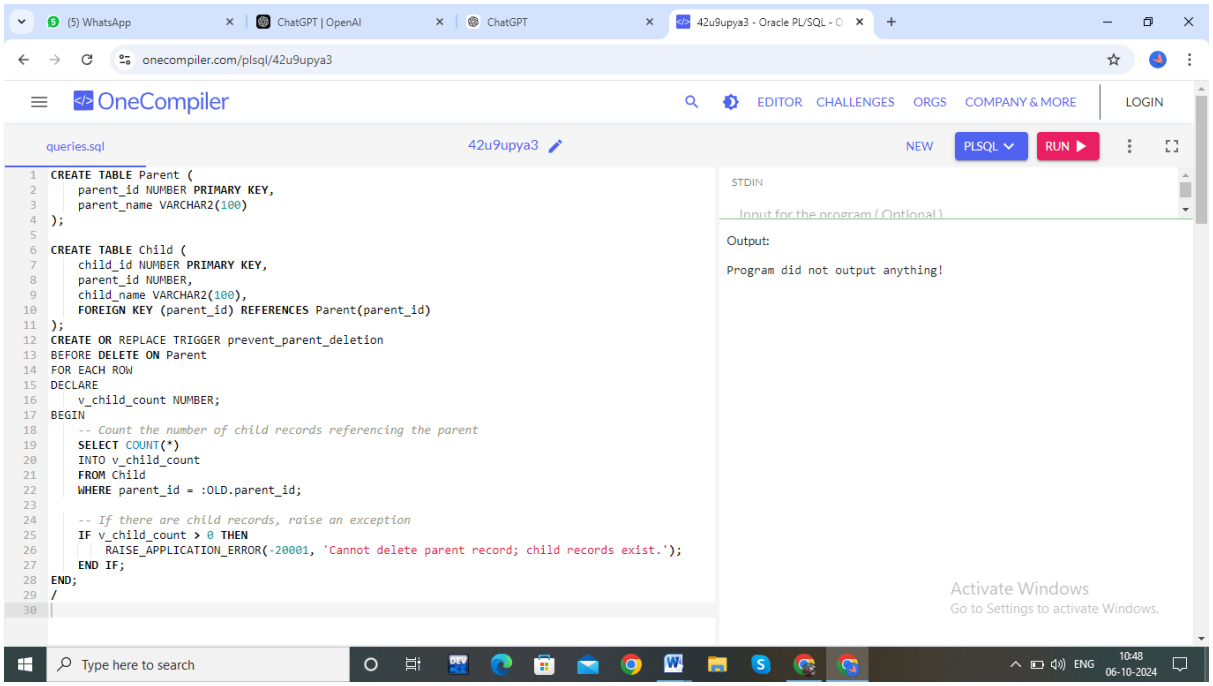
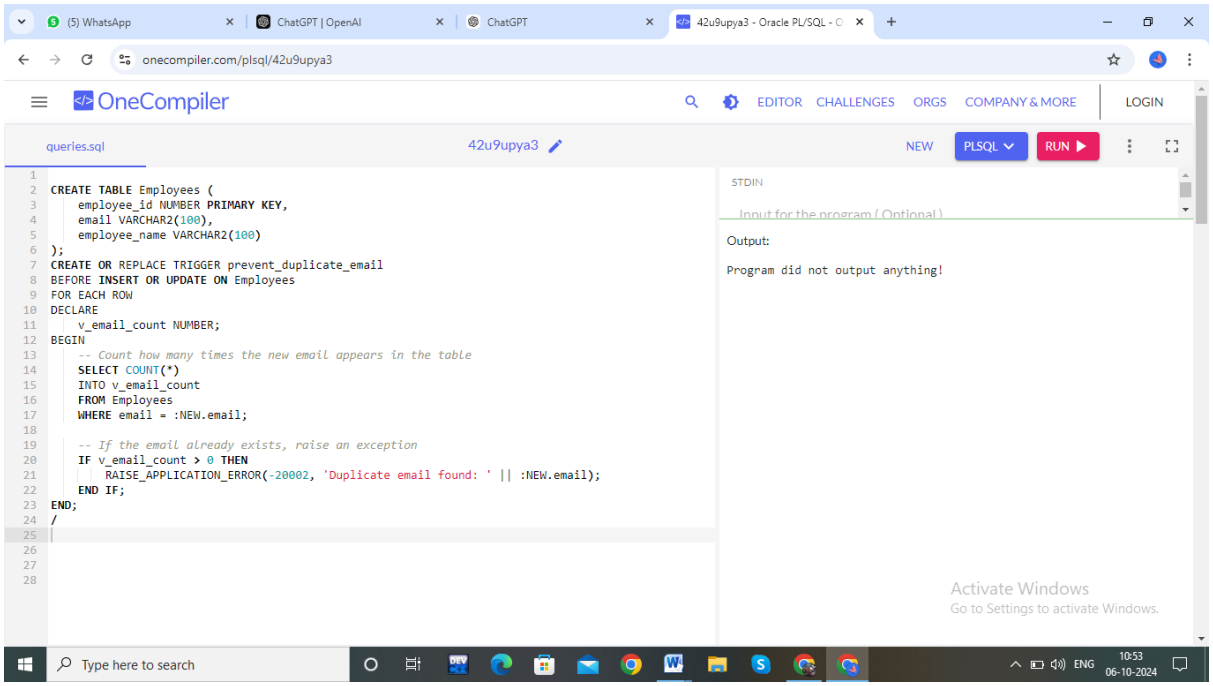
Excerise: 13

231801175

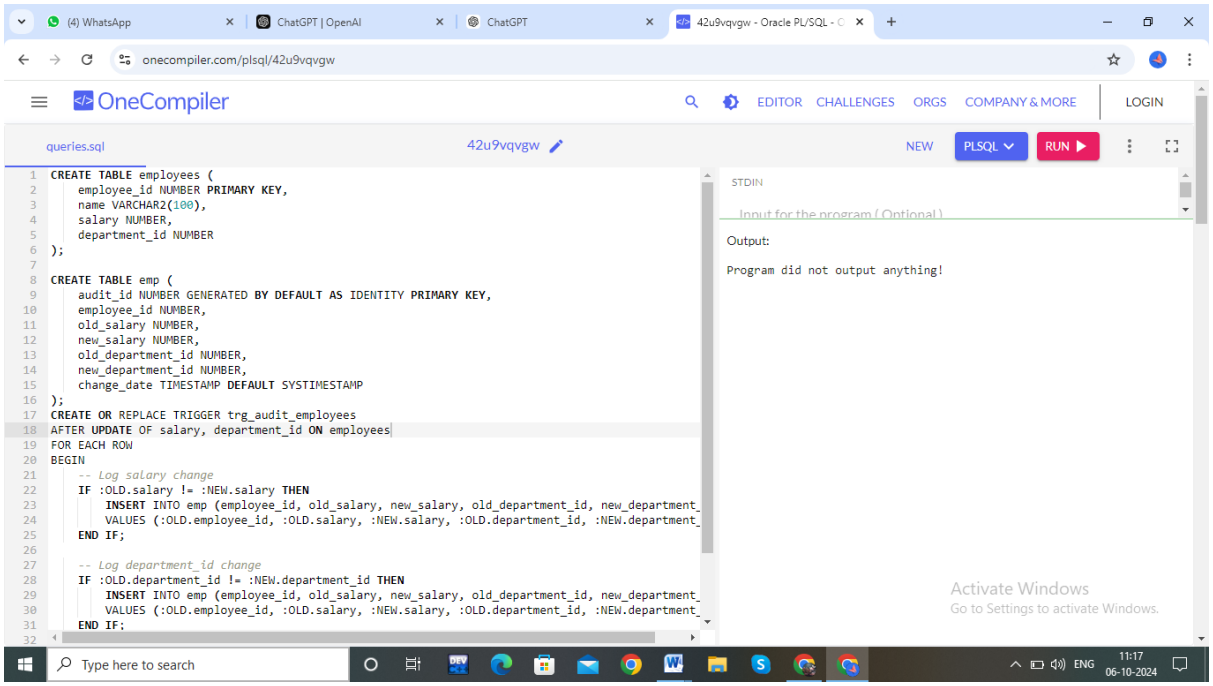


231801175

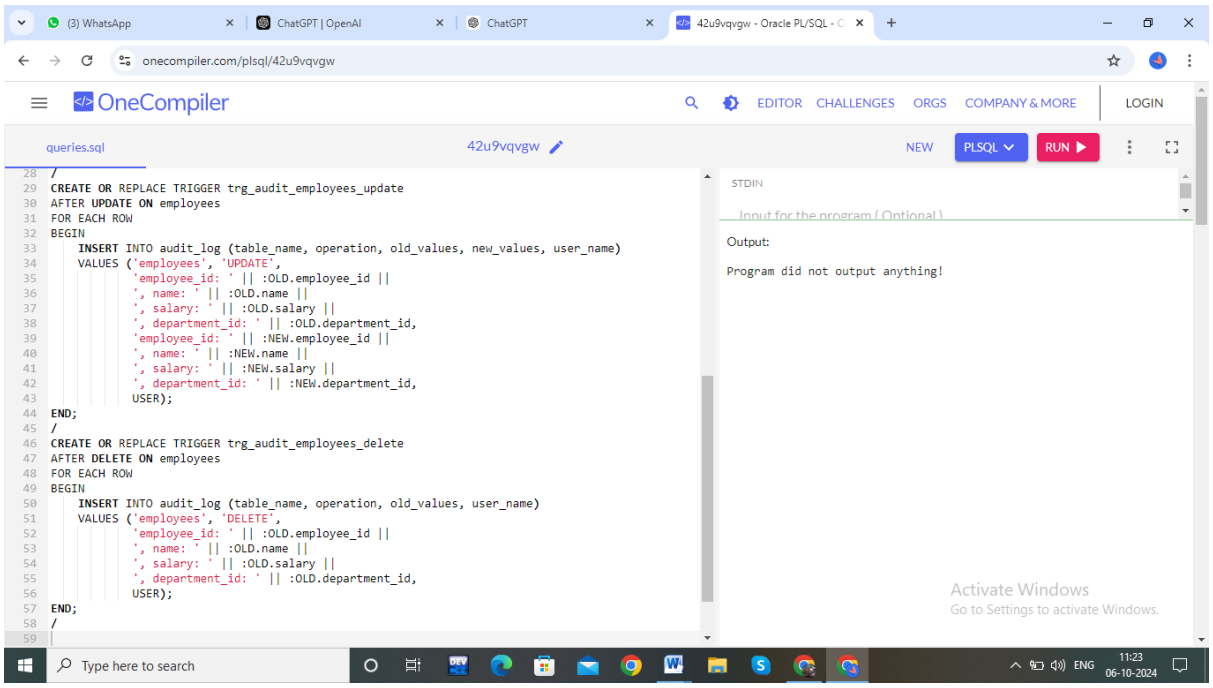


231801175

Excerise: 13



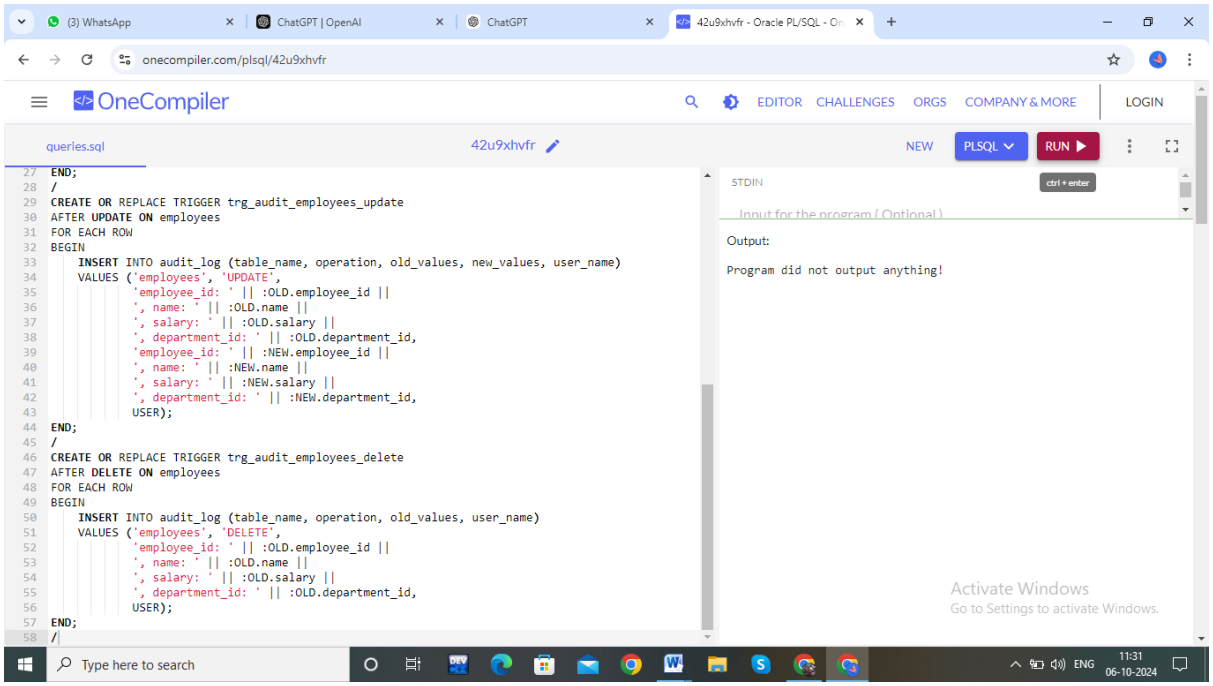
```
CREATE TABLE employees (
    employee_id NUMBER PRIMARY KEY,
    name VARCHAR2(100),
    salary NUMBER,
    department_id NUMBER
);

CREATE TABLE emp (
    audit_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    employee_id NUMBER,
    old_salary NUMBER,
    new_salary NUMBER,
    old_department_id NUMBER,
    new_department_id NUMBER,
    change_date TIMESTAMP DEFAULT SYSTIMESTAMP
);
CREATE OR REPLACE TRIGGER trg_audit_employees
AFTER UPDATE OF salary, department_id ON employees
FOR EACH ROW
BEGIN
    -- Log salary change
    IF :OLD.salary != :NEW.salary THEN
        INSERT INTO emp (employee_id, old_salary, new_salary, old_department_id, new_department_
        VALUES (:OLD.employee_id, :OLD.salary, :NEW.salary, :OLD.department_id, :NEW.department_
    END IF;

    -- Log department_id change
    IF :OLD.department_id != :NEW.department_id THEN
        INSERT INTO emp (employee_id, old_salary, new_salary, old_department_id, new_department_
        VALUES (:OLD.employee_id, :OLD.salary, :NEW.salary, :OLD.department_id, :NEW.department_
    END IF;
```



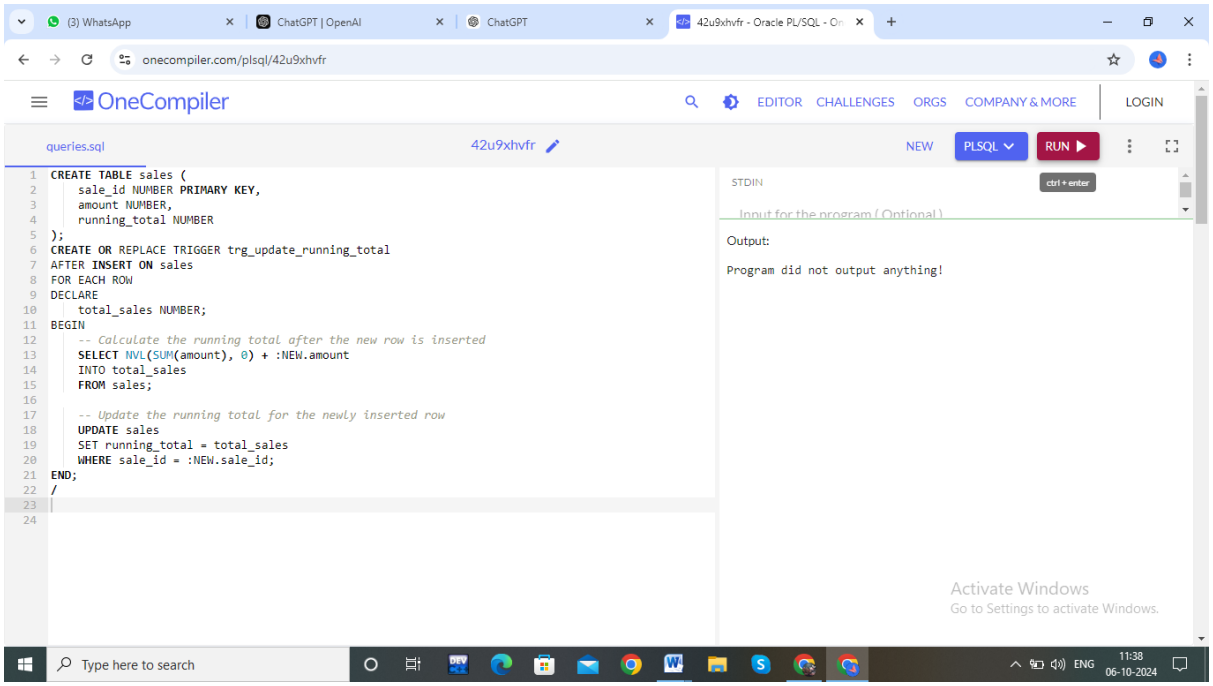
```
/
CREATE OR REPLACE TRIGGER trg_audit_employees_update
AFTER UPDATE ON employees
FOR EACH ROW
BEGIN
    INSERT INTO audit_log (table_name, operation, old_values, new_values, user_name)
    VALUES ('employees', 'UPDATE',
            'employee_id: ' || :OLD.employee_id ||
            ', name: ' || :OLD.name ||
            ', salary: ' || :OLD.salary ||
            ', department_id: ' || :OLD.department_id,
            'employee_id: ' || :NEW.employee_id ||
            ', name: ' || :NEW.name ||
            ', salary: ' || :NEW.salary ||
            ', department_id: ' || :NEW.department_id,
            USER);
END;
/
CREATE OR REPLACE TRIGGER trg_audit_employees_delete
AFTER DELETE ON employees
FOR EACH ROW
BEGIN
    INSERT INTO audit_log (table_name, operation, old_values, user_name)
    VALUES ('employees', 'DELETE',
            'employee_id: ' || :OLD.employee_id ||
            ', name: ' || :OLD.name ||
            ', salary: ' || :OLD.salary ||
            ', department_id: ' || :OLD.department_id,
            USER);
END;
/
```

Excerise: 13

231801175




```
END;
/
CREATE OR REPLACE TRIGGER trg_audit_employees_update
AFTER UPDATE ON employees
FOR EACH ROW
BEGIN
    INSERT INTO audit_log (table_name, operation, old_values, new_values, user_name)
    VALUES ('employees', 'UPDATE',
            'employee_id: ' || :OLD.employee_id ||
            ', name: ' || :OLD.name ||
            ', salary: ' || :OLD.salary ||
            ', department_id: ' || :OLD.department_id,
            'employee_id: ' || :NEW.employee_id ||
            ', name: ' || :NEW.name ||
            ', salary: ' || :NEW.salary ||
            ', department_id: ' || :NEW.department_id,
            USER);
END;
/
CREATE OR REPLACE TRIGGER trg_audit_employees_delete
AFTER DELETE ON employees
FOR EACH ROW
BEGIN
    INSERT INTO audit_log (table_name, operation, old_values, user_name)
    VALUES ('employees', 'DELETE',
            'employee_id: ' || :OLD.employee_id ||
            ', name: ' || :OLD.name ||
            ', salary: ' || :OLD.salary ||
            ', department_id: ' || :OLD.department_id,
            USER);
END;
/
```




```
CREATE TABLE sales (
    sale_id NUMBER PRIMARY KEY,
    amount NUMBER,
    running_total NUMBER
);
CREATE OR REPLACE TRIGGER trg_update_running_total
AFTER INSERT ON sales
FOR EACH ROW
DECLARE
    total_sales NUMBER;
BEGIN
    -- Calculate the running total after the new row is inserted
    SELECT NVL(SUM(amount), 0) + :NEW.amount
    INTO total_sales
    FROM sales;

    -- Update the running total for the newly inserted row
    UPDATE sales
    SET running_total = total_sales
    WHERE sale_id = :NEW.sale_id;
END;
/
```