

EXERCISE-17

TRIGGER

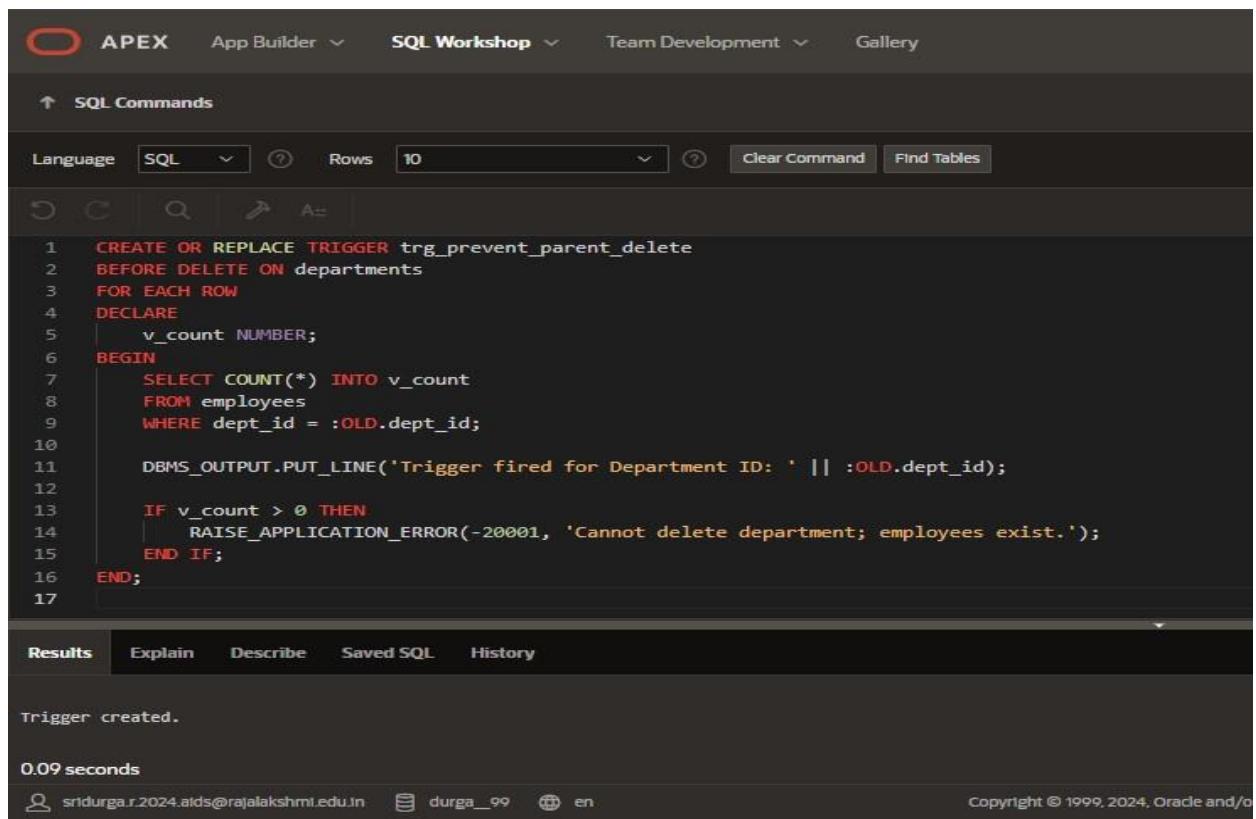
NAME: Vidhyashree.J

ROLL NO: 241801310

DEPARTMENT: AI&DS

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.



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop' (selected), 'Team Development', and 'Gallery'. The main area is titled 'SQL Commands' with a 'Language' dropdown set to 'SQL'. Below the title are various toolbar icons. The code editor contains the following PL/SQL code:

```
1 CREATE OR REPLACE TRIGGER trg_prevent_parent_delete
2 BEFORE DELETE ON departments
3 FOR EACH ROW
4 DECLARE
5     v_count NUMBER;
6 BEGIN
7     SELECT COUNT(*) INTO v_count
8     FROM employees
9     WHERE dept_id = :OLD.dept_id;
10
11    DBMS_OUTPUT.PUT_LINE('Trigger fired for Department ID: ' || :OLD.dept_id);
12
13    IF v_count > 0 THEN
14        RAISE_APPLICATION_ERROR(-20001, 'Cannot delete department; employees exist.');
15    END IF;
16
17 END;
```

The 'Results' tab is selected at the bottom, showing the message 'Trigger created.' and a execution time of '0.09 seconds'. The status bar at the bottom right indicates the user is 'durga_99' and the session is 'en'.

The screenshot shows the Oracle APEX SQL Workshop interface. In the SQL Commands tab, a PL/SQL block is run:

```
1 BEGIN
2   DELETE FROM departments WHERE dept_id = 1;
3 EXCEPTION
4   WHEN OTHERS THEN
5     DBMS_OUTPUT.PUT_LINE(SQLERRM);
6 END;
```

The results show the error output:

```
Trigger fired for Department ID: 1
ORA-20001: Cannot delete department; employees exist.
ORA-06512: at "WKSP_DURGA99.TRG_PREVENT_PARENT_DELETE", line 11
ORA-04088: error during execution of trigger 'WKSP_DURGA99.TRG_PREVENT_PARENT_DELETE'

1 row(s) deleted.

0.05 seconds
```

At the bottom, session details and copyright information are visible.

Program 2

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.

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

```
1 CREATE OR REPLACE TRIGGER trg_prevent_duplicate_roll
2 BEFORE INSERT OR UPDATE ON students
3 FOR EACH ROW
4 DECLARE
5     v_count NUMBER;
6 BEGIN
7     SELECT COUNT(*) INTO v_count
8     FROM students
9     WHERE roll_no = :NEW.roll_no
10    AND ( :NEW.student_id IS NULL OR student_id != :NEW.student_id );
11
12    DBMS_OUTPUT.PUT_LINE('Trigger fired for Roll No: ' || :NEW.roll_no);
13
14    IF v_count > 0 THEN
15        RAISE_APPLICATION_ERROR(-20002, 'Duplicate roll number not allowed.');
16    END IF;
17
18 END;
```

Results Explain Describe Saved SQL History

Trigger created.

0.07 seconds

sridurga.r.2024.alds@rajalakshmi.edu.in durga_99 en Copyright © 1999,

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

```
1 BEGIN
2     INSERT INTO students VALUES (3, 101, 'Charlie');
3 EXCEPTION
4     WHEN OTHERS THEN
5         DBMS_OUTPUT.PUT_LINE(SQLERRM);
6 END;
```

Results Explain Describe Saved SQL History

Trigger fired for Roll No: 101
ORA-20002: Duplicate roll number not allowed.
ORA-06512: at "WKSP_DURGA99.TRG_PREVENT_DUPLICATE_ROLL", line 12
ORA-04088: error during execution of trigger 'WKSP_DURGA99.TRG_PREVENT_DUPLICATE_ROLL'

1 row(s) inserted.

0.01 seconds

Program 3

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.

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, the 'SQL Workshop' tab is selected. The main area displays the following PL/SQL code:

```
1 CREATE OR REPLACE TRIGGER trg_limit_total_sales
2 BEFORE INSERT ON sales
3 FOR EACH ROW
4 DECLARE
5     v_total NUMBER;
6     v_threshold CONSTANT NUMBER := 10000; -- threshold limit
7 BEGIN
8     SELECT NVL(SUM(amount), 0) INTO v_total FROM sales;
9
10    DBMS_OUTPUT.PUT_LINE('Current total: ' || v_total);
11
12    IF v_total + :NEW.amount > v_threshold THEN
13        RAISE_APPLICATION_ERROR(-20003, 'Cannot insert. Total sales would exceed limit of ' || v_threshold);
14    END IF;
15
16
```

Below the code, the 'Results' tab is active, showing the output: "Trigger created." and "0.07 seconds". The bottom status bar shows the user's email as sridurga.r.2024.alds@rajalakshmi.edu.in and the session ID as durga_99.

The screenshot shows the Oracle SQL Workshop interface. In the top navigation bar, the 'SQL Workshop' tab is selected. The main area displays the following PL/SQL code:

```
1 BEGIN
2     INSERT INTO sales VALUES (4, 2000);
3 EXCEPTION
4     WHEN OTHERS THEN
5         DBMS_OUTPUT.PUT_LINE(SQLERRM);
6 END;
```

Below the code, the 'Results' tab is active, showing the output: "Current total: 9500", "ORA-20003: Cannot insert. Total sales would exceed limit of 10000", "ORA-06512: at "WKSP_DURGA99.TRG_LIMIT_TOTAL_SALES", line 10", "ORA-04088: error during execution of trigger 'WKSP_DURGA99.TRG_LIMIT_TOTAL_SALES'", and "1 row(s) inserted." The bottom status bar shows the user's email as sridurga.r.2024.alds@rajalakshmi.edu.in and the session ID as durga_99.

Program 4

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

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

```

1 CREATE OR REPLACE TRIGGER trg_audit_employee_changes
2 AFTER UPDATE OF dept_name, salary ON employees
3 FOR EACH ROW
4 BEGIN
5   DBMS_OUTPUT.PUT_LINE('Audit Trigger Fired for Emp ID: ' || :OLD.emp_id);
6
7   INSERT INTO audit_employees (emp_id, old_dept, new_dept,
8                               old_salary, new_salary,
9                               changed_on, changed_by)
10  VALUES (:OLD.emp_id,
11           :OLD.dept_name, :NEW.dept_name,
12           :OLD.salary, :NEW.salary,
13           SYSDATE, USER);
14
15 END;

```

Results Explain Describe Saved SQL History

Trigger created.

0.09 seconds

APEX App Builder SQL Workshop Team Development Gallery Schema WKSP_DURGA99

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

```

1 SELECT * FROM audit_employees;
2

```

Results Explain Describe Saved SQL History

AUDIT_ID	EMP_ID	OLD_DEPT	NEW_DEPT	OLD_SALARY	NEW_SALARY	CHANGED_ON	CHANGED_BY
21	1	HR	Finance	5000	5000	11/3/2025	APEX_PUBLIC_USER
1	2	IT	IT	6000	6500	11/3/2025	APEX_PUBLIC_USER

2 rows returned in 0.01 seconds Download

Program 5

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.

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

```

1 CREATE OR REPLACE TRIGGER trg_student_audit
2 AFTER INSERT OR UPDATE OR DELETE ON students
3 FOR EACH ROW
4 BEGIN
5   IF INSERTING THEN
6     DBMS_OUTPUT.PUT_LINE('Inserted student: ' || :NEW.stud_name);
7     INSERT INTO student_audit (stud_id, operation, new_name, new_course, changed_on, changed_by)
8       VALUES (:NEW.stud_id, 'INSERT', :NEW.stud_name, :NEW.course, SYSDATE, USER);
9
10  ELSIF UPDATING THEN
11    DBMS_OUTPUT.PUT_LINE('Updated student: ' || :OLD.stud_name);
12    INSERT INTO student_audit (stud_id, operation, old_name, new_name, old_course, new_course, changed_on, changed_by)
13      VALUES (:OLD.stud_id, 'UPDATE', :OLD.stud_name, :NEW.stud_name, :OLD.course, :NEW.course, SYSDATE, USER);
14
15  ELSIF DELETING THEN
16    DBMS_OUTPUT.PUT_LINE('Deleted student: ' || :OLD.stud_name);
17    INSERT INTO student_audit (stud_id, operation, old_name, old_course, changed_on, changed_by)
18      VALUES (:OLD.stud_id, 'DELETE', :OLD.stud_name, :OLD.course, SYSDATE, USER);
19  END IF;
20
21 END;

```

Results Explain Describe Saved SQL History

Trigger created.

0.07 seconds

sridurga.r.2024.aids@rajalakshmi.edu.in durga_99 en Copyright © 1999, 2024, Oracle and/or its affiliates.

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables Save Run

```

1 SELECT * FROM student_audit;
2

```

Results Explain Describe Saved SQL History

AUDIT_ID	STUD_ID	OPERATION	OLD_NAME	NEW_NAME	OLD_COURSE	NEW_COURSE	CHANGED_ON	CHANGED_BY
22	1	DELETE	Alice	-	AI	-	11/5/2025	APEX_PUBLIC_USER
1	1	INSERT	-	Alice	-	AI	11/5/2025	APEX_PUBLIC_USER
2	2	UPDATE	Bob	Bob	ML	Data Science	11/5/2025	APEX_PUBLIC_USER
21	2	INSERT	-	Bob	-	ML	11/5/2025	APEX_PUBLIC_USER

4 rows returned in 0.02 seconds Download

sridurga.r.2024.aids@rajalakshmi.edu.in durga_99 en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.2.8

Program 7

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.

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

```

1 CREATE OR REPLACE TRIGGER trg_update_running_total
2 BEFORE INSERT ON sales_summary
3 FOR EACH ROW
4 DECLARE
5     v_total NUMBER;
6 BEGIN
7     -- get the current running total
8     SELECT NVL(MAX(running_total), 0)
9     INTO v_total
10    FROM sales_summary;
11
12    -- add the new amount to the total
13    :NEW.running_total := v_total + :NEW.amount;
14
15    DBMS_OUTPUT.PUT_LINE('Running total updated to: ' || :NEW.running_total);
16
17 END;

```

Results Explain Describe Saved SQL History

Trigger created.
0.08 seconds

sridurgur2024.aids@rajalakshmi.edu.in durga_99 en Copyright

APEX App Builder SQL Workshop Team Development Gallery search Schema

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

```

1 SELECT * FROM sales_summary;
2

```

Results Explain Describe Saved SQL History

SALE_ID	AMOUNT	RUNNING_TOTAL
1	1000	1000
2	2000	3000
3	1500	4500

3 rows returned in 0.01 seconds Download

Program 8

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.

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

```
1 CREATE OR REPLACE TRIGGER trg_validate_order_stock
2 BEFORE INSERT ON orders
3 FOR EACH ROW
4 DECLARE
5     v_stock NUMBER;
6 BEGIN
7     SELECT stock_qty INTO v_stock
8     FROM inventory
9     WHERE item_id = :NEW.item_id;
10
11    IF :NEW.quantity > v_stock THEN
12        RAISE_APPLICATION_ERROR(-20008, 'Not enough stock available for this item.');
13    ELSE
14        DBMS_OUTPUT.PUT_LINE('Order accepted: ' || :NEW.quantity || ' items.');
15    END IF;
16 EXCEPTION
17    WHEN NO_DATA_FOUND THEN
18        RAISE_APPLICATION_ERROR(-20009, 'Item does not exist in inventory.');
19
20
```

Results Explain Describe Saved SQL History

Trigger created.

0.07 seconds

sridurga.r2024.aids@rajalakshmi.edu.in durga_99 en Copyright © 1999, 2024, Oracle and/or its affiliates.

APEX App Builder SQL Workshop Team Development Gallery

↑ SQL Commands

Language SQL Rows 10 Clear Command Find Tables

```
1 BEGIN
2     INSERT INTO orders VALUES (103, 1, 10, SYSDATE);
3 EXCEPTION
4     WHEN OTHERS THEN
5         DBMS_OUTPUT.PUT_LINE(SQLERRM);
6 END;
```

Results Explain Describe Saved SQL History

ORA-20008: Not enough stock available for this item.
ORA-06512: at "WKSP_DURGA99.TRG_VALIDATE_ORDER_STOCK", line 9
ORA-04088: error during execution of trigger 'WKSP_DURGA99.TRG_VALIDATE_ORDER_STOCK'

1 row(s) inserted.

0.00 seconds

sridurga.r2024.aids@rajalakshmi.edu.in durga_99 en Copyright © 1999, 2024, Oracle and/or its affiliates.

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language SQL Rows 10 Clear Command Find Tables

SELECT * FROM orders;

Results Explain Describe Saved SQL History

ORDER_ID	ITEM_ID	QUANTITY	ORDER_DATE
101	1	2	11/3/2025
102	2	5	11/3/2025

2 rows returned in 0.02 seconds [Download](#)

The screenshot shows the Oracle SQL Workshop interface. At the top, there are navigation links: APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. Below the navigation is a search bar and some user icons. The main area is titled "SQL Commands". Under "Language", "SQL" is selected. The "Rows" dropdown is set to 10. There are buttons for "Clear Command" and "Find Tables". Below this, a code editor shows the SQL query: "SELECT * FROM orders;". The results tab is active, showing a table with four columns: ORDER_ID, ITEM_ID, QUANTITY, and ORDER_DATE. The data returned is two rows: one with ORDER_ID 101, ITEM_ID 1, QUANTITY 2, and ORDER_DATE 11/3/2025; and another with ORDER_ID 102, ITEM_ID 2, QUANTITY 5, and ORDER_DATE 11/3/2025. At the bottom, it says "2 rows returned in 0.02 seconds" and has a "Download" link.