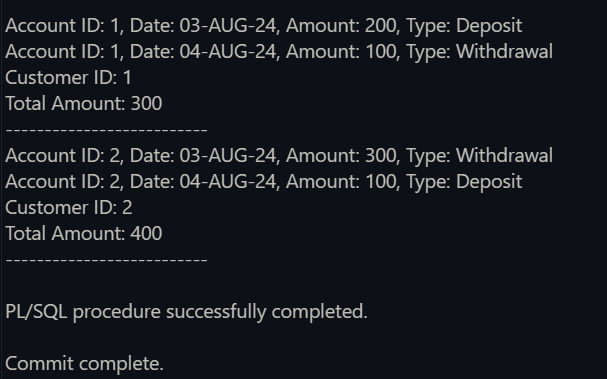
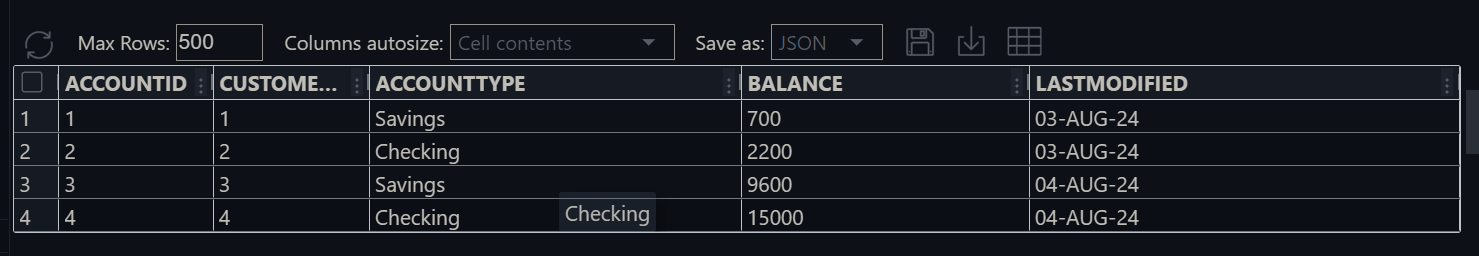
**Exercise 6: Cursors**

* **Scenario 1:** Generate monthly statements for all customers.
* **Question:** Write a PL/SQL block using an explicit cursor ***GenerateMonthlyStatements*** that retrieves all transactions for the current month and prints a statement for each customer.
* The code defines a cursor *GenerateMonthlyStatements* to fetch transaction data for the current month, joined with account information and ordered by *CUSTOMERID* and *TRANSACTIONDATE*.
* It declares variables to hold fetched data and a few control variables for customer tracking and total amount calculation.
* It opens the cursor, iterates through each fetched transaction, and checks if the transaction belongs to a new customer to reset totals and print the previous customer's statement.
* It prints each transaction's details and accumulates the total transaction amount for each customer.
* After the loop, it prints the statement
*  for the last customer and closes the cursor.

**Here is the Output of the code:**

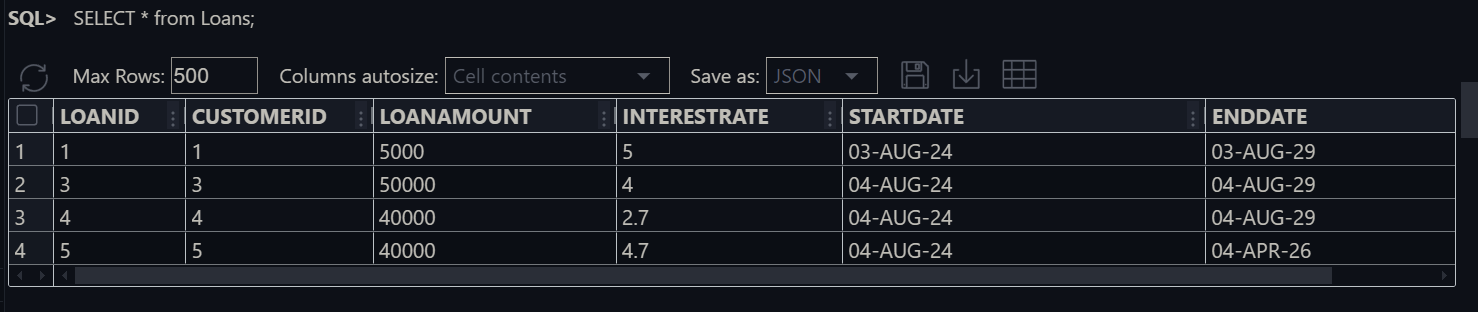
* **Scenario 2:** Apply annual fee to all accounts.
* **Question:** Write a PL/SQL block using an explicit cursor ***ApplyAnnualFee*** that deducts an annual maintenance fee from the balance of all accounts.
* **Cursor Declaration:** A cursor *ApplyAnnualFee* is declared to select *ACCOUNTID* and *BALANCE* from the *ACCOUNTS* table for updating the BALANCE.
* **Variable Initialization:** Variables *v\_account\_id* and *v\_balance* are declared to store the fetched *ACCOUNTID* and *BALANCE*, and a constant *v\_annual\_fee* is set to 50.
* **Cursor Opening:** The cursor *ApplyAnnualFee* is opened to start processing the rows.
* **Loop Processing:** A loop is used to fetch each row from the cursor, deduct the annual fee from the balance, update the balance in the *ACCOUNTS* table, and optionally log the deduction.
* **Cursor Closing:** The cursor *ApplyAnnualFee* is closed after processing all rows.

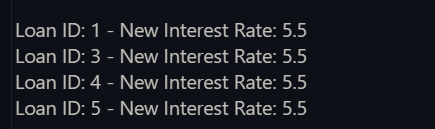
**The Code output:**

this is the ACCOUNT table before using the cursor

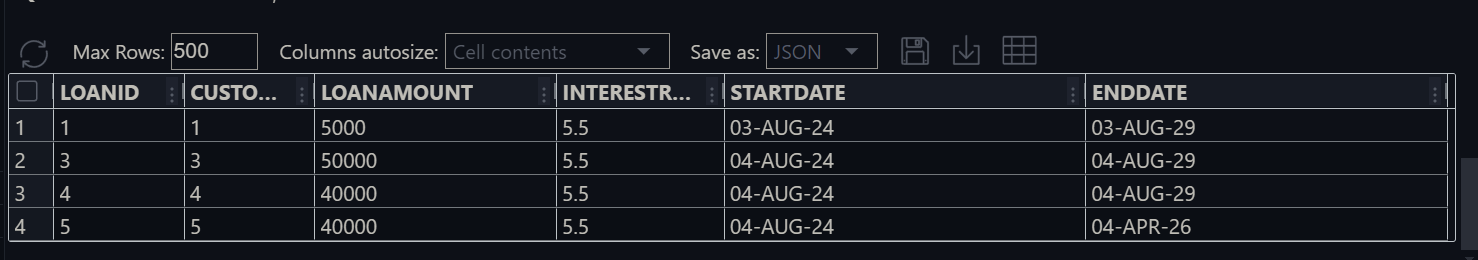
This is the Balance after use the cursor ***ApplyAnnualFee***

* **Scenario 3:** Update the interest rate for all loans based on a new policy.
* **Question:** Write a PL/SQL block using an explicit cursor ***UpdateLoanInterestRates*** that fetches all loans and updates their interest rates based on the new policy
* **Cursor Declaration:** The *UpdateLoanInterestRates* cursor selects *LOANID* and *INTERESTRATE* from the LOANS table and locks the INTERESTRATE column for update.
* **Variable Declarations:** Declares variables *v\_loan\_id* and v\_interest\_*rate* to store cursor output and a constant *v\_new\_interest\_rate* set to 5.5.
* **Cursor Opening:** Opens the *UpdateLoanInterestRates* cursor for processing.
* **Loop Through Records:** Fetches each row into variables, applies a new interest rate, updates the table, and optionally logs the changes.
* **Cursor Closing:** Closes the *UpdateLoanInterestRates* cursor after processing all records.

**The Output of the code:** Before apply the ***UpdateLoanInterestRates.*** The LOANS Table

****

Apply the ***UpdateLoanInterestRates*** in the LOANS table and update the Interest rate

The new LOANS table after update the Interest rate