



Bansilal Ramnath Agarwal Charitable
Trust's Vishwakarma Institute of
Information Technology

Department of

Name: Yuvraj Prabhakar Sankilwar

Class: TY

Division: A

Roll No: 392052

Semester: V

Academic Year: 2024-2025

Subject Name & Code: ECI3120G - Mainframe Technologies

Title of Assignment: 2. Create two PS file

Date of Performance: 01/04/2025

Date of Submission: 02/04/2025

Aim: Create two PS files:

Problem Statement:

File1: Contains 50 records, each 80 bytes long, with the primary key located in columns 13 to 20.

File2: Contains 50 records, each 80 bytes long, with the primary key located in columns 21 to 28.

Task: Write a COBOL program that:

Reads both files (File1 and File2).

Creates a new file, File3, which will contain only the matching records from File1 and

File2, based on the primary key found in both files

Background Information:

In modern enterprise computing environments, data integration and record reconciliation are vital components of day-to-day operations. Organizations often maintain vast volumes of data across multiple flat files, and merging or comparing these files based on business keys is a common requirement. This assignment explores one such scenario through the use of COBOL—a legacy but still widely used programming language in mainframe environments—for performing **file matching operations**.

The task involves working with **two physical sequential (PS) files**, each containing 50 fixed-length records of 80 bytes. The key focus is to extract and compare the **primary key fields** embedded within each record—located at different column positions in each file—and then create a third output file that retains only the matching records from the first file. This type of operation is critical in numerous real-world applications, such as synchronizing customer records, reconciling transaction logs, merging billing information, and aligning inventory data across departments.

Understanding PS Files and Key Fields

A PS (Physical Sequential) file is one of the simplest and most commonly used datasets in the IBM z/OS mainframe environment. It stores data in a linear, sequential format, which makes it suitable for line-by-line reading and writing. Each record in this scenario is 80 bytes long, using **Fixed Block (FB)** record format, which ensures each record has the exact same length—this simplifies processing and positioning of key fields.

In this assignment:

- **File1** contains the key from **columns 13 to 20** (8 characters).
- **File2** contains the key from **columns 21 to 28** (8 characters). These primary keys are typically used to uniquely identify a record (like an ID number, account number, or reference code).

This design reflects a real-world situation where datasets might store similar information in slightly different formats or layouts, requiring programs to account for positional differences when comparing records.

Importance of COBOL in File Processing

COBOL (Common Business Oriented Language) is renowned for its readability and structured syntax, making it a preferred choice for financial and administrative systems. One of COBOL's major strengths lies in **sequential file processing**, where it can efficiently read, compare, and manipulate large volumes of fixed-format records.

In this assignment, COBOL is used to:

- Open and read two PS files sequentially.
- Extract and compare the key fields from each file.
- Output matching records into a third file.

This aligns with typical COBOL use cases such as payroll processing, bank reconciliation, inventory matching, and policy verification, where line-by-line record comparison is essential.

Matching Records – Real World Relevance

Matching records across files is a key operation in data engineering and integration pipelines. For instance:

- In **banking**, customer transactions from different systems may be cross-referenced.
- In **retail**, sales and shipment logs are reconciled to verify deliveries.
- In **insurance**, policyholder data is matched with claim files for validation.

In all these scenarios, accurate comparison of keys ensures data integrity and prevents mismatches or duplication. COBOL remains a reliable tool for such tasks due to its compatibility with VSAM, PS files, and batch processing systems.

Technical Concepts Involved

To accomplish this file matching task, a number of key technical concepts are applied:

- **Sequential File Access:** Reading files one record at a time in a top-down fashion.
- **Key Extraction:** Using string slicing or offset referencing to isolate keys within a fixed-length record.
- **Condition Matching:** Comparing two fields for equality and conditionally writing results.
- **File Control:** Managing multiple files simultaneously using the FILE-CONTROL and FD declarations in COBOL.

The output file only includes matching records from File1, reinforcing the importance of conditional logic and precise file manipulation

[GitHub Repo Link:](#)

<https://github.com/yuvrajofficials/mainframe-assignments.git>

Conclusion:

This assignment demonstrated the use of COBOL for processing and comparing sequential files based on key fields located at different positions. By reading two PS files and generating an output file with only the matching records, it highlighted essential concepts such as file handling, key extraction, and conditional logic. The task reinforced practical skills in COBOL programming and mainframe file operations while showcasing real-world applications like data reconciliation and record validation. Overall, it provided a solid foundation in structured data processing and the importance of accuracy in enterprise-level batch processing systems.