



Florida Department of Education

Education Data Warehouse

FUNCTIONAL RULES

Version 5.1



DMR Consulting
Guylaine Poirier
Dominique Boisclair

April 22, 2002

TABLE OF CONTENTS

1.	Introduction.....	3
2.	General architectural rules	4
	RG-001 Managing parameters in UNIX	4
	RG-002 Designing web pages using WEBDB	5
	RG-003 Creating identifiers for the Education Data Warehouse	5
	RG-004 Updating the data warehouse and data marts	5
	RG-005 Short code description.....	6
3.	Rules governing the extraction process.....	7
	RE-001 Data to be extracted	7
	RE-002 Data compression	7
	RE-003 Selective extraction.....	7
	RE-004 Empty data element “low value”, “high value” or “null value”	7
	RE-005 Cleansing of last name, first name, middle name and middle initial	8
	RE-006 Creation of SSN-ID and NON-SSN-ID at extraction	8
	Creation of SSN-ID and Non-SSN-ID for PK12	8
	Creation of SSN-ID and Non-SSN-ID for CC	10
	Creation of SSN-ID and Non-SSN-ID for SUS	10
	RE-007 Validation’s rules for SSN.....	12
	RE-008 Upper Case Transformation.....	12
	RE-009 Dates Format	13
4.	Rules governing data cleansing and transformation.....	13
	RCT-001 Dates	13
	RCT-002 Upper Case Transformation.....	13
	RCT-003 Transformation or Validation of data value	13
	RCT-004 Replace “blanks” and “nines” to “null value”	14
	RCT-005 Rejected orphan data.....	14
	RCT-006 Generating a unique sequential number for transactions or non-significant keys	14
	RCT-007 Generating significant zeroes.....	15
	RCT-008 Indicator data elements	15
	RCT-009 Data element format conversion	15
	RCT-010 Character data element concatenation.....	16
	RCT-011 Identifying a student with demographic information.....	16
	RCT-012 Linking a student’s non-demographic data to a Student Education DW ID.....	26
	RCT-014 Generating attributes Create System Source and Update System Source	27

1. INTRODUCTION

This document presents basic functional specifications to be applied in all processes involved in the Education Data Warehouse. The objective of these rules is twofold:

- 1) Provide guidance/orientations/constraints to all EDW architects and analysts;
- 2) Be a common extension to any EDW-related functional specification, thus eliminating redundancy of written information in several documents and facilitating their updates.

Each rule is attributed a rule number (Ex. RG-001) structured in this way:

- "R" stands for Rule
- followed by a rule scope indicator (G for General rules, E for Extraction rules, CT for Cleansing/Transforming rules
- followed by a three digits sequential number (Ex. 001).

This document presents the first set of functional rules applicable to the EDW. Many rules will need to be further updated or created in the future during the development process.

2. GENERAL ARCHITECTURAL RULES

RG-001 MANAGING PARAMETERS IN UNIX

Unix parameters are stored in an Oracle table. These parameters are under the responsibility of the Data Warehouse Management Team who is able to add, modify or delete these parameters using the meta data application.

Parameters are defined by the following data elements:

Data Element	Description
Parameter Name	Name describing the parameter
Parameter Type	Format of the parameter Possible values are: <ul style="list-style-type: none"> • ALphanumeric, • DY (date YYYY), • DD(date MM-DD-YYYY), • DM (date MM-YYYY), • EN (integer), • HN (hour HH:MM:SS), • Indicator standard (Yes/No), • AMount, • PErcentage, • REal, • RAte.
Parameter Description	Short description/parameter detail(s)
First Parameter Identifier	First part of key to find parameter value Name of the file
Second Parameter Identifier	Second part of key to find parameter value Parameter Description (Upper case)
Third Parameter Identifier	Third part of key to find parameter value Process step (Matching and Cleansing or Transforming and Aggregating)
Start Date	Start Date indicates when the parameter starts being effective.

End Date	End Date indicates when the parameter stops being effective.
Date of Last Update	Date of Last Update indicates the last time someone updated the parameter.
Parameter Value	Parameter value.

RG-002 DESIGNING WEB PAGES USING WEBDB

The Department of Education standards for Web site development will be used. These standards can be found at the following URL:
[http://www.firn.edu/doe/webdev /](http://www.firn.edu/doe/webdev/)

The Data Warehouse Web site must respect these DOE standards while considering the limitations of the Oracle WEBDB tool.

RG-003 CREATING IDENTIFIERS FOR THE EDUCATION DATA WAREHOUSE

An identifier in the Education Data Warehouse (normalized model) must be unique (only one meaning for each value). To ensure this uniqueness of identifiers, independent identifiers must be created. These identifiers are also recommended for addressing security issues, especially when an identifier can be used to help recognize a person.

Example: Institutions, Student ID, Staff ID, etc.

RG-004 UPDATING THE DATA WAREHOUSE AND DATA MARTS

The Education Data Warehouse must be updated on the weekend to decrease negative impacts on users since the data warehouse must be shut down during that time.

Data marts must be updated as soon as possible after the data warehouse is updated.

Data marts can be updated fully or partially, depending on the nature of the aggregation.

Creating or updating data marts involves a period of non-availability that should be no longer than one day.

As the creation of data marts uses computer resources allocated to user queries, the number of data marts created at the same time must be limited. A priority for updating data marts must be established based on users' needs.

Data marts will be accessible until the time they are updated even though the data warehouse and data mart are not completely synchronized. Users should be informed of this temporary situation.

RG-005 SHORT CODE DESCRIPTION

Codes must have a usable description for reporting. This short description must contain a maximum of 20 characters .

3. RULES GOVERNING THE EXTRACTION PROCESS

RE-001 DATA TO BE EXTRACTED

Beginning years to be extracted (date) must be a parameter the Data Warehouse Administration team can change. A beginning year must be defined for every extraction.

This approach will be useful for the first extractions, five years, and the normal extraction of the current year. It will also be useful for re-extraction when a major modification is made .

Example: beginning date of extraction: 01-09-1995 for the first extraction or
01-09-2000 for the current year

RE-002 DATA COMPRESSION

For files on the mainframe, compressed data elements (COMP) must be decompressed before being transferred to the Unix server. This is necessary because compressed data elements are not recognized on the Unix platform.

RE-003 SELECTIVE EXTRACTION

To save space and processing time, all data elements are not extracted. Only data elements that will be loaded into the data warehouse or needed for cleansing and transformation are extracted (applicable for Phase 1&2 and exceptionally on other Phases).

RE-004 EMPTY DATA ELEMENT “LOW VALUE”, “HIGH VALUE” OR “NULL VALUE”

When the value of a data element is a missing value (low value in a mainframe file, null in a relational database), this value must be replaced with:

blanks for a character field,
nines for a numeric field.

Blanks and nines will be loaded as “null” into the data warehouse.

Other values can be used to indicate that a data element is not applicable if the nines are a possible value. In these circumstances, the functional specification and the meta data must identify the precise value to be used.

In addition, all fields must be scanned to verify that no low value character is found within a field. Any low value character must be replaced by a blank.

RE-005 CLEANSING OF LAST NAME, FIRST NAME, MIDDLE NAME AND MIDDLE INITIAL

For those fields, a cleansing must be done and new fields should be created at the extraction. First name, last name, middle name and middle initial are kept as they are and the new values are put in clean first name, clean last name, clean middle name and clean middle initial.

For PK12, when there is a separate field for the appendage, this field is concatenated to last-name.

The process of cleaning is described as follows :

- All blank characters should be removed.
- All non-alphabetic characters should be removed:

Data Element Name	Before Application of Rule	After Application of Rule
Last name:	<EM&GEE;;>	<EMGEE>
First name:	<ANNE(-MARIE>	<ANNEMARIE>
Middle name:	<W. "MO">	<WMO>

The cleansing should be done at the extraction. Cleaned names are used only for the identification process and should not be kept into the data warehouse.

RE-006 CREATION OF SSN-ID AND NON-SSN-ID AT EXTRACTION

Creation of SSN-ID and Non-SSN-ID for PK12

SSN-ID

If the last digit of the Student Number Identifier is an "X", it is a Social Security Number (SSN).

Apply validation rules (see : Rule RE-007 SSN validation rules) to verify if SSN is valid. If it is a valid SSN then copy the first 9 digits of the Student Number Identifier to a SSN-ID.

If it is invalid, consider it as a Non-SSN.

The SSN-ID will be used in the Record to Match file for the Identification process.

Non-SSN-ID

A Non-SSN-ID is always created for every student's record, even if they have a valid SSN.

The Non-SSN-ID is composed as follows :

Data Source X(04) (ex : PKxx) (where xx is the District Number)
Student Number Identifier X(10)

If the last digit of the Student Number Identifier is an "X", it is omitted in the Non-SSN-ID and is initialized to blank.

Alias SSN-ID

If the last digit of the Student Number Identifier - Alias is an "X", it is an Alias - Social Security Number (Alias-SSN).

Apply validation rules (see : Rule RE-007 SSN validation rules) to verify if the Alias-SSN is valid.

If it is a valid Alias-SSN then copy the first 9 digits of the Student Number Identifier-Alias to an Alias-SSN-ID.

If it is invalid, consider it as an Alias-Non-SSN.

The Alias-SSN-ID will be used in the Record to Match file for the Identification process.

Alias Non-SSN-ID

An Alias-Non-SSN-ID is created when there is a Student Number Identifier – Alias, even if they have a valid Alias-SSN.

The Alias-Non-SSN-ID is composed as follows :

Data Source X(04) (ex : PKxx) (where xx is the District Number)
Student Number Identifier-Alias X(10)

If the last digit of the Student Number Identifier-Alias is an "X", it is omitted in the Alias-Non-SSN-ID and is initialized to blank.

The Alias-Non-SSN-ID will be used in the Record to Match file for the Identification process.

Creation of SSN-ID and Non-SSN-ID for CC

SSN-ID

If the first digit of the Student Identification Number (PSNID) is a numeric character, it is a Social Security Number (SSN).

If the number used is not a social security number, the Student Identification Number must begin with an alphabetic character.

Apply validation rules (see : Rule RE-007 SSN validation rules) to verify if SSN is valid. If it is a valid SSN then copy the 9 digits of the Student Identification Number (PSNID) to a SSN-ID.

If it is invalid, consider it as a Non-SSN.

The SSN-ID will be used in the Record to Match file for the Identification process.

Non-SSN-ID

A Non-SSN-ID is always created for every student's record, even if they have a valid SSN.

The Non-SSN-ID is composed as follows :

Data Source X(04) (ex : CC) (Two last digits are filled with blank)
Student Identification Number (PSNID) X(10)

Note. When the first digit of the Student Identification Number (PSNID) is an alphabetic character, the length of the PSNID can vary (from 7 to 10 digits). In this case, to create a Non-SSN-ID the Student Identification Number should be used as it is the file and if the length is less than 10 digits, it has to be filled with blanks.

Note. Alias does not apply to Community College.

Creation of SSN-ID and Non-SSN-ID for SUS

SSN-ID

The Person Identification Number which is the student identifier is supposed to be a Social Security Number (SSN).

Apply validation rules (see : Rule RE-007 SSN validation rules) to verify if SSN is valid. If it is a valid SSN then copy the 9 digits of the Person Identification Number to a SSN-ID.

If it is invalid, consider it as a Non-SSN

The SSN-ID will be used in the Record to Match file for the Identification process.

Non-SSN-ID

A Non-SSN-ID is always created for every student's record, even if they have a valid SSN.

The Non-SSN-ID is composed as follows :

Data Source X(04) (ex : SUS) (Last digit is filled with blank)
Person Identification Number X(10) (Last digit is filled with blank)

The Non-SSN-ID will be used in the Record to Match file for the Identification process.

Alias SSN-ID

When a Person Identification Number Previous (which is the previous student identifier) is present, it is considered as a Previous Social Security Number (Alias-SSN).

Apply validation rules (see : Rule RE-007 SSN validation rules) to verify if SSN is valid. If it is a valid SSN then copy the 9 digits of the Person Identification Number Previous to an Alias-SSN-ID.

If it is invalid, consider it as an Alias-Non-SSN

The Alias-SSN-ID will be used in the Record to Match file for the Identification process.

Alias-Non-SSN-ID

An Alias-Non-SSN-ID is created when there is a Person Identification Number Previous, even if they have a valid Alias-SSN.

The Alias-Non-SSN-ID is composed as follow :

Data Source X(04) (ex : SUS) (Last digit is filled with blank)
Person Identification Number Previous X(10) (Last digit filled with blank)

The Alias-Non-SSN-ID will be used in the Record to Match file for the Identification process.

Some SUS files follow the same pattern as PK-12 files, meaning that they use an “X” in the last digit of the Student Id to confirm that it is a Social Security Number (SSN). For these special cases, it should be mentioned in the extracted rule of the field.

RE-007 VALIDATION’S RULES FOR SSN

The nine-digit Social Security Number (SSN) is composed of three parts :

- The first set of three digits is called “Area Number” and is assigned by the geographical region. Actually, the last new area number created is 772.
- The second set of two digits is called the “Group Number”. Within each area, the group number ranges from 01 to 99 but is not assigned in consecutive order.
- The final set of four digits is the Serial Number. Within each group, the serial numbers run consecutively from 0001 to 9999.

To validate a SSN, some rules should be applied :

- a) Length and format : 9 numeric characters
- b) Invalid if equal to “000000000”
- c) Three first digits (Area Number) different from “000” (ex : 000123456 not valid).
- d) Digit 4 and 5 (Group Number) different from “00” (ex : 123006789 not valid)
- e) Last four digits (Serial Number) different from “0000”
- f) The first digit should be less than 8 (801123456 not valid) (Last area number created is 772 so there is a range for creation of new areas which is not too frequent)

Note : If low values or high values are found in SSN validation then the record must be rejected.

RE-008 UPPER CASE TRANSFORMATION

All character data elements must be transformed into upper case at the extraction. For the Identification process, it is necessary to have all character data elements in upper case to be able to match them.

RE-009 DATES FORMAT

Applicable for SQL Server, Oracle, Excel)

All dates must be transformed into format MMDDYYYY.

If a date is not valid, it should be replaced with nines and loaded as “null” into the warehouse.

If a date is filled with nulls, it should be converted to nines only if nines is not a valid value. Otherwise, the replacement value should be specified.

4. RULES GOVERNING DATA CLEANSING AND TRANSFORMATION

RCT-001 DATES

All dates must be transformed into the format MMDDYYYY.

If a date is not valid, it is replaced with nines and loaded as “null” into Oracle. If the century is missing, a specific algorithm will establish if the century is set to 19 or 20, depending on the date’s context.

RCT-002 UPPER CASE TRANSFORMATION

All character data elements must be transformed into upper case. This simplifies querying because Oracle is case-sensitive.

RCT-003 TRANSFORMATION OR VALIDATION OF DATA VALUE

The transformation rule for a specific element must indicate what is the process to apply to reach the target value.

Each data element which has a list of values in the source has to be validated or transformed.

Transformation applies only when Source data element values are different than Target data element values. This transformation must be specified in the transformation rule section of the functional specifications.

If the Source data element values are the same than the Target data element values, a validation is required. This validation must be specified in the transformation rule section of the functional specifications.

When the code value of a data element is an unknown (the code value is not found in the corresponding Source data element values), then the data element is set to a default value. This transformation must be specified in the transformation rule section of the functional specifications.

RCT-004 REPLACE “BLANKS” AND “NINES” TO “NULL VALUE”

In the “Extract “ process, when the value of a data element was replaced with “blanks” or “nines” this value has to be loaded as “null” into the data warehouse.

RCT-005 REJECTED ORPHAN DATA

When a transaction (such as “course”) cannot be linked to the student demographic information, the record is not loaded into the Data Warehouse. In theory this situation is not supposed to happen.

RCT-006 GENERATING A UNIQUE SEQUENTIAL NUMBER FOR TRANSACTIONS OR NON-SIGNIFICANT KEYS

Unique numbers are generated sequentially and by range to allow several processes to work in parallel.

This sequential number is assigned using a generalized table that stores the last number used and the last usable number. If the last usable number is reached, an error message is sent and action must be taken. Information on sequential numbers is stored in an Oracle table named SEQ_GAP.

NAME	GAP	Quantity allowed by call to subroutines (range)
Student Identifier (K20_EDW_ID)	100000000001 to 999999999999	10 000
Course	000001 to 999999	100
Staff Identifier	100000000001 to 999999999999	10 000

Institutions	000001 to 999999	100
More to come		

RCT-007 GENERATING SIGNIFICANT ZEROES

By default, all numeric fields with zeroes are considered as a significant zero. Non-significant or non-applicable numbers must be specifically identified, and these fields will be converted to null when being loaded into the data warehouse.

RCT-008 INDICATOR DATA ELEMENTS

The only authorized values for an indicator data element in the Data Warehouse are Y (Yes), N (No) or null. A specific transformation rule must be applied for every different input value.

RCT-009 DATA ELEMENT FORMAT CONVERSION

This rule specifies the way data element formats must be converted, mainly when an element is converted from a numeric format to a character format.

This transformation rule must be applied as shown in the following examples:

Source Format	Source Value	Target Format	Target Value
9(3)	1 10 100 0	X(5)	00001 00010 00100 00000 or blank ¹
X(4)	A AA AAA AAAA	X(8) char	A_____ or _ = blank AA_____ AAA_____ AAAA_____
X(4)	A AA AAA AAAA	X(8) varchar	A AA AAA AAAA

1 If the source value is 0, the resulting value depends on the situation. If 0 is a significant value, the target value must be 0000. If not, the target value must be blank to be loaded as null into Oracle.

RCT-010 CHARACTER DATA ELEMENT CONCATENATION

When data elements must be concatenated, the process must remove the blank characters at the end of elements (keep one blank between the elements). This is required to save space in the Data Warehouse.

The transformation rule must be applied as in the following examples:

Source Format	Source Value	Target Format	Target Value
Last name X(20) First name X(20)	'MCCARTNEY ' 'PAUL '	NAME X(40)	PAUL MCCARTNEY

RCT-011 IDENTIFYING A STUDENT WITH DEMOGRAPHIC INFORMATION

This rule identifies and links a unique data warehouse student identifier for each student record containing demographic data from various data sources such as PK12, Community Colleges, State Universities, Talented 20, Vocational and others. As a result of this unique identification, all information related to an individual student such as courses, grades, financial aid and so on is linked to the student longitudinally across the various educational institutions the student may attend over a period of time.

Student identification is done by comparing demographic data elements to a “Student Identification Reference” table which contains a unique student identifier used internally for the data warehouse.

Some data sources should create a unique EDW Id and some others should not. A parameter in the extraction file must tell if an identifier should be created or not.

In order to establish a good comparison between data elements, some of the following data should be cleaned before applying the identification process.
(for more details, see Extraction rule RE-004)

The “Record to Match “ file extracted from PK-12, Community Colleges, State Universities, etc., used to identify a student is as follows:

Input File Format for Student Identification (Records to be linked)			
Data element	Description	Format	List of values
Sequence Number	Contains a unique number which is given by the system .	X(14)	Concatenation of the following data elements: <ul style="list-style-type: none"> Task Unit (UT) X(04)

Input File Format for Student Identification (Records to be linked)			
Data element	Description	Format	List of values
			<ul style="list-style-type: none"> • Sub Task Unit X(2) • Sequential number given by the system. 9(8)
Access data mode	From the Record to Match File. Indicate if the record can or cannot create a K20_EDW_ID. Only the Update mode can create.	X(1)	U = Update mode Q = Query mode
Student SSN_ID	May contain the social security number used to identify a student.	9(9)	Extracted from Student identification (Rule RE-005)
Student alias SSN_ID	May contain previous social security number used to identify a student. Note : Doesn't apply to CC.	9(9)	Extracted from Previous Student identification (Rule RE-005)
Student N_SSN_ID	May contain a N_SSN student identification. Derived for each student. Always present.	X(14)	Concatenation of the following data elements: <ul style="list-style-type: none"> • Data source X(04) which is SUS, CC PK+District, etc. • Student identification number assigned. (Rule RE-005)
Student alias N_SSN_ID	May contain a N_SSN student identification composed of the concatenation of data source and the previous student identification number assigned. Note : Doesn't apply to CC.	X(14)	Concatenation of the following data elements: <ul style="list-style-type: none"> • Data source x(04), • Previous student identification number assigned. (Rule RE-005)
Last name	Student's last name	X(30)	For PK12, Concatenation of the following data elements: <ul style="list-style-type: none"> • Last name • Appendage
Clean Last Name	Student's last name without invalid characters and blanks	X(30)	(Rule RE-004)
First name	Student's first name	X(30)	
Clean First name	Student's first name without invalid characters and blanks	X(30)	(Rule RE-004)
Middle name	Student's middle name	X(30)	
Clean Middle name	Student's middle name without invalid characters and blanks	X(30)	(Rule RE-004)
Middle initial	Student's middle initial	X(01)	
Clean Middle initial	Student's middle initial without invalid characters and blanks	X(01)	(Rule RE-004)
Birth date	Student's birth date	MMDDYYYY	
Gender	Student's gender	X(01)	

Input File Format for Student Identification (Records to be linked)			
Data element	Description	Format	List of values
Racial category	Student's racial category	X(01)	
Institution code	Identifies the institution	X(04)	

The “Student Identification Reference” table (Oracle table) contains a unique student identifier which is given by the system and this data is used to associate a unique student identifier to a student in the data warehouse. The data elements stored in this table are described as follows:

Student Identification Reference Table			
Data element	Description	Format	List of values
Unique student identifier	Internal unique student identifier assigned to the student for the data warehouse	X(8)	Created by the system
Student SSN_ID	May contain the actual social security number used to identify a student. Used to create a K20 EDW ID.	9(9)	
Student N_SSN_ID	Contains a N_SSN student identification. Used to create a K20 EDW ID.	X(14)	
Last name	Student's last name	X(30)	
Clean last name	Clean student's last name	X(30)	
First name	Student's first name	X(30)	
Clean first name	Clean Student's first name	X(30)	
Middle name	Student's middle name	X(30)	
Clean middle name	Clean student's middle name	X(30)	
Middle initial	Student's middle initial	X(01)	
Clean middle initial	Clean student's middle initial	X(01)	
Birth date	Student's birth date	MMDDYYYY	
Gender	Student's gender	X(01)	
Racial category	Student's racial category	X(01)	
Institution code	Identifies the institution	X(04)	

As well, a parameter-driven table is required to determine what rules (rules 1 through 13 explained below) are applied to the student identification process for the source data in question.

An output file is generated as a result of processing the input file by associating a unique student identifier with each input record. The output file format is as follows:

Output File Format for Student Identification (Linked Records)			
Data Element	Description	Format	List of Values
Sequence Number	Contains a unique number which	X(14)	Concatenation of the

Output File Format for Student Identification (Linked Records)			
Data Element	Description	Format	List of Values
	is given by the system .		following data elements: <ul style="list-style-type: none"> • Task Unit (UT) X(04) • Sub Task Unit X(2) • Sequential number given by the system. 9(8)
Unique student identifier	Internal unique student identifier assigned to the student for the data warehouse	X(8)	Created by the system

Rules for Student identification:

The following matching rules are applied in the order they are presented in the following pages. Whenever a data element is not present for the search in question, the next hierarchical rule in the list is applied.

The search rules are as follows:

Search by N_SSN

1. Search by student N_SSN_ID (For 2000 and over). Not applicable for CC.
 - 1.a Search by student N_SSN_ID id and last name (before 2000 and for PK12 only) Not applicable for CC.
2. Search by student alias N_SSN_ID and birth date
3. Search by student alias N_SSN_ID, last name and first name.
4. Search by student alias N_SSN_ID, clean last name and clean first name

Search by SSN

5. Search by student SSN_ID and birth date
6. Search by student SSN_ID, last name, first name
7. Search by student SSN_ID, clean last name, clean first name
8. Search by student alias SSN_ID and birth date

9. Search by student alias SSN_ID, last name and first name
10. Search by student alias SSN_ID, clean last name and clean first name

Search by Name

11. Search by last name, first name and birth date
12. Search by clean last name, clean first name and birth date

Special rules

13. Search by N_SSN_ID and institution code.
14. Search by student alias N_SSN_ID and birth date
15. Search by student alias N_SSN_ID, last name and first name.
16. Search by student alias N_SSN_ID, clean last name and clean first name

Note.

- ◇ For SUS, there is no complete birth date at this time (prior to 2001). In this special case, rules with birth date won't apply.
- ◇ Rule #1a will apply for extracted years before 2000. This rule is applied only before 2000 because quality of data was a potential problem. Match with last-name is used to be more precise.
- ◇ A special rule (rule #13) is specific for files which don't have any demographic items. This rule will be used for Community College instead of Rule 1.
- ◇ Rules 14, 15 and 16 are required for PK12 but could apply to others.
- ◇ Search by date is performed only when the date is valid.

Search Results :

There are three possible outcomes to any of the above-mentioned searches. They are as follows:

- A **No match found** result occurs when a search returns no unique student identifier (K20_EDW_ID) from the “Student Identification Reference” table.
 - Execute the next hierarchical search rule until a match is found or all searches are performed.
 - When all searches are performed and there is still no match then:
 - ◇ if the **access data mode flag is Q (query)**, the student record is not created and a reject file is created.
 - ◇ if the **access data mode flag is U (update)**, a new unique student identifier (K20_EDW_ID) is created in the “Student Identification Reference” table using the student’s demographic data to create the occurrence, and the student record will use this new unique student identifier.

 - A **Match found with one unique identifier** result occurs when a search returns one unique identifier from the “Student Identification Reference” table.
 - The unique student identifier found is assigned to the student record in question.
 - ◇ Compare the other criteria and check if one or more parameters have changed (gender, race, middle initial, etc.). If something changes, create a record with the new elements. This new record will be linked to the student identifier found.

 - A **Match found with several unique identifiers** result occurs when a search returns more than one unique identifier from the “Student Identification Reference” table.
 - The search is refined by using the remaining identifying data elements to search for a unique student identifier match.
 - ◇ If one unique identifier is found after refining the search, the unique identifier found is linked to the student record in question.
 - ◇ When all refined searches are exhausted and a single unique student identifier is not found, the most recently created unique identifier is linked to the student record.
- (For more details, see Identifying a student with demographic information when more than one student id is found)*

PROCESS CONTROL

Functional controls are required to control the processing of each source file and are defined as follows :

A- RECORDS READ from source file xxxxxx	:	999,999,999
B- RECORDS WRITTEN to Identification Student file xxxxxx	:	999,999,999
C- REJECTED RECORDS (not found in query mode or rejected for other reasons)	:	999,999,999
D- RECORDS MATCHED IN SEQUENTIAL FILE	:	999,999,999
E- RECORDS MATCHED BY RULE 1 AND 1A	:	999,999,999
F- RECORDS MATCHED BY RULE 2	:	999,999,999
G- RECORDS MATCHED BY RULE 3	:	999,999,999
H- RECORDS MATCHED BY RULE 4	:	999,999,999
I- RECORDS MATCHED BY RULE 5	:	999,999,999
J- RECORDS MATCHED BY RULE 6	:	999,999,999
K- RECORDS MATCHED BY RULE 7	:	999,999,999
L- RECORDS MATCHED BY RULE 8	:	999,999,999
M- RECORDS MATCHED BY RULE 9	:	999,999,999
N- RECORDS MATCHED BY RULE 10	:	999,999,999
O- RECORDS MATCHED BY RULE 11	:	999,999,999
P- RECORDS MATCHED BY RULE 12	:	999,999,999
Q- RECORDS MATCHED BY RULE 13	:	999,999,999
R- RECORDS MATCHED BY RULE 14	:	999,999,999
S- RECORDS MATCHED BY RULE 15	:	999,999,999
T- RECORDS MATCHED BY RULE 16	:	999,999,999
U - STUDENTS NOT FOUND IN QUERY MODE	:	999,999,999
V – STUDENT NOT FOUND IN UPDATE MODE	:	999,999,999
W- STUDENT IDENTIFIER (K20_EDW_ID) PREVIOUSLY EXISTING: (D+E+F+G+H+I+J+K+L+M+N+O+P+Q+R+S+T)	:	999,999,999
X- STUDENT IDENTIFIER (K20_EDW_ID) NEWLY CREATED	:	999,999,999
Y- NUMBER OF RANDOM USED	:	999,999,999

Z- NUMBER OF RECORDS READ FROM IDENTIFICATION	:	999,999,999
AA- NUMBER OF DUPLICATED ADDED ROWS	:	999,999,999
BB- TOTAL ADDED ROWS	:	999,999,999

BALANCING

EQUATION 01	:	A = B + C
EQUATION 02	:	V = X
EQUATION 03	:	B = V+W

Identifying a student with demographic information when more than one student id is found

A SEARCH BY STUDENT NON-SSN ID IS PERFORMED

Rule #1 - Search by student non-ssn id

Rule #1a - Search by student non-ssn id, last name

Rule #2 - Search by student alias non-ssn id and birth date

Rule #3 - Search by student alias non-ssn id, last name and first name

Rule #4 - Search by student alias non-ssn, clean last name and clean first name

- If **N (more than 1) unique identifiers found**, the rule governing a match with N unique identifiers applies. Refine the search according to the refined search criteria for N_SSN_ID rules defined below.

Refined search

- 1- Search by SSN ;
- 2- Search by clean last name and clean first name ;
- 3- Search by birth date ;
- 4- Search by gender ;
- 5- Search by race ;
- 6- Search by clean middle name ;
- 7- Search by clean middle initial.

- ◇ If one unique identifier is found after refining the search, the unique identifier found is linked to the student record in question.
- ◇ When all refined searches are exhausted and a single unique student identifier is not found, the most recently created unique identifier is linked to the student record.

A SEARCH BY STUDENT SSN ID IS PERFORMED

Rule #5 - Search by student ssn id and birth date

Rule #6 - Search by student ssn id last name, first name

Rule #7 - Search by student ssn id clean last name, clean first name

Rule #8 - Search by student alias ssn id and birth date

Rule #9 - Search by student alias ssn id, last name and first name

Rule #10 - Search by student alias ssn id, clean last name and clean first name

- If **N (more than 1) unique identifiers found**, the rule governing a match with N unique identifiers applies. Refine the search according to the refined search criteria for SSN rules defined below.

Refined search

- 1- Search by clean last name and clean first name ;
- 2- Search by birth date ;
- 3- Search by gender ;
- 4- Search by race ;
- 5- Search by clean middle name ;
- 6- Search by clean middle initial.

- ◇ If one unique identifier is found after refining the search, the unique identifier found is linked to the student record in question.
- ◇ When all refined searches are exhausted and a single unique student identifier is not found, the most recently created unique identifier is linked to the student record.

A SEARCH BY STUDENT NAME ID IS PERFORMED

Rule #11 - Search by last name, first name and birth date

Rule #12 - Search by clean last name, clean first name and birth date

- If **N (more than 1) unique identifiers found**, the rule governing a match with N unique identifiers applies. Refine the search according to the refined search criteria for name rules defined below.

Refined search

- 1- Search by birth date ;
- 2- Search by gender ;
- 3- Search by race ;
- 4- Search by clean middle name ;
- 5- Search by clean middle initial.

- ◇ If one unique identifier is found after refining the search, the unique identifier found is linked to the student record in question.
- ◇ When all refined searches are exhausted and a single unique student identifier is not found, the most recently created unique identifier is linked to the student record.

- ◇ *Note.* There is no refined search for rule #13. In this kind of file, no demographic data is present so it is impossible to perform a refined search. If more than one unique student identifier is found, the most recently created unique identifier is linked to the student record.

RCT-012 LINKING A STUDENT'S NON-DEMOGRAPHIC DATA TO A STUDENT EDUCATION DW ID

This rule links a student's non-demographic record, such as a student's courses or grades, to be linked to its associated student demographic record's unique student identifier assigned by the previous functional rule RCT-011. A balance-line algorithm is used to link each non-demographic record to a student's demographic record. Once linked, the unique student identifier assigned to the student's demographic record is also assigned to the student's non-demographic record.

If a non-demographic file is expected to have referential integrity with a demographic file, but a student non-demographic record cannot be linked to a unique student identifier, the entire non-demographic file is rejected, the record at fault is identified on a rejection report, but the entire file is still analyzed completely to identify any other data integrity problem.

When a non-demographic file does not have the referential integrity with the demographic file, faulty records only are rejected. Depending on the Unit Task, a rejection report may be produced.

RCT-014 GENERATING ATTRIBUTES CREATE SYSTEM SOURCE AND UPDATE SYSTEM SOURCE

The attributes CREATE SYSTEM SOURCE and UPDATE SYSTEM SOURCE are 5 character codes identifying the source system that created or modified a table entry. These two fields are present in all data warehouse tables. The create attribute is mandatory and the update attribute is required when the entity is modified. Values to be used are identified by each Cleanse and Match Unit Task.