Create a **PL/SQL package** that reads two input files every day:

* A **CSV** containing stock ownership per user received from an **exchange**.
* A **CSV** containing stock ownership per user received from a **depository**.

**Requirements:**

1. **Database Design**: Design a database schema to store the data from both CSV files. The schema should be flexible enough to handle potential changes in data structure or format.
2. **Data Processing**: The PL/SQL package should:
   1. **Compare** the stock ownership statements per user from both the exchange and depository.
   2. **Identify mismatches**: A mismatch occurs if the ownership details (e.g., stock count, stock ID, etc.) differ between the two sources for the same user.
   3. **Log the outcome** of the comparison (i.e., matches, mismatches) into a separate table for further analysis.
3. **File Inputs**: The package should accept **FilePaths** as inputs for the CSVs.

**Additional Requirements:**

1. **Error Handling and Edge Cases**:
   1. Implement error handling for cases such as:
      1. Missing or corrupt files.
      2. Incomplete rows in the CSV (e.g., missing stock data or user details).
      3. Format errors (e.g., invalid CSV structure).
2. **Performance Considerations**:
   1. Ensure the solution scales for **large datasets** (e.g., millions of records per day).
3. **Testing and Validation**:
   1. Write a set of **test cases** to validate the functionality of the PL/SQL package. Test cases should include:
      1. A scenario where all data matches perfectly.
      2. Scenarios with various types of mismatches (e.g., stock count differences, stock missing in one source, etc.).
      3. Handling of edge cases like missing or corrupt data.

**Example Inputs:**

**exchange\_data.csv**

User\_ID,Stock\_ID,Stock\_Name,Stock\_Count

1001,STK001,Apple,50

1001,STK002,Microsoft,20

1002,STK003,Google,30

1002,STK004,Amazon,25

1003,STK001,Apple,40

1003,STK003,Google,15

1004,STK002,Microsoft,50

1004,STK005,Facebook,10

1005,STK006,Tesla,35

1006,STK007,Netflix,45

1007,STK002,Microsoft,10

1008,STK001,Apple,60

1009,STK003,Google,25

1010,STK004,Amazon,55

1011,STK005,Facebook,5

1012,STK006,Tesla,80

1013,STK007,Netflix,20

1014,STK008,Intel,40

1015,STK009,AMD,35

1016,STK010,NVIDIA,50

1017,STK011,Salesforce,25

1018,STK012,Shopify,60

1019,STK013,Adobe,70

1020,STK014,Oracle,80

1021,STK015,Cisco,90

1022,STK016,Qualcomm,45

1023,STK017,IBM,30

1024,STK018,SAP,40

1025,STK019,Siemens,50

**depository\_data.csv**

User\_ID,Stock\_ID,Stock\_Name,Stock\_Count

1001,STK001,Apple,50

1001,STK002,Microsoft,22

1002,STK003,Google,30

1002,STK004,Amazon,25

1003,STK001,Apple,45

1003,STK003,Google,15

1004,STK002,Microsoft,50

1004,STK005,Facebook,12

1005,STK006,Tesla,30

1006,STK007,Netflix,45

1007,STK002,Microsoft,10

1008,STK001,Apple,60

1009,STK003,Google,20

1010,STK004,Amazon,55

1011,STK005,Facebook,7

1012,STK006,Tesla,80

1013,STK007,Netflix,25

1014,STK008,Intel,45

1015,STK009,AMD,33

1016,STK010,NVIDIA,50

1017,STK011,Salesforce,28

1018,STK012,Shopify,65

1019,STK013,Adobe,70

1020,STK014,Oracle,80

1021,STK015,Cisco,85

1022,STK016,Qualcomm,45

1023,STK017,IBM,35

1024,STK018,SAP,40

1025,STK019,Siemens,48