Legacy COBOL Business Logic Analysis

Business Requirements

System Functionality Overview

This document outlines the functionality of a system designed to manage and display employee records from a database, specifically focusing on interaction with the EMPLOYEE database table. The system is intended to facilitate user interaction for sequential viewing of employee records, while ensuring appropriate handling of null values, particularly in the commission field. The following sections detail the input and output fields, business rules, and assumptions or constraints governing the system's operation.

Business Objectives

The primary objectives of the system are as follows:

- Efficiently retrieve and display employee records from the EMPLOYEE database table.
- Enable users to interact with the system to view records in a sequential manner.
- Appropriately manage and display null values in the commission field to ensure data integrity and clarity.

Input Fields

The system utilizes a single input field to control record retrieval:

- **ANSS**: This character-type field accepts input to determine whether to continue fetching records. It recognizes 'Y' or 'y' to proceed with the next record retrieval, while any other character input will terminate the loop.

Output Fields

The system provides several output fields to display employee information and operational status:

- **DISP-RATE**: This field presents a formatted display of the employee's pay rate.
- **DISP-COM**: This field shows a formatted display of the employee's commission.
- **DISP-CODE**: This field indicates the status of SQL operations, providing insight into the success or failure of database interactions.
- Additional details displayed include the employee's department, last name, first name, street address, city, state, zip code, pay rate, and commission status.

Business Rules

The system follows a structured process to manage the retrieval and display of employee records:

1. **Initialize and Open Cursor**:
  - A cursor is declared to select all records from the EMPLOYEE table, ordered by the last name.
  - The cursor is opened, and the resulting SQL status code is displayed via the `DISP-CODE` field to inform users of the operation's status.

2. **Fetch and Display Records**:
  - Records are fetched from the cursor into designated working-storage variables.
  - The SQL status code is updated and displayed in `DISP-CODE` for each operation.
  - The system loops through the records until no further data is available, as indicated by SQLCODE values less than 0 or equal to 100.
  - For each record, the system:
    - Transfers the `PAYRATE` to `DISP-RATE` and `COM` to `DISP-COM`.
    - Displays comprehensive employee details, including department, name, address, and pay rate.
    - Checks the `COM-NULL-IND` indicator to determine if the commission is null and displays an appropriate message if so.
    - Prompts the user to decide whether to fetch the next record, proceeding upon 'Y' or 'y' input, or exiting the loop otherwise.

3. **Close Cursor**:
  - Once all records have been processed or the user opts to stop, the cursor is closed to conclude the database interaction.

Assumptions and Constraints

The system operates under the following assumptions and constraints:

- The EMPLOYEE table comprises fields including ENO, LNAME, FNAME, STREET, CITY, ST, ZIP, DEPT, PAYRATE, and COM.
- It is assumed that users will input 'Y' or 'y' to continue fetching records; any other input will result in the termination of the loop.
- The system's error handling is limited to displaying SQL status codes, with no additional exception management.
- Null values in the commission field are addressed by checking the `COM-NULL-IND` indicator to ensure accurate data representation.

This documentation provides stakeholders with a comprehensive understanding of the

system's functionality, ensuring clarity in its operation and interaction with the EMPLOYEE database table.

Generated Test Cases

Test Case 1: Successful Retrieval and Display of Employee Records

Test Case Description:
Test the system's ability to retrieve and display employee records sequentially from the EMPLOYEE database table, handling null values for commission appropriately.

Input Values:
- ANSS: 'Y' (to continue fetching records)

Expected Output:
- DISP-CODE: '0' (indicating successful SQL operation)
- Employee details displayed: Department, Last Name, First Name, Street, City, State, Zip Code, Pay Rate, and Commission.
- DISP-RATE: Formatted pay rate of the employee.
- DISP-COM: Formatted commission or a message indicating null commission if applicable.
- Prompt for user input after each record.

Reasoning:
This test case ensures that the system can successfully open a cursor, fetch records, and display them sequentially. It verifies that the system correctly formats and displays employee details, including handling null commissions. The test also checks that the loop continues fetching records when the user inputs 'Y'.

Test Case 2: Handling Null Commission Values

Test Case Description:
Test the system's ability to handle and display null commission values correctly when fetching employee records.

Input Values:
- ANSS: 'Y' (to continue fetching records)

Expected Output:
- DISP-CODE: '0' (indicating successful SQL operation)
- Employee details displayed with a specific message for null commission.
- DISP-RATE: Formatted pay rate of the employee.

- DISP-COM: Message indicating that the commission is null.
- Prompt for user input after each record.

Reasoning:
This test case focuses on verifying the system's capability to identify and display null commission values. It checks that the `COM-NULL-IND` is correctly interpreted and that an appropriate message is displayed instead of a numeric value for commission.

Test Case 3: User Termination of Record Fetching

Test Case Description:
Test the system's response when the user decides to stop fetching records by entering a character other than 'Y' or 'y'.

Input Values:
- ANSS: 'N' (or any character other than 'Y' or 'y')

Expected Output:
- DISP-CODE: '0' (indicating successful SQL operation)
- Display of the current employee details before termination.
- No further records fetched after the input.
- Cursor closed successfully.

Reasoning:
This test case ensures that the system correctly exits the record-fetching loop when the user inputs a character other than 'Y' or 'y'. It verifies that the system stops fetching additional records and closes the cursor appropriately, confirming that user interaction controls the flow as intended.