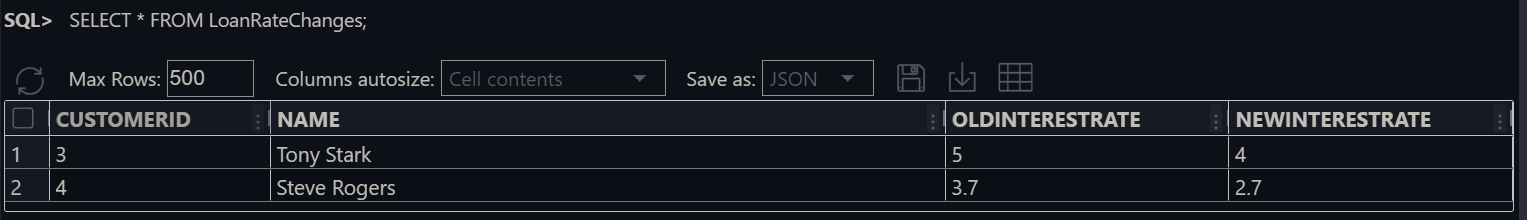
**Exercise 1: Control Structures**

* **Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old.
* **Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates
* **Table Creation**: Creates a *LoanRateChange*s table to record customer loan interest rate changes, including *CustomerID*, *Name*, *OldInterestRate*, and *NewInterestRate*.
* **Cursor Declaration**: Declares a cursor *customer*\_cursor to fetch *CustomerID*, *Name*, *DOB*, *LoanID*, and *InterestRate* from *Customers* and *Loans* tables.
* **Variable Declarations**: Declares variables *v*\_*age*, v\_*oldInterestRate*, and v\_*newInterestRate* to store the customer's age, old interest rate, and new interest rate, respectively.
* **Cursor Loop**: Iterates through each record in the cursor, calculates the age of the customer in years using the *DOB* and current date.
* **Interest Rate Update**: If the customer is older than 60, updates their loan's interest rate by reducing it by 1%, and logs the change in the *LoanRateChanges* table.
* **Commit Changes**: Commits the transaction to save the updates to the database and inserts the changes.

**The output of the code:**

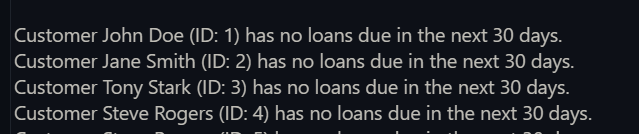
* **Scenario 2:** A customer can be promoted to VIP status based on their balance.
* **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.
* A cursor customer\_*cursor* is declared to select *CustomerID* and *Balance* from the *Customers* table.
* The cursor iterates over each record in the *Customers* table.
* If a customer's *Balance* exceeds 10,000, the *ISVIP* status is set to **YES**; otherwise, it is set to **NO**.
* The changes are committed to the database, and a *SELECT \* FROM CUSTOMERS* statement is executed to display the updated table.

**The output of the code**

****

* **Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.
* **Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.
* **Set Server Output:** The *SET SERVEROUTPUT ON;* command enables the display of output messages from PL/SQL blocks.
* **Customer Cursor Declaration:** A cursor *customer\_cursor* is declared to select the *CustomerID* and *Name* from the *Customers* table.
* **Loan Cursor Declaration:** A cursor *loan\_cursor* is declared to select *LoanID* and *EndDate* from the Loans table for a given *CustomerID*, where the loan end date falls within the next 30 days.
* **Customer Loop:** A loop iterates through each record in the *customer\_cursor*, processing each customer.
* **Loan Loop and Reminder Output:** For each customer, another loop iterates through the *loan\_cursor* to check for loans due in the next 30 days. If a loan is found, a reminder message is printed, and a flag *v\_loan\_due* is set to **TRUE**.
* **No Loans Due Message:** If no loans are due for a customer within the next 30 days (*v\_loan\_due* remains **FALSE**), a message stating that the customer has no loans due is printed.

**The output of the code :**

****