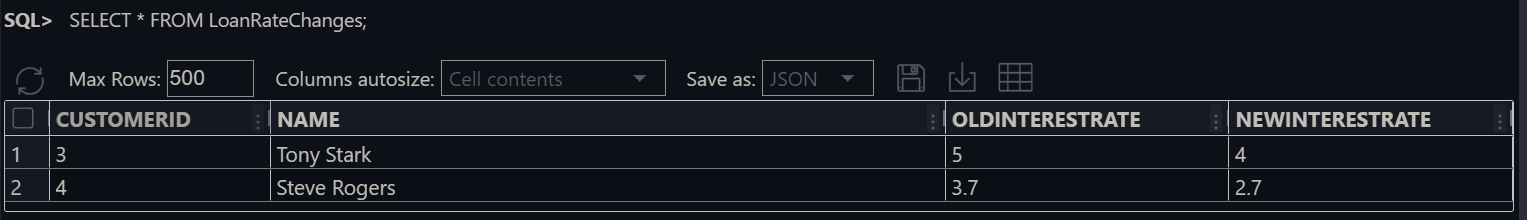
**Exercise 1: Control Structures**

* **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
* The code starts by creating a `LoanRateChanges` table to log changes in loan interest rates, including columns for `CustomerID`, `Name`, `OldInterestRate`, and `NewInterestRate`. It then declares a cursor, `customer\_cursor`, to fetch `CustomerID`, `Name`, `DOB`, `LoanID`, and `InterestRate` from the `Customers` and `Loans` tables. Variables are declared to hold the customer's age, the old interest rate, and the new interest rate.
* The cursor loops through each record, calculating the customer's age based on their date of birth and the current date. If the customer is over 60 years old, their loan's interest rate is reduced by 1%, and the change is logged in the `LoanRateChanges` table. Finally, the transaction is committed to save the updates, ensuring that all changes are recorded in the database.

**The OUTPUT:**

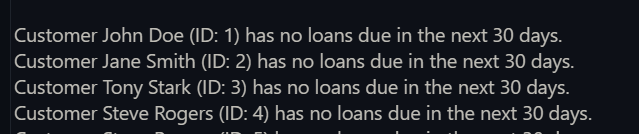
* **Scenario 2:** A customer can be promoted to VIP status based on their balance.
* **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.
* A cursor named `customer\_cursor` is declared to select `CustomerID` and `Balance`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

**The OUTPUT :**

****

* **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.
* The `SET SERVEROUTPUT ON;` command is used to enable the display of output messages from PL/SQL blocks. A cursor named `customer\_cursor` is declared to select `CustomerID` and `Name` from the `Customers` table, and another cursor, `loan\_cursor`, is declared to select `LoanID` and `EndDate` from the `Loans` table for a specific `CustomerID`, where the loan end date falls within the next 30 days.
* The code processes each customer by iterating through `customer\_cursor`. For each customer, it then iterates through `loan\_cursor` to check if any loans are due in the next 30 days. If a due loan is found, a reminder message is printed, and a flag `v\_loan\_due` is set to `TRUE`. If no loans are due (i.e., `v\_loan\_due` remains `FALSE`), a message indicating that the customer has no loans due is printed.

**The OUTPUT :**

****