EXERCISE-17

TRIGGER

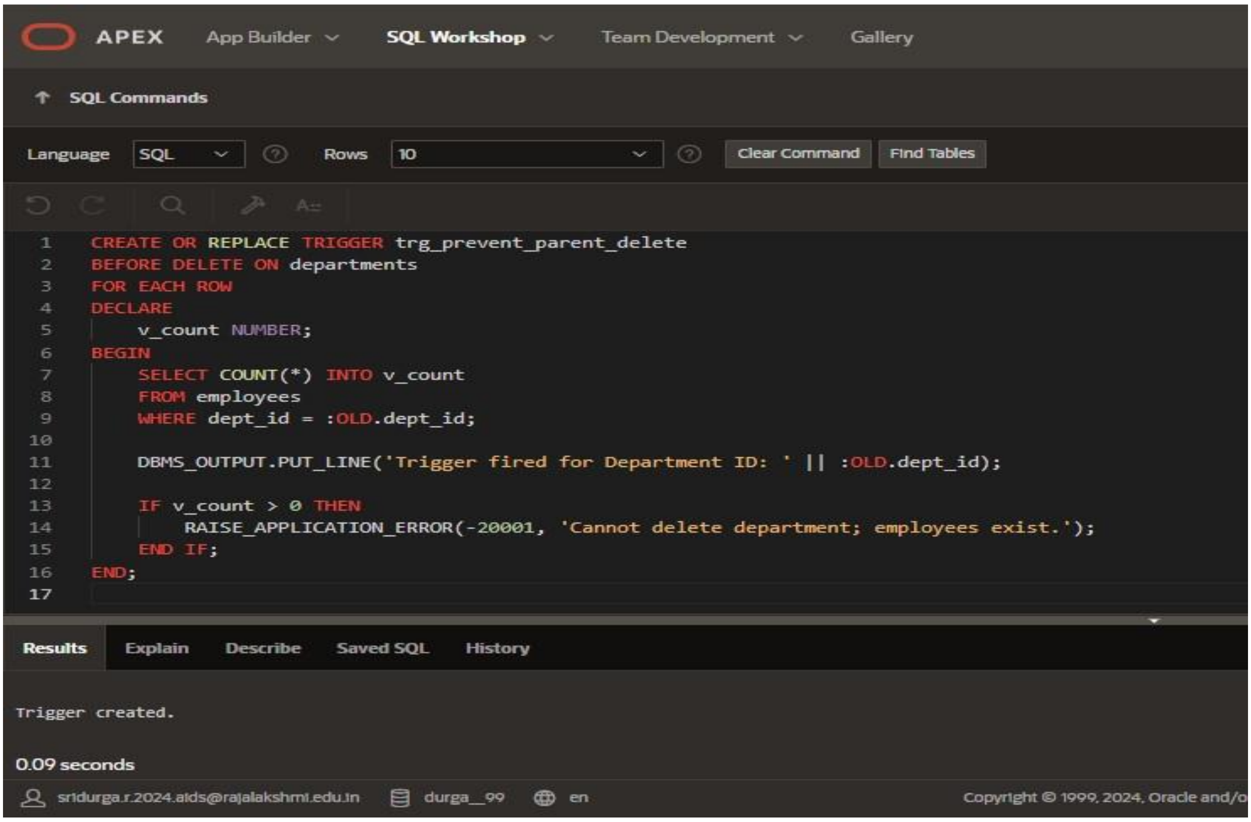
NAME: Vidhyashree.J

ROLL NO: 241801310

DEPARTMENT: AI&DS

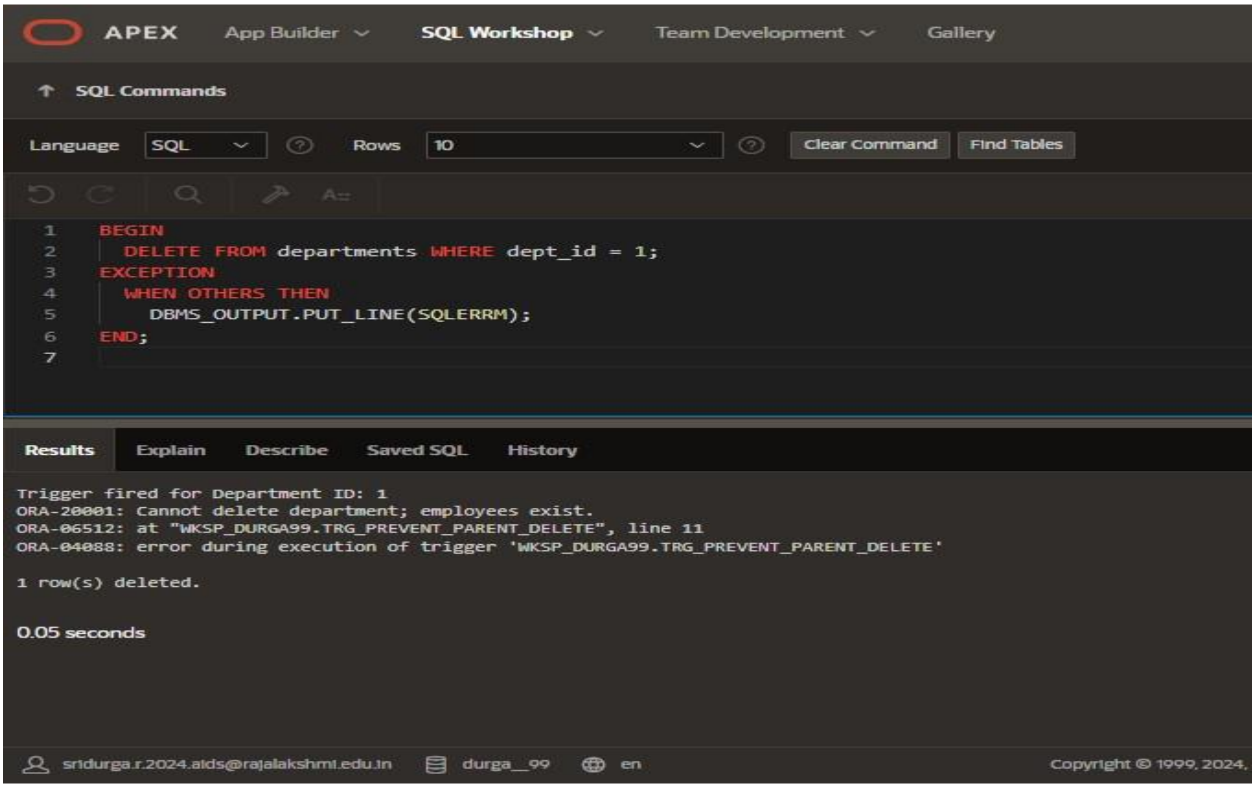
Program 1

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.



```
CREATE OR REPLACE TRIGGER trg_prevent_parent_delete
BEFORE DELETE ON departments
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM employees
    WHERE dept_id = :OLD.dept_id;

    DBMS_OUTPUT.PUT_LINE('Trigger fired for Department ID: ' || :OLD.dept_id);

    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Cannot delete department; employees exist.');
    END IF;
END;
```

```
Trigger created.

0.09 seconds
```



Language SQL

Rows 10

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

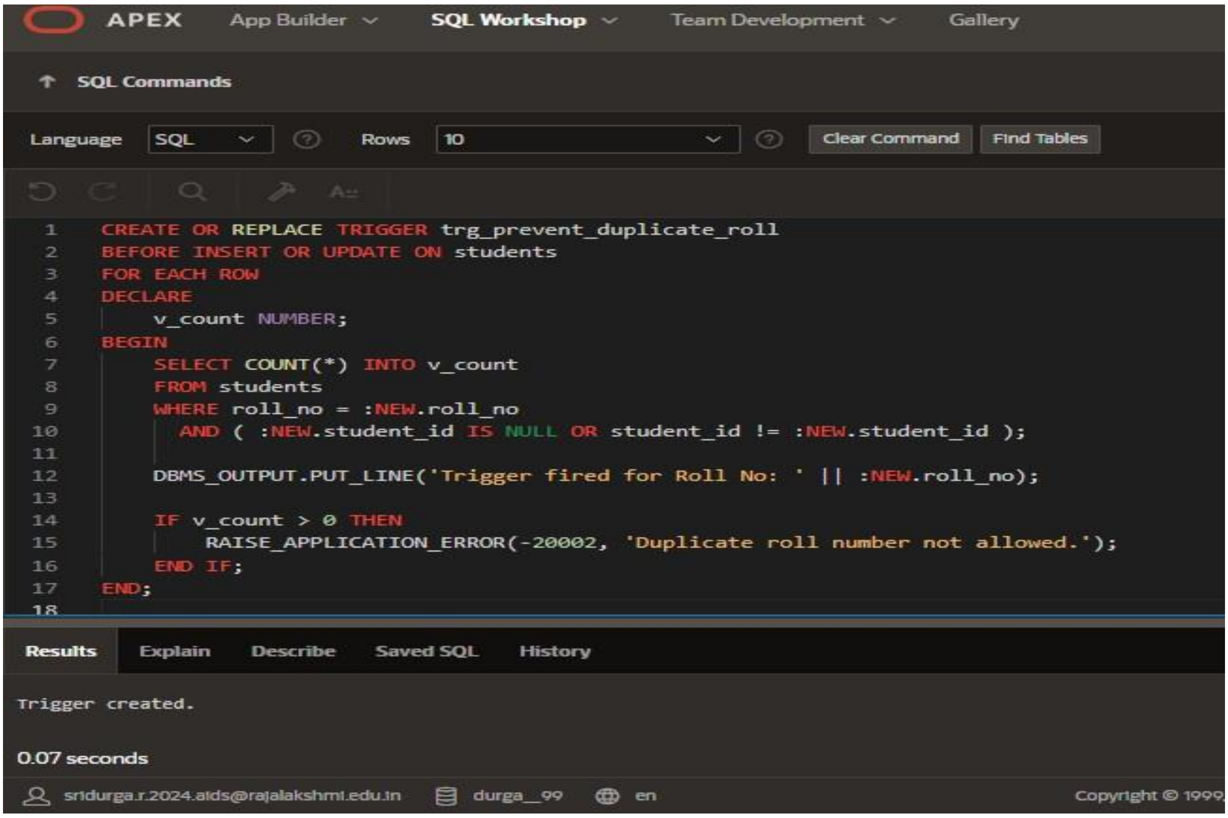
```
Trigger fired for Department ID: 1
ORA-20001: Cannot delete department; employees exist.
ORA-06512: at "WKSP_DURGA99.TRG_PREVENT_PARENT_DELETE", line 11
ORA-04088: error during execution of trigger 'WKSP_DURGA99.TRG_PREVENT_PARENT_DELETE'

1 row(s) deleted.
```

0.05 seconds

sridurga.r.2024.aids@rajalakshmi.edu.in durga__99 en Copyright © 1999, 2024,

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.



Language SQL

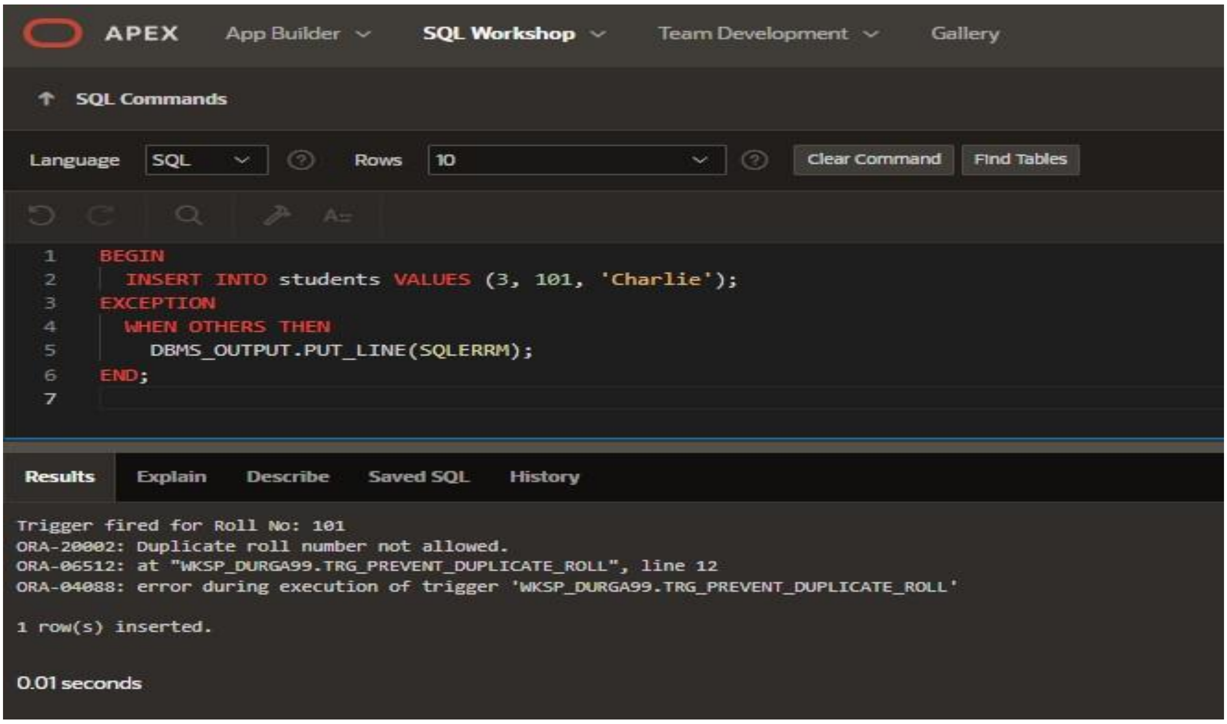
Rows 10

```
CREATE OR REPLACE TRIGGER trg_prevent_duplicate_roll
BEFORE INSERT OR UPDATE ON students
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM students
    WHERE roll_no = :NEW.roll_no
      AND ( :NEW.student_id IS NULL OR student_id != :NEW.student_id );

    DBMS_OUTPUT.PUT_LINE('Trigger fired for Roll No: ' || :NEW.roll_no);

    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Duplicate roll number not allowed.');
    END IF;
END;
```

```
Trigger created.
```

0.07 seconds

sridurga.r.2024.aids@rajalakshmi.edu.in durga__99 en Copyright © 1999



Language SQL

Rows 10

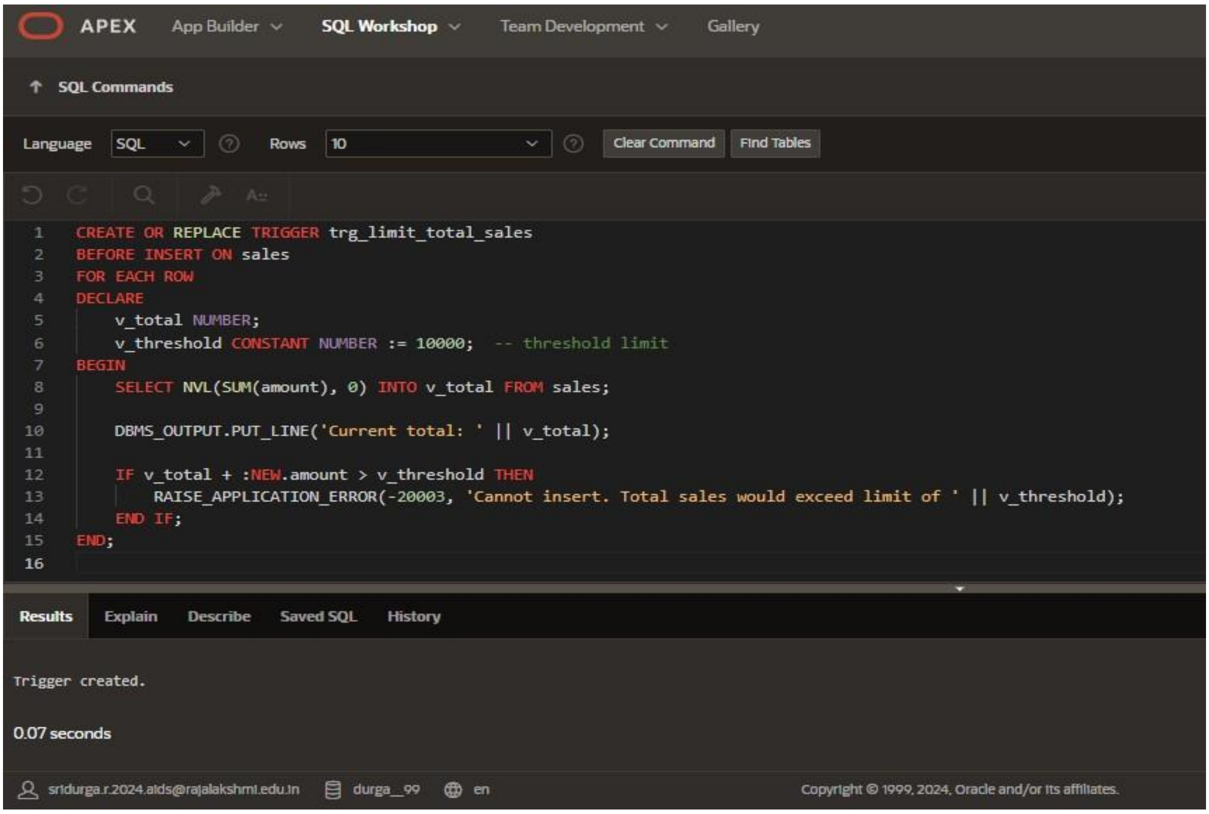
```
BEGIN
  INSERT INTO students VALUES (3, 101, 'Charlie');
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;
```

```
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.



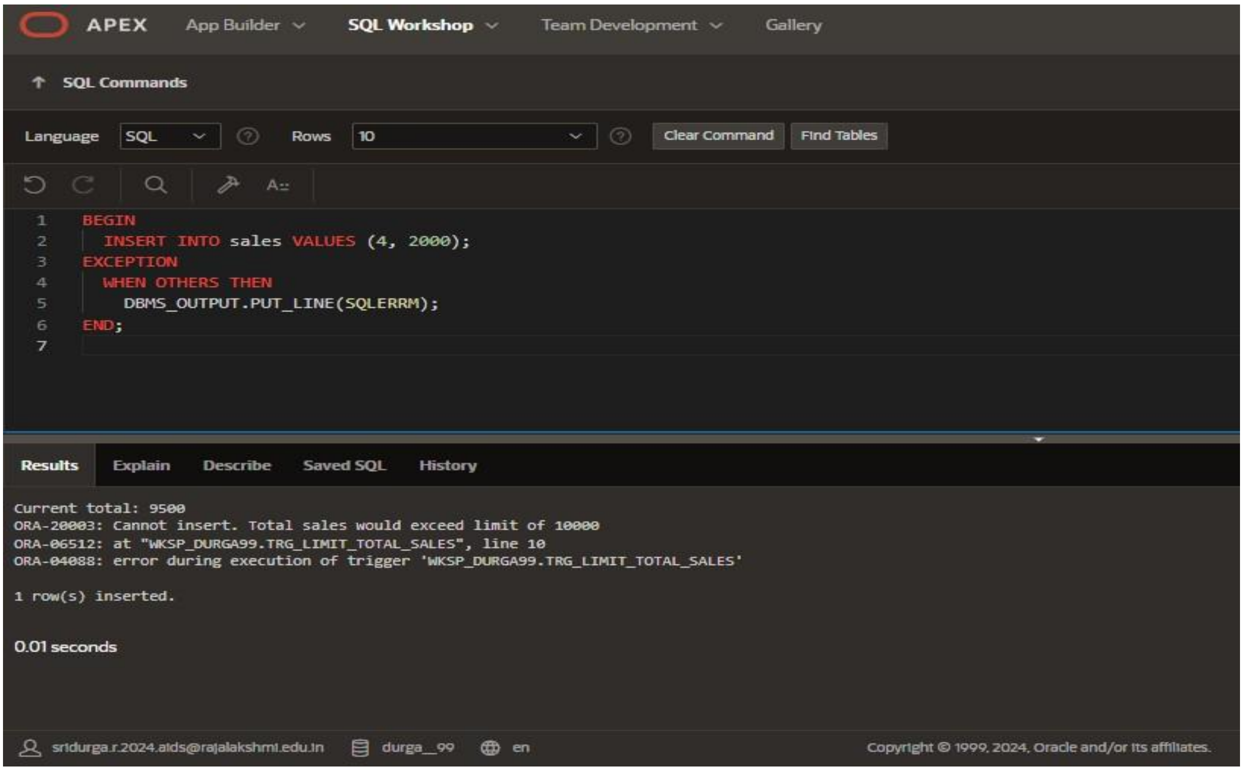
Language SQL

Rows 10

```
CREATE OR REPLACE TRIGGER trg_limit_total_sales
BEFORE INSERT ON sales
FOR EACH ROW
DECLARE
    v_total NUMBER;
    v_threshold CONSTANT NUMBER := 10000;  -- threshold limit
BEGIN
    SELECT NVL(SUM(amount), 0) INTO v_total FROM sales;

    DBMS_OUTPUT.PUT_LINE('Current total: ' || v_total);

    IF v_total + :NEW.amount > v_threshold THEN
        RAISE_APPLICATION_ERROR(-20003, 'Cannot insert. Total sales would exceed limit of ' || v_threshold);
    END IF;
END;
```

```
Trigger created.
```

0.07 seconds



Language SQL

Rows 10

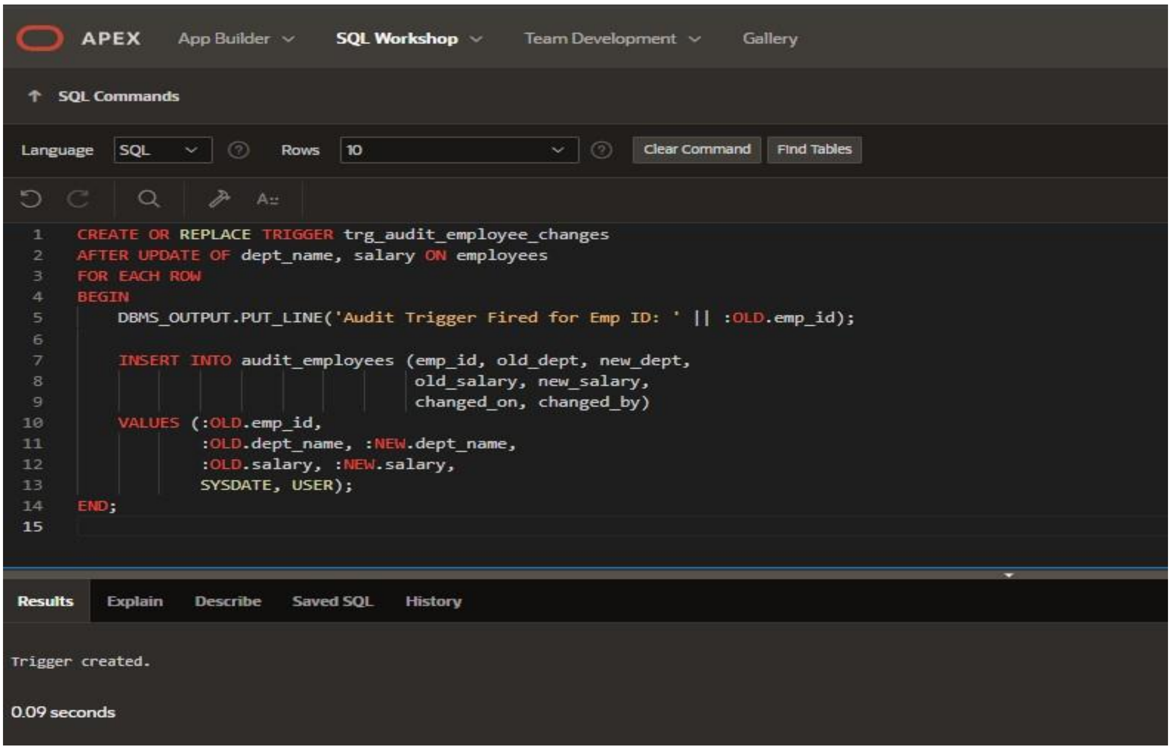
```
BEGIN
  INSERT INTO sales VALUES (4, 2000);
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;
```

```
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

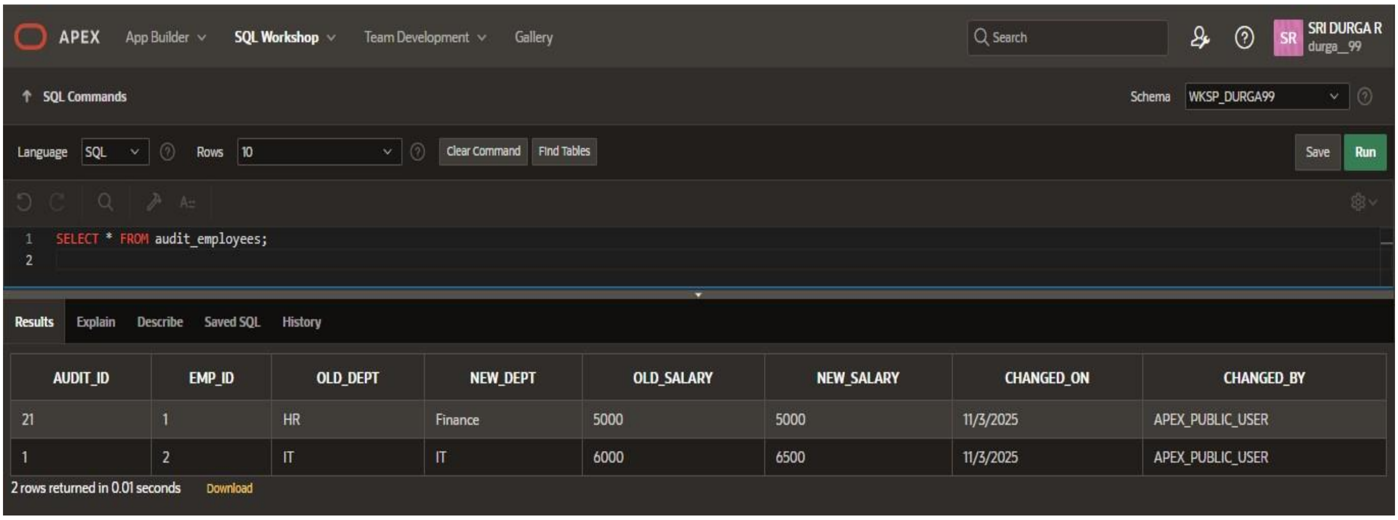
Program 4

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



```sql
CREATE OR REPLACE TRIGGER trg_audit_employee_changes
AFTER UPDATE OF dept_name, salary ON employees
FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('Audit Trigger Fired for Emp ID: ' || :OLD.emp_id);

    INSERT INTO audit_employees (emp_id, old_dept, new_dept,
                                 old_salary, new_salary,
                                 changed_on, changed_by)
    VALUES (:OLD.emp_id,
            :OLD.dept_name, :NEW.dept_name,
            :OLD.salary, :NEW.salary,
            SYSDATE, USER);
END;
```

```
Trigger created.

0.09 seconds
```



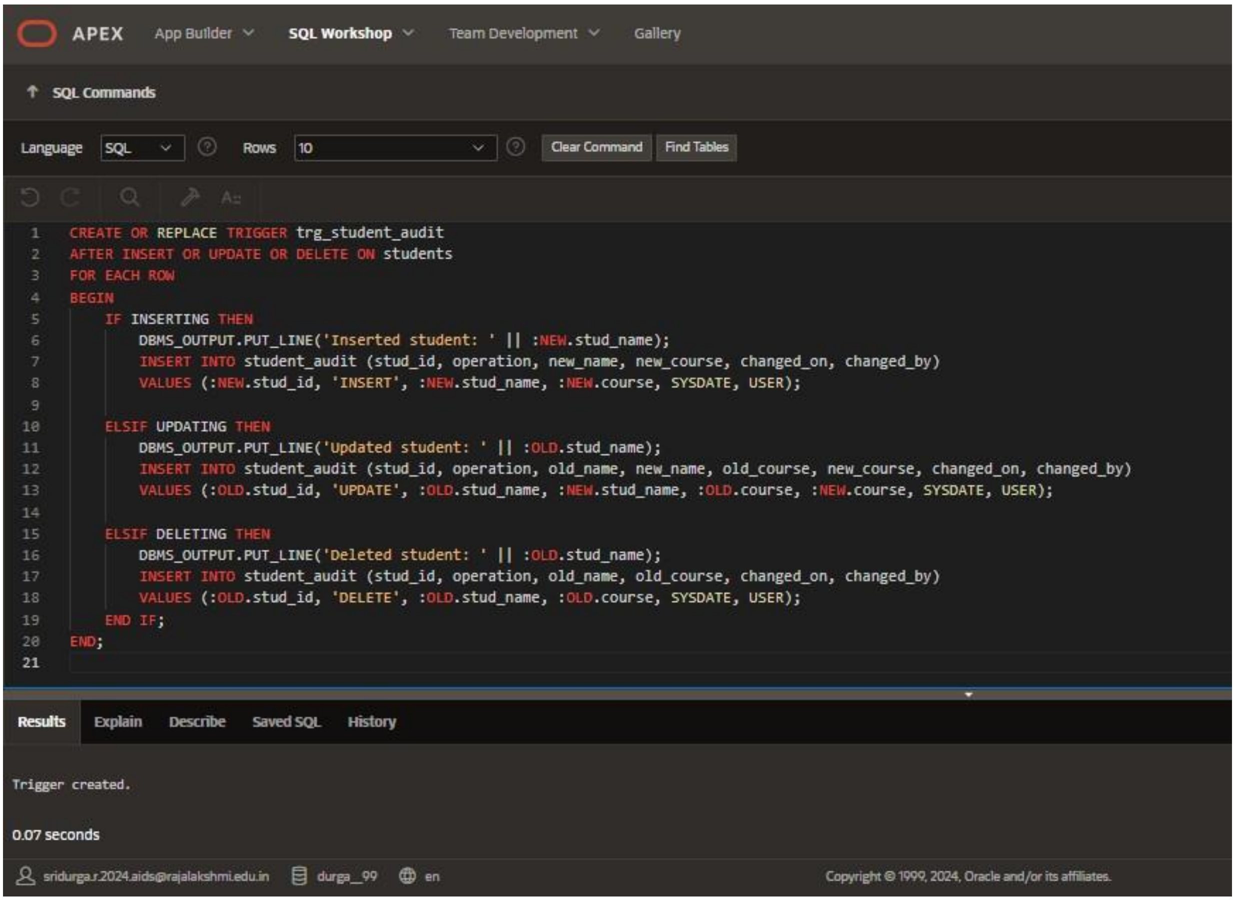
```sql
SELECT * FROM audit_employees;
```

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

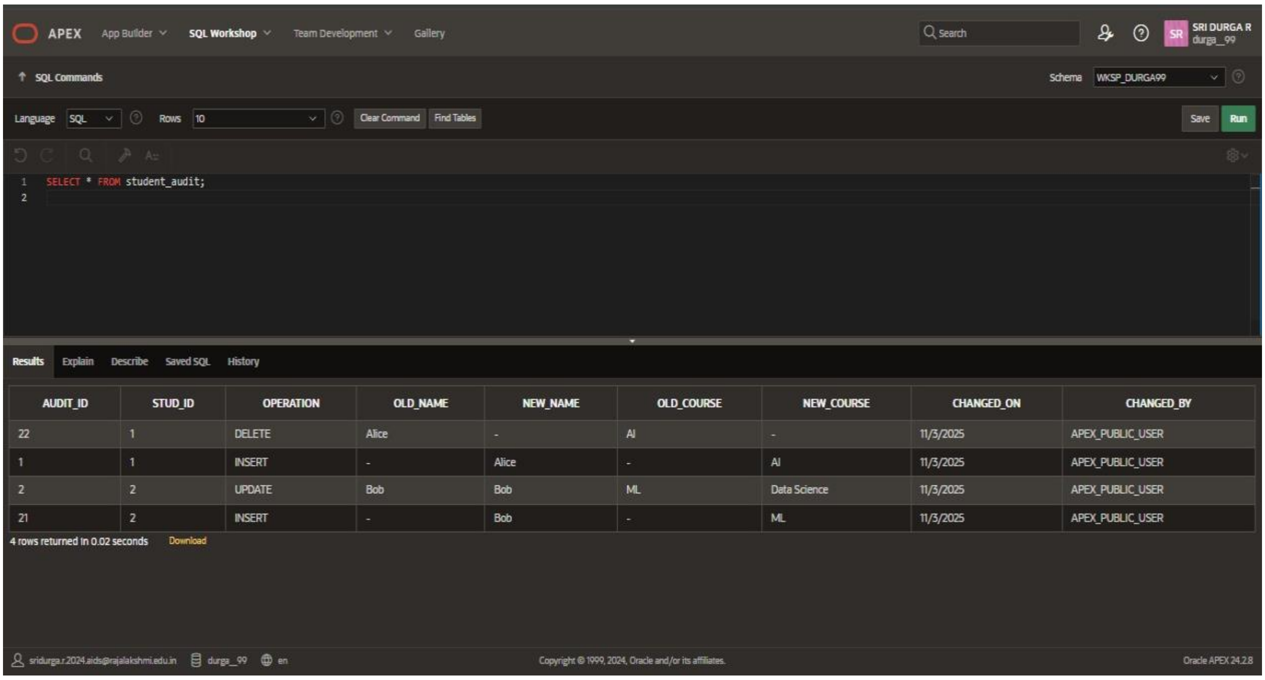
Program 5

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.



```sql
CREATE OR REPLACE TRIGGER trg_student_audit
AFTER INSERT OR UPDATE OR DELETE ON students
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        DBMS_OUTPUT.PUT_LINE('Inserted student: ' || :NEW.stud_name);
        INSERT INTO student_audit (stud_id, operation, new_name, new_course, changed_on, changed_by)
        VALUES (:NEW.stud_id, 'INSERT', :NEW.stud_name, :NEW.course, SYSDATE, USER);

    ELSIF UPDATING THEN
        DBMS_OUTPUT.PUT_LINE('Updated student: ' || :OLD.stud_name);
        INSERT INTO student_audit (stud_id, operation, old_name, new_name, old_course, new_course, changed_on, changed_by)
        VALUES (:OLD.stud_id, 'UPDATE', :OLD.stud_name, :NEW.stud_name, :OLD.course, :NEW.course, SYSDATE, USER);

    ELSIF DELETING THEN
        DBMS_OUTPUT.PUT_LINE('Deleted student: ' || :OLD.stud_name);
        INSERT INTO student_audit (stud_id, operation, old_name, old_course, changed_on, changed_by)
        VALUES (:OLD.stud_id, 'DELETE', :OLD.stud_name, :OLD.course, SYSDATE, USER);
    END IF;
END;
```

```
Trigger created.

0.07 seconds
```



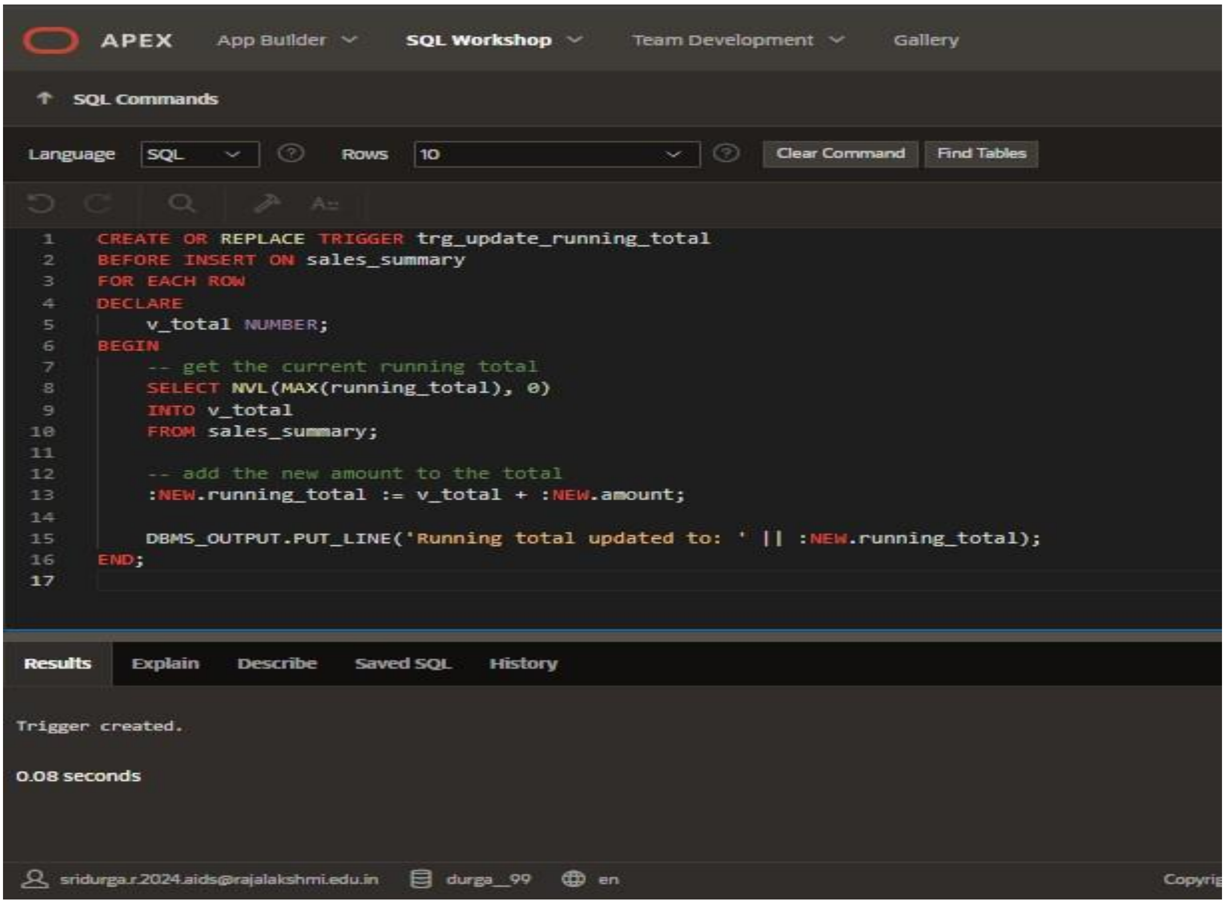
```sql
SELECT * FROM student_audit;
```

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

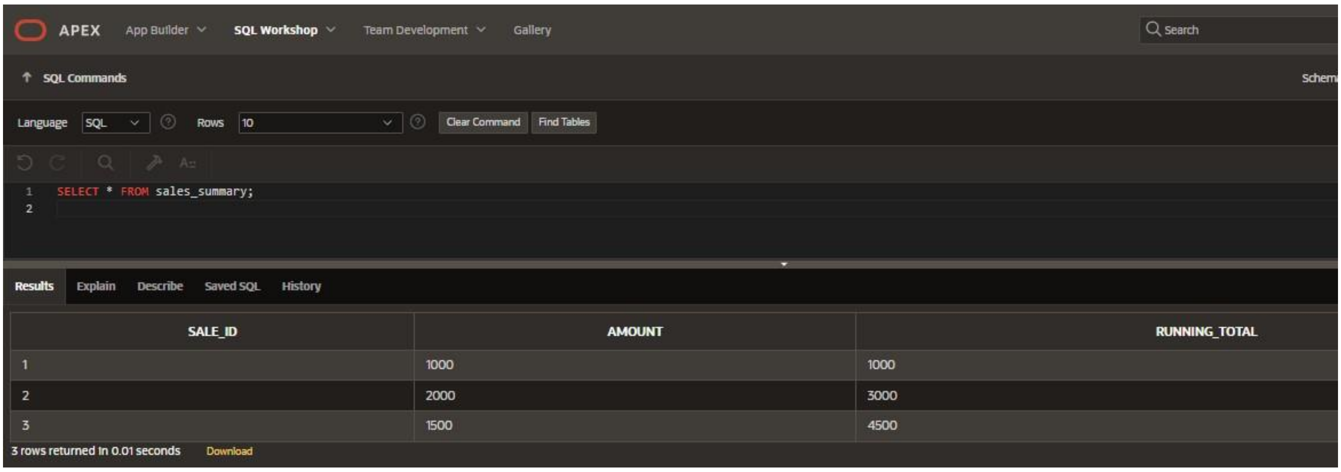
Program 7

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.



```
CREATE OR REPLACE TRIGGER trg_update_running_total
BEFORE INSERT ON sales_summary
FOR EACH ROW
DECLARE
    v_total NUMBER;
BEGIN
    -- get the current running total
    SELECT NVL(MAX(running_total), 0)
    INTO v_total
    FROM sales_summary;

    -- add the new amount to the total
    :NEW.running_total := v_total + :NEW.amount;

    DBMS_OUTPUT.PUT_LINE('Running total updated to: ' || :NEW.running_total);
END;
```

```
Trigger created.

0.08 seconds
```



```
SELECT * FROM sales_summary;
```

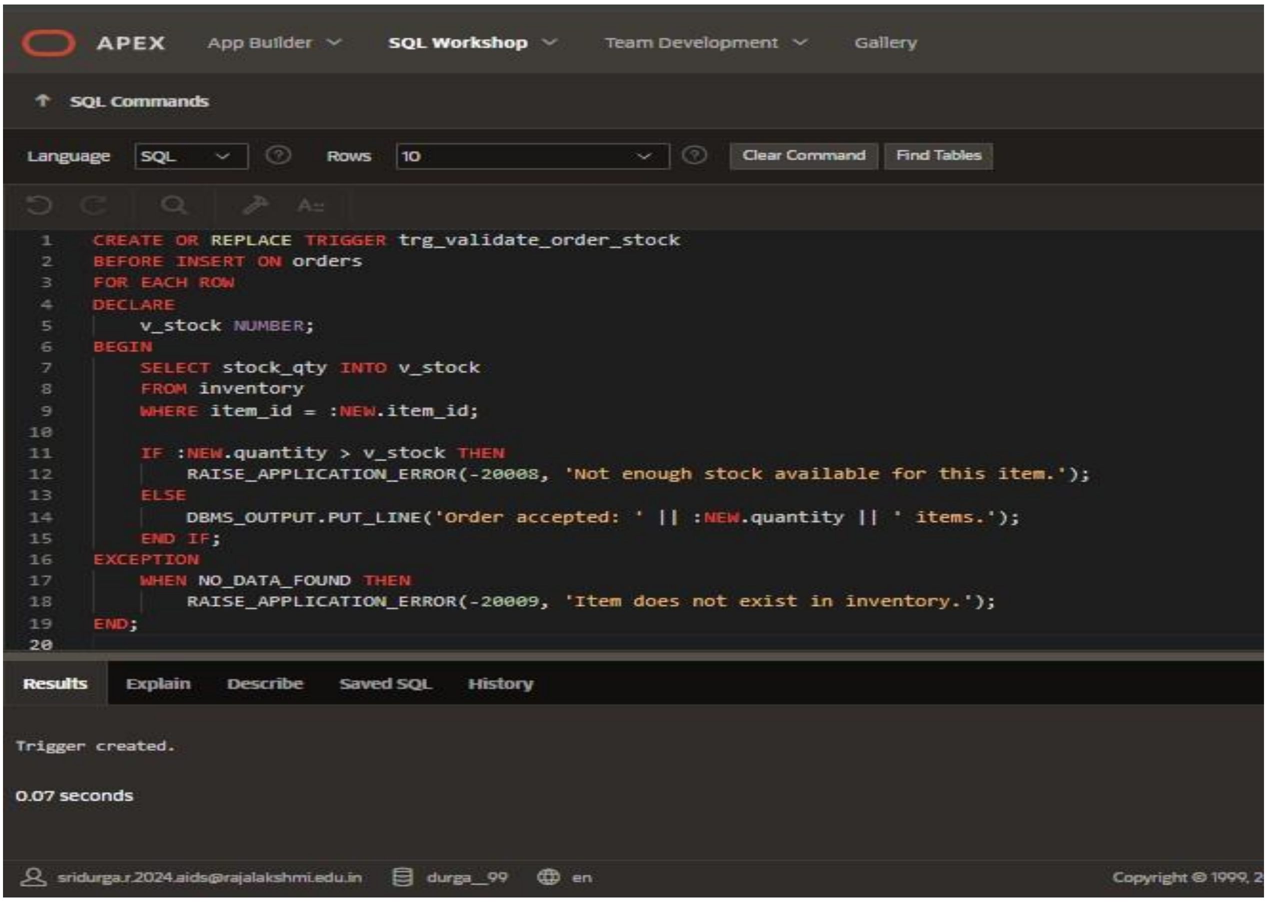
SALE_ID	AMOUNT	RUNNING_TOTAL
1	1000	1000
2	2000	3000
3	1500	4500

3 rows returned in 0.01 seconds

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.



Language SQL

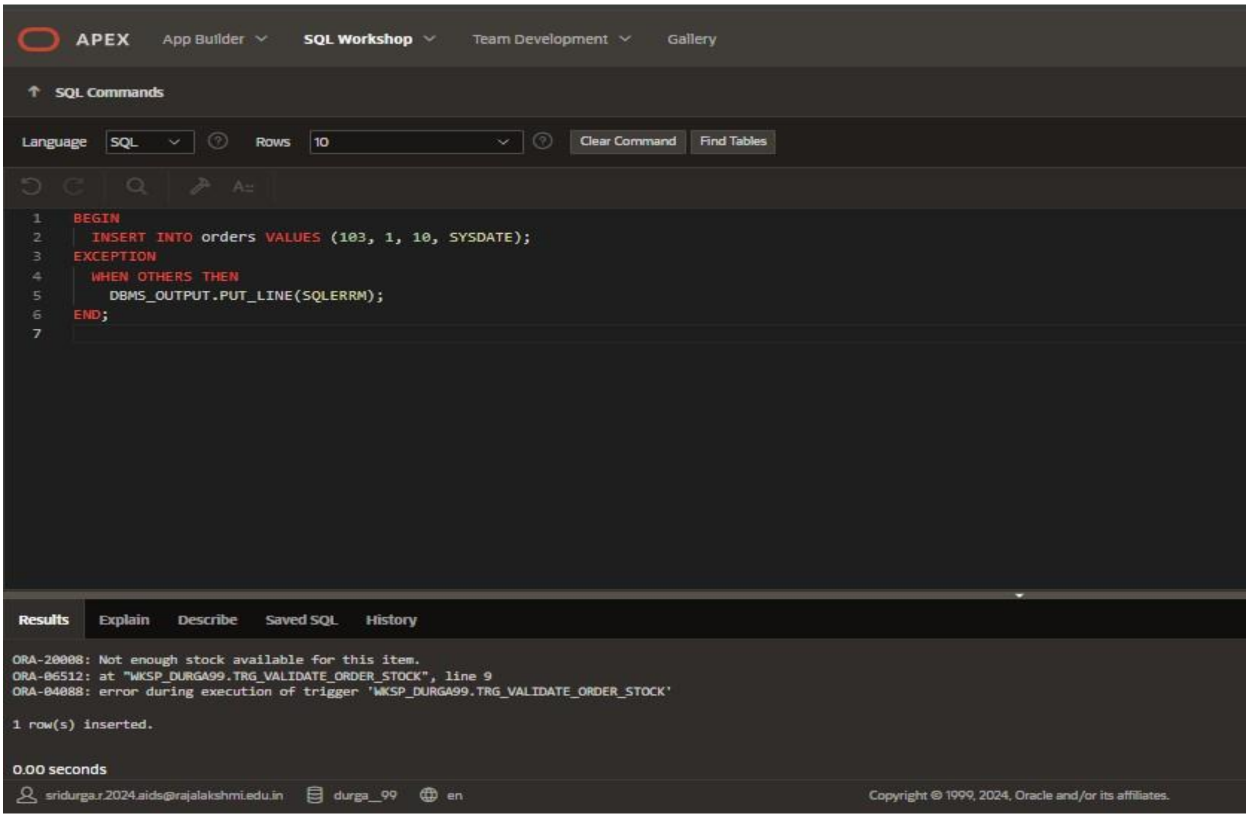
Rows 10

```
CREATE OR REPLACE TRIGGER trg_validate_order_stock
BEFORE INSERT ON orders
FOR EACH ROW
DECLARE
    v_stock NUMBER;
BEGIN
    SELECT stock_qty INTO v_stock
    FROM inventory
    WHERE item_id = :NEW.item_id;

    IF :NEW.quantity > v_stock THEN
        RAISE_APPLICATION_ERROR(-20008, 'Not enough stock available for this item.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Order accepted: ' || :NEW.quantity || ' items.');
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20009, 'Item does not exist in inventory.');
END;
```

```
Trigger created.

0.07 seconds
```

sridurga.r.2024.aids@rajalakshmi.edu.in durga__99 en

Copyright © 1999, 2



Language SQL

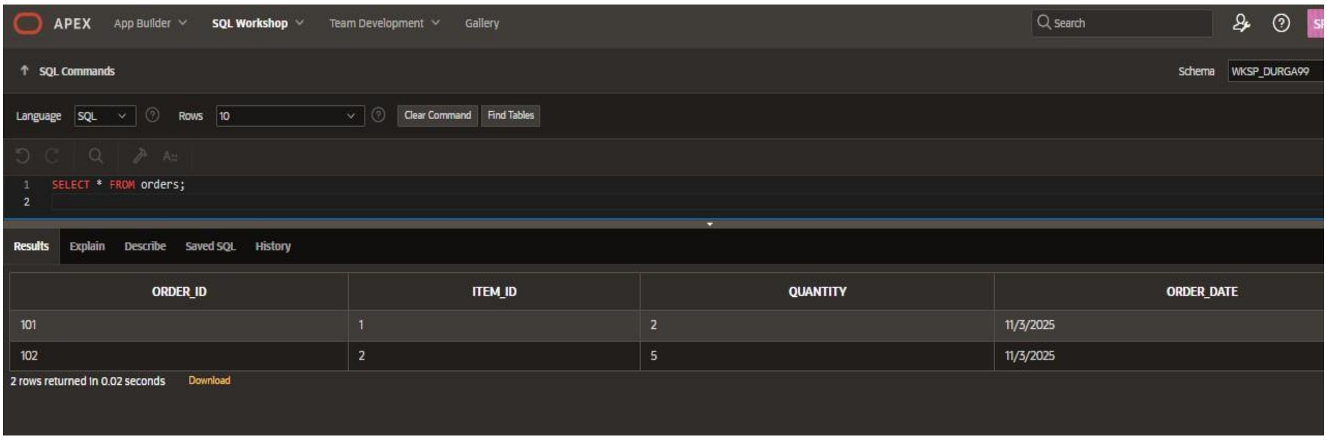
Rows 10

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

```
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.



```
SELECT * FROM orders;
```

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