Advanced Databases Ice Task 4

```
--Question 1
-- Create the CUSTOMER table
CREATE TABLE CUSTOMER (
   customer_id VARCHAR(10) PRIMARY KEY,
   fullname VARCHAR(50) NOT NULL,
   contact VARCHAR(15) NOT NULL,
   email VARCHAR(50) NOT NULL
);

-- Create the OUTLET table
CREATE TABLE OUTLET (
   outlet_id VARCHAR(10) PRIMARY KEY,
   floor VARCHAR(5) NOT NULL,
   outlet VARCHAR(50) NOT NULL,
   floor_size VARCHAR(20) NOT NULL
);

-- Create the RENTALAGREEMENT table
CREATE TABLE RENTALAGREEMENT (
   rental_id VARCHAR(10) PRIMARY KEY,
   amount VARCHAR(10) NOT NULL,
   rental_date DATE NOT NULL,
   customer_id VARCHAR(10),
   outlet_id VARCHAR(10),
   FOREIGN KEY (customer_id) REFERENCES CUSTOMER(customer_id),
```

```
   FOREIGN KEY (outlet_id) REFERENCES OUTLET(outlet_id)
);

-- Insert data into CUSTOMER table
INSERT INTO CUSTOMER (customer_id, fullname, contact, email)
VALUES
('101123', 'Cory Coleman', '076 465 1256', 'cc@zsat.co.za'),
('101124', 'Raquel Turner', '072 545 4587', 'rt@ytd.co.za'),
('101125', 'Daniella Perkins', '083 254 9871', 'perd@zsat.co.za'),
('101126', 'Nikhil Michael', '071 554 4589', 'nik@m@ytd.co.za'),
('101127', 'Lucy Moore', '073 555 8987', 'lumor@klm.co.za');

-- Insert data into OUTLET table
INSERT INTO OUTLET (outlet_id, floor, outlet, floor_size)
VALUES
('OUT1', '1', 'A', '20mx60m'),
('OUT2', '1', 'B', '18mx50m'),
('OUT3', '2', 'A', '25mx60m'),
('OUT4', '2', '1', '15mx40m'),
('OUT5', '3', 'A', '20mx60m');

-- Insert data into RENTALAGREEMENT table
INSERT INTO RENTALAGREEMENT (rental_id, amount, rental_date, customer_id, outlet_id)
VALUES
('A101', 'R5 800', TO_DATE('03-09-2019', 'DD-MM-YYYY'), '101123', 'OUT2'),
('A102', 'R3 000', TO_DATE('05-09-2019', 'DD-MM-YYYY'), '101125', 'OUT3'),
```

```
('A103', 'R5 995', TO_DATE('09-09-2019', 'DD-MM-YYYY'), '101127', 'OUT1');



--Question 2
SELECT
   c.fullname AS CUSTOMER,
   o.floor_size AS FLOOR_SIZE,
   r.amount AS AMOUNT
FROM
   CUSTOMER c
JOIN
   RENTALAGREEMENT r ON c.customer_id = r.customer_id
JOIN
   OUTLET o ON r.outlet_id = o.outlet_id
ORDER BY
   TO_NUMBER(SUBSTR(r.amount, 2)) ASC;


--Question 3
SELECT
   c.customer_id AS CUSTOMERID,
   c.contact AS CONTACT_NUMBER,
   r.outlet_id AS OUTLET_ID,
   r.amount AS ORIGINAL_AMOUNT,
   TO_NUMBER(SUBSTR(r.amount, 2)) * 0.20 AS INCREASE_AMOUNT,
   TO_NUMBER(SUBSTR(r.amount, 2)) * 1.20 AS TOTAL_INCREASED_AMOUNT
FROM
```

```
    CUSTOMER c
JOIN
    RENTALAGREEMENT r ON c.customer_id = r.customer_id
ORDER BY
    c.customer_id;



--Question 4
DECLARE
    CURSOR rental_cursor IS
        SELECT
            c.fullname AS customer_name,
            o.floor_size AS floor_size,
            TO_CHAR(r.rental_date, 'DD-MON-YY') AS rental_date
        FROM
            CUSTOMER c
        JOIN
            RENTALAGREEMENT r ON c.customer_id = r.customer_id
        JOIN
            OUTLET o ON r.outlet_id = o.outlet_id
        ORDER BY
            r.rental_date;

    rental_record rental_cursor%ROWTYPE;
BEGIN
    -- Open the cursor
```

```
  OPEN rental_cursor;

  -- Loop through each record
  LOOP
    FETCH rental_cursor INTO rental_record;
    EXIT WHEN rental_cursor%NOTFOUND;

    -- Display the results
    DBMS_OUTPUT.PUT_LINE('CUSTOMER NAME: ' || rental_record.customer_name);
    DBMS_OUTPUT.PUT_LINE('FLOOR SIZE: ' || rental_record.floor_size);
    DBMS_OUTPUT.PUT_LINE('RENTAL DATE: ' || rental_record.rental_date);
    DBMS_OUTPUT.PUT_LINE('----------------------------');
  END LOOP;

  -- Close the cursor
  CLOSE rental_cursor;
END;
/

--Question 5
DECLARE
  CURSOR rental_cursor IS
    SELECT
      r.customer_id,
      r.outlet_id,
      TO_CHAR(r.rental_date, 'DD-MON-YY') AS rental_date
```

```
    FROM
      RENTALAGREEMENT r
    WHERE
      r.rental_date BETWEEN TO_DATE('05-SEP-2019', 'DD-MON-YYYY') AND TO_DATE('10-SEP-2019', 'DD-MON-YYYY')
    ORDER BY
      r.rental_date DESC;

  rental_record rental_cursor%ROWTYPE;
BEGIN
  -- Open the cursor
  OPEN rental_cursor;

  -- Loop through each record
  LOOP
    FETCH rental_cursor INTO rental_record;
    EXIT WHEN rental_cursor%NOTFOUND;

    -- Display the results
    DBMS_OUTPUT.PUT_LINE('CUSTOMER ID: ' || rental_record.customer_id);
    DBMS_OUTPUT.PUT_LINE('OUTLET ID: ' || rental_record.outlet_id);
    DBMS_OUTPUT.PUT_LINE('RENTAL DATE: ' || rental_record.rental_date);
    DBMS_OUTPUT.PUT_LINE('----------------------------');
  END LOOP;

  -- Close the cursor
```

```
    CLOSE rental_cursor;
END;
/


--Question 6
CREATE VIEW vwincreaseAmount AS
SELECT
    c.customer_id AS CUSTOMER_ID,
    r.outlet_id AS OUTLET_ID,
    o.floor_size AS FLOOR_SIZE,
    r.amount AS ORIGINAL_AMOUNT,
    TO_NUMBER(SUBSTR(r.amount, 2)) * 1.10 AS INCREASED_AMOUNT
FROM
    CUSTOMER c
JOIN
    RENTALAGREEMENT r ON c.customer_id = r.customer_id
JOIN
    OUTLET o ON r.outlet_id = o.outlet_id;
```

```sql
--Question 2
SELECT
    c.fullname AS CUSTOMER,
    o.floor_size AS FLOOR_SIZE,
    r.amount AS AMOUNT
FROM
    CUSTOMER c
JOIN
    RENTALAGREEMENT r ON c.customer_id = r.customer_id
JOIN
    OUTLET o ON r.outlet_id = o.outlet_id
ORDER BY
    TO_NUMBER(SUBSTR(r.amount, 2)) ASC;

--Question 3
SELECT
    c.customer_id AS CUSTOMERID,
    c.contact AS CONTACT_NUMBER,
    r.outlet_id AS OUTLET_ID,
    r.amount AS ORIGINAL_AMOUNT,
    TO_NUMBER(SUBSTR(r.amount, 2)) * 0.20 AS INCREASE_AMOUNT,
```

Script Output x | Query Result x

SQL | All Rows Fetched: 0 in 0.003 seconds

CUSTOMER	FLOOR_S...	AMOUNT

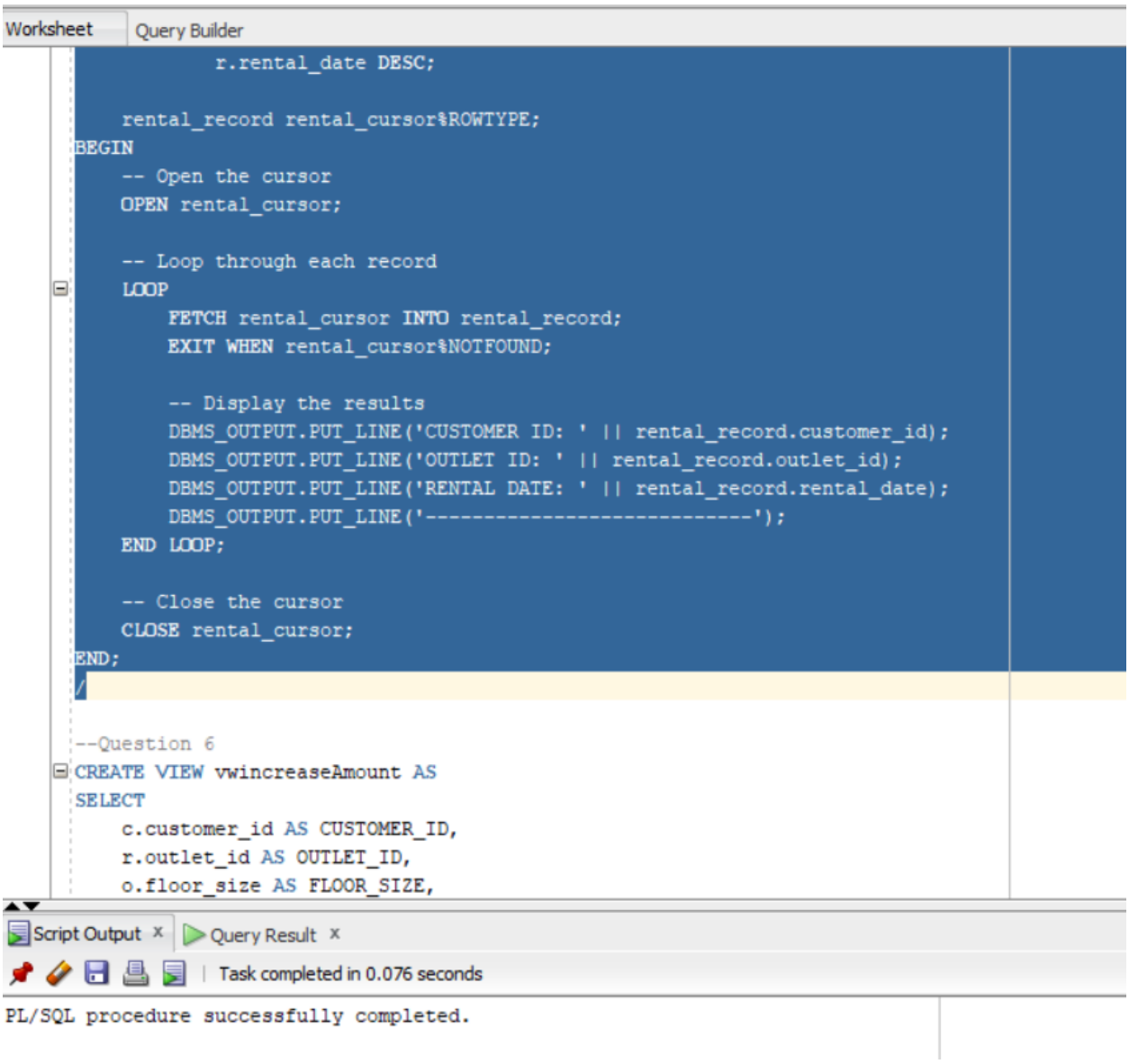
```sql
--Question 3
SELECT
    c.customer_id AS CUSTOMERID,
    c.contact AS CONTACT_NUMBER,
    r.outlet_id AS OUTLET_ID,
    r.amount AS ORIGINAL_AMOUNT,
    TO_NUMBER(SUBSTR(r.amount, 2)) * 0.20 AS INCREASE_AMOUNT,
    TO_NUMBER(SUBSTR(r.amount, 2)) * 1.20 AS TOTAL_INCREASED_AMOUNT
FROM
    CUSTOMER c
JOIN
    RENTALAGREEMENT r ON c.customer_id = r.customer_id
ORDER BY
    c.customer_id;


--Question 4
DECLARE
    CURSOR rental_cursor IS
        SELECT
            c.fullname AS customer_name,
            o.floor_size AS floor_size,
```

ot Output x | Query Result x

SQL | All Rows Fetched: 0 in 0.009 seconds

CUSTOME...	CONTACT...	OUTLET_ID	ORIGINAL...	INCREAS...	TOTAL_IN...

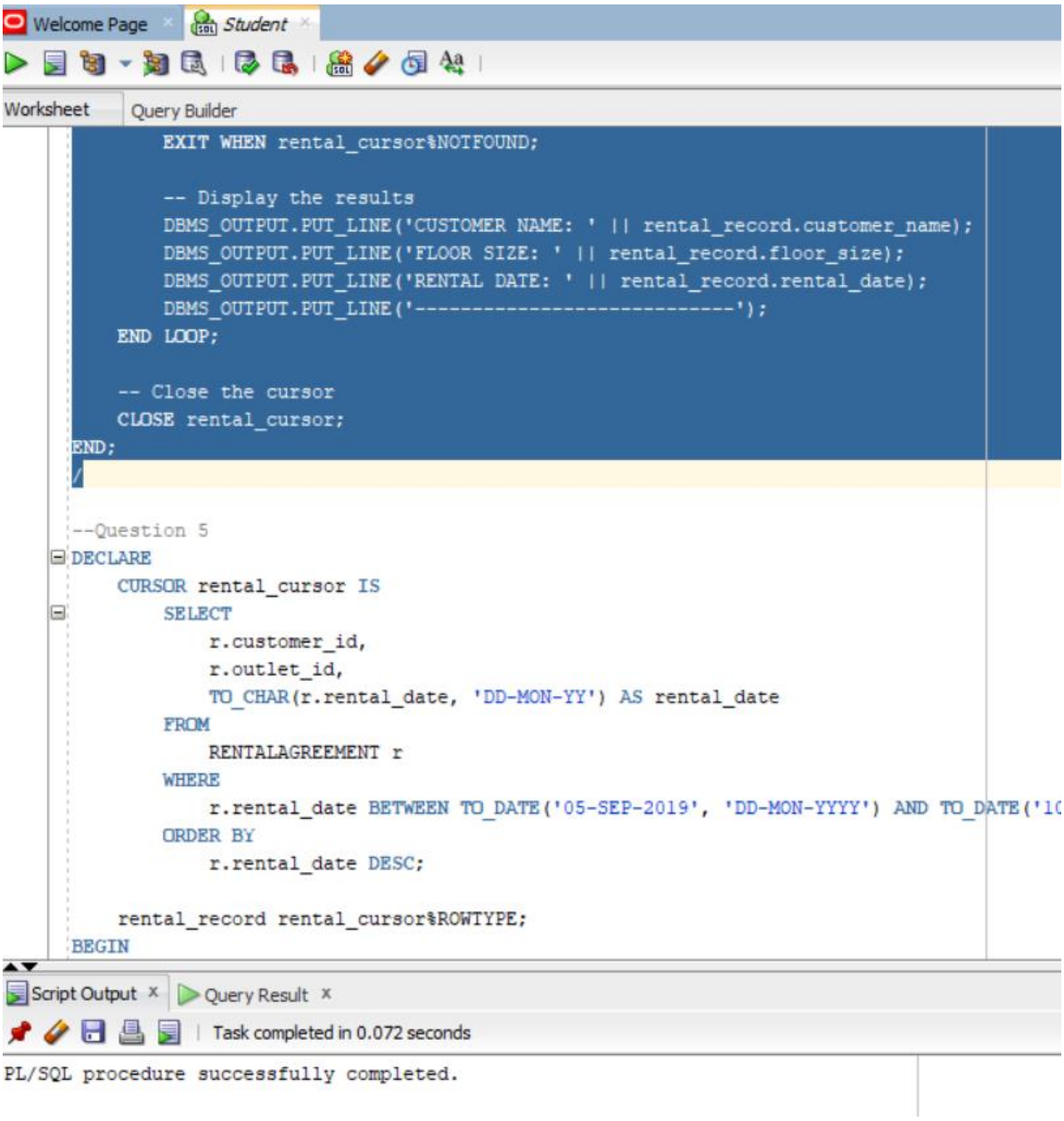

Worksheet
Query Builder
r.rental_date DESC;
rental_record rental_cursor%ROWTYPE;
BEGIN
-- Open the cursor
OPEN rental_cursor;
-- Loop through each record
LOOP
FETCH rental_cursor INTO rental_record;
EXIT WHEN rental_cursor%NOTFOUND;
-- Display the results
DBMS_OUTPUT.PUT_LINE('CUSTOMER ID: ' || rental_record.customer_id);
DBMS_OUTPUT.PUT_LINE('OUTLET ID: ' || rental_record.outlet_id);
DBMS_OUTPUT.PUT_LINE('RENTAL DATE: ' || rental_record.rental_date);
DBMS_OUTPUT.PUT_LINE('----------');
END LOOP;
-- Close the cursor
CLOSE rental_cursor;
END;
/
--Question 6
CREATE VIEW vwincreaseAmount AS
SELECT
c.customer_id AS CUSTOMER_ID,
r.outlet_id AS OUTLET_ID,
o.floor_size AS FLOOR_SIZE,
Script Output
Query Result
Task completed in 0.076 seconds
PL/SQL procedure successfully completed.

```sql
--Question 6
CREATE VIEW vwincreaseAmount AS
SELECT
    c.customer_id AS CUSTOMER_ID,
    r.outlet_id AS OUTLET_ID,
    o.floor_size AS FLOOR_SIZE,
    r.amount AS ORIGINAL_AMOUNT,
    TO_NUMBER(SUBSTR(r.amount, 2)) * 1.10 AS INCREASED_AMOUNT
FROM
    CUSTOMER c
JOIN
    RENTALAGREEMENT r ON c.customer_id = r.customer_id
JOIN
    OUTLET o ON r.outlet_id = o.outlet_id;
```

Script Output x | Query Result x

Task completed in 0.093 seconds

```
View VWINCREASEAMOUNT created.
```


Welcome Page
Student
Worksheet
Query Builder
EXIT WHEN rental_cursor%NOTFOUND;
-- Display the results
DBMS_OUTPUT.PUT_LINE('CUSTOMER NAME: ' || rental_record.customer_name);
DBMS_OUTPUT.PUT_LINE('FLOOR SIZE: ' || rental_record.floor_size);
DBMS_OUTPUT.PUT_LINE('RENTAL DATE: ' || rental_record.rental_date);
DBMS_OUTPUT.PUT_LINE('----------');
END LOOP;
-- Close the cursor
CLOSE rental_cursor;
END;
/
--Question 5
DECLARE
CURSOR rental_cursor IS
SELECT
r.customer_id,
r.outlet_id,
TO_CHAR(r.rental_date, 'DD-MON-YY') AS rental_date
FROM
RENTALAGREEMENT r
WHERE
r.rental_date BETWEEN TO_DATE('05-SEP-2019', 'DD-MON-YYYY') AND TO_DATE('1
ORDER BY
r.rental_date DESC;
rental_record rental_cursor%ROWTYPE;
BEGIN
Script Output
Query Result
Task completed in 0.072 seconds
PL/SQL procedure successfully completed.