Name: Braden Caleb Perumal

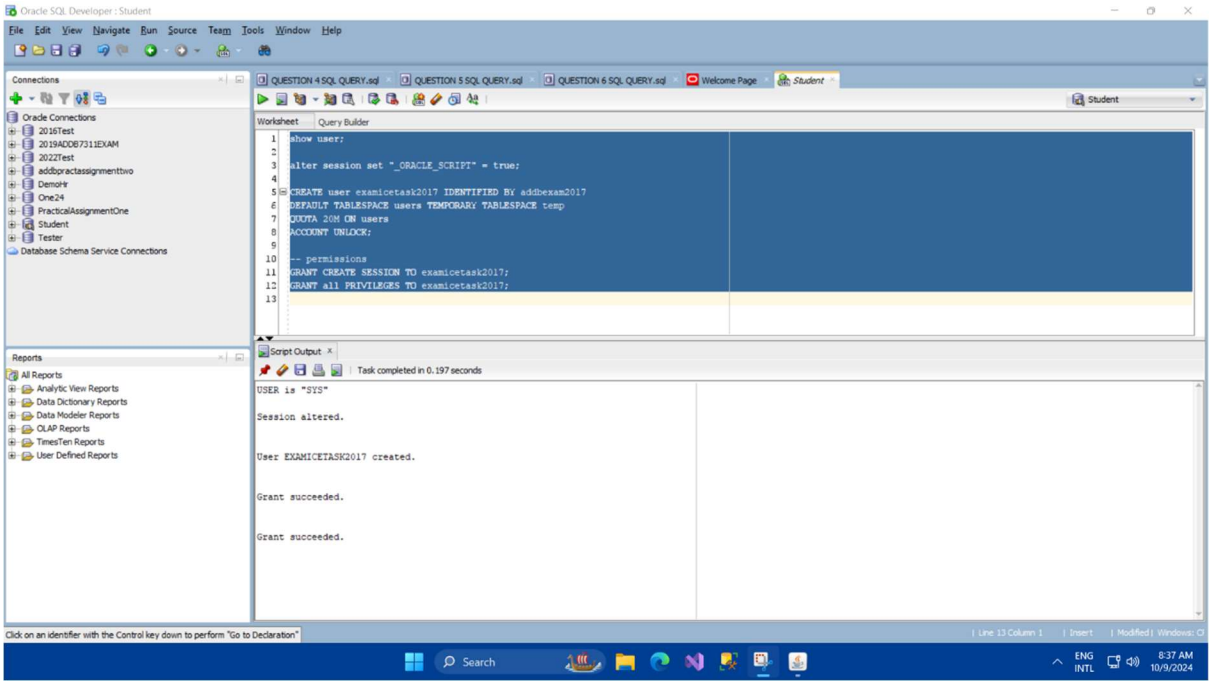
Student Number: ST10287165

Lecturer: Ms N Ramckurran

Module: ADDB 7311

ADDB7311 ICE TASK 3

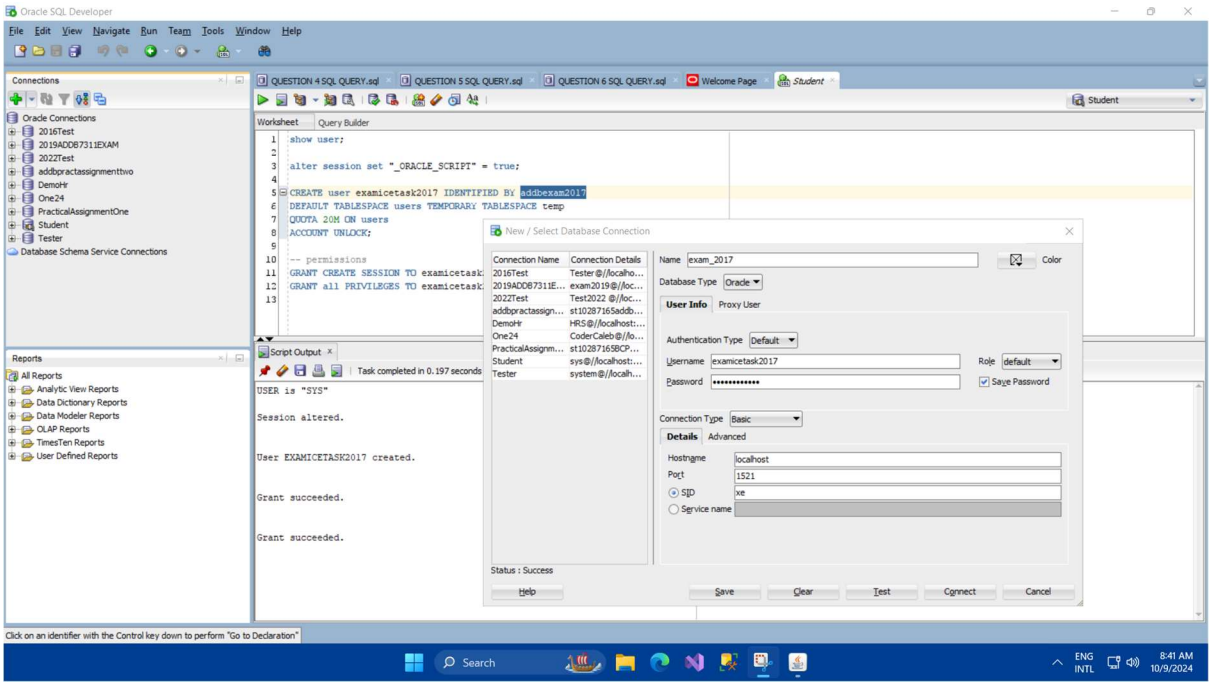
Creating a new user with privileges



SQL Query:

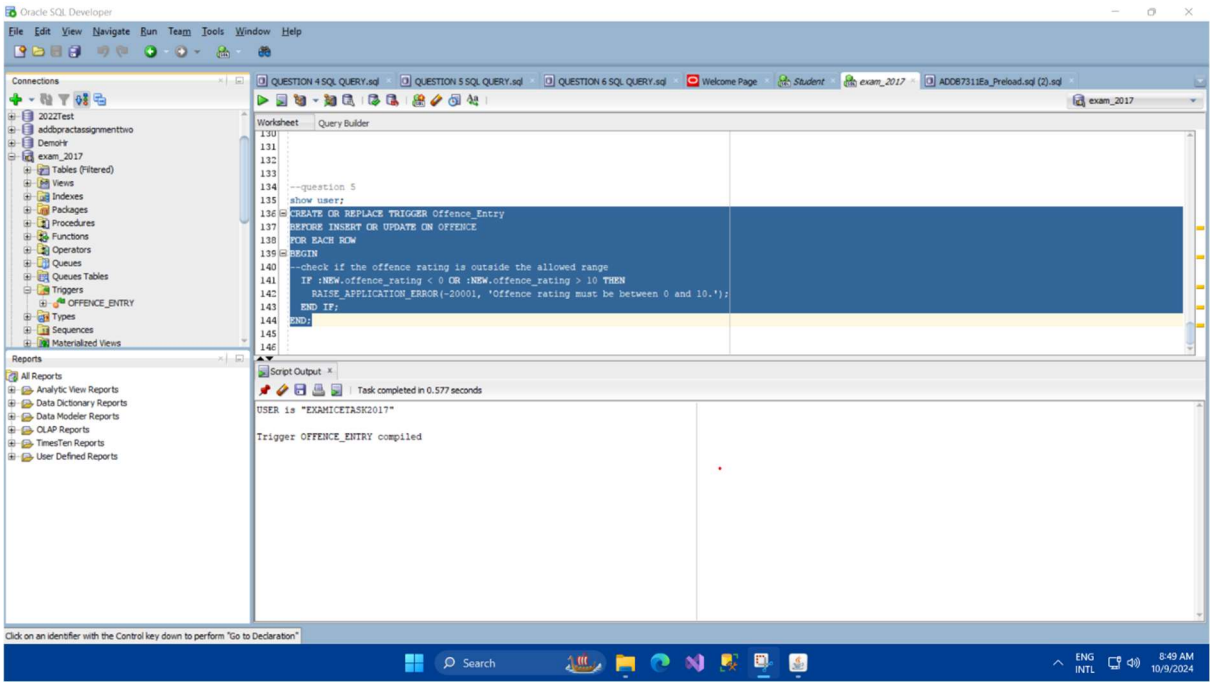
```
show user;

alter session set "_ORACLE_SCRIPT" = true;

CREATE user examicetask2017 IDENTIFIED BY addbexam2017
DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp
QUOTA 20M ON users
ACCOUNT UNLOCK;

-- permissions
GRANT CREATE SESSION TO examicetask2017;
GRANT all PRIVILEGES TO examicetask2017;
```

Creating new database:

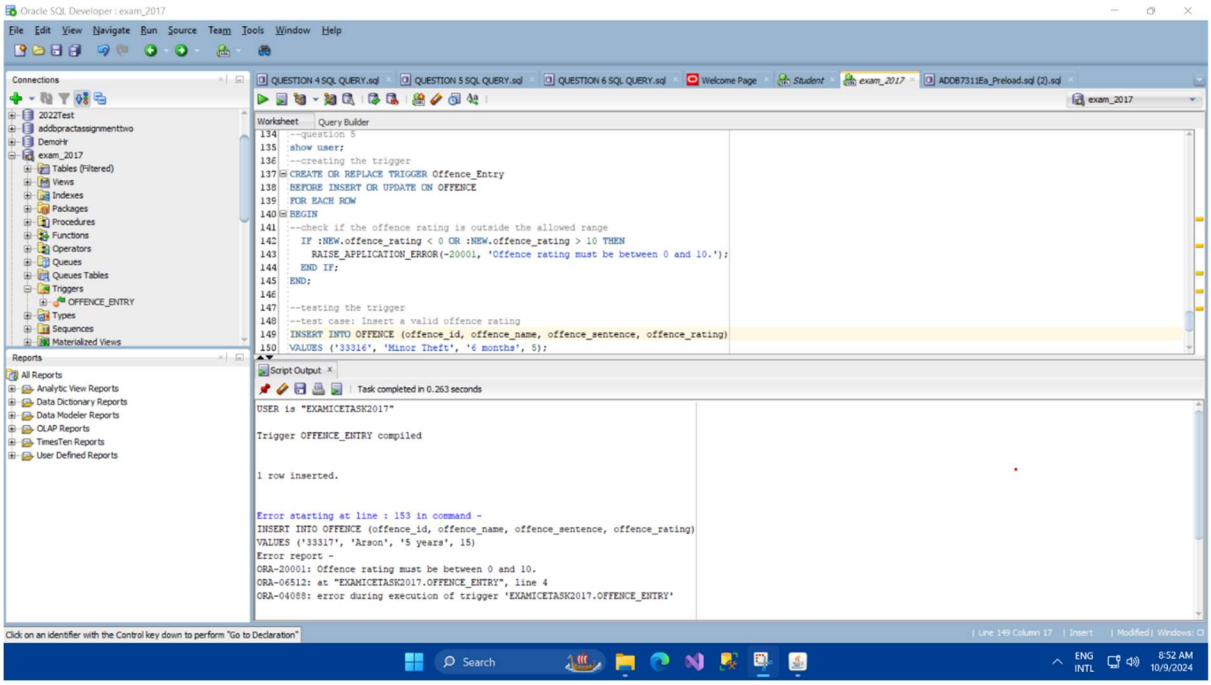


Question Five

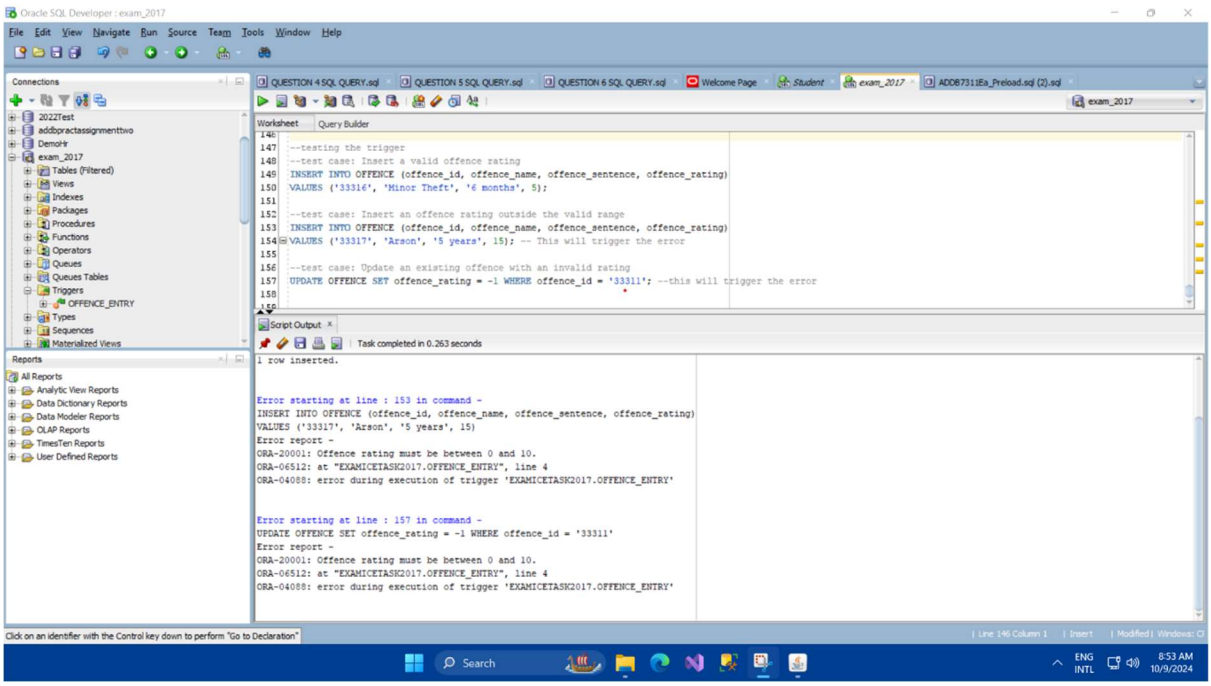
Creating the trigger:



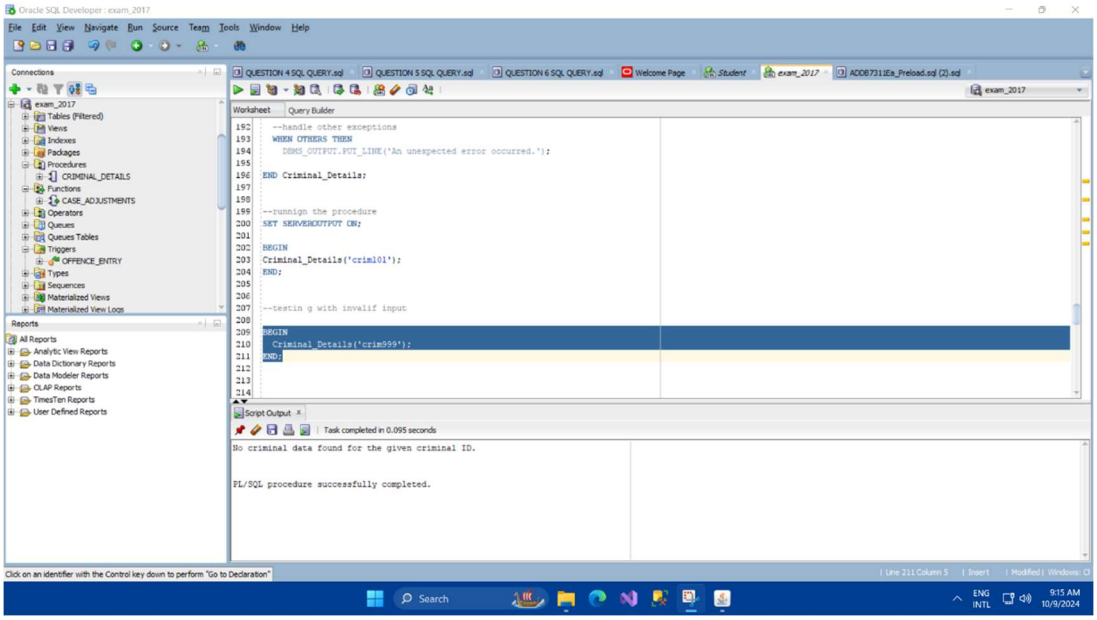
Testing the trigger:



Testing the trigger with valid input:



Testing the trigger with invalid input:

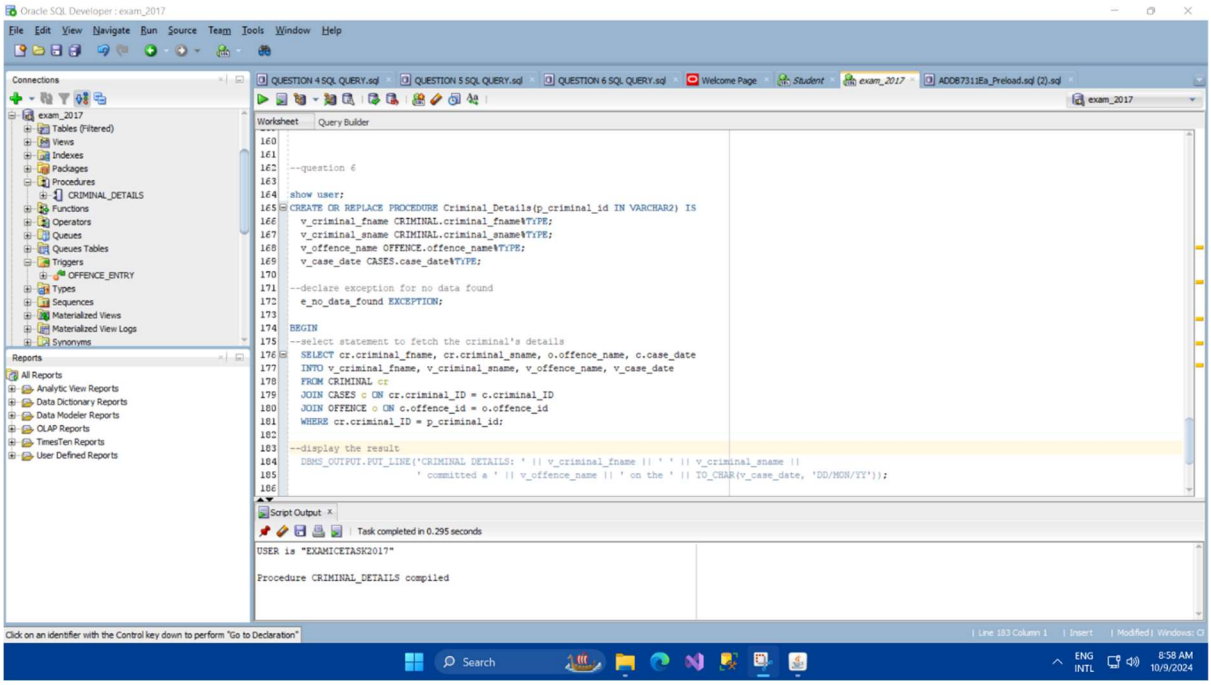


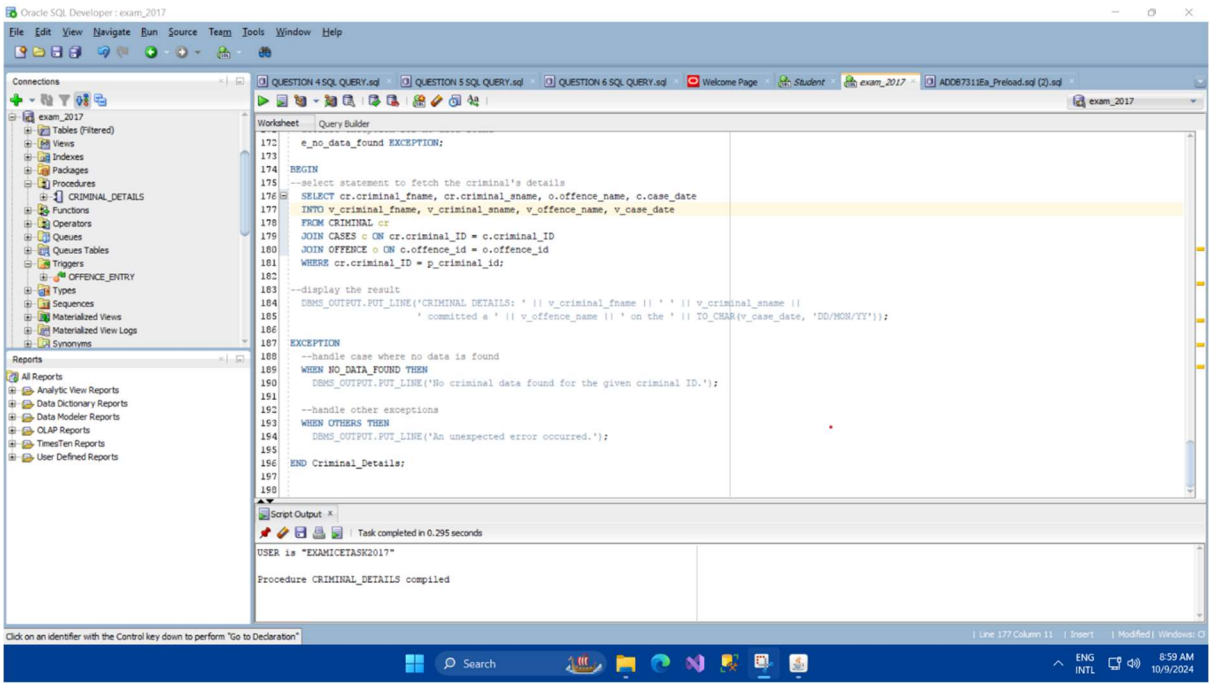
SQL Query:

```
--question 5
show user;
--creating the trigger
CREATE OR REPLACE TRIGGER Offence_Entry
BEFORE INSERT OR UPDATE ON OFFENCE
FOR EACH ROW
BEGIN
--check if the offence rating is outside the allowed range
  IF :NEW.offence_rating < 0 OR :NEW.offence_rating > 10 THEN
    RAISE_APPLICATION_ERROR(-20001, 'Offence rating must be between 0 and 10.');
  END IF;
END;

--testing the trigger
--test case: Insert a valid offence rating
INSERT INTO OFFENCE (offence_id, offence_name, offence_sentence, offence_rating)
VALUES ('33316', 'Minor Theft', '6 months', 5);

--test case: Insert an offence rating outside the valid range
INSERT INTO OFFENCE (offence_id, offence_name, offence_sentence, offence_rating)
VALUES ('33317', 'Arson', '5 years', 15); -- This will trigger the error

--test case: Update an existing offence with an invalid rating
UPDATE OFFENCE SET offence_rating = -1 WHERE offence_id = '33311'; --this will trigger the error
--testin g with invalif input

BEGIN
  Criminal_Details('crim999');
END;
```

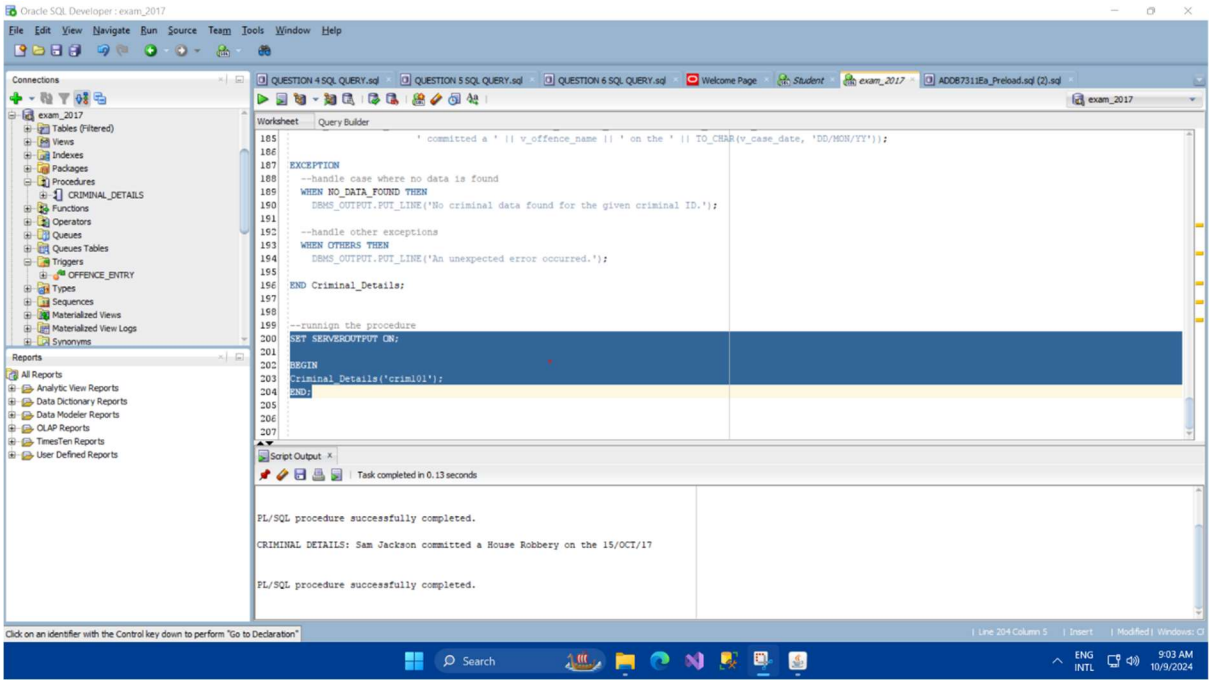
Question Six

Creating procedure:





Executing the procedure:



SQL Query:

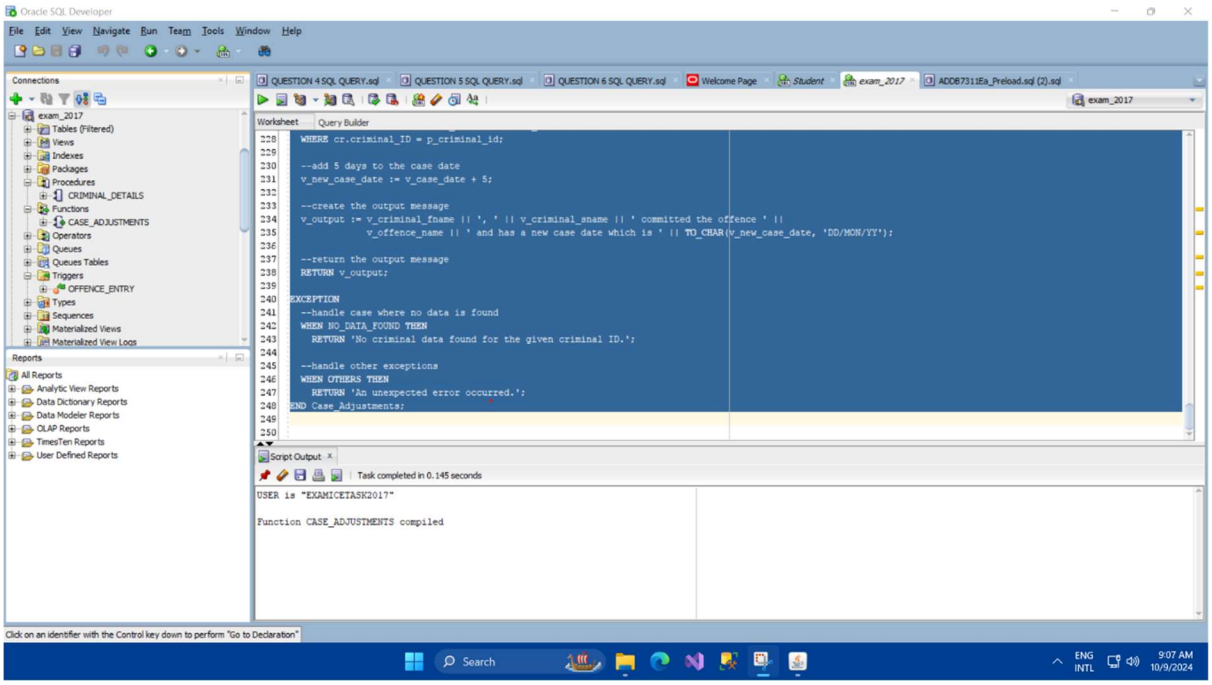
```
--question 6

show user;
CREATE OR REPLACE PROCEDURE Criminal_Details(p_criminal_id IN VARCHAR2) IS
  v_criminal_fname CRIMINAL.criminal_fname%TYPE;
  v_criminal_sname CRIMINAL.criminal_sname%TYPE;
  v_offence_name OFFENCE.offence_name%TYPE;
  v_case_date CASES.case_date%TYPE;

--declare exception for no data found
  e_no_data_found EXCEPTION;

BEGIN
--select statement to fetch the criminal's details
  SELECT cr.criminal_fname, cr.criminal_sname, o.offence_name, c.case_date
  INTO v_criminal_fname, v_criminal_sname, v_offence_name, v_case_date
  FROM CRIMINAL cr
  JOIN CASES c ON cr.criminal_ID = c.criminal_ID
  JOIN OFFENCE o ON c.offence_id = o.offence_id
  WHERE cr.criminal_ID = p_criminal_id;

--display the result
  DBMS_OUTPUT.PUT_LINE('CRIMINAL DETAILS: ' || v_criminal_fname || ' ' || v_criminal_sname ||
```

```
                ' committed a ' || v_offence_name || ' on the ' || TO_CHAR(v_case_date,
'DD/MON/YY'));
```

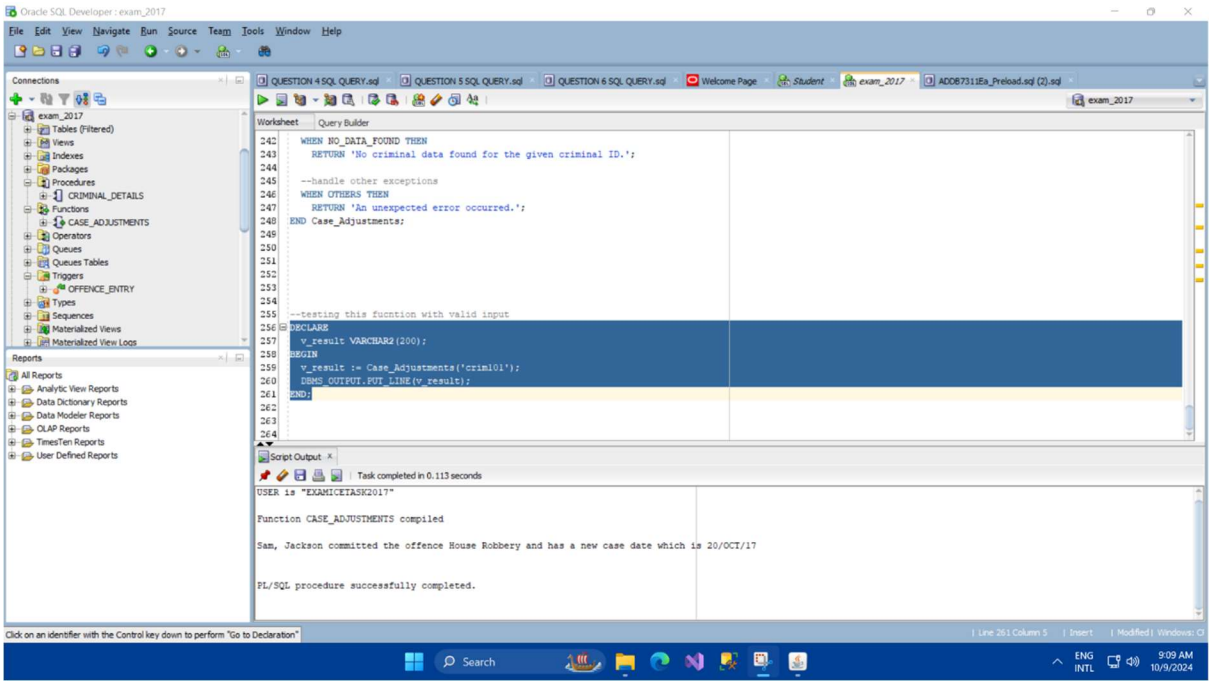
```
EXCEPTION
  --handle case where no data is found
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No criminal data found for the given criminal ID.');

  --handle other exceptions
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An unexpected error occurred.');

END Criminal_Details;


--runnign the procedure
SET SERVEROUTPUT ON;

BEGIN
Criminal_Details('crim101');
END;
```

Question Seven:

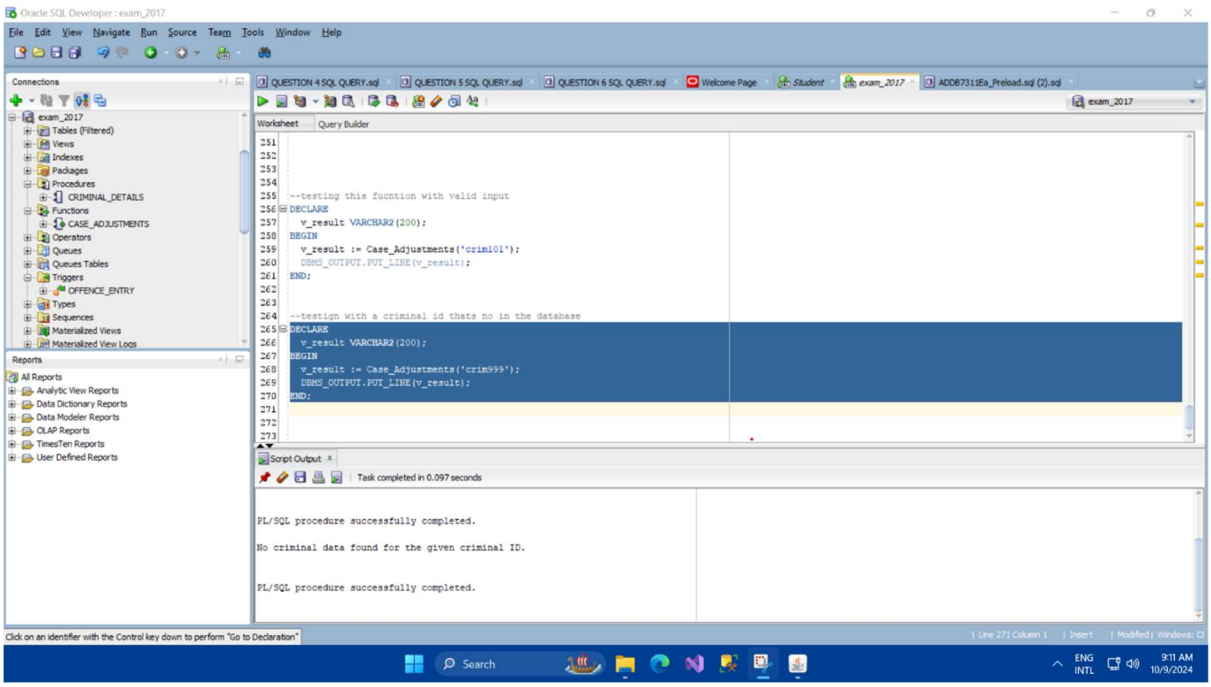
Creating function:



Testing the function with valid input:



Testing the function with invalid input:



SQL Query:

```
--question 7
show user;
CREATE OR REPLACE FUNCTION Case_Adjustments(p_criminal_id IN VARCHAR2)
RETURN VARCHAR2 IS
  v_criminal_fname CRIMINAL.criminal_fname%TYPE;
  v_criminal_sname CRIMINAL.criminal_sname%TYPE;
  v_offence_name OFFENCE.offence_name%TYPE;
  v_case_date CASES.case_date%TYPE;
  v_new_case_date CASES.case_date%TYPE;
  v_output VARCHAR2(200);

  --declare exception for no data found
  e_no_data_found EXCEPTION;

BEGIN
  --select statement to fetch the criminal's details
  SELECT cr.criminal_fname, cr.criminal_sname, o.offence_name, c.case_date
  INTO v_criminal_fname, v_criminal_sname, v_offence_name, v_case_date
  FROM CRIMINAL cr
  JOIN CASES c ON cr.criminal_ID = c.criminal_ID
  JOIN OFFENCE o ON c.offence_id = o.offence_id
  WHERE cr.criminal_ID = p_criminal_id;

  --add 5 days to the case date
  v_new_case_date := v_case_date + 5;

  --create the output message
  v_output := v_criminal_fname || ', ' || v_criminal_sname || ' committed the offence ' ||
          v_offence_name || ' and has a new case date which is ' || TO_CHAR(v_new_case_date,
'DD/MON/YY');

  --return the output message
  RETURN v_output;

EXCEPTION
  --handle case where no data is found
  WHEN NO_DATA_FOUND THEN
    RETURN 'No criminal data found for the given criminal ID.';

  --handle other exceptions
  WHEN OTHERS THEN
    RETURN 'An unexpected error occurred.';
END Case_Adjustments;
```

```
--testing this fucntion with valid input
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := Case_Adjustments('crim101');
  DBMS_OUTPUT.PUT_LINE(v_result);
END;


--testign with a criminal id thats no in the database
DECLARE
  v_result VARCHAR2(200);
BEGIN
  v_result := Case_Adjustments('crim999');
  DBMS_OUTPUT.PUT_LINE(v_result);
END;
```