ADDB 7311

Advanced Databases

Assignment 1

Nivad Ramdass

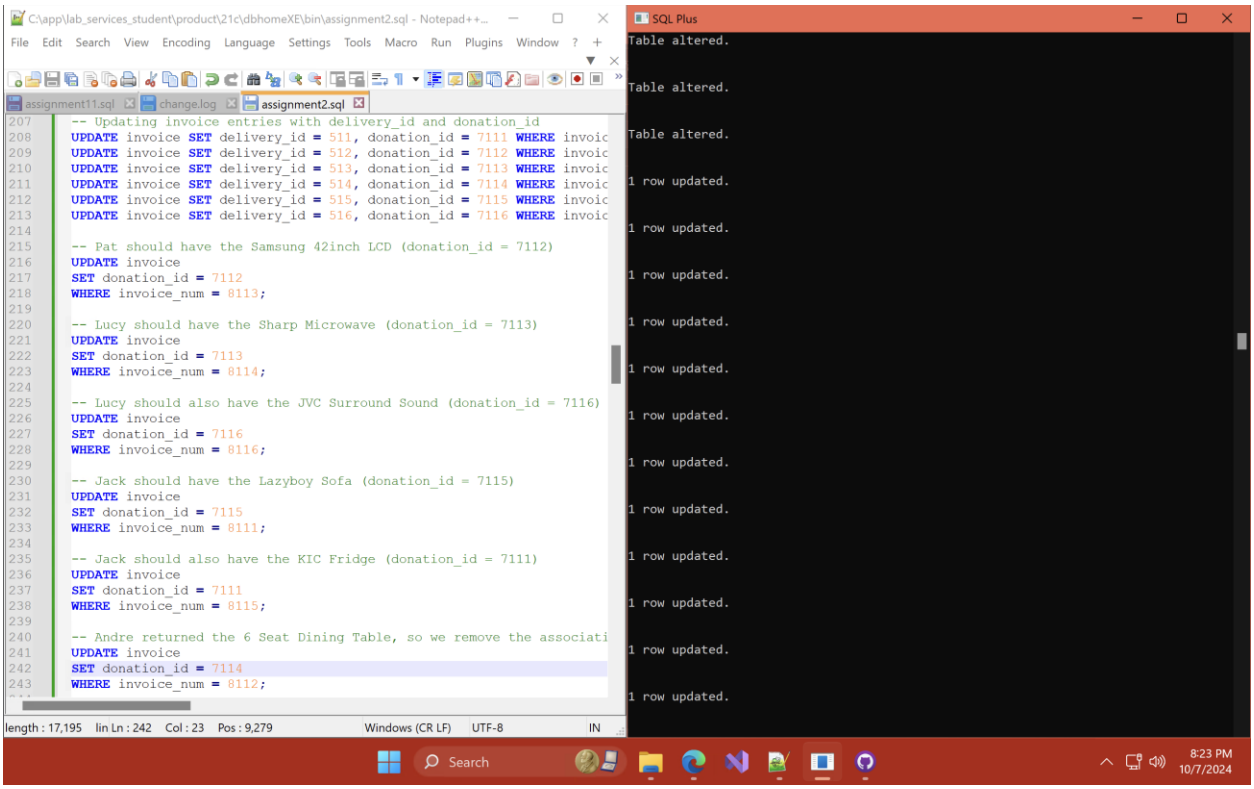
By submitting this assignment, I acknowledge that I have read and understood all the rules as per the terms in the registration contract, in particular the assignment and assessment rules in The IIE Assessment Strategy and Policy (IE009), the intellectual integrity and plagiarism rules in the Intellectual Integrity and Property Rights Policy (IE023), as well as any rules and regulations published in the student portal.

Plagiarism Statements

Statement	Signature	Statement	Signature
I have read the assessment rules provided in this declaration.		I have not shared this assessment with any other student.	
This assessment is my own work.		I have not presented the work of published sources as my own work.	
I have not copied any other student's work in this assessment.		I have correctly cited all my sources of information.	
I have not uploaded the assessment question to any website or App offering assessment assistance.		My referencing is technically correct, consistent, and congruent.	
I have not downloaded my assessment response from a website.		I have acted in an academically honest way in this assessment.	
I have not used any AI tool without reviewing, re-writing, and re-working this information, and referencing any AI tools in my work.			

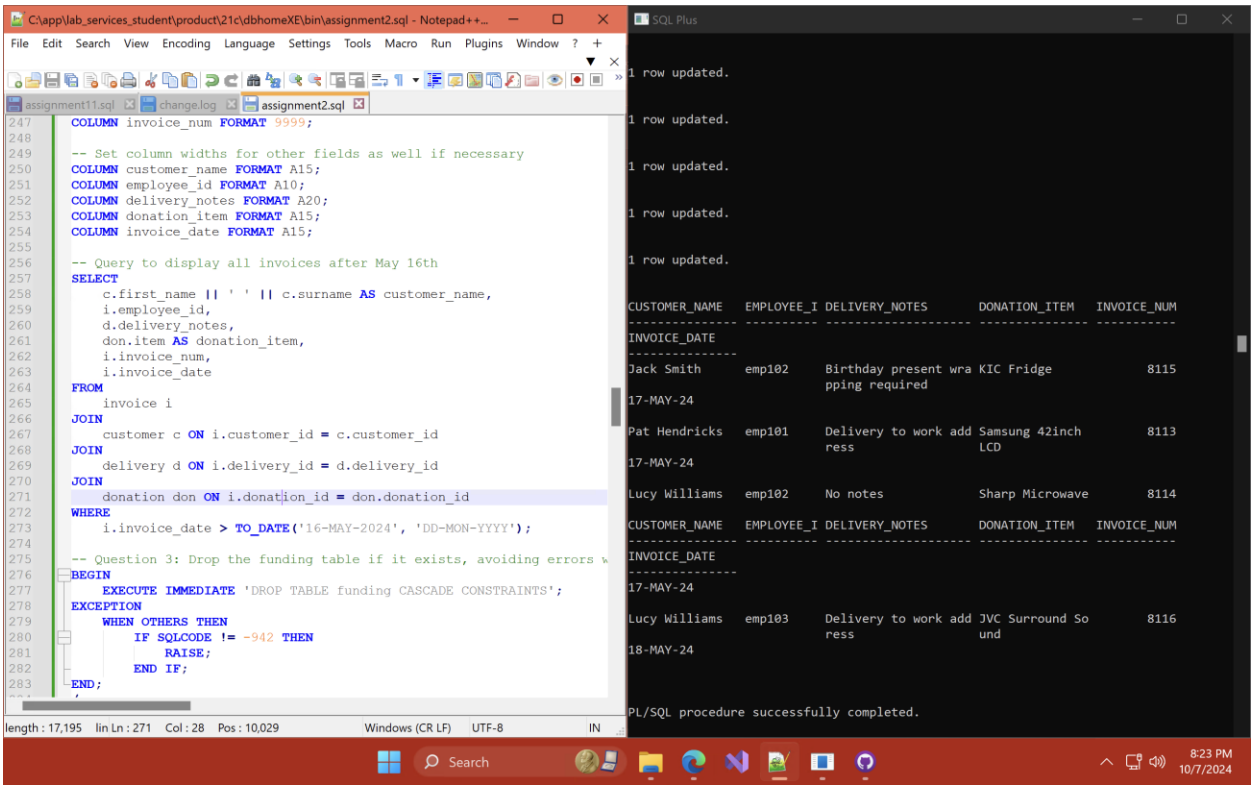
Question 1: Dropping Tables

For this question, I wrote PL/SQL blocks using EXECUTE IMMEDIATE to drop existing tables if they existed. I used exception handling to ensure no errors were raised if the tables were already dropped.



Question 2: Creating and Populating Tables

I created several tables including customer, employee, donator, donation, delivery, and invoice, then inserted sample data into each. Additionally, I added donation_id and delivery_id columns to the invoice table to link it with the donation and delivery tables, ensuring proper relationships between these entities.

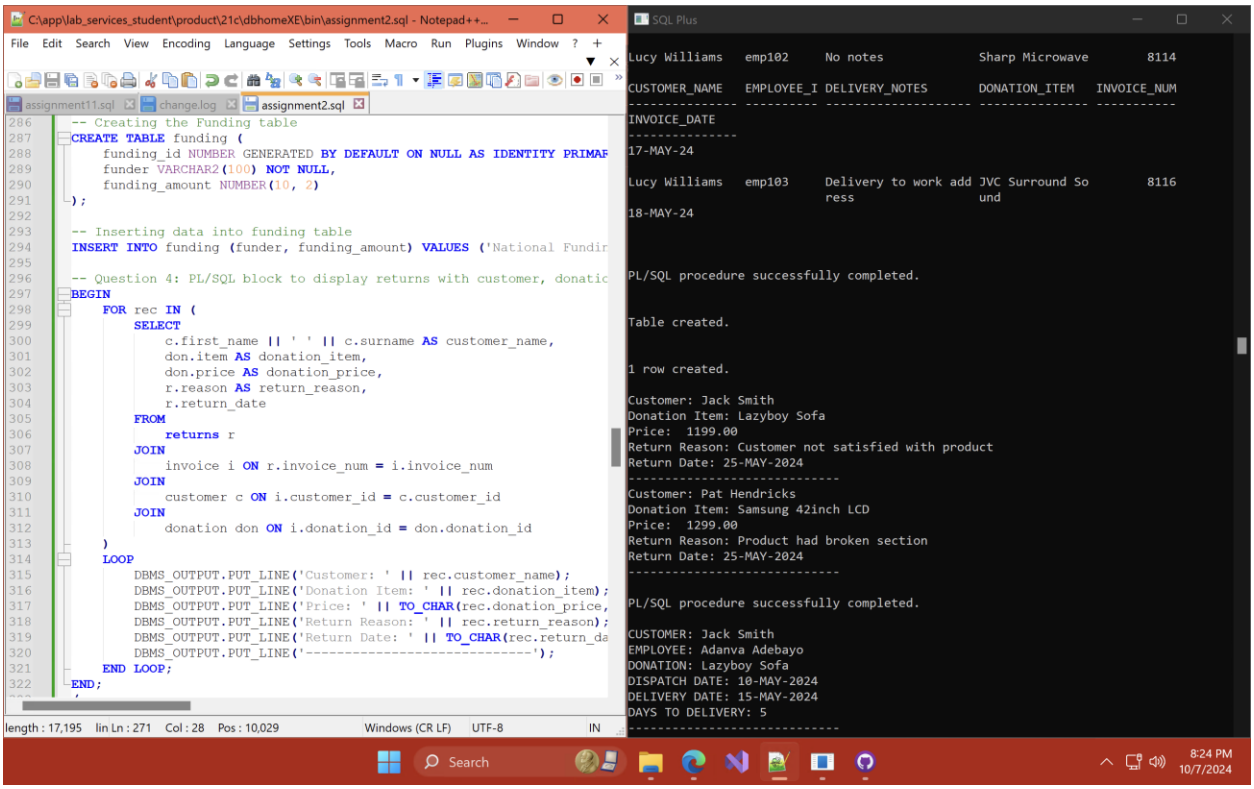


Question 3: Dropping and Creating the Funding Table

I dropped the funding table (if it existed) using EXECUTE IMMEDIATE, and then created a new funding table with an auto-incrementing funding_id. I inserted a row into the table to demonstrate its functionality.

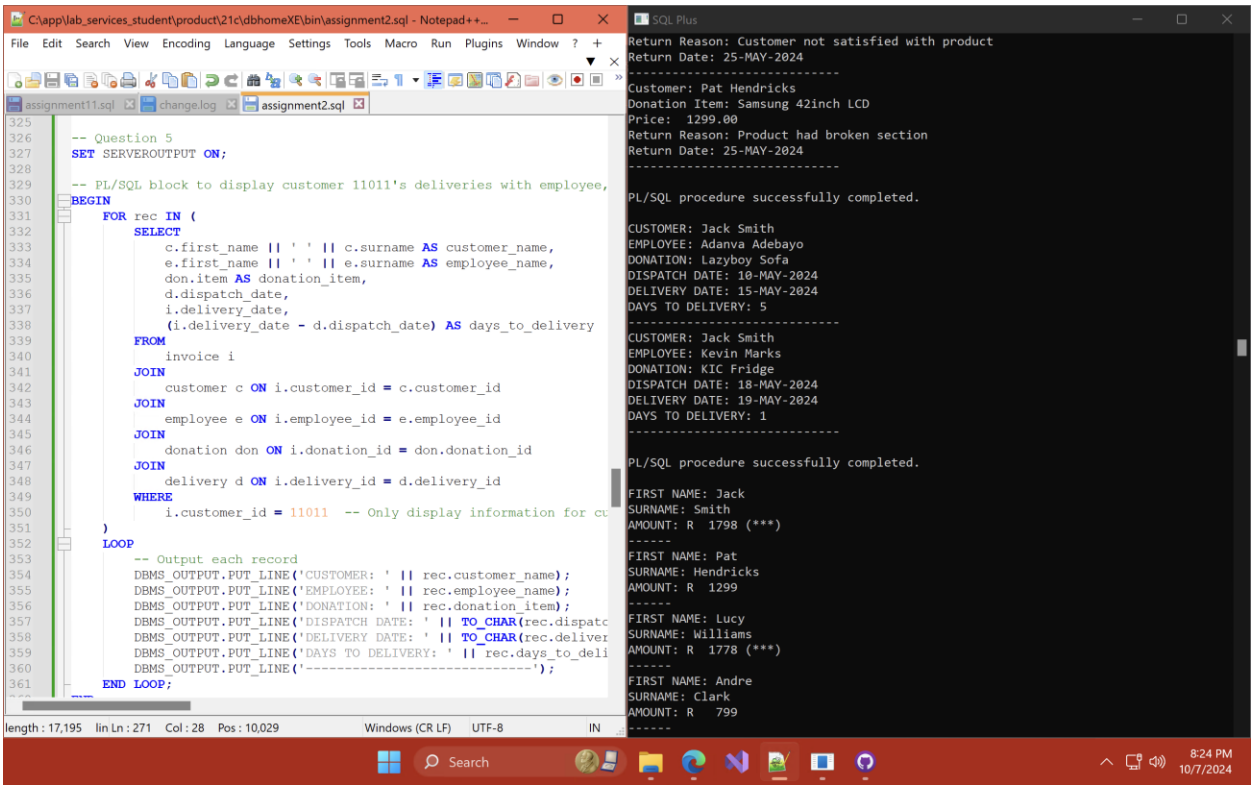
Question 4: Displaying Returns

In this question, I used a PL/SQL block with a FOR loop to query customer returns. The query joined the customer, donation, and returns tables to display the customer's name, donation item, price, and reason for the return, using DBMS_OUTPUT to display the information.



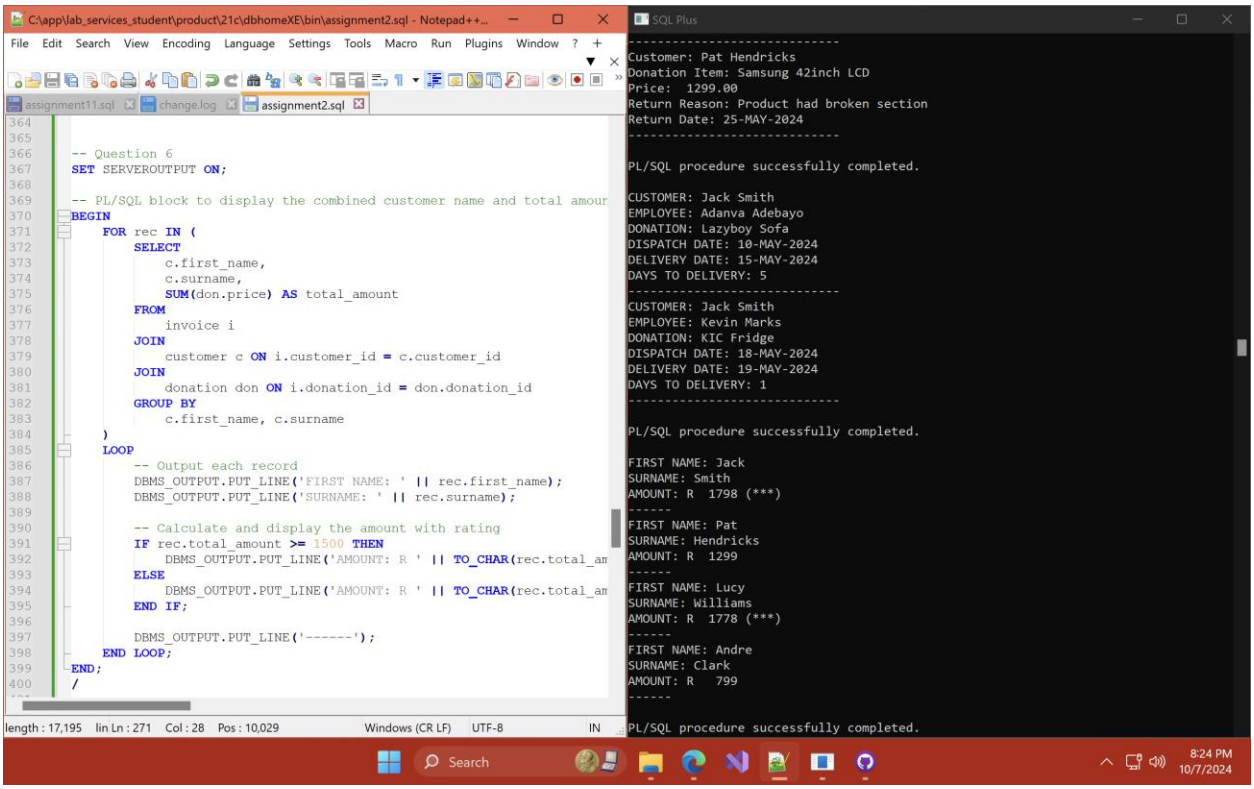
Question 5: Customer Deliveries

I wrote a PL/SQL block to display deliveries for a specific customer (Jack). The block retrieved the customer's name, employee handling the delivery, the donation item, and the delivery dates, along with the number of days between dispatch and delivery. This information was output using DBMS_OUTPUT.



Question 6: Total Amount Spent by Each Customer

This PL/SQL block aggregated the total amount spent by each customer based on their donations. I grouped by the customer's name and used SUM to calculate the total donation value, applying a rating system for customers based on the amount they spent. The results were displayed using DBMS_OUTPUT.



Question 7.1: %TYPE Attribute Example

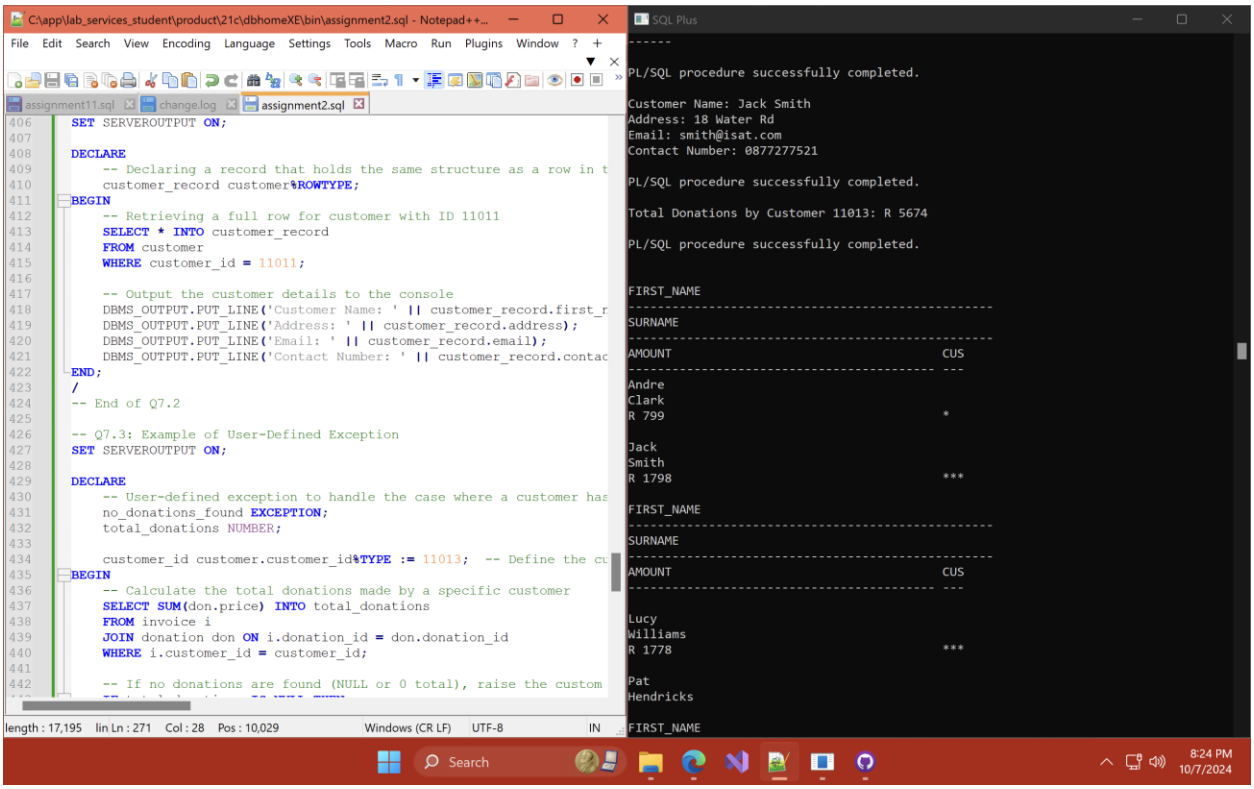
In this question, we used the %TYPE attribute to declare a variable of the same data type as the email column in the customer table. We assigned a specific customer ID (Jack's customer_id = 11011) and retrieved his email address using a SELECT INTO query, then displayed it using DBMS_OUTPUT.

Question 7.2: %ROWTYPE Attribute Example

Here, we used the %ROWTYPE attribute to declare a record that mirrors the structure of a row from the customer table. We retrieved all of Jack's details (name, address, email, etc.) and displayed them using DBMS_OUTPUT.

Question 7.3: User-Defined Exception Example

This question involved creating a custom PL/SQL exception. We queried the total donations for a customer and raised the custom no_donations_found exception if the customer had not made any donations. The exception was handled with a DBMS_OUTPUT message indicating no donations were made.



Question 8: Customer Ratings and Amounts

We used a query to calculate the total amount spent by each customer on donations and applied a rating system based on the total amount. The query included a CASE statement to assign star ratings and displayed the customer's name, total amount spent, and their rating.

C:\app\lab_services_student\product\21c\dbhomeXE\bin\assignment2.sql - Notepad++...

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? +

assignment11.sql | change.log | assignment2.sql

```sql
-- Question 8
SELECT
    c.first_name,
    c.surname,
    'R ' || SUM(don.price) AS amount,
    CASE
        WHEN SUM(don.price) >= 1500 THEN '***'  -- 3-star rating
        WHEN SUM(don.price) BETWEEN 1000 AND 1499 THEN '**'  -- 2-sta
        ELSE '*'  -- 1-star rating
    END AS customer_rating
FROM
    invoice i
JOIN
    customer c ON i.customer_id = c.customer_id
JOIN
    donation don ON i.donation_id = don.donation_id
GROUP BY
    c.first_name, c.surname
ORDER BY
    c.first_name;


SELECT
    c.first_name || ' ' || c.surname AS customer_name,
    don.item AS donation_item,
    i.invoice_date
FROM
    customer c
JOIN
    invoice i ON c.customer_id = i.customer_id
JOIN
    donation don ON i.donation_id = don.donation_id
ORDER BY
    c.first_name, c.surname;
```

length : 17,195 lin Ln : 271 Col : 28 Pos : 10,029 Windows (CR LF) UTF-8 IN

SQL Plus

```
Contact Number: 0877277521

PL/SQL procedure successfully completed.

Total Donations by Customer 11013: R 5674

PL/SQL procedure successfully completed.


FIRST_NAME
--------------------------------------------------
SURNAME
--------------------------------------------------
AMOUNT                                    CUS
----------------------------------------- ---
Andre
Clark
R 799                                     *

Jack
Smith
R 1798                                    ***

FIRST_NAME
--------------------------------------------------
SURNAME
--------------------------------------------------
AMOUNT                                    CUS
----------------------------------------- ---

Lucy
Williams
R 1778                                    ***

Pat
Hendricks

FIRST_NAME
--------------------------------------------------
SURNAME
--------------------------------------------------
AMOUNT                                    CUS
----------------------------------------- ---
R 1299                                    **
```

Search 8:24 PM 10/7/2024