ADVANCED DATABASE

Practicum

Siyabonga Terry Jnr Maluleke
ST10202185

--Question 1

ST10202185_PRACTICUM.CUSTOMER

P *	CUST_ID	VARCHAR2 (26 BYTE)
	CUST_FNAME	VARCHAR2 (26 BYTE)
	CUST_SNAME	VARCHAR2 (26 BYTE)
	CUST_ADDRESS	VARCHAR2 (26 BYTE)
	CUST_CONTACT	VARCHAR2 (26 BYTE)

CUSTOMER_PK (CUST_ID)

CUSTOMER_PK (CUST_ID)

ST10202185_PRACTICUM.INSTRUCTOR

P *	INS_ID	NUMBER (38)
	INS_FNAME	VARCHAR2 (26 BYTE)
	INS_SNAME	VARCHAR2 (26 BYTE)
	INS_CONTACT	VARCHAR2 (26 BYTE)
	INS_LEVEL	NUMBER (38)

INSTRUCTOR_PK (INS_ID)

INSTRUCTOR_PK (INS_ID)

ST10202185_PRACTICUM.DIVE

DIVE_ID	NUMBER (38)
DIVE_NAME	VARCHAR2 (26 BYTE)
DIVE_DURATION	VARCHAR2 (26 BYTE)
DIVE_LOCATION	VARCHAR2 (26 BYTE)
DIVE_EXP_LEVEL	NUMBER (38)
DIVE_COST	NUMBER (38)

ST10202185_PRACTICUM.DIVE_EVENT

DIVE_EVENT_ID	VARCHAR2 (26 BYTE)
DIVE_DATE	DATE
DIVE_PARTICIPANTS	NUMBER (38)
INS_ID	NUMBER (38)
CUST_ID	VARCHAR2 (26 BYTE)
DIVE_ID	NUMBER (38)

--Question 2

```
CREATE ROLE ADMINISTRATOR;

GRANT ALL PRIVILEGES ON ST10202185_Practicum.* TO ADMINISTRATOR;

CREATE USER "JOHN SMITH" IDENTIFIED BY Password1;

GRANT ADMINISTRATOR TO "JOHN SMITH";


CREATE ROLE GENERAL_USER;

GRANT SELECT, INSERT, UPDATE, DELETE ON ST10202185_Practicum TO GENERAL_USER;

CREATE USER "SIYA" IDENTIFIED BY Password1;

GRANT GENERAL_USER TO "JOHN SMITH";
```

--Question 3

```sql
SELECT i.INS_FNAME || ' ' || i.INS_SNAME AS "INSTRUCTOR",

        c.CUST_FNAME || ' ' || c.CUST_SNAME AS "CUSTOMER",

        de.DIVE_ID,

        de.DIVE_DATE,

        de.DIVE_PARTICIPANTS AS "DIVE_PARTICIPANTS"

FROM Dive_Event de

JOIN Instructor I ON de.INS_ID = i.INS_ID

JOIN Customer c ON de.CUST_ID = c.CUST_ID

WHERE de.DIVE_PARTICIPANTS >= 8;
```

Query Result ×

SQL | All Rows Fetched: 3 in 0.028 seconds

	INSTRUCTOR	CUSTOMER	DIVE_ID	DIVE_DATE	DIVE_PARTICIPANTS
1	Bob Du Preez	Alex Hanson	552	18-JUL-17	8
2	Simon Jones	Heinrich Willis	554	25-JUL-17	10
3	Simon Jones	Thando Zolani	556	22-JUL-17	8

--Question 4

```sql
DECLARE

    V_dive_name VARCHAR2(100);

    V_dive_date DATE;

    V_participants NUMBER;

BEGIN

    FOR dive_rec IN (SELECT d.DIVE_NAME, de.DIVE_DATE, de.DIVE_PARTICIPANTS

                    FROM Dive_Event de

                    JOIN Dive d ON de.DIVE_ID = d.DIVE_ID
```

```
                    WHERE de.DIVE_PARTICIPANTS >= 10)

    LOOP

        V_dive_name := dive_rec.DIVE_NAME;

        V_dive_date := dive_rec.DIVE_DATE;

        V_participants := dive_rec.DIVE_PARTICIPANTS;


        DBMS_OUTPUT.PUT_LINE('DIVE NAME: ' || v_dive_name);

        DBMS_OUTPUT.PUT_LINE('DIVE DATE: ' || TO_CHAR(v_dive_date,
'DD/MON/YY'));

        DBMS_OUTPUT.PUT_LINE('PARTICIPANTS: ' || v_participants);

    END LOOP;

END;
```
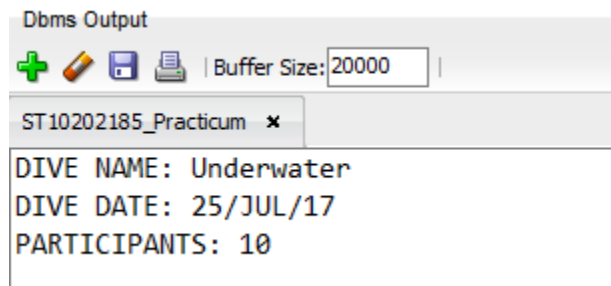
Dbms Output

➕ ✏️ 💾 🖨️ | Buffer Size: 20000 |

ST10202185_Practicum ✕

```
DIVE NAME: Underwater
DIVE DATE: 25/JUL/17
PARTICIPANTS: 10
```

```
--Question 5


DECLARE

    V_customer_name VARCHAR2(100);

    V_dive_name VARCHAR2(100);

    V_participants NUMBER;

    V_instructors_required NUMBER;

BEGIN
```

```sql
    FOR dive_rec IN (SELECT c.CUST_FNAME || ', ' || c.CUST_SNAME AS
customer_name,
                            d.DIVE_NAME,
                            de.DIVE_PARTICIPANTS,
                            d.DIVE_COST
                    FROM Dive_Event de
                    JOIN Customer c ON de.CUST_ID = c.CUST_ID
                    JOIN Dive d ON de.DIVE_ID = d.DIVE_ID
                    WHERE d.DIVE_COST > 500)
    LOOP
        V_customer_name := dive_rec.customer_name;

        V_dive_name := dive_rec.DIVE_NAME;

        V_participants := dive_rec.DIVE_PARTICIPANTS;


        IF V_participants <= 4 THEN

            V_instructors_required := 1;

        ELSIF V_participants BETWEEN 5 AND 7 THEN

            V_instructors_required := 2;

        ELSE

            V_instructors_required := 3;

        END IF;


        DBMS_OUTPUT.PUT_LINE('CUSTOMER: ' || V_customer_name);

        DBMS_OUTPUT.PUT_LINE('DIVE NAME: ' || V_dive_name);

        DBMS_OUTPUT.PUT_LINE('PARTICIPANTS: ' || V_participants);

        DBMS_OUTPUT.PUT_LINE('STATUS: ' || V_instructors_required || '
instructors required.');

    END LOOP;

END;
```

```
CUSTOMER: Heinrich, Willis
DIVE NAME: Underwater
PARTICIPANTS: 10
STATUS: 3 instructors required.
CUSTOMER: Alex, Hanson
DIVE NAME: Wave
PARTICIPANTS: 8
STATUS: 3 instructors required.
CUSTOMER: Thando, Zolani
DIVE NAME: Rough Seas
PARTICIPANTS: 8
STATUS: 3 instructors required.
CUSTOMER: Sarah, Jones
DIVE NAME: Wave
PARTICIPANTS: 5
STATUS: 2 instructors required.
```

--Question 6

CREATE VIEW Vw_Dive_Event AS

SELECT de.INS_ID,

       de.CUST_ID,

       c.CUST_ADDRESS,

       d.DIVE_DURATION,

       de.DIVE_DATE

FROM Dive_Event de

JOIN Customer c ON de.CUST_ID = c.CUST_ID

JOIN Dive d ON de.DIVE_ID = d.DIVE_ID

```
WHERE de.DIVE_DATE < TO_DATE('2017-07-19', 'YYYY-MM-DD');
```

View VW_DIVE_EVENT created.

--Question 7

```
CREATE OR REPLACE TRIGGER New_Dive_Event

BEFORE INSERT ON Dive_Event

FOR EACH ROW

BEGIN

    IF :NEW.DIVE_PARTICIPANTS <= 0 OR :NEW.DIVE_PARTICIPANTS > 20 THEN

        RAISE_APPLICATION_ERROR(-20001, 'Invalid number of participants.
Please enter a value between 1 and 20.');

    END IF;

END;


INSERT INTO Dive_Event (DIVE_EVENT_ID, DIVE_DATE, DIVE_PARTICIPANTS, INS_ID,
CUST_ID, DIVE_ID)

VALUES ('de110', '2017-07-30', 0, '101', 'C115', 554);
```
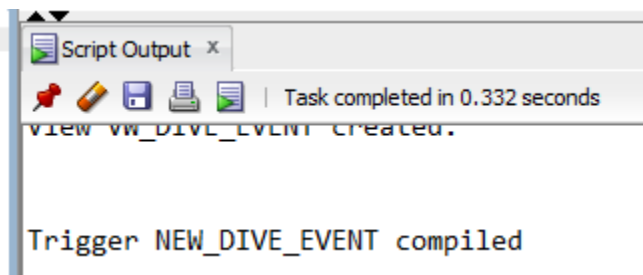
Script Output ×

Task completed in 0.332 seconds

View VW_DIVE_EVENT created.

Trigger NEW_DIVE_EVENT compiled

```
--Question 8

CREATE OR REPLACE PROCEDURE sp_Customer_Details(
    P_cust_id IN VARCHAR2,
    P_dive_date IN DATE
)
IS
    V_customer_name VARCHAR2(100);
    V_dive_name VARCHAR2(100);
BEGIN
    SELECT c.CUST_FNAME || ' ' || c.CUST_SNAME, d.DIVE_NAME
    INTO V_customer_name, V_dive_name
    FROM Dive_Event de
    JOIN Customer c ON de.CUST_ID = c.CUST_ID
    JOIN Dive d ON de.DIVE_ID = d.DIVE_ID
    WHERE de.CUST_ID = P_cust_id
    AND de.DIVE_DATE = P_dive_date;


    DBMS_OUTPUT.PUT_LINE('CUSTOMER DETAILS: ' || V_customer_name || ' booked
for the ' || V_dive_name || ' on the ' || TO_CHAR(P_dive_date, 'DD/MON/YY')
|| '.');
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No customer found for the given customer ID and
dive date.');
END;
/
```
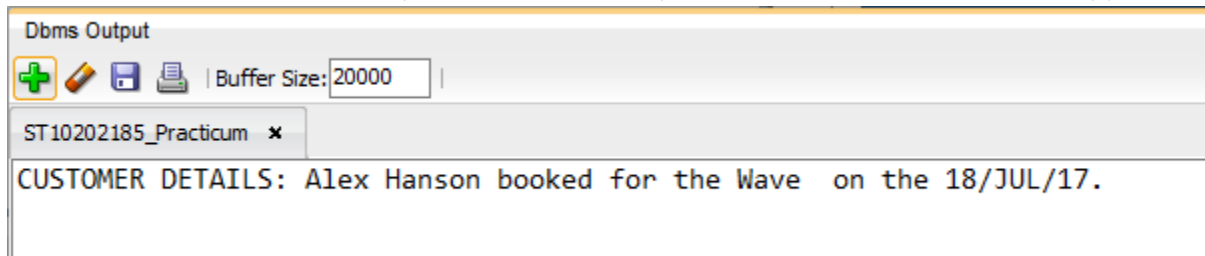
```
EXEC sp_Customer_Details('C118', TO_DATE('2017-07-18', 'YYYY-MM-DD'));
```

Dbms Output

➕ ✏️ 💾 🖨️ | Buffer Size: 20000 |

ST10202185_Practicum ✖

CUSTOMER DETAILS: Alex Hanson booked for the Wave on the 18/JUL/17.
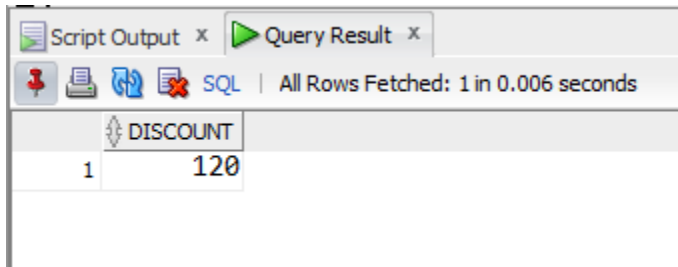
--Question 9

```
Function:CREATE OR REPLACE FUNCTION calculate_discount(
    P_price IN NUMBER
) RETURN NUMBER
IS
    V_discount NUMBER;
BEGIN
    IF p_price <= 500 THEN
        V_discount := p_price * 0.1; -- 10% discount
    ELSE
        V_discount := p_price * 0.2; -- 20% discount
    END IF;

    RETURN v_discount;
EXCEPTION
    WHEN OTHERS THEN
        RETURN NULL; -- Return NULL in case of any exception
END;
SELECT calculate_discount(600) AS discount FROM dual;
```

Function CALCULATE_DISCOUNT compiled

>>Query Run In:Query Result

Script Output ×   Query Result ×

SQL | All Rows Fetched: 1 in 0.006 seconds

	DISCOUNT
1	120

--Question10

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Report Viewer</title>

</head>

<body>

    <h1>Report Viewer</h1>

    <button onclick="displayCustomerDetails()">Customer Details</button>

    <button onclick="displayDiveAdjustments()">Dive Adjustments</button>


    <script>

        Function displayCustomerDetails() {

            // Code to display customer details report

            Alert("Displaying Customer Details Report…");
```

```
            }


        Function displayDiveAdjustments() {

            // Code to display dive adjustments report

            Alert("Displaying Dive Adjustments Report…");

        }

    </script>

</body>

</html>
```