

EXERCISE-17

TRIGGER

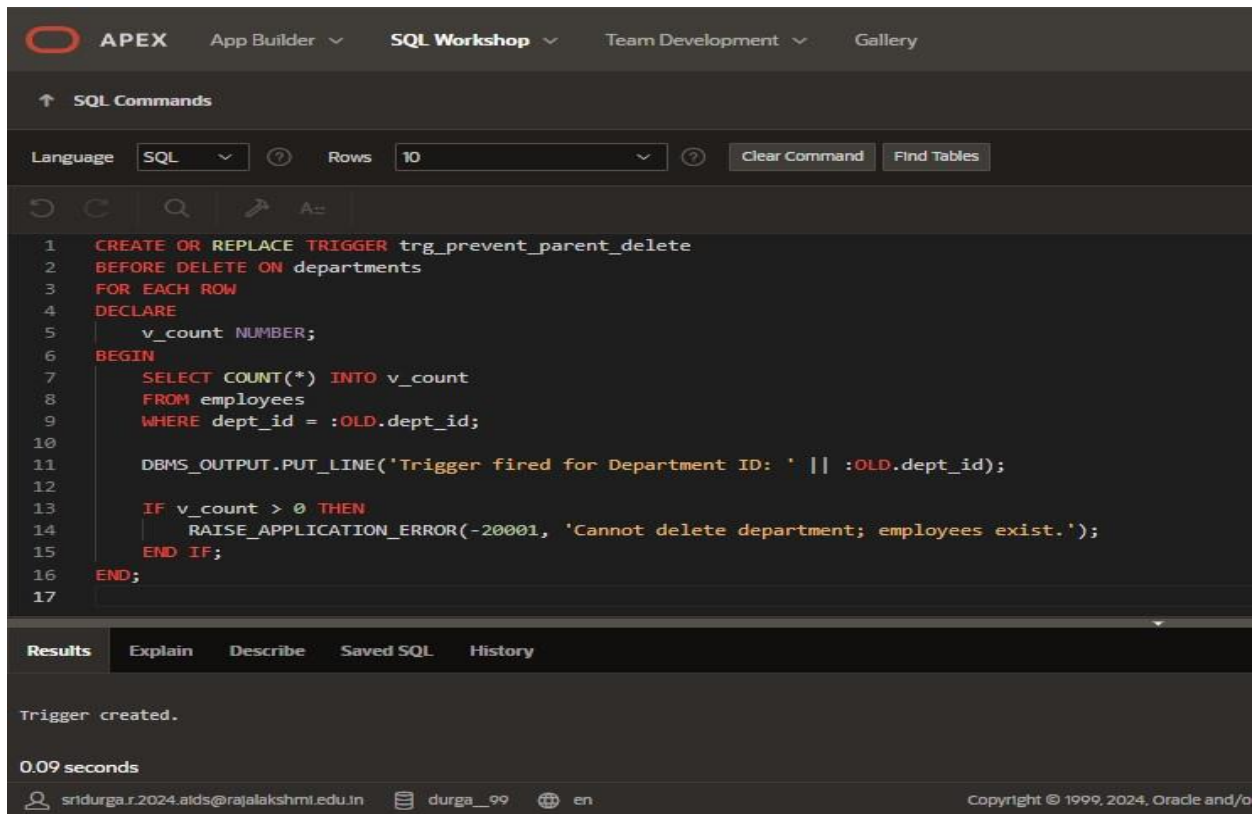
NAME: Vidhyashree.J

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 displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' section is active, showing a code editor with 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.');

Below the code editor, the 'Results' tab is selected, showing the message 'Trigger created.' and a execution time of '0.09 seconds'. The bottom status bar indicates the user 'sridurga.r.2024.aids@rajalakshmi.edu.in', the session 'durga_99', and the language 'en'. A copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates. All rights reserved.' is also visible.


```

The screenshot displays the APEX SQL Workshop interface. At the top, there are navigation tabs: APEX, App Builder, SQL Workshop (selected), Team Development, and Gallery. Below these, the 'SQL Commands' section is active, showing a script editor with the following PL/SQL code:

```
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' tab is selected, showing the execution 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
```

The bottom status bar shows the user 'sridurga.r.2024.aids@rajalakshmi.edu.in', the workspace 'durga_99', and the language 'en'. A copyright notice 'Copyright © 1999, 2024.' is also 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.

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' section is active, showing a SQL script to create a trigger named 'trg_prevent_duplicate_roll'. The script is as follows:

```
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.');

The 'Results' tab is selected, displaying the message 'Trigger created.' and the execution time '0.07 seconds'. The bottom status bar shows the user 'sridurga.r.2024.ajds@rajalakshmi.edu.in', the workspace 'durga_99', and the language 'en'.


```

The screenshot shows the APEX SQL Workshop interface. The top navigation bar is the same as the previous screenshot. The 'SQL Commands' section is active, showing a SQL script to insert a row into the 'students' table:

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

The 'Results' tab is selected, displaying the output of the execution:

```
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.
```

The execution time is '0.01seconds'. The bottom status bar shows the user 'sridurga.r.2024.ajds@rajalakshmi.edu.in', the workspace 'durga_99', and the language 'en'.

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.

APEXApp BuilderSQL WorkshopTeam DevelopmentGallery

↑ SQL Commands

LanguageSQLRows10Clear CommandFind Tables

↶↷🔍🔗A..

```
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 END;
16
```

ResultsExplainDescribeSaved SQLHistory

Trigger created.

0.07 seconds

sridurga.r.2024.alds@rajalakshmi.edu.in

durga_99

en

Copyright © 1999, 2024, Oracle and/or its affiliates.

APEXApp BuilderSQL WorkshopTeam DevelopmentGallery

↑ SQL Commands

LanguageSQLRows10Clear CommandFind Tables

↶↷🔍🔗A..

```
1 BEGIN
2     INSERT INTO sales VALUES (4, 2000);
3 EXCEPTION
4     WHEN OTHERS THEN
5         DBMS_OUTPUT.PUT_LINE(SQLERRM);
6 END;
7
```

ResultsExplainDescribeSaved SQLHistory

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'

1 row(s) inserted.

0.01 seconds

sridurga.r.2024.alds@rajalakshmi.edu.in

durga_99

en

Copyright © 1999, 2024, Oracle and/or its affiliates.

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.

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, the 'SQL Commands' tab is active. The 'Language' dropdown is set to 'SQL', and the 'Rows' dropdown is set to '10'. The 'Clear Command' and 'Find Tables' buttons are visible. The SQL editor contains the following code:

```
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 END;
15
```

Below the SQL editor, the 'Results' tab is active, showing the message 'Trigger created.' and the execution time '0.09 seconds'.

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, the 'SQL Commands' tab is active. The 'Language' dropdown is set to 'SQL', and the 'Rows' dropdown is set to '10'. The 'Clear Command' and 'Find Tables' buttons are visible. The SQL editor contains the following code:

```
1 SELECT * FROM audit_employees;
2
```

Below the SQL editor, the 'Results' tab is active, showing a table with 2 rows. The table has 8 columns: AUDIT_ID, EMP_ID, OLD_DEPT, NEW_DEPT, OLD_SALARY, NEW_SALARY, CHANGED_ON, and CHANGED_BY. The first row has values: 21, 1, HR, Finance, 5000, 5000, 11/3/2025, and APEX_PUBLIC_USER. The second row has values: 1, 2, IT, IT, 6000, 6500, 11/3/2025, and APEX_PUBLIC_USER. The message '2 rows returned in 0.01 seconds' and a 'Download' button are visible below the table.

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

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 END;
21

```

Results Explain Describe Saved SQL History

Trigger created.

0.07 seconds

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

APEX App Builder SQL Workshop Team Development Gallery

Search

SQL Workshop Scheme WKSP_DURGA99

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	-	11/3/2025	APEX_PUBLIC_USER
1	1	INSERT	-	Alice	-	AI	11/3/2025	APEX_PUBLIC_USER
2	2	UPDATE	Bob	Bob	ML	Data Science	11/3/2025	APEX_PUBLIC_USER
21	2	INSERT	-	Bob	-	ML	11/3/2025	APEX_PUBLIC_USER

4 rows returned in 0.02 seconds Download

snidurga.r.2024.aids@irajalakshmi.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.

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' section is active, showing a SQL script for creating a trigger. The script is as follows:

```
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 END;
```

Below the script, the 'Results' tab is selected, displaying the message 'Trigger created.' and the execution time '0.08 seconds'. The bottom status bar shows the user 'sridurga.r.2024.aids@rajalakshmi.edu.in', the workspace 'durga_99', and the language 'en'.

The screenshot shows the APEX SQL Workshop interface with the 'Results' tab selected. The query executed is 'SELECT * FROM sales_summary;'. The results are displayed in a table with three columns: 'SALE_ID', 'AMOUNT', and 'RUNNING_TOTAL'. The table contains three rows of data. The status bar at the bottom indicates '3 rows returned in 0.01 seconds' and provides a 'Download' link.

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

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

LanguageSQLRows10Clear CommandFind Tables

A=

1CREATE OR REPLACE TRIGGER trg_validate_order_stock

2BEFORE INSERT ON orders

3FOR EACH ROW

4DECLARE

5v_stock NUMBER;

6BEGIN

7SELECT stock_qty INTO v_stock

8FROM inventory

9WHERE item_id = :NEW.item_id;

10

11IF :NEW.quantity > v_stock THEN

12RAISE_APPLICATION_ERROR(-20008, 'Not enough stock available for this item.');

13ELSE

14DBMS_OUTPUT.PUT_LINE('Order accepted: ' || :NEW.quantity || ' items.');

15END IF;

16EXCEPTION

17WHEN NO_DATA_FOUND THEN

18RAISE_APPLICATION_ERROR(-20009, 'Item does not exist in inventory.');

19END;

20

Results

ExplainDescribeSaved SQLHistory

Trigger created.

0.07 seconds

sridurga.r.2024.aids@rajalakshmi.edu.in

durga_99

en

Copyright © 1999, 2

APEX

App Builder

SQL Workshop

Team Development

Gallery

↑ SQL Commands

LanguageSQLRows10Clear CommandFind Tables

A=

1BEGIN

2INSERT INTO orders VALUES (103, 1, 10, SYSDATE);

3EXCEPTION

4WHEN OTHERS THEN

5DBMS_OUTPUT.PUT_LINE(SQLERRM);

6END;

7

Results

ExplainDescribeSaved SQLHistory

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.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

Search

SQL Commands

Schema WKSP_DURGA99

Language SQL Rows 10 Clear Command Find Tables

A-Z

1 SELECT * FROM orders;

2

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