**KARABO NICHOLAS BALOYI**

**ST10033503**

**ADDB6321**

**PRACTICUM (EXAM)**

**24 JUNE 2024**

**QUESTION 1: SET UP OF DATABASE**

CREATE TABLE INSTRUCTOR (

INS\_ID VARCHAR2(10) PRIMARY KEY,

INS\_FNAME VARCHAR2(50),

INS\_SNAME VARCHAR2(50),

INS\_CONTACT VARCHAR2(20),

INS\_LEVEL NUMBER

);

INSERT INTO INSTRUCTOR (INS\_ID, INS\_FNAME, INS\_SNAME , INS\_CONTACT, INS\_LEVEL)

VALUES ('101', 'James', 'Willis', '0843569851', '7');

INSERT INTO INSTRUCTOR (INS\_ID, INS\_FNAME, INS\_SNAME , INS\_CONTACT, INS\_LEVEL)

VALUES ('102', 'Sam', 'Wait', '0763698521', '2');

INSERT INTO INSTRUCTOR (INS\_ID, INS\_FNAME, INS\_SNAME , INS\_CONTACT, INS\_LEVEL)

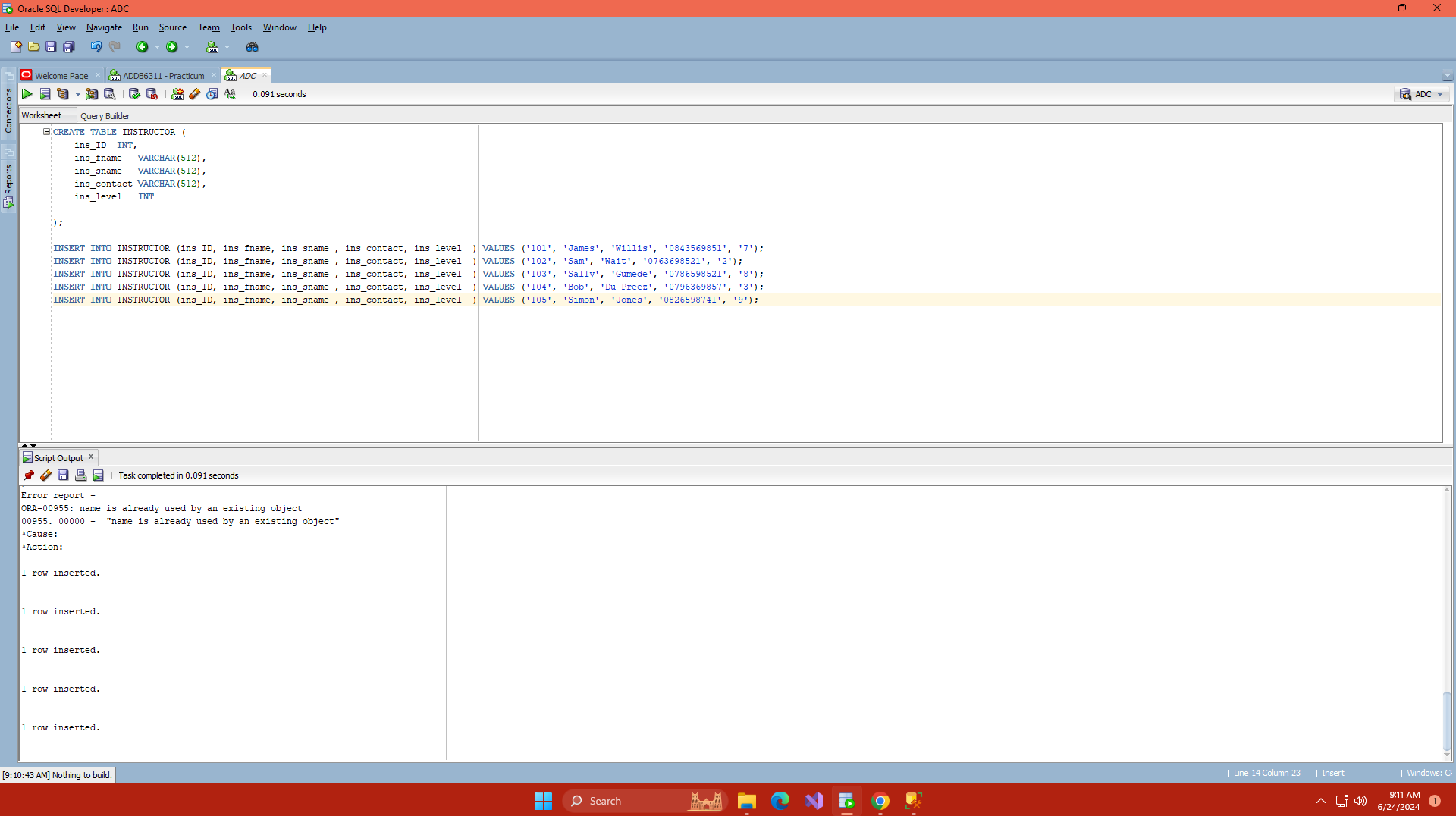
VALUES ('103', 'Sally', 'Gumede', '0786598521', '8');

INSERT INTO INSTRUCTOR (INS\_ID, INS\_FNAME, INS\_SNAME , INS\_CONTACT, INS\_LEVEL)

VALUES ('104', 'Bob', 'Du Preez', '0796369857', '3');

INSERT INTO INSTRUCTOR (INS\_ID, INS\_FNAME, INS\_SNAME , INS\_CONTACT, INS\_LEVEL)

VALUES ('105', 'Simon', 'Jones', '0826598741', '9');



CREATE TABLE CUSTOMERS (

CUST\_ID VARCHAR2(10) PRIMARY KEY,

CUST\_FNAME VARCHAR2(50),

CUST\_SNAME VARCHAR2(50),

CUST\_ADDRESS VARCHAR2(200),

CUST\_CONTACT VARCHAR2(15)

);

INSERT INTO CUSTOMERS (CUST\_ID , CUST\_FNAME , CUST\_SNAME, CUST\_ADDRESS , CUST\_CONTACT) VALUES ('C115', 'Heinrich', 'Willis', '3 Main Road', '0821253659');

INSERT INTO CUSTOMERS (CUST\_ID , CUST\_FNAME , CUST\_SNAME, CUST\_ADDRESS , CUST\_CONTACT) VALUES ('C116', 'David', 'Watson', '13 Cape Road', '0769658547');

INSERT INTO CUSTOMERS (CUST\_ID , CUST\_FNAME , CUST\_SNAME, CUST\_ADDRESS , CUST\_CONTACT) VALUES ('C117', 'Waldo', 'Smith', '3 Mountain Road', '0863256574');

INSERT INTO CUSTOMERS (CUST\_ID , CUST\_FNAME , CUST\_SNAME, CUST\_ADDRESS , CUST\_CONTACT) VALUES ('C118', 'Alex', 'Hanson', '8 Circle Road', '0762356587');

INSERT INTO CUSTOMERS (CUST\_ID , CUST\_FNAME , CUST\_SNAME, CUST\_ADDRESS , CUST\_CONTACT) VALUES ('C119', 'Kuhle', 'Bitterhout', '15 Main Road', '0821235258');

INSERT INTO CUSTOMERS (CUST\_ID , CUST\_FNAME , CUST\_SNAME, CUST\_ADDRESS , CUST\_CONTACT) VALUES ('C120', 'Thando', 'Zolani', '88 Summer Road', '0847541254');

INSERT INTO CUSTOMERS (CUST\_ID , CUST\_FNAME , CUST\_SNAME, CUST\_ADDRESS , CUST\_CONTACT) VALUES ('C121', 'Philip', 'Jackson', '3 Long Road', '0745556658');

INSERT INTO CUSTOMERS (CUST\_ID , CUST\_FNAME , CUST\_SNAME, CUST\_ADDRESS , CUST\_CONTACT) VALUES ('C122', 'Sarah', 'Jones', '7 Sea Road', '0814745745');

INSERT INTO CUSTOMERS (CUST\_ID , CUST\_FNAME , CUST\_SNAME, CUST\_ADDRESS , CUST\_CONTACT) VALUES ('C123', 'Catherine', 'Howard', '31 Lake Side Road', '0822232521');

(Independent Institute of Education, 2024)

A screenshot of a computer

Description automatically generated

CREATE TABLE DIVE (

DIVE\_ID VARCHAR2(10) PRIMARY KEY,

DIVE\_NAME VARCHAR2(50),

DIVE\_DURATION VARCHAR2(50),

DIVE\_LOCATION VARCHAR2(50),

DIVE\_EXP\_LEVEL NUMBER,

DIVE\_COST VARCHAR2(10)

);

INSERT INTO DIVE (DIVE\_ID , DIVE\_NAME , DIVE\_DURATION , DIVE\_LOCATION , DIVE\_EXP\_LEVEL , DIVE\_COST )

VALUES ('550', 'Shark Dive ', '3 hours ', 'Shark Point ', '8', '500');

INSERT INTO DIVE (DIVE\_ID , DIVE\_NAME , DIVE\_DURATION , DIVE\_LOCATION , DIVE\_EXP\_LEVEL , DIVE\_COST )

VALUES ('551', 'Coral Dive ', '1 hour ', 'Break Point ', '7', '300');

INSERT INTO DIVE (DIVE\_ID , DIVE\_NAME , DIVE\_DURATION , DIVE\_LOCATION , DIVE\_EXP\_LEVEL , DIVE\_COST )

VALUES ('552', 'Wave Crescent', '2 hours ', 'Ship wreck ally ', '3', '800');

INSERT INTO DIVE (DIVE\_ID , DIVE\_NAME , DIVE\_DURATION , DIVE\_LOCATION , DIVE\_EXP\_LEVEL , DIVE\_COST )

VALUES ('553', 'Underwater Exploration', '1 hour ', 'Coral ally ', '2', '250');

INSERT INTO DIVE (DIVE\_ID , DIVE\_NAME , DIVE\_DURATION , DIVE\_LOCATION , DIVE\_EXP\_LEVEL , DIVE\_COST )

VALUES ('554', 'Underwater Adventure', '3 hours ', 'Sandy Beach ', '3', '750');

INSERT INTO DIVE (DIVE\_ID , DIVE\_NAME , DIVE\_DURATION , DIVE\_LOCATION , DIVE\_EXP\_LEVEL , DIVE\_COST )

VALUES ('555', 'Deep Blue Ocean', '30 minutes ', 'Lazy Waves ', '2', '120');

INSERT INTO DIVE (DIVE\_ID , DIVE\_NAME , DIVE\_DURATION , DIVE\_LOCATION , DIVE\_EXP\_LEVEL , DIVE\_COST )

VALUES ('556', 'Rough Seas ', '1 hour ', 'Pipe ', '9', '700');

INSERT INTO DIVE (DIVE\_ID , DIVE\_NAME , DIVE\_DURATION , DIVE\_LOCATION , DIVE\_EXP\_LEVEL , DIVE\_COST )

VALUES ('557', 'White Water ', '2 hours ', 'Drifts ', '5', '200');

INSERT INTO DIVE (DIVE\_ID , DIVE\_NAME , DIVE\_DURATION , DIVE\_LOCATION , DIVE\_EXP\_LEVEL , DIVE\_COST )

VALUES ('558', 'Current Adventure', '2 hours ', 'Rock Lands ', '3', '150');

A screenshot of a computer

Description automatically generated

CREATE TABLE DIVE\_EVENT (

DIVE\_EVENT\_ID VARCHAR2(10),

DIVE\_DATE DATE,

DIVE\_PARTICIPANTS NUMBER,

INS\_ID VARCHAR2(10),

CUST\_ID VARCHAR2(10),

DIVE\_ID VARCHAR2(10),

FOREIGN KEY (INS\_ID) REFERENCES INSTRUCTOR(INS\_ID),

FOREIGN KEY (CUST\_ID) REFERENCES CUSTOMERS(CUST\_ID),

FOREIGN KEY (DIVE\_ID) REFERENCES DIVE(DIVE\_ID)

);

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID , DIVE\_DATE, DIVE\_PARTICIPANTS , INS\_ID , CUST\_ID , DIVE\_ID) VALUES ('de\_101', '15-Jul-17', '5', '103', 'C115', '558');

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID , DIVE\_DATE, DIVE\_PARTICIPANTS , INS\_ID , CUST\_ID , DIVE\_ID) VALUES ('de\_102', '16-Jul-17', '7', '102', 'C117', '555');

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID , DIVE\_DATE, DIVE\_PARTICIPANTS , INS\_ID , CUST\_ID , DIVE\_ID) VALUES ('de\_103', '18-Jul-17', '8', '104', 'C118', '552');

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID , DIVE\_DATE, DIVE\_PARTICIPANTS , INS\_ID , CUST\_ID , DIVE\_ID) VALUES ('de\_104', '19-Jul-17', '3', '101', 'C119', '551');

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID , DIVE\_DATE, DIVE\_PARTICIPANTS , INS\_ID , CUST\_ID , DIVE\_ID) VALUES ('de\_105', '21-Jul-17', '5', '104', 'C121', '558');

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID , DIVE\_DATE, DIVE\_PARTICIPANTS , INS\_ID , CUST\_ID , DIVE\_ID) VALUES ('de\_106', '22-Jul-17', '8', '105', 'C120', '556');

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID , DIVE\_DATE, DIVE\_PARTICIPANTS , INS\_ID , CUST\_ID , DIVE\_ID) VALUES ('de\_107', '25-Jul-17', '10', '105', 'C115', '554');

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID , DIVE\_DATE, DIVE\_PARTICIPANTS , INS\_ID , CUST\_ID , DIVE\_ID) VALUES ('de\_108', '27-Jul-17', '5', '101', 'C122', '552');

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID , DIVE\_DATE, DIVE\_PARTICIPANTS , INS\_ID , CUST\_ID , DIVE\_ID) VALUES ('de\_109', '28-Jul-17', '3', '102', 'C123', '553');

A screenshot of a computer

Description automatically generated

**QUESTION 3**

SELECT

I.INS\_FNAME || ' ' || I.INS\_SNAME AS INSTRUCTOR\_NAME,

C.CUST\_FNAME || ' ' || C.CUST\_SNAME AS CUSTOMER\_NAME,

D.DIVE\_LOCATION,

DE.DIVE\_PARTICIPANTS

FROM DIVE\_EVENT DE

JOIN

INSTRUCTOR I ON DE.INS\_ID = I.INS\_ID

JOIN

CUSTOMERS C ON DE.CUST\_ID = C.CUST\_ID

JOIN

DIVE D ON DE.DIVE\_ID = D.DIVE\_ID

WHERE

DE.DIVE\_PARTICIPANTS BETWEEN 8 AND 10;

(Independent Institute of Education, 2024)

A screenshot of a computer

Description automatically generated

**QUESTION 4**

SET SERVEROUTPUT ON;

DECLARE

CURSOR dive\_cursor IS

SELECT

D.DIVE\_NAME,

DE.DIVE\_DATE

FROM

DIVE\_EVENT DE

JOIN

DIVE D ON DE.DIVE\_ID = D.DIVE\_ID

WHERE

DE.DIVE\_PARTICIPATIONS >=10;

dive\_record dive\_cursor%ROWTYPE;

BEGIN

OPEN dive\_cursor;

LOOP

FETCH dive\_cursor INTO dive\_record;

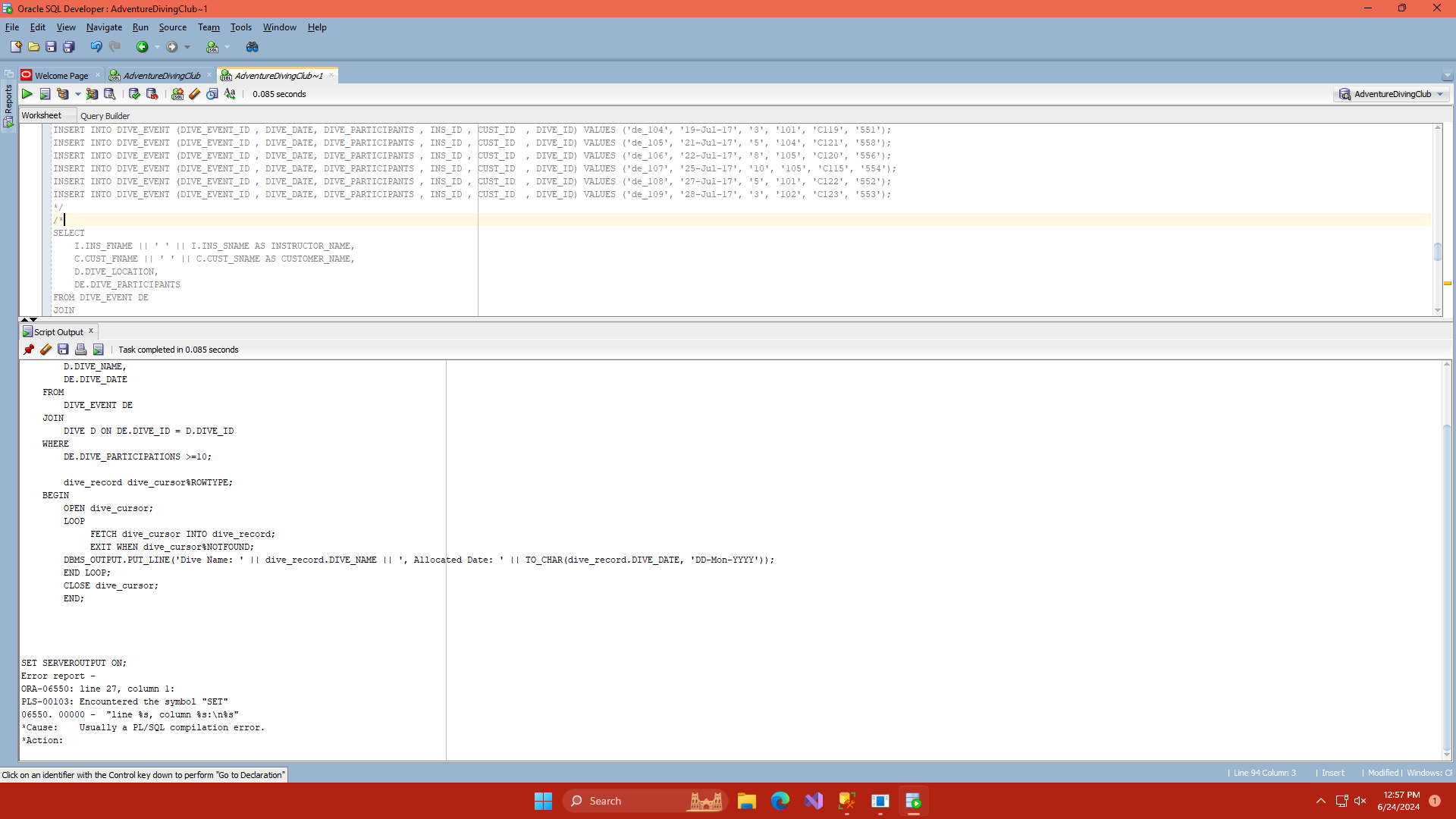
EXIT WHEN dive\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Dive Name: ' || dive\_record.DIVE\_NAME || ', Allocated Date: ' || TO\_CHAR(dive\_record.DIVE\_DATE, 'DD-Mon-YYYY'));

END LOOP;

CLOSE dive\_cursor;

END;



**QUESTION 5**

DECLARE

CURSOR dive\_cursor IS

SELECT C.CUST\_FNAME || ', ' || C.CUST\_SNAME AS CUSTOMER\_NAME,

D.DIVE\_NAME,

DE.DIVE\_PARTICIPANTS,

D.DIVE\_COST

FROM DIVE\_EVENT DE

JOIN CUSTOMERS C ON DE.CUST\_ID = C.CUST\_ID

JOIN DIVE D ON DE.DIVE\_ID = D.DIVE\_ID

WHERE TO\_NUMBER(D.DIVE\_COST) > 500;

dive\_record dive\_cursor%ROWTYPE;

num\_instructors NUMBER;

BEGIN

-- Open the cursor

OPEN dive\_cursor;

-- Loop through each record in the cursor

LOOP

FETCH dive\_cursor INTO dive\_record;

EXIT WHEN dive\_cursor%NOTFOUND;

-- Determine the number of instructors required based on the policy

IF dive\_record.DIVE\_PARTICIPANTS <= 4 THEN

num\_instructors := 1;

ELSIF dive\_record.DIVE\_PARTICIPANTS BETWEEN 5 AND 7 THEN

num\_instructors := 2;

ELSE

num\_instructors := 3;

END IF;

-- Display the customer name, dive name, participants, and number of instructors required

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER: ' || dive\_record.CUSTOMER\_NAME);

DBMS\_OUTPUT.PUT\_LINE('DIVE NAME: ' || dive\_record.DIVE\_NAME);

DBMS\_OUTPUT.PUT\_LINE('PARTICIPANTS: ' || dive\_record.DIVE\_PARTICIPANTS);

DBMS\_OUTPUT.PUT\_LINE('STATUS: ' || num\_instructors || ' Instructors required');

DBMS\_OUTPUT.PUT\_LINE('-------------------------------------------');

END LOOP;

CLOSE dive\_cursor;

END;

A screenshot of a computer

Description automatically generated

**QUESTION 6**

CREATE OR REPLACE VIEW Vw\_Dive\_Event AS

SELECT

DE.INS\_ID AS INSTRUCTOR\_ID,

DE.CUST\_ID AS CUSTOMER\_ID,

C.CUST\_ADDRESS AS CUSTOMER\_ADDRESS,

D.DIVE\_DURATION

FROM

DIVE\_EVENT DE

JOIN CUSTOMERS C ON DE.CUST\_ID = C.CUST\_ID

JOIN DIVE D ON DE.DIVE\_ID = D.DIVE\_ID

WHERE

DE.DIVE\_DATE < TO\_DATE('19-Jul-2017', 'DD-Mon-YYYY');

A screenshot of a computer

Description automatically generated

**QUESTION 7**

CREATE OR REPLACE TRIGGER New\_Dive\_Event

BEFORE INSERT OR UPDATE ON DIVE\_EVENT

FOR EACH ROW

BEGIN

IF :NEW.DIVE\_PARTICIPANTS <= 0

OR :NEW.DIVE\_PARTICIPANTS > 20

THEN

RAISE\_APPLICATION\_ERROR(-20001, 'The number of participants must be between 1 and 20');

end if;

END;

(sharper, 2017)

A screenshot of a computer

Description automatically generated

**QUESTION 8**

CREATE OR REPLACE PROCEDURE sp\_Customer\_Details (

p\_CUST\_ID IN CUSTOMERS.CUST\_ID%TYPE,

p\_DIVE\_DATE IN DIVE\_EVENT.DIVE\_DATE%TYPE

) IS

v\_CUST\_FNAME CUSTOMERS.CUST\_FNAME%TYPE;

v\_CUST\_SNAME CUSTOMERS.CUST\_SNAME%TYPE;

v\_DIVE\_NAME DIVE.DIVE\_NAME%TYPE;

BEGIN

SELECT C.CUST\_FNAME, C.CUST\_SNAME, D.DIVE\_NAME

INTO v\_CUST\_FNAME, v\_CUST\_SNAME, v\_DIVE\_NAME

FROM CUSTOMERS C

JOIN DIVE\_EVENT DE ON C.CUST\_ID = DE.CUST\_ID

JOIN DIVE D ON DE.DIVE\_ID = D.DIVE\_ID

WHERE C.CUST\_ID = p\_CUST\_ID

AND DE.DIVE\_DATE = p\_DIVE\_DATE;

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER DETAILS: ' || v\_CUST\_FNAME || ' ' || v\_CUST\_SNAME || ' booked for ' || v\_DIVE\_NAME || ' on ' || TO\_CHAR(p\_DIVE\_DATE, 'DD/Month/YY'));

EXCEPTION

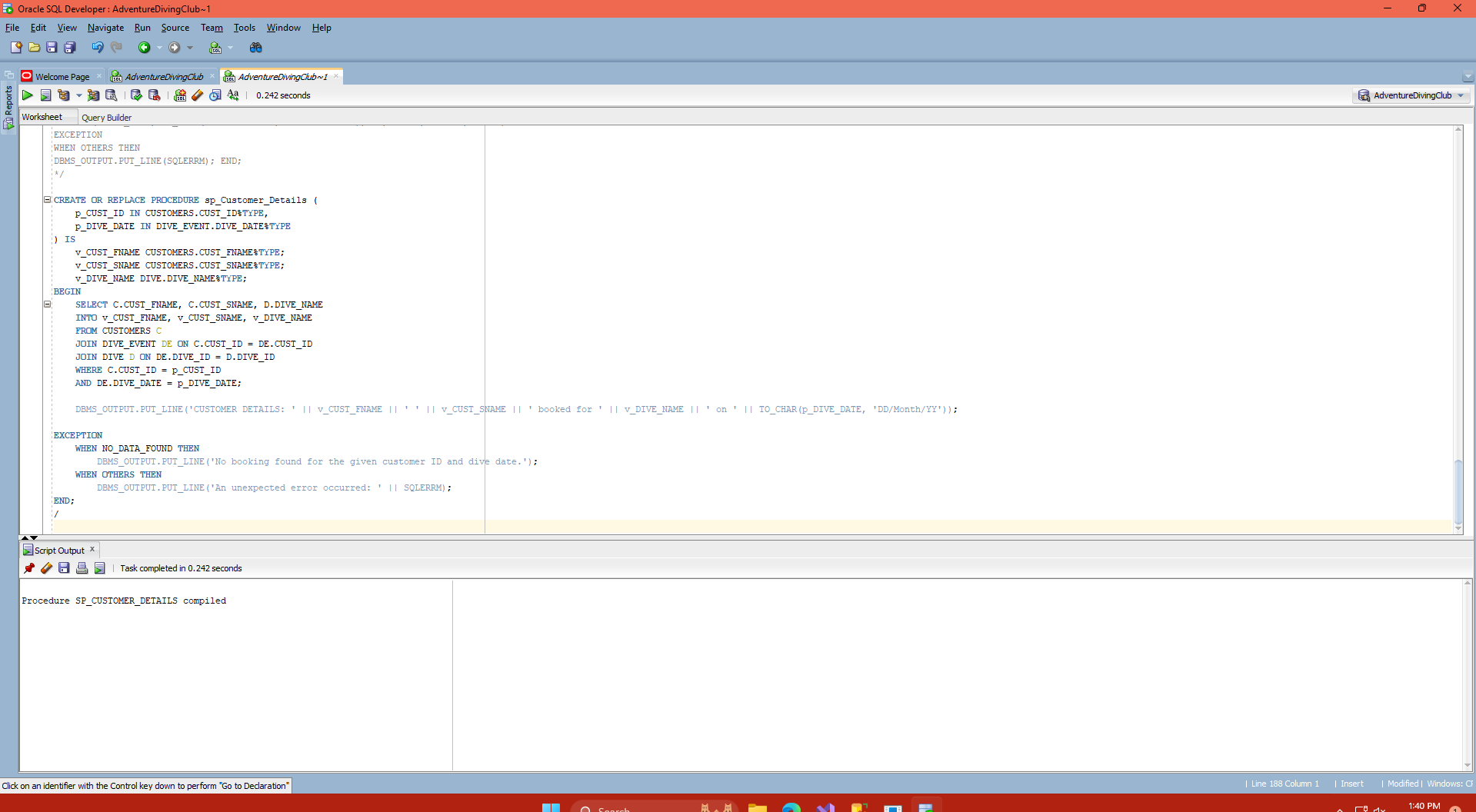
WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No booking found for the given customer ID and dive date.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An unexpected error occurred: ' || SQLERRM);

END;



(alex, 2021)

**QUESTION 9**

BEGIN

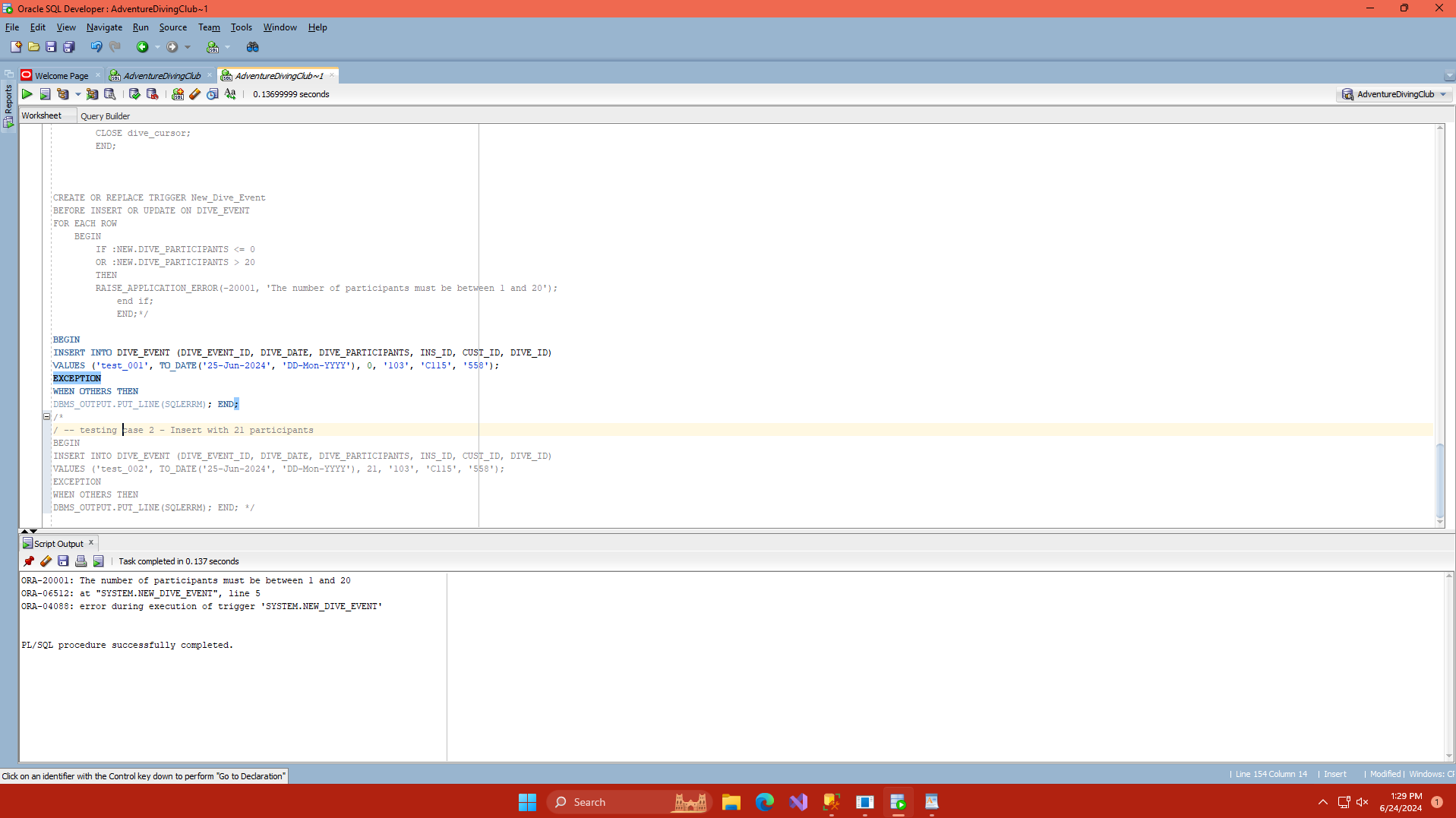
INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID, DIVE\_DATE, DIVE\_PARTICIPANTS, INS\_ID, CUST\_ID, DIVE\_ID)

VALUES ('test\_001', TO\_DATE('25-Jun-2024', 'DD-Mon-YYYY'), 0, '103', 'C115', '558');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE(SQLERRM); END;



/ -- testing case 2 - Insert with 21 participants

BEGIN

INSERT INTO DIVE\_EVENT (DIVE\_EVENT\_ID, DIVE\_DATE, DIVE\_PARTICIPANTS, INS\_ID, CUST\_ID, DIVE\_ID)

VALUES ('test\_002', TO\_DATE('25-Jun-2024', 'DD-Mon-YYYY'), 21, '103', 'C115', '558');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE(SQLERRM); END;

(LinkedIn Learning, 2022)

A screenshot of a computer

Description automatically generated

DECLARE

v\_revenue NUMBER;

BEGIN

v\_revenue := get\_dive\_event\_revenue('de\_101');

IF v\_revenue IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Total Revenue for the dive event: ' || v\_revenue);

END IF;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An unexpected error occurred while executing the function: ' || SQLERRM);

END;

QUESTION 11: MINI PRESENTATION

*Independent Institute of Education: Module Manual & Lecture Slides (2024)*

*Advanced SQL tutorial: Stored procedures + use cases* (2021) *YouTube*. Available at: https://youtu.be/NrBJmtD0kEw?si=6jkPHqSQWZJGJ2rI (Accessed: 24 June 2024).

*How to create insert trigger using SQL server* (2017) *YouTube*. Available at: https://youtu.be/KLJ29y3qqyk?si=Q99SHjPnAfSfxn4Z (Accessed: 24 June 2024).

*SQL tutorial - using functions* (2022) *YouTube*. Available at: https://youtu.be/8OCM0zL4QeI?si=\_lWaUuPUuiXmYqE0 (Accessed: 24 June 2024).