23K-6005

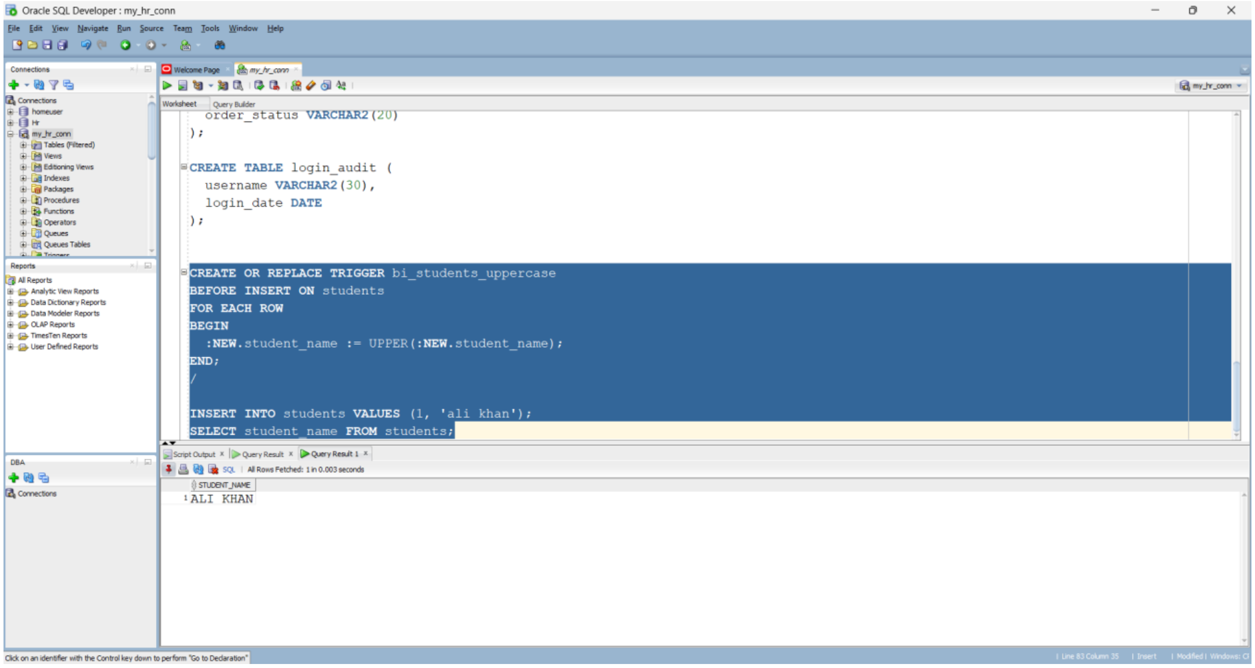
HARIS AHMED

BSAI-5A

DATABASE LAB 05:

LAB TASKS

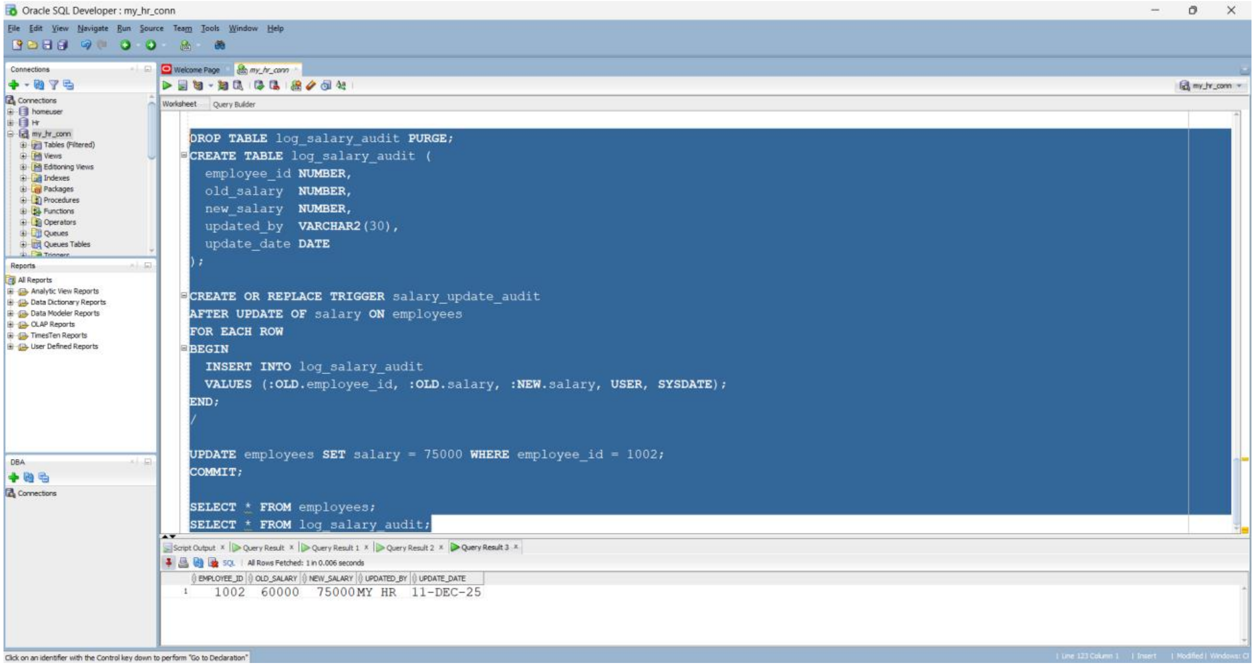
1)



```sql
  order_status VARCHAR2(20)
);

CREATE TABLE login_audit (
  username VARCHAR2(30),
  login_date DATE
);


CREATE OR REPLACE TRIGGER bi_students_uppercase
BEFORE INSERT ON students
FOR EACH ROW
BEGIN
  :NEW.student_name := UPPER(:NEW.student_name);
END;
/

INSERT INTO students VALUES (1, 'ali khan');
SELECT student_name FROM students;
```

```
All Rows Fetched: 1 in 0.003 seconds

  STUDENT_NAME
1 ALI KHAN
```

2)



Oracle SQL Developer : my_hr_conn

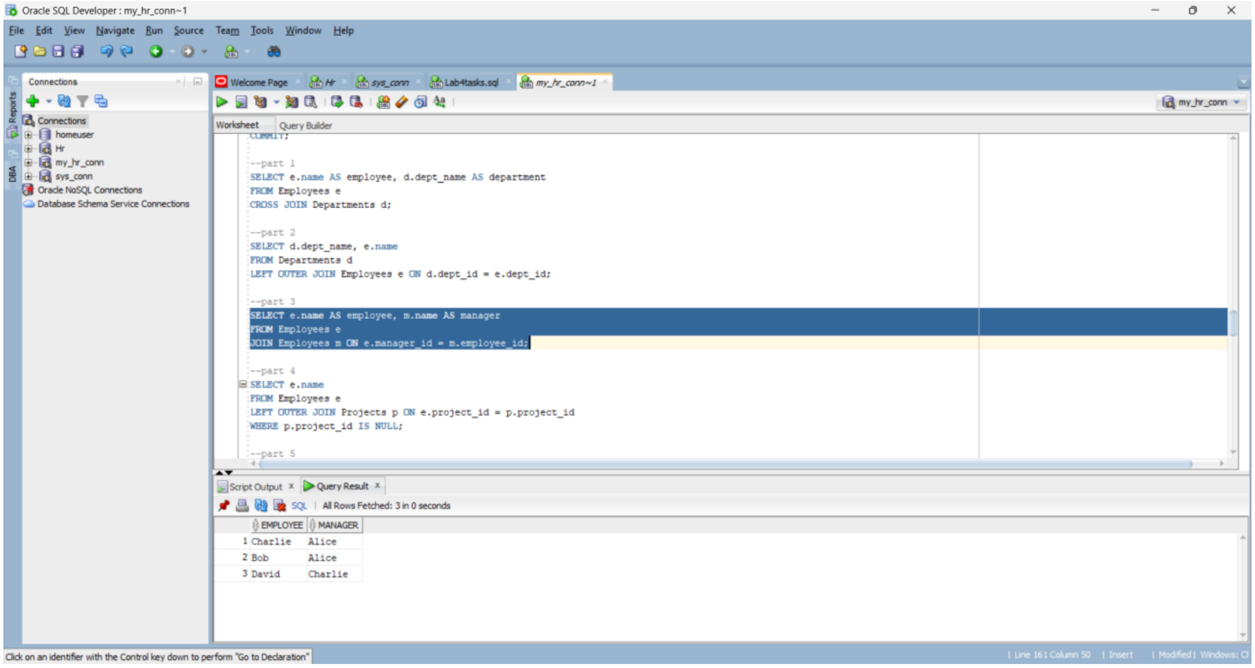
```
DROP TABLE log_salary_audit PURGE;
CREATE TABLE log_salary_audit (
  employee_id NUMBER,
  old_salary  NUMBER,
  new_salary  NUMBER,
  updated_by  VARCHAR2(30),
  update_date DATE
);

CREATE OR REPLACE TRIGGER salary_update_audit
AFTER UPDATE OF salary ON employees
FOR EACH ROW
BEGIN
  INSERT INTO log_salary_audit
  VALUES (:OLD.employee_id, :OLD.salary, :NEW.salary, USER, SYSDATE);
END;
/

UPDATE employees SET salary = 75000 WHERE employee_id = 1002;
COMMIT;

SELECT * FROM employees;
SELECT * FROM log_salary_audit;
```

All Rows Fetched: 1 in 0.006 seconds

```
    EMPLOYEE_ID OLD_SALARY NEW_SALARY UPDATED_BY UPDATE_DATE
1          1002      60000      75000MY HR     11-DEC-25
```

Script Output
Query Result
Query Result 1
Query Result 2
Query Result 3
Click on an identifier with the Control key down to perform "Go to Declaration"

3)



Oracle SQL Developer : my_hr_conn~1

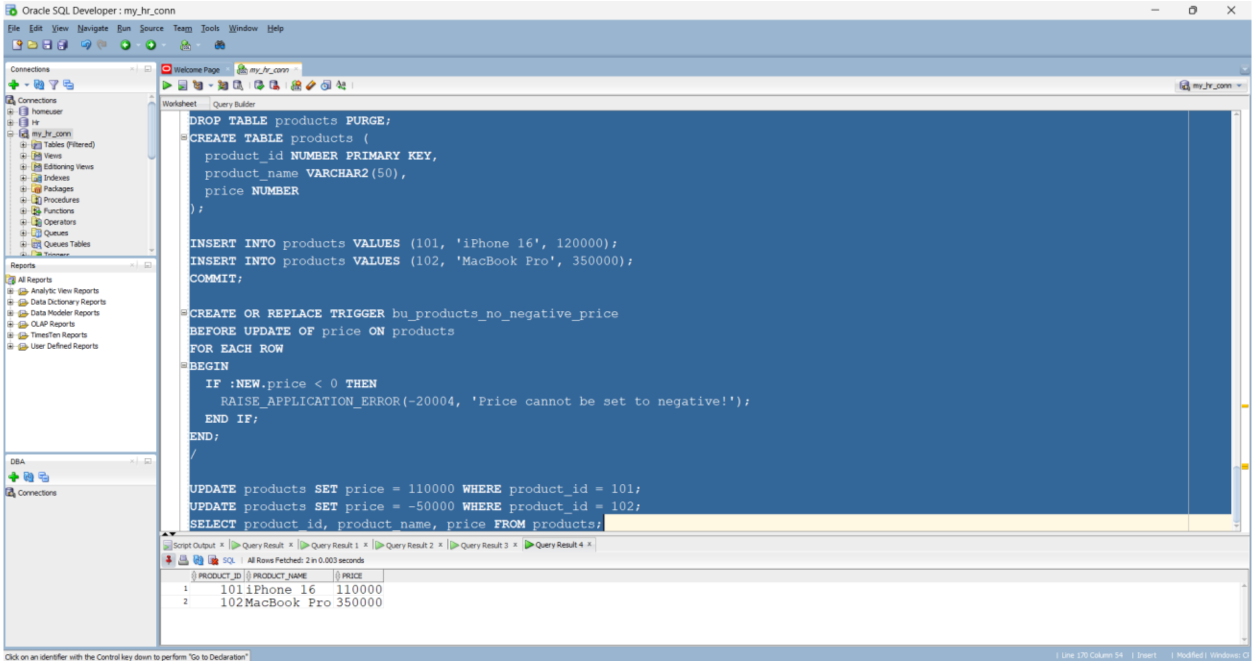
```
COMMIT;

--part 1
SELECT e.name AS employee, d.dept_name AS department
FROM Employees e
CROSS JOIN Departments d;

--part 2
SELECT d.dept_name, e.name
FROM Departments d
LEFT OUTER JOIN Employees e ON d.dept_id = e.dept_id;

--part 3
SELECT e.name AS employee, m.name AS manager
FROM Employees e
JOIN Employees m ON e.manager_id = m.employee_id;

--part 4
SELECT e.name
FROM Employees e
LEFT OUTER JOIN Projects p ON e.project_id = p.project_id
WHERE p.project_id IS NULL;

--part 5
```

All Rows Fetched: 3 in 0 seconds

```
  EMPLOYEE MANAGER
1 Charlie  Alice
2 Bob      Alice
3 David    Charlie
```

Welcome Page
Hr
sys_conn
Lab4tasks.sql
my_hr_conn~1
Script Output
Query Result
Click on an identifier with the Control key down to perform "Go to Declaration"

4)



```sql
DROP TABLE products PURGE;
CREATE TABLE products (
  product_id NUMBER PRIMARY KEY,
  product_name VARCHAR2(50),
  price NUMBER
);

INSERT INTO products VALUES (101, 'iPhone 16', 120000);
INSERT INTO products VALUES (102, 'MacBook Pro', 350000);
COMMIT;

CREATE OR REPLACE TRIGGER bu_products_no_negative_price
BEFORE UPDATE OF price ON products
FOR EACH ROW
BEGIN
  IF :NEW.price < 0 THEN
    RAISE_APPLICATION_ERROR(-20004, 'Price cannot be set to negative!');
  END IF;
END;
/

UPDATE products SET price = 110000 WHERE product_id = 101;
UPDATE products SET price = -50000 WHERE product_id = 102;
SELECT product_id, product_name, price FROM products;
```

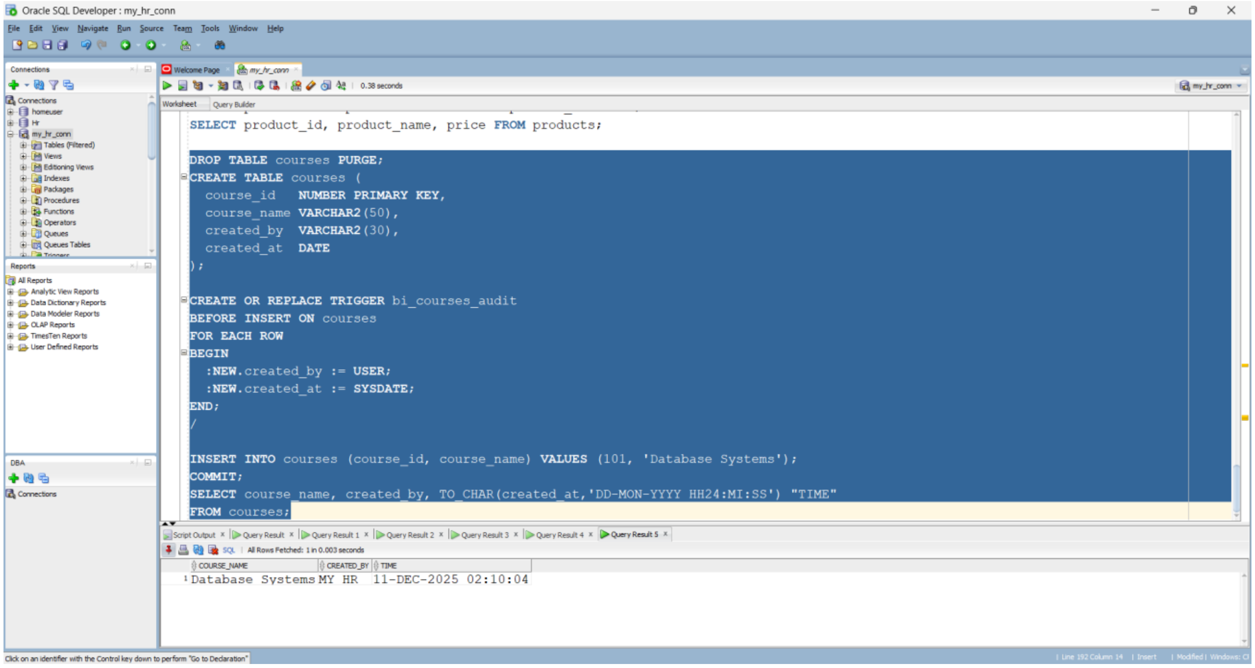
All Rows Fetched: 2 in 0.003 seconds

```
   PRODUCT_ID PRODUCT_NAME PRICE
1         101 iPhone 16    110000
2         102 MacBook Pro  350000
```

Click on an identifier with the Control key down to perform "Go to Declaration"

Line 170 Column 54 | Insert | Modified | Windows: CR

5)



```sql
SELECT product_id, product_name, price FROM products;

DROP TABLE courses PURGE;
CREATE TABLE courses (
  course_id   NUMBER PRIMARY KEY,
  course_name VARCHAR2(50),
  created_by  VARCHAR2(30),
  created_at  DATE
);

CREATE OR REPLACE TRIGGER bi_courses_audit
BEFORE INSERT ON courses
FOR EACH ROW
BEGIN
  :NEW.created_by := USER;
  :NEW.created_at := SYSDATE;
END;
/

INSERT INTO courses (course_id, course_name) VALUES (101, 'Database Systems');
COMMIT;
SELECT course_name, created_by, TO_CHAR(created_at,'DD-MON-YYYY HH24:MI:SS') "TIME"
FROM courses;
```

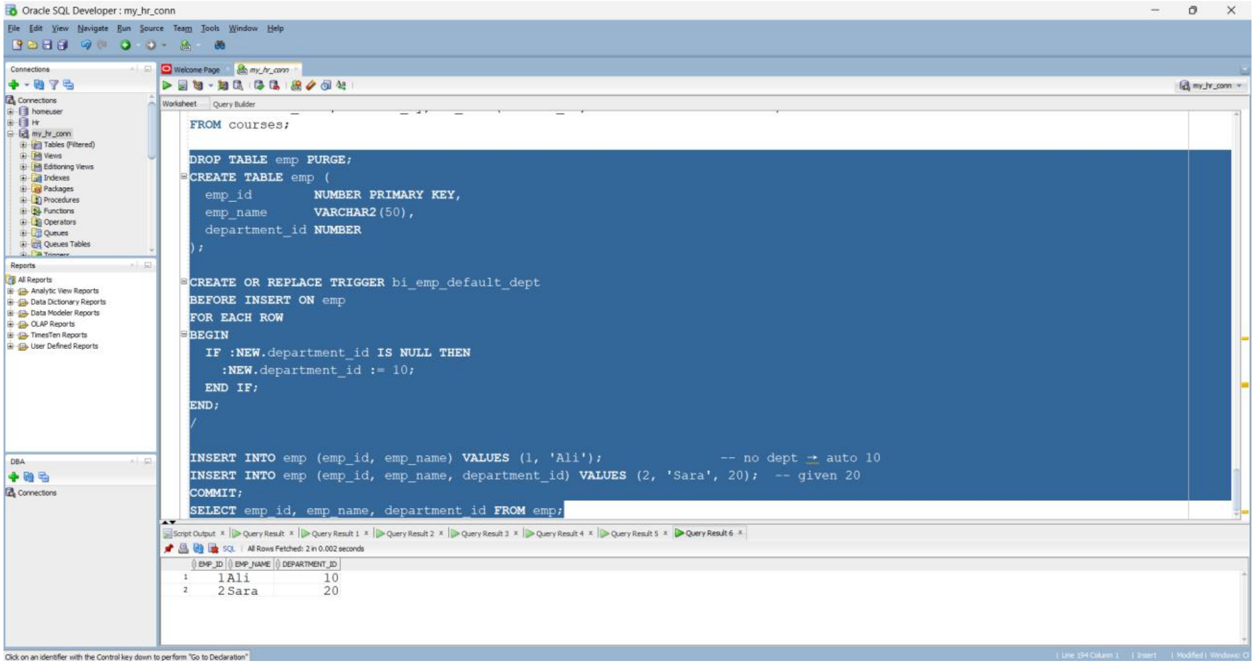
All Rows Fetched: 1 in 0.003 seconds

```
  COURSE_NAME      CREATED_BY TIME
1 Database Systems MY HR      11-DEC-2025 02:10:04
```

Click on an identifier with the Control key down to perform "Go to Declaration"

Line 192 Column 14 | Insert | Modified | Windows: CR

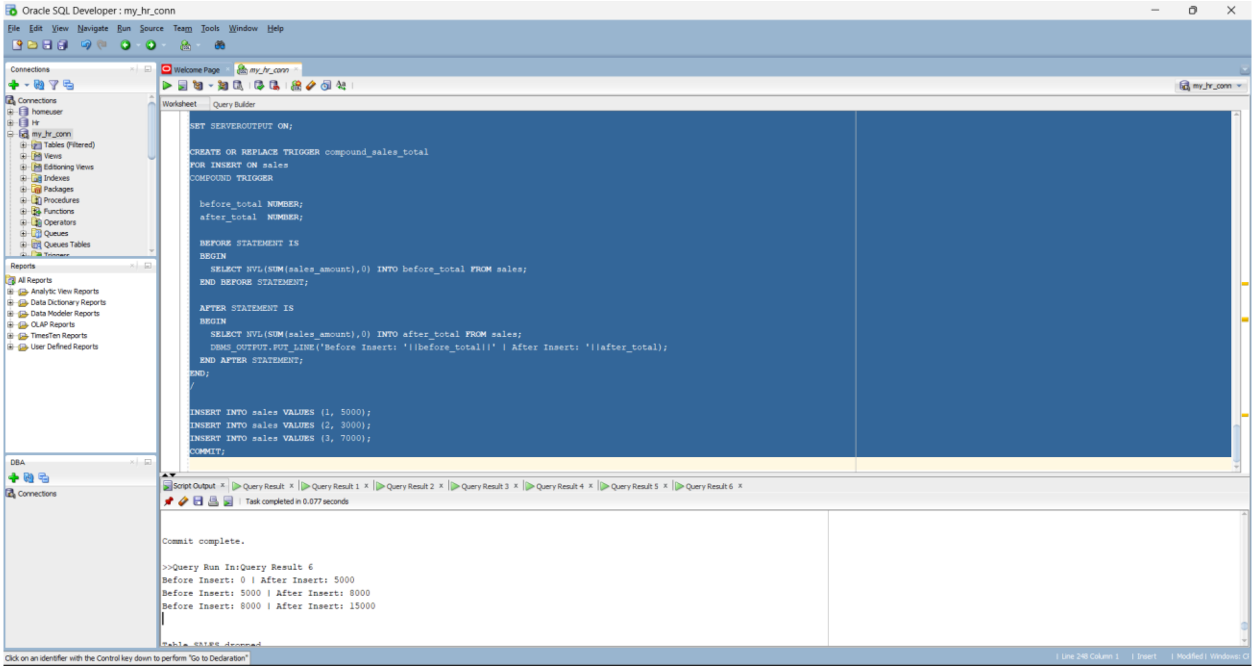
6)



```sql
FROM courses;

DROP TABLE emp PURGE;
CREATE TABLE emp (
  emp_id        NUMBER PRIMARY KEY,
  emp_name      VARCHAR2(50),
  department_id NUMBER
);

CREATE OR REPLACE TRIGGER bi_emp_default_dept
BEFORE INSERT ON emp
FOR EACH ROW
BEGIN
  IF :NEW.department_id IS NULL THEN
    :NEW.department_id := 10;
  END IF;
END;
/

INSERT INTO emp (emp_id, emp_name) VALUES (1, 'Ali');                -- no dept → auto 10
INSERT INTO emp (emp_id, emp_name, department_id) VALUES (2, 'Sara', 20);  -- given 20
COMMIT;
SELECT emp_id, emp_name, department_id FROM emp;
```

	EMP_ID	EMP_NAME	DEPARTMENT_ID
1	1	Ali	10
2	2	Sara	20

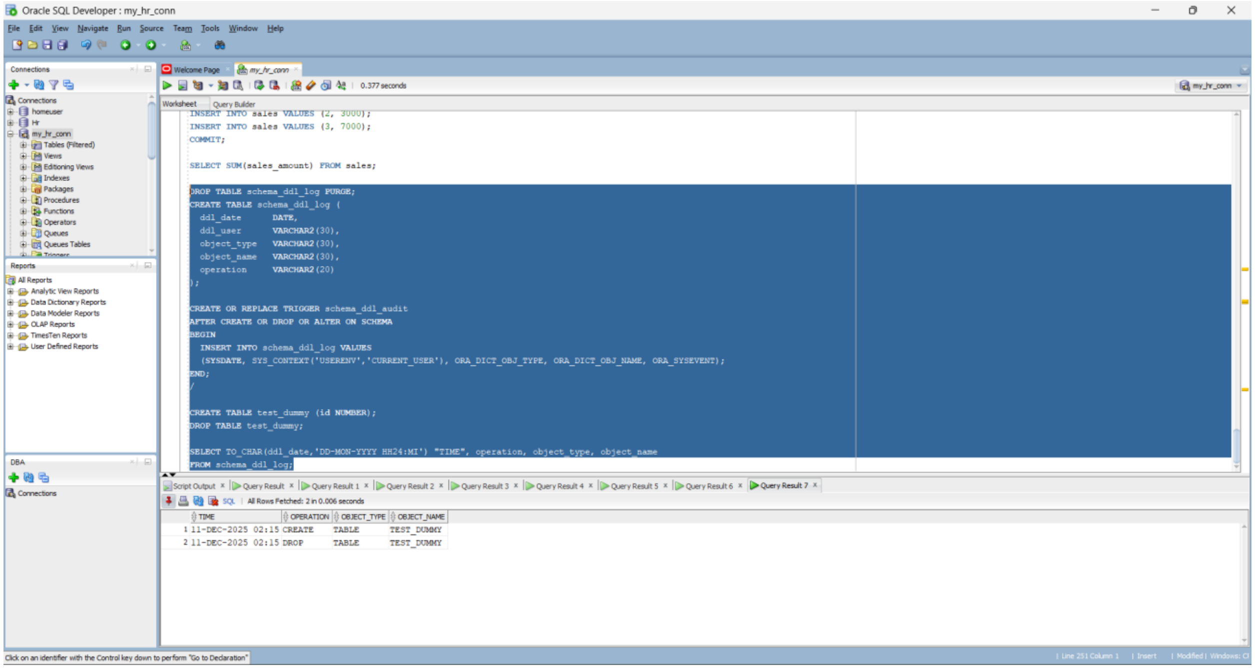
7)



```sql
SET SERVEROUTPUT ON;

CREATE OR REPLACE TRIGGER compound_sales_total
FOR INSERT ON sales
COMPOUND TRIGGER

  before_total NUMBER;
  after_total  NUMBER;

  BEFORE STATEMENT IS
  BEGIN
    SELECT NVL(SUM(sales_amount),0) INTO before_total FROM sales;
  END BEFORE STATEMENT;

  AFTER STATEMENT IS
  BEGIN
    SELECT NVL(SUM(sales_amount),0) INTO after_total FROM sales;
    DBMS_OUTPUT.PUT_LINE('Before Insert: '||before_total||' | After Insert: '||after_total);
  END AFTER STATEMENT;
END;
/

INSERT INTO sales VALUES (1, 5000);
INSERT INTO sales VALUES (2, 3000);
INSERT INTO sales VALUES (3, 7000);
COMMIT;
```

```
Commit complete.

>>Query Run In:Query Result 6
Before Insert: 0 | After Insert: 5000
Before Insert: 5000 | After Insert: 8000
Before Insert: 8000 | After Insert: 15000


Table SALES dropped
```

8)

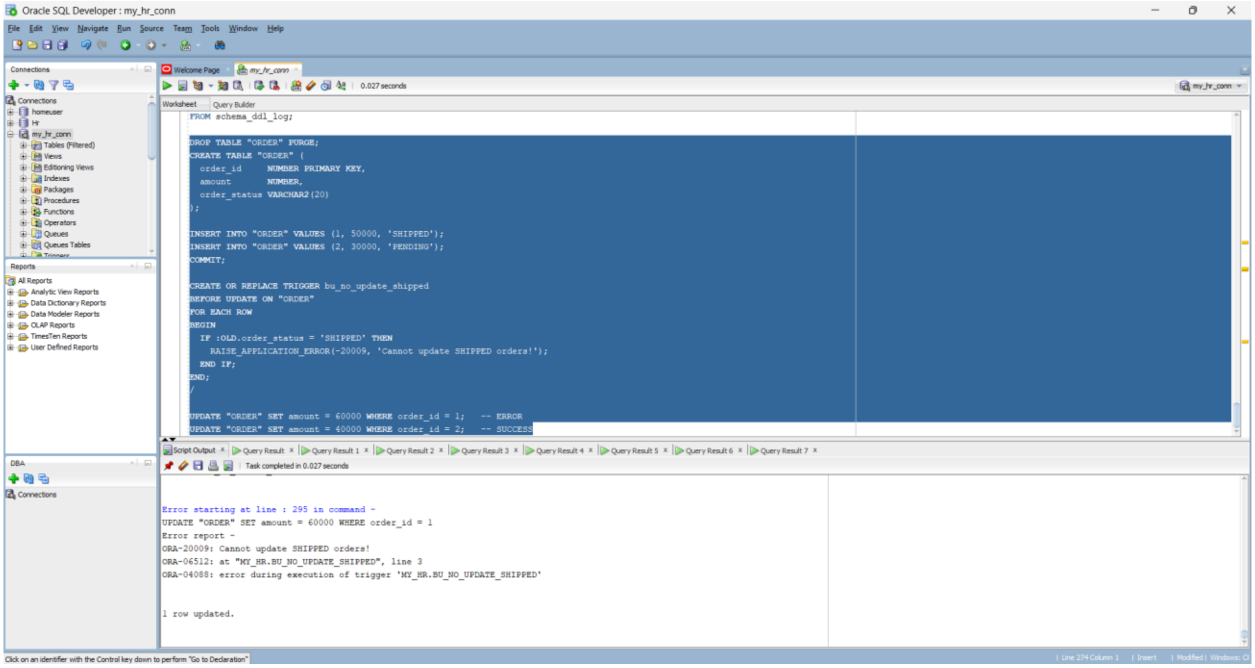


```sql
INSERT INTO sales VALUES (2, 3000);
INSERT INTO sales VALUES (3, 7000);
COMMIT;

SELECT SUM(sales_amount) FROM sales;

DROP TABLE schema_ddl_log PURGE;
CREATE TABLE schema_ddl_log (
  ddl_date      DATE,
  ddl_user      VARCHAR2(30),
  object_type   VARCHAR2(30),
  object_name   VARCHAR2(30),
  operation     VARCHAR2(20)
);

CREATE OR REPLACE TRIGGER schema_ddl_audit
AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
  INSERT INTO schema_ddl_log VALUES
  (SYSDATE, SYS_CONTEXT('USERENV','CURRENT_USER'), ORA_DICT_OBJ_TYPE, ORA_DICT_OBJ_NAME, ORA_SYSEVENT);
END;
/

CREATE TABLE test_dummy (id NUMBER);
DROP TABLE test_dummy;

SELECT TO_CHAR(ddl_date,'DD-MON-YYYY HH24:MI') "TIME", operation, object_type, object_name
FROM schema_ddl_log;
```

All Rows Fetched: 2 in 0.006 seconds

	TIME	OPERATION	OBJECT_TYPE	OBJECT_NAME
1	11-DEC-2025 02:15	CREATE	TABLE	TEST_DUMMY
2	11-DEC-2025 02:15	DROP	TABLE	TEST_DUMMY

9)

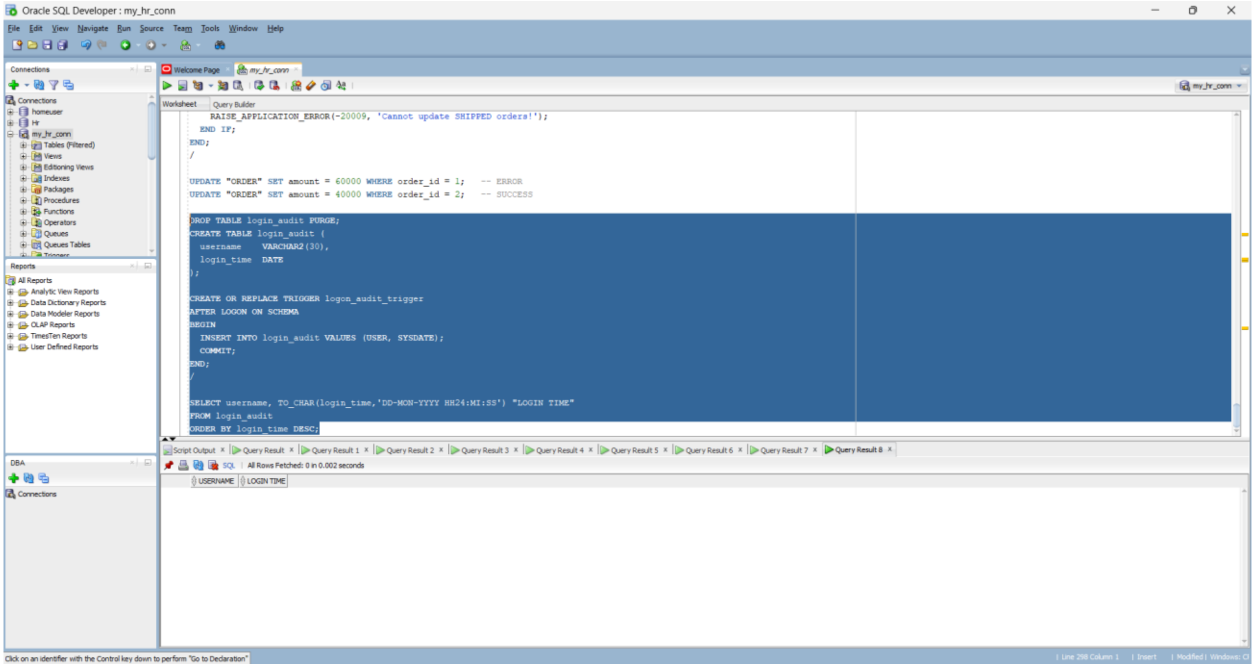


```sql
FROM schema_ddl_log;

DROP TABLE "ORDER" PURGE;
CREATE TABLE "ORDER" (
  order_id     NUMBER PRIMARY KEY,
  amount       NUMBER,
  order_status VARCHAR2(20)
);

INSERT INTO "ORDER" VALUES (1, 50000, 'SHIPPED');
INSERT INTO "ORDER" VALUES (2, 30000, 'PENDING');
COMMIT;

CREATE OR REPLACE TRIGGER bu_no_update_shipped
BEFORE UPDATE ON "ORDER"
FOR EACH ROW
BEGIN
  IF :OLD.order_status = 'SHIPPED' THEN
    RAISE_APPLICATION_ERROR(-20009, 'Cannot update SHIPPED orders!');
  END IF;
END;
/

UPDATE "ORDER" SET amount = 60000 WHERE order_id = 1;   -- ERROR
UPDATE "ORDER" SET amount = 40000 WHERE order_id = 2;   -- SUCCESS
```

Task completed in 0.027 seconds

```
Error starting at line : 295 in command -
UPDATE "ORDER" SET amount = 60000 WHERE order_id = 1
Error report -
ORA-20009: Cannot update SHIPPED orders!
ORA-06512: at "MY_HR.BU_NO_UPDATE_SHIPPED", line 3
ORA-04088: error during execution of trigger 'MY_HR.BU_NO_UPDATE_SHIPPED'


1 row updated.
```

10)



```
    RAISE_APPLICATION_ERROR(-20009, 'Cannot update SHIPPED orders!');
  END IF;
END;
/

UPDATE "ORDER" SET amount = 60000 WHERE order_id = 1;   -- ERROR
UPDATE "ORDER" SET amount = 40000 WHERE order_id = 2;   -- SUCCESS

DROP TABLE login_audit PURGE;
CREATE TABLE login_audit (
  username    VARCHAR2(30),
  login_time  DATE
);

CREATE OR REPLACE TRIGGER logon_audit_trigger
AFTER LOGON ON SCHEMA
BEGIN
  INSERT INTO login_audit VALUES (USER, SYSDATE);
  COMMIT;
END;
/

SELECT username, TO_CHAR(login_time,'DD-MON-YYYY HH24:MI:SS') "LOGIN TIME"
FROM login_audit
ORDER BY login_time DESC;
```

All Rows Fetched: 0 in 0.002 seconds

USERNAME	LOGIN TIME