**Project Name: Part and Supplier**

**Requirement Document**

This is small batch Cobol Application. As part of this project following components need to create.

1. Customer provide the input file format. This input format has Parts, Supplier, Supplier address and Purchase order.
2. Sort the input Part and supplier file based on Part-no.
3. Use the above file and validate the all fields in the input file and generate separate files Parts, Supplier, Supplier Address and Purchase order.
4. Parts file contain all parts information. To generate this file following conditions need to validate.

* Required fields: PART-NUMBER/PART-NAME/ VEHICLE-MAKE, VEHICLE-MODEL

VEHICLE-YEAR

* VEHICLE-MAKE must be one of the listed 88-level fields.
* VEHICLE-YEAR must be between 1990 and 2019
* WEEKS-LEAD-TIME must be numeric and between 1 and 4

1. Supplier file contain information on the suppliers for given part. To generate this file following conditions need to validate.

* Required fields: SUPPLIER-CODE, SUPPLIER-TYPE, SUPPLIER-NAME, SUPPLIER-PERF
* SUPPLIER-TYPE, SUPPLIER-RATING, SUPPLIER-STATUS must be one of the listed 88-level values
* SUPPLIER-ACT-DATE -Optional but if a value exists in the field ensure that the date is valid
* If SUPPLIER-TYPE is SUBCONTRACTOR SUPPLIER-RATING must be Highest Quality
* SUPPLIER-STATUS -must be one of the listed 88-level fields

1. Supplier Address file contain 3 different address for each supplier. To generate this file following conditions need to validate.

* Required fields: ADDRESS-1, CITY, ADDR-STATE

and ZIP-CODE

* ADDRESS-TYPE must be one of the 88-level fields
* ZIP-CODE and ADDR-STATE must match in the STATEZIP file

1. Purchase order file contain 3 purchase orders cut to a supplier for given part. To generate this file following conditions need to validate.

* Required fields: PO-NUMBER, BUYER-CODE, ORDER-DATE, QUANTITY
* QUANTITY must be between 0 and 999,999
* If QUANTITY is > 0, UNIT-PRICE must be > 0.
* UNIT-PRICE must be between $1 and $1,000,000.00
* ORDER-DATE must be a valid date
* DELIVERY-DATE is optional but if there is data, it must be a valid date and the date must be later than ORDER-DATE

1. All Required fields cannot be blank
2. All records should be passes all the following sub-program

* Parts subprogram
* Supplier subprogram
* Supplier Address Subprogram
* Purchase Order Subprogram

1. If any input records are invalid, then above subprograms will send the return code 8.
2. All Invalid records should be written into error file

If errors are < 4 then

* The bad record must include error messages in the output file.
* Meaning that your output file record layout needs additional space for the error message(s) that accompany the invalid data fields in the record.

If errors are > 3 then

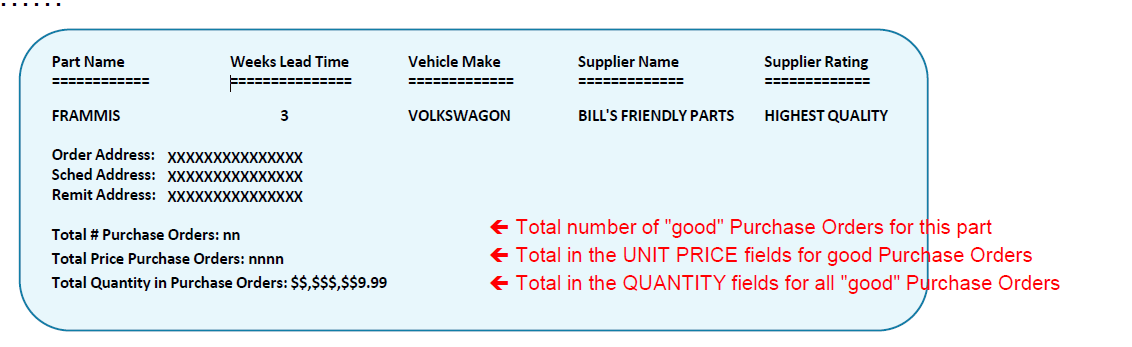
* The bad record is written to the output file with a single message that the record is completely invalid.

1. If input records are good, then above sub-program will send the return code 0 and good records are written into Parts, Supplier, Supplier address, Purchase order file and good records output PARTSUPP file.
2. For date validation use IBM “CEEDAYS” program.
3. Address State and ZIP validation use State Zip file and this file have valid State Abbreviation and ZIP code combination.

Zip file: DDS0001.LEARN.STATE.ADDRESS.ZIP

1. Control Break report need to generate for Good records.

Sample Report format:



1. Display statement can be used for ABEND Purposes