Design Doc – Risk Processing

**Aim:** Process to calculate final risk score of all students and create a file containing details of students with lower risk scores.

S3 object

Processing

Aurora db

**Development:**

For each student we will calculate the score for each category based on rules and configurations in the below files.

  

1. 02-School Fuel - At-Risk Definitions file contains the conditions based on which scores can be calculated for each category.
2. AtRiskControlFile contains what scores should be assigned when a particular condition is met for particular schoolID. This is a config file.
3. Final scores table structure file contains structure of final table in which scores of all students of the district will be placed.

**Additional logic to S3 to aurora load process:**

Whenever a file is loaded to auroradb we will be inserting a record into the table described below. This record indicates what file was received for a particular month.

**Table Name:** risk\_processing\_config

**Schema:** Processing

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **possible values** |
| Filename | varchar | Name of the file (student.csv) |
| Received\_Date | date | MM/DD/YYYY |
| Aurora\_load\_flag | varchar | Y' or 'N' |
| Risk\_Processing\_flag | varchar | Y' or 'N' |

Example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Filename**   |  | | --- | |  | | **Received\_Date** | **Aurora\_load** | **Risk\_Processing** |
| Student.csv | 8/1/2022 | Y | N |
| StudentPerformance.csv | 8/1/2022 | Y | N |

If record is already present for a month and we are reinserting a new file, update the column “Risk\_Processing” of the record to ‘N’.

Adding config table to aurora:

We need to add below table to aurora to store config information from “AtRiskControlFile” file.

Table structure is present in “AtRiskControlFile” file above.

When a config file is placed in path {s3 bucket name}/inbound-files/config-files, we need to trigger the lambda function to execute sf\_s3\_to\_aurora\_load.py script in which we will call another script (sf\_risk\_config\_aurora\_load.py) to load this config file to the above table.`

**Processing Flow:**

We will use AWS step function to orchestrate the risk processing. Everyday step function will check if all required files were received and if risk processing is done for that month or not using risk\_processing\_flag table. If risk process needs to be done the step function will trigger required lambda function. The flow is shown as below.

sf\_atrisk\_aurora\_to\_s3.py (Lambda)

YES

sf\_risk\_processing.py (Lambda)

NO

Aurora\_load=’Y’ (For all Fact tables) && Risk\_Processing = ‘N’ for at least one record for that month

**Processing Scripts:**

**sf\_risk\_processing.py:** This script should be placed in the lambda function (schoolfuel-{entity\_id}-lambda-risk-processing). This script sequentially uses imported functions from scripts (sf\_{category}\_risk\_processing.py) to calculate risk scores for each category (eg: sf\_guacs\_risk\_processing.py).

In sf\_risk\_processing.py:

1. Check if there is data in “student\_risk\_scores” for current month & Year using process\_Date column. If no data, insert all students into the table with the current processing date.
2. If records are already present for current month continue the process with no action
3. Next sequentially call functions to process scores for each category.

**sf\_guacs\_risk\_processing.py** (For Gifted under achievement category):

1. Identify student with Gifted = ‘N’ in student table. For these students update element1 = ‘1-GUACS-001’ and update Element1\_score = NULL.
2. For rest of the students process the records in StudentsPerformance table for last 90 days.
3. For each record calculate score based on logic given in file (02-School Fuel - At-Risk Definitions). Assign score based on config file (AtRiskControlFile).
4. Average the score by grouping all records of a student.
5. This average score should be updated to Element1\_score of the “student\_risk\_scores” table.
6. Update the Final\_score column as well based on current score
7. Update the process\_Date

Similar process should be followed for other categories based on Logic given and config files.

At the end update the “risk\_processing” column to ‘Y’ in “risk\_processing\_flag” table.

**sf\_atrisk\_aurora\_to\_s3.py**: This script to is identify students at risk based on scores and load these students data to S3 in folder {S3 bucket}/outbound-files, as csv files.

The students at risk should be identified based on “Risk\_cutoff3” column value in “risk\_control\_config” table.