

Name: Braden Caleb Perumal

Student Number: ST10287165

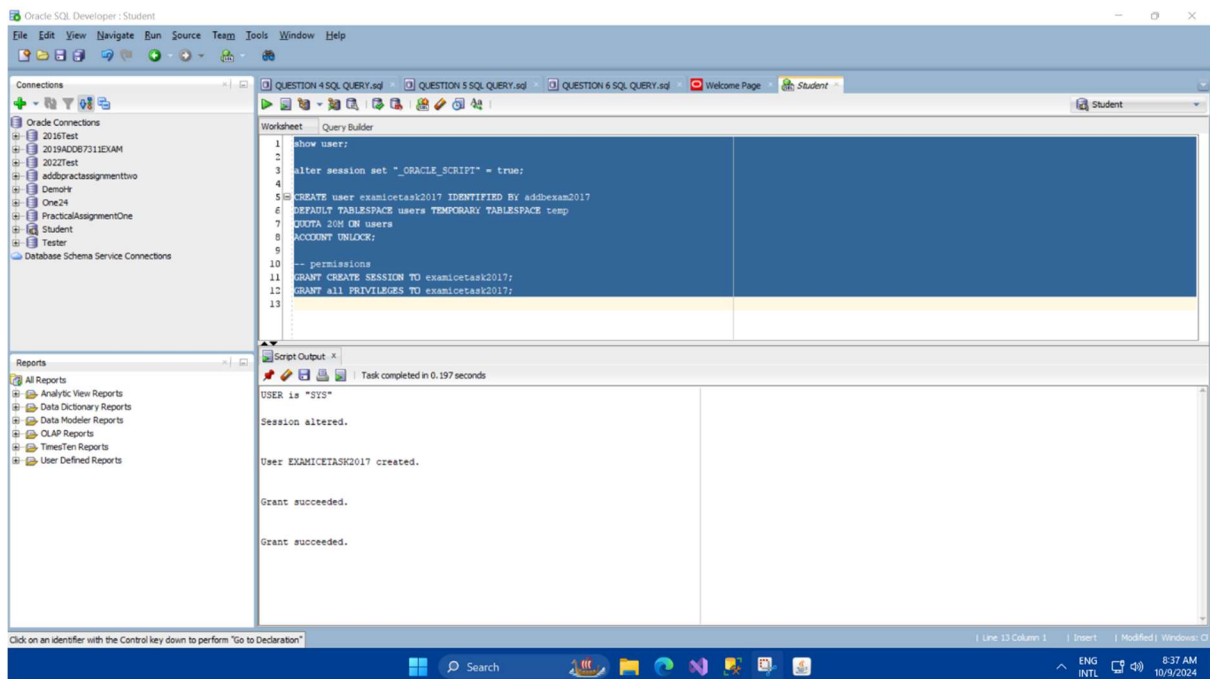
Lecturer: Ms N Ramckurran

Module: ADDB 7311

ADDB7311 ICE TASK 3



Creating a new user with privileges



SQL Query:

show user;

alter session set "_ORACLE_SCRIPT" = true;

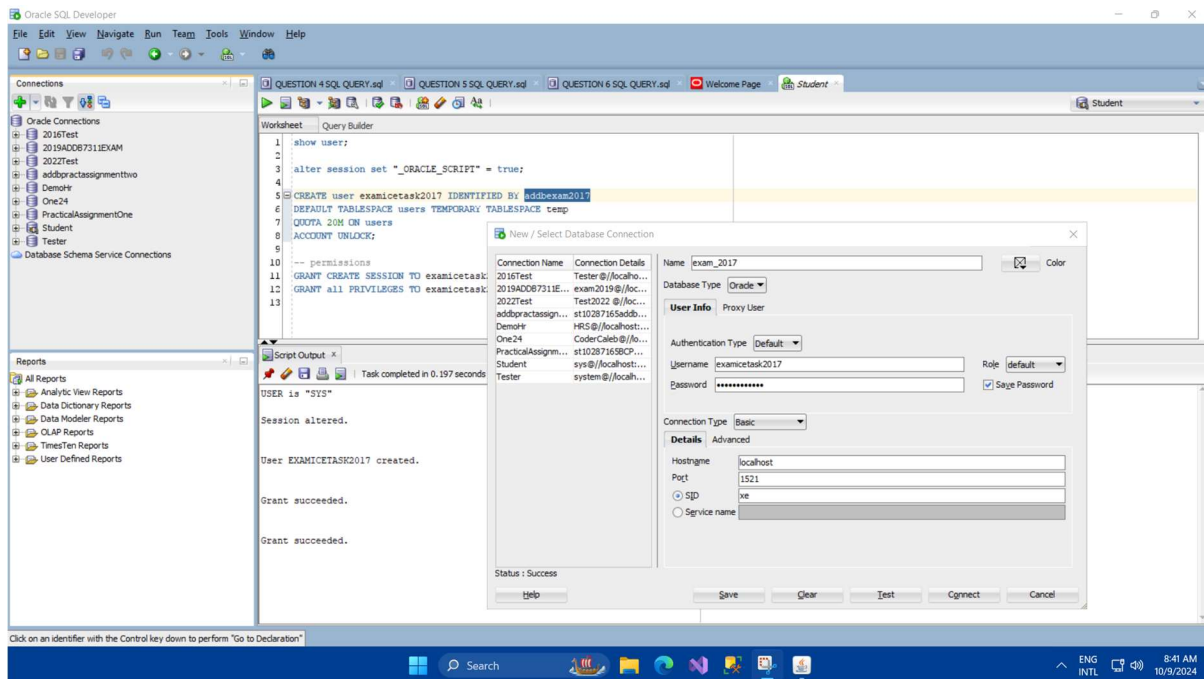
CREATE user examicetask2017 IDENTIFIED BY addbexam2017
DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp
QUOTA 20M ON users
ACCOUNT UNLOCK;

-- permissions

GRANT CREATE SESSION TO examicetask2017;

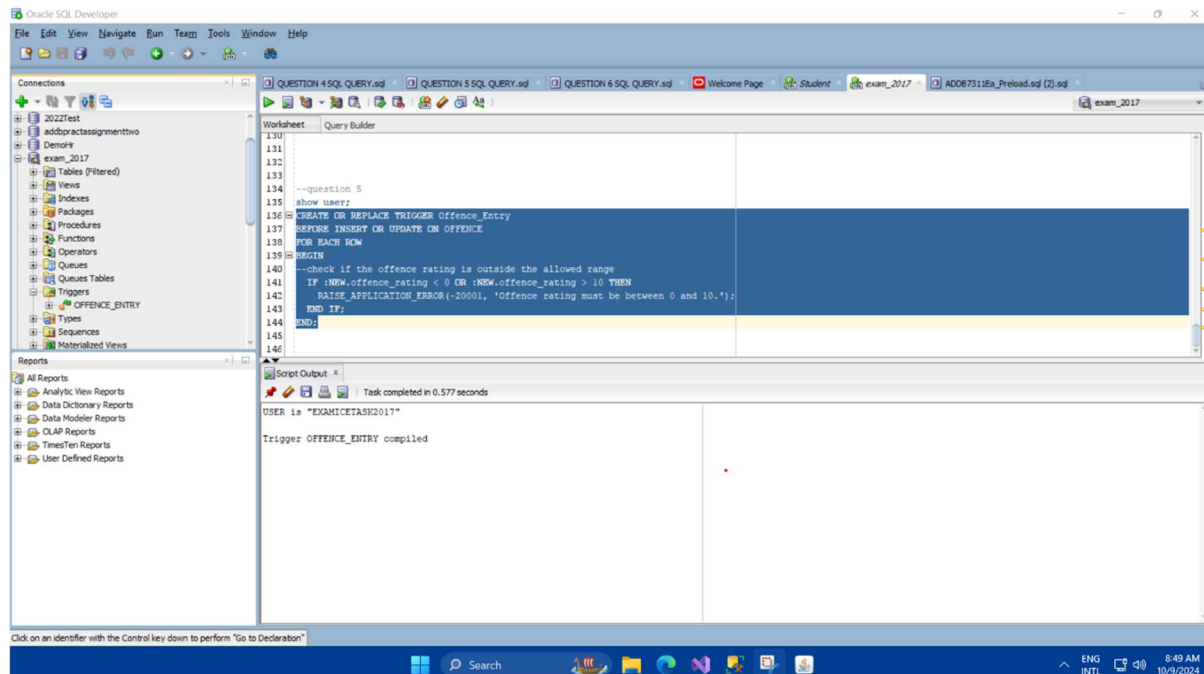
GRANT all PRIVILEGES TO examicetask2017;

Creating new database:



Question Five

Creating the trigger:



Testing the trigger:

Oracle SQL Developer: exam_2017

Connections: 2022test, addopractassgmmenthwo, Demoir, exam_2017 (Tables (Filtered), Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Tables, Triggers, OFFENCE_ENTRY, Types, Sequences, Materialized Views), Reports (All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports)

Worksheet: Query Builder

```
134 --Question 5
135 show user;
136 --creating the trigger
137 CREATE OR REPLACE TRIGGER OFFENCE_Entry
138 BEFORE INSERT OR UPDATE ON OFFENCE
139 FOR EACH ROW
140 BEGIN
141 --check if the offence rating is outside the allowed range
142 IF :NEW.offence_rating < 0 OR :NEW.offence_rating > 10 THEN
143 RAISE_APPLICATION_ERROR(-20001, 'Offence rating must be between 0 and 10.');
```

Script Output:

Task completed in 0.263 seconds

USER is "EXAMICETASK2017"

Trigger OFFENCE_ENTRY compiled

1 row inserted.

Error starting at line : 153 in command -

```
INSERT INTO OFFENCE (offence_id, offence_name, offence_sentence, offence_rating)
VALUES ('33317', 'Arson', '5 years', 15)
```

Error report -

ORA-20001: Offence rating must be between 0 and 10.

ORA-06512: at "EXAMICETASK2017.OFFENCE_ENTRY", line 4

ORA-04088: error during execution of trigger 'EXAMICETASK2017.OFFENCE_ENTRY'

Testing the trigger with valid input:

Oracle SQL Developer: exam_2017

Connections: 2022test, addopractassgmmenthwo, Demoir, exam_2017 (Tables (Filtered), Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Tables, Triggers, OFFENCE_ENTRY, Types, Sequences, Materialized Views), Reports (All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports)

Worksheet: Query Builder

```
146 --testing the trigger
147 --test case: Insert a valid offence rating
148 INSERT INTO OFFENCE (offence_id, offence_name, offence_sentence, offence_rating)
149 VALUES ('33316', 'Minor Theft', '6 months', 5);
150
151 --test case: Insert an offence rating outside the valid range
152 INSERT INTO OFFENCE (offence_id, offence_name, offence_sentence, offence_rating)
153 VALUES ('33317', 'Arson', '5 years', 15); -- This will trigger the error
154
155 --test case: Update an existing offence with an invalid rating
156 UPDATE OFFENCE SET offence_rating = -1 WHERE offence_id = '33311'; --this will trigger the error
157
```

Script Output:

Task completed in 0.263 seconds

1 row inserted.

Error starting at line : 153 in command -

```
INSERT INTO OFFENCE (offence_id, offence_name, offence_sentence, offence_rating)
VALUES ('33317', 'Arson', '5 years', 15)
```

Error report -

ORA-20001: Offence rating must be between 0 and 10.

ORA-06512: at "EXAMICETASK2017.OFFENCE_ENTRY", line 4

ORA-04088: error during execution of trigger 'EXAMICETASK2017.OFFENCE_ENTRY'

Error starting at line : 157 in command -

```
UPDATE OFFENCE SET offence_rating = -1 WHERE offence_id = '33311'
```

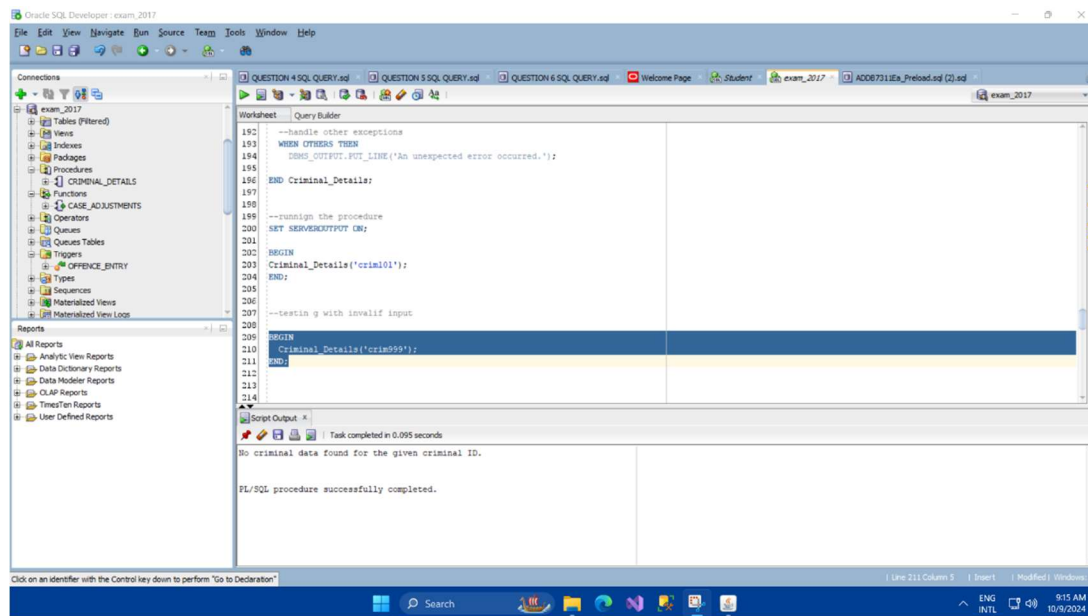
Error report -

ORA-20001: Offence rating must be between 0 and 10.

ORA-06512: at "EXAMICETASK2017.OFFENCE_ENTRY", line 4

ORA-04088: error during execution of trigger 'EXAMICETASK2017.OFFENCE_ENTRY'

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:

The screenshot shows the Oracle SQL Developer interface with the 'Query Builder' window open. The procedure being created is named 'CRIMINAL_DETAILS' and is designed to fetch criminal details based on a provided ID. The code includes variable declarations for criminal frame, name, offence name, and case date. It uses a SELECT statement with JOINs to fetch data from the CRIMINAL, CASES, and OFFENCE tables. The result is displayed using DBMS_OUTPUT.PUT_LINE. An exception is declared for 'e_no_data_found'.

```
160
161
162 --question 6
163
164 show user;
165 CREATE OR REPLACE PROCEDURE Criminal_Details(p_criminal_id IN VARCHAR2) IS
166   v_criminal_fname CRIMINAL.criminal_fname%TYPE;
167   v_criminal_sname CRIMINAL.criminal_sname%TYPE;
168   v_offence_name OFFENCE.offence_name%TYPE;
169   v_case_date CASES.case_date%TYPE;
170
171 --declare exception for no data found
172   e_no_data_found EXCEPTION;
173
174 BEGIN
175   --select statement to fetch the criminal's details
176   SELECT cr.criminal_fname, cr.criminal_sname, o.offence_name, c.case_date
177   INTO v_criminal_fname, v_criminal_sname, v_offence_name, v_case_date
178   FROM CRIMINAL cr
179   JOIN CASES c ON cr.criminal_id = c.criminal_id
180   JOIN OFFENCE o ON c.offence_id = o.offence_id
181   WHERE cr.criminal_id = p_criminal_id;
182
183   --display the result
184   DBMS_OUTPUT.PUT_LINE('CRIMINAL DETAILS: ' || v_criminal_fname || ' ' || v_criminal_sname ||
185     ' committed a ' || v_offence_name || ' on the ' || TO_CHAR(v_case_date, 'DD/MON/YY'));
186
187   --handle case where no data is found
188   WHEN NO_DATA_FOUND THEN
189     DBMS_OUTPUT.PUT_LINE('No criminal data found for the given criminal ID.');
```

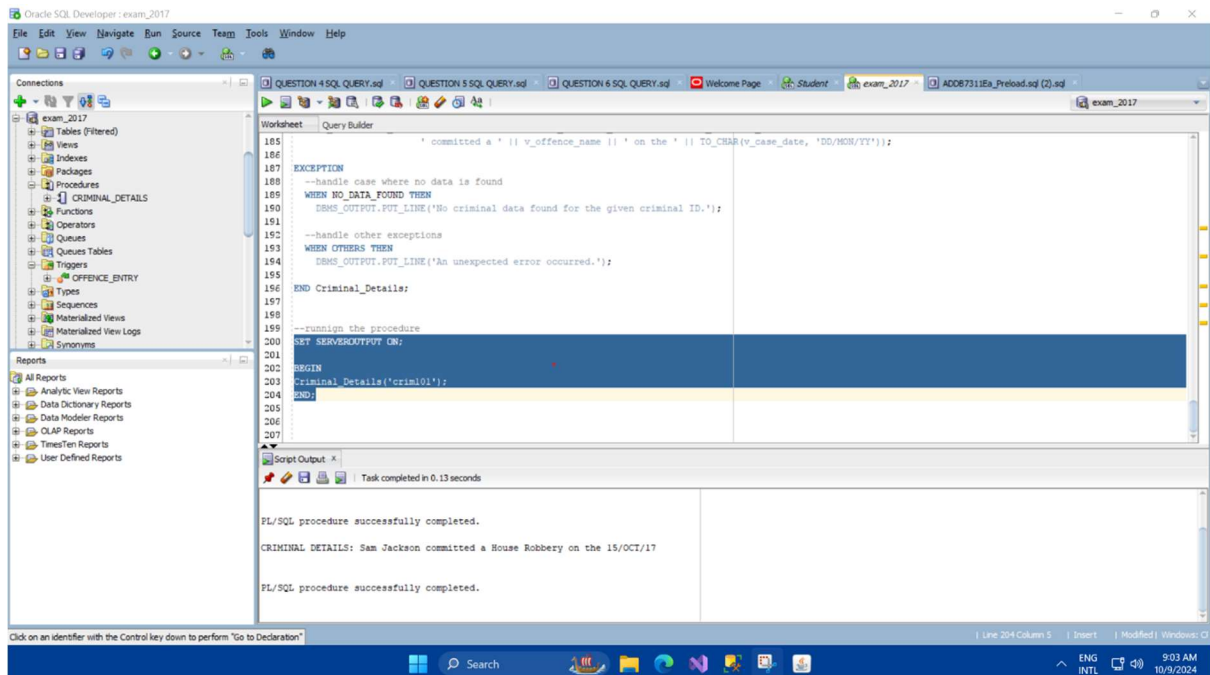
Script Output: Task completed in 0.295 seconds
USER is "EXAMICETASK2017"
Procedure CRIMINAL_DETAILS compiled

This screenshot shows the same Oracle SQL Developer interface, but the procedure code is now complete. It includes the exception handling logic for 'e_no_data_found' and 'OTHERS'.

```
172   e_no_data_found EXCEPTION;
173
174 BEGIN
175   --select statement to fetch the criminal's details
176   SELECT cr.criminal_fname, cr.criminal_sname, o.offence_name, c.case_date
177   INTO v_criminal_fname, v_criminal_sname, v_offence_name, v_case_date
178   FROM CRIMINAL cr
179   JOIN CASES c ON cr.criminal_id = c.criminal_id
180   JOIN OFFENCE o ON c.offence_id = o.offence_id
181   WHERE cr.criminal_id = p_criminal_id;
182
183   --display the result
184   DBMS_OUTPUT.PUT_LINE('CRIMINAL DETAILS: ' || v_criminal_fname || ' ' || v_criminal_sname ||
185     ' committed a ' || v_offence_name || ' on the ' || TO_CHAR(v_case_date, 'DD/MON/YY'));
186
187   --handle case where no data is found
188   WHEN NO_DATA_FOUND THEN
189     DBMS_OUTPUT.PUT_LINE('No criminal data found for the given criminal ID.');
```

Script Output: Task completed in 0.295 seconds
USER is "EXAMICETASK2017"
Procedure CRIMINAL_DETAILS compiled

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;

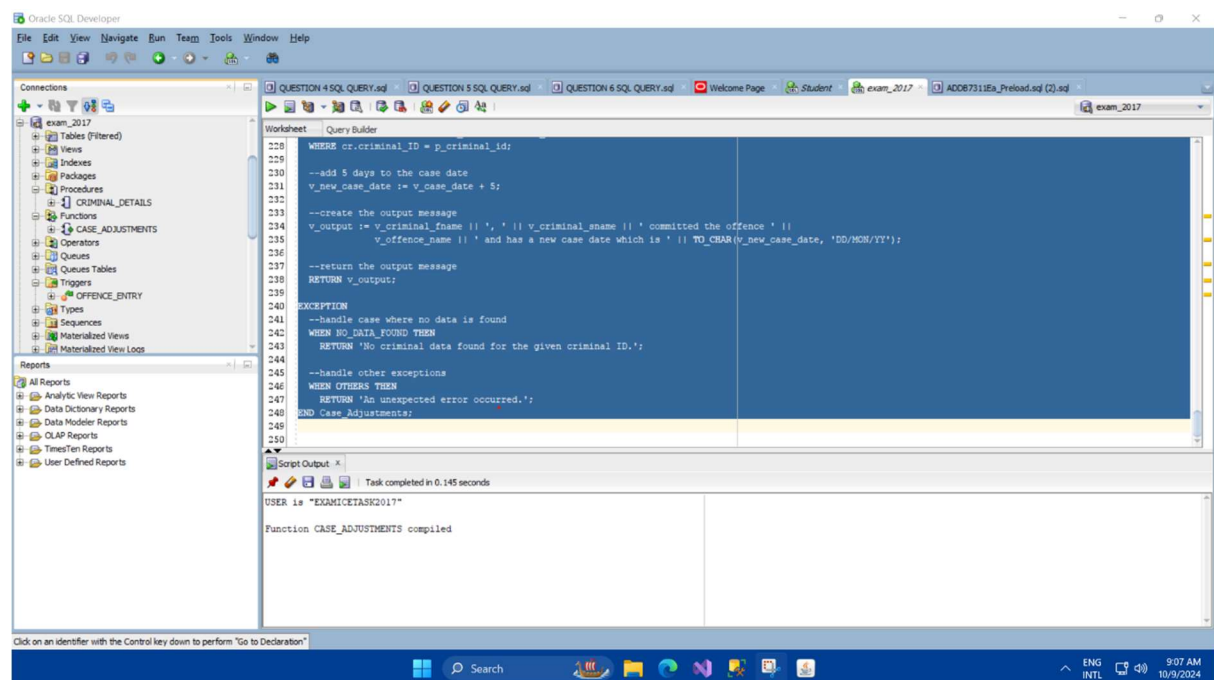
BEGIN

Criminal_Details('crim101');

END;

Question Seven:

Creating function:



Testing the function with valid input:

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for 'exam_2017', including tables, views, packages, procedures, functions, operators, queues, queues tables, triggers, types, sequences, materialized views, and materialized view logs. The main workspace is divided into a 'Worksheet' and a 'Script Output' pane. The 'Worksheet' contains a PL/SQL function named 'CASE_ADJUSTMENTS' with the following code:

```
242 WHEN NO_DATA_FOUND THEN
243 RETURN 'No criminal data found for the given criminal ID.';
244
245 --handle other exceptions
246 WHEN OTHERS THEN
247 RETURN 'An unexpected error occurred.';
248 END CASE_ADJUSTMENTS;
249
250
251
252
253
254
255 --testing this function with valid input
256 DECLARE
257 v_result VARCHAR2(200);
258 BEGIN
259 v_result := Case_Adjustments('crim101');
260 DBMS_OUTPUT.PUT_LINE(v_result);
261 END;
```

The 'Script Output' pane shows the results of the function execution:

```
Task completed in 0.113 seconds
USER is "EXAMICETASK2017"
Function CASE_ADJUSTMENTS compiled
Sam, Jackson committed the offence House Robbery and has a new case date which is 20/OCT/17
PL/SQL procedure successfully completed.
```

Testing the function with invalid input:

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for 'exam_2017'. The main workspace is divided into a 'Worksheet' and a 'Script Output' pane. The 'Worksheet' contains a PL/SQL function named 'CASE_ADJUSTMENTS' with the following code:

```
251
252
253
254
255 --testing this function with valid input
256 DECLARE
257 v_result VARCHAR2(200);
258 BEGIN
259 v_result := Case_Adjustments('crim101');
260 DBMS_OUTPUT.PUT_LINE(v_result);
261 END;
262
263
264 --testing with a criminal id thats no in the database
265 DECLARE
266 v_result VARCHAR2(200);
267 BEGIN
268 v_result := Case_Adjustments('crim999');
269 DBMS_OUTPUT.PUT_LINE(v_result);
270 END;
271
272
273
```

The 'Script Output' pane shows the results of the function execution:

```
Task completed in 0.097 seconds
PL/SQL procedure successfully completed.
No criminal data found for the given criminal ID.
PL/SQL procedure successfully completed.
```

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;