have been modified

IF :OLD.column\_name1 != :NEW.column\_name1 OR :OLD.column\_name2 !=

:NEW.column\_name2 THEN

-- Insert the old and new values into the audit table

INSERT INTO audit\_table (user\_id, change\_time, old\_value, new\_value) VALUES (USER, SYSDATE, :OLD.column\_name1 || ', ' || :OLD.column\_name2,

:NEW.column\_name1 || ', ' || :NEW.column\_name2); END IF;

END;

/

Output:

## User\_ID Change\_Time Old\_Value New\_Value

SYSTEM 2024-09-19

10:05:00

OldValue1, OldValue2

NewValue, AnotherNewValue

Write a code in PL/SQL to implement a trigger that records user activity (inserts, updates, deletes) in an audit log for a given set of tables.

CREATE OR REPLACE TRIGGER audit\_user\_activity AFTER INSERT OR UPDATE OR DELETE ON table\_name FOR EACH ROW

BEGIN

IF INSERTING THEN

INSERT INTO audit\_log (user\_id, operation, record\_id, change\_time) VALUES (USER, 'INSERT', :NEW.id\_column, SYSDATE);

ELSIF UPDATING THEN

INSERT INTO audit\_log (user\_id, operation, record\_id, change\_time) VALUES (USER, 'UPDATE', :NEW.id\_column, SYSDATE);

ELSIF DELETING THEN

INSERT INTO audit\_log (user\_id, operation, record\_id, change\_time) VALUES (USER, 'DELETE', :OLD.id\_column, SYSDATE);

END IF;

END;

/

|  |  |  |  |
| --- | --- | --- | --- |
| **User\_ID** | **Operation** | **Record\_ID** | **Change\_Time** |
| SYSTEM | INSERT | 1 | 2024-09-19  10:10:00 |
| SYSTEM | UPDATE | 1 | 2024-09-19  10:15:00 |
| SYSTEM | DELETE | 1 | 2024-09-19  10:20:00 |

Write a code in PL/SQL to implement a trigger that automatically calculates and updates a running total column for a table whenever new rows are inserted.

CREATE OR REPLACE TRIGGER update\_running\_total AFTER INSERT ON table\_name

FOR EACH ROW BEGIN

-- Update the running total column in the total\_table UPDATE total\_table

SET running\_total = running\_total + :NEW.value\_column WHERE total\_id = :NEW.total\_id;

END;

/

Output:

## Total\_ID Running\_Total

1 1500

Write a code in PL/SQL to create a trigger that validates the availability of items before allowing an order to be placed, considering stock levels and pending orders.

CREATE OR REPLACE TRIGGER validate\_item\_availability BEFORE INSERT ON orders

FOR EACH ROW DECLARE

v\_stock\_level NUMBER; v\_pending\_orders NUMBER; BEGIN

SELECT stock INTO v\_stock\_level FROM inventory WHERE item\_id = :NEW.item\_id;

-- Check pending orders

SELECT SUM(quantity) INTO v\_pending\_orders FROM orders

WHERE item\_id = :NEW.item\_id AND status = 'Pending';

-- Ensure stock is available for the order

IF v\_stock\_level - v\_pending\_orders < :NEW.order\_quantity THEN RAISE\_APPLICATION\_ERROR(-20004, 'Insufficient stock available for this

order.'); END IF; END;

/

## Output:

ORA-20004: Insufficient stock available for this order.