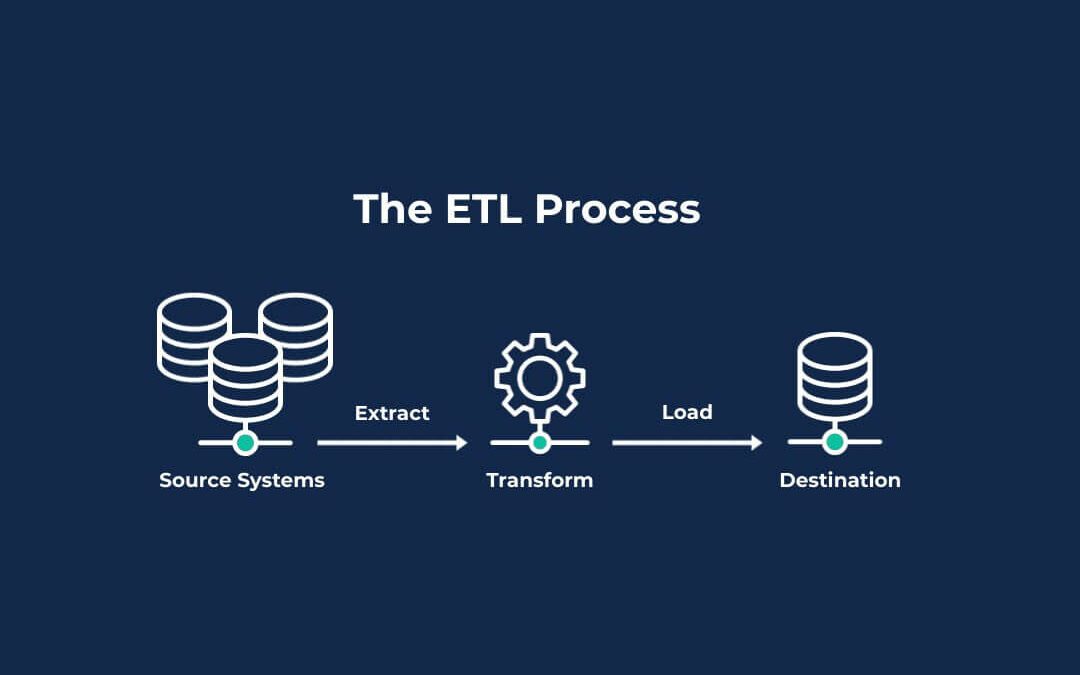
**Interim Project Report**

**STUDENT REPORT**

****

**PREPARED BY:**

|  |  |
| --- | --- |
| **ASSOCIATE ID** | **ASSOCIATE NAME** |
| 2387458 | Shreeraj Patil |
| 2387503 | Khalid Ahamed |
| 2387452 | Dinesh Mummidivarapu |
| 2387460 | Yukesh Vasudevan |
| 2387419 | Priyanshi |
| 2387415 | Samrudhi Parte |

**GUIDED BY:**

**Mentor Trainer**

Kavin Kumar G Kavin Kumar G

Associate Associate

AIA - Cloud Data Integration AIA - Cloud Data Integration

**CONTENTS**

|  |  |
| --- | --- |
| * Introduction | 6 |
| * Purpose of the Project | 6 |
| * About the Software System | 6-7 |
| * Scope of the System | 7 |
| * Architecture Diagram | 8 |
| * Flowchart | 10 |
| * Functional Requirement 1: Department Dimension | 11-14 |
| * Functional Requirement 2: Student Dimension | 15-18 |
| * Functional Requirement 3: Marks Fact | 19-23 |
| * Functional Requirement 4: Subject Average Aggregate | 24-27 |
| * Functional Requirement 5: Subject Pass Percent Aggregate | 28-31 |
| * Task View and Session Statistics | 32 |
| * Design Constraints | 33 |
| * Conclusion and Future Scope | 34 |

**ABBREVIATIONS**

|  |  |
| --- | --- |
| **TERM** | **DESCRIPTION** |
| ETL | Extraction, Transformation and Loading / Extract, Transform and Load |
| SRC | Source |
| TGT | Target |
| SQ | Source Qualifier |
| M | Mapping |
| WF | Workflow |
| S | Session |
| EXP | Expression |
| DIM | Dimension |
| AGG | Aggregate |
| AVG | Average |

**TABLES**

Table 1: Department Dimension Source Description

Table 2: Department Dimension Target Description

Table 3: Department Dimension Transformations

Table 4: Student Dimension Source Description

Table 5: Student Dimension Target Description

Table 6: Student Dimension Transformation

Table 7: Marks Fact Source Description

Table 8: Marks Fact Target Description

Table 9: Marks Fact Transformations

Table 10: Subject Average Aggregate Source Description

Table 11: Subject Average Aggregate Target Description

Table 12: Subject Average Aggregate Transformation

Table 13: Subject Pass Percentage Aggregate Source Description

Table 14: Subject Pass Percentage Aggregate Target Description

Table 15: Subject Pass Percentage Aggregate Transformations

Table 16: Design Constraints

**FIGURES**

Fig. 1: Architecture Diagram

1. Physical Architecture
2. Logical Architecture

Fig. 2: Flow Chart

Fig. 3: Department Dimension Target Definition

Fig. 4: Department Dimension Mapping

Fig. 5: Department Dimension Workflow

Fig. 6: Department Dimension Target Table

Fig. 7: Student Dimension Target Definition

Fig. 8: Student Dimension Mapping

Fig. 9: Student Dimension Workflow

Fig. 10: Student Dimension Target Table

Fig. 11: Marks Fact Target Definition

Fig. 12: Marks Fact Mapping

Fig. 13: Marks Fact Workflow

Fig. 14: Marks Fact Target Table

Fig. 15: Subject Average Aggregate Target Definition

Fig. 16: Subject Average Aggregate Mapping

Fig. 17: Subject Average Aggregate Workflow

Fig. 18: Subject Average Aggregate Target Table

Fig. 19: Subject Pass Percentage Aggregate Target Definition

Fig. 20: Subject Pass Percentage Aggregate Mapping

Fig. 21: Subject Pass Percentage Aggregate Workflow

Fig. 22: Subject Pass Percentage Aggregate Target Table

Fig. 23: Final Workflow

Fig. 24: Task Details

**INTRODUCTION**

The software requirements document's objective is to methodically record the needs for the project and the "Student Report" system that will be created. This document gives a clear and thorough grasp of the expected functionality by outlining the system's functional requirements. It acts as the cornerstone for project scoping, guaranteeing that all relevant factors are taken into account prior to moving forward with the design and implementation stages. This document also aids in the identification of possible dependencies and hazards, which promotes improved resource allocation and planning.

This document's scope is restricted to needs from non-functional, quality, and user viewpoints. In order to make sure the produced system satisfies the required standards, it seeks to record user wants, expectations, and restrictions. In order to provide a solid and dependable system, the document highlights the significance of non-functional needs including performance, security, and usability. This document purposefully omits design elements in order to keep requirements front and centre and prevent misunderstandings throughout the development process.

**PURPOSE OF THE PROJECT**

The "Student Report" project aims to provide an all-inclusive system that effectively collects and handles student performance data. In order to guarantee that both students and educators have access to correct and current information, this project intends to develop an intuitive platform where educators may enter, monitor, and evaluate student reports. The project aims to lessen administrative responsibilities and improve the overall educational experience by automating and optimizing the report generating process.

Because this system offers a consolidated repository of student performance data, it will also help instructors, students, and parents communicate more effectively. By providing insights and statistics on student achievement, it seeks to enhance educational outcomes and facilitate data-driven decision-making. The goal of the "Student Report" project is to develop a solid, scalable, and dependable system that can accommodate the various needs of its users and support ongoing enhancements to the educational process.

**ABOUT THE SOFTWARE SYSTEM**

Effective management and analysis of student performance data is the goal of the "Student Report" software system. Teachers may enter, monitor, and provide comprehensive reports on students' academic progress using this system's centralized platform. The system seeks to improve the accuracy and dependability of student performance records while lowering administrative burdens by optimizing the data collection and report generating process.

User authentication, data input forms, report production modules, and data analytics tools are just a few of the functional components that make up the software system. Teachers, administrators, and other stakeholders may simply explore and utilize its capabilities because of its user-friendly design. Additionally, the system facilitates the integration of data from many sources, providing a thorough understanding of student performance across subjects and evaluation periods.

Furthermore, the system places a high priority on data security and privacy, putting strong safeguards in place to guard private student data. It is designed to be flexible and scalable, meeting the changing requirements of educational establishments. The "Student Report" software system gives teachers the ability to make well-informed decisions, pinpoint areas that need development, and ultimately improve the overall educational experience for students by offering timely insights and analytics.

**SCOPE OF THE SYSTEM**

The "Student Report" software system's scope includes a number of crucial elements to guarantee that it satisfies the various demands of its customers. With an emphasis on effectively gathering and handling student performance data, the system is made to meet functional requirements. Data input forms for capturing student information, modules for creating comprehensive reports, analytics tools to offer insights into student development and performance trends, and user authentication and authorization functions are all included. By offering an intuitive user interface and simple navigation, the system seeks to meet the demands of educators, administrators, and other stakeholders. It guarantees thorough and accurate student performance records by enabling the smooth integration of data from various sources. Additionally, the system prioritizes non-functional needs like privacy and data security, guaranteeing that private student data is safeguarded by strong security protocols.

**ARCHITECTURE DIAGRAM**

1. **PHYSICAL ARCHITECTURE:**

The physical architecture of the "Student Report" is composed of physical components, including system elements and interfaces, designed to effectively manage customer data. This architecture aims to meet the logical architecture elements and system requirements.

The physical architecture is divided into two primary layers:

**Source Layer**: This layer comprises four source systems that supply raw data. These sources are linked to the data warehouse layer through various data loading processes.

**Data Warehouse Layer**: This layer contains independent dimensions and fact tables that store the processed data.

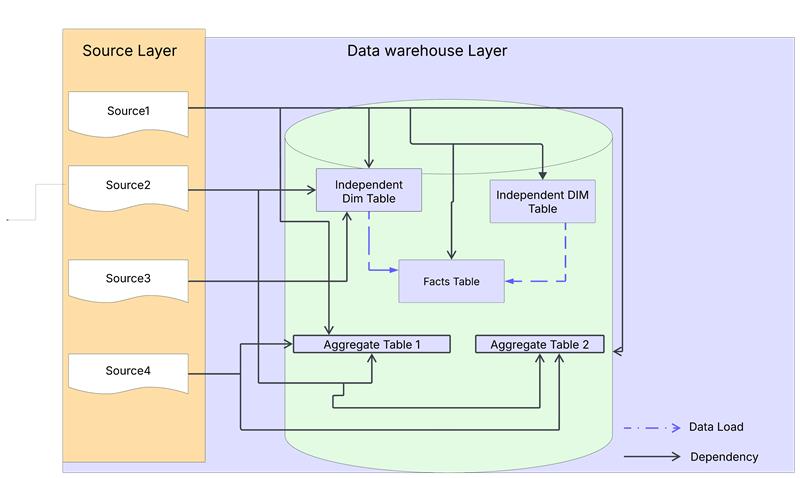


Fig. 1a) Physical Architecture Diagram

1. **LOGICAL ARCHITECTURE:**

The logical architecture of the "Student Report” outlines the processes and activities necessary to deliver user services. This architecture is technology-agnostic, concentrating solely on the logical flow of data and procedures.

The architecture comprises two principal layers: the source layer and the data warehouse layer.

**Source Layer:** This layer consists of three sources that supply the raw data.

**Data Warehouse Layer**: This layer encompasses dimensions and fact tables that store the processed data.

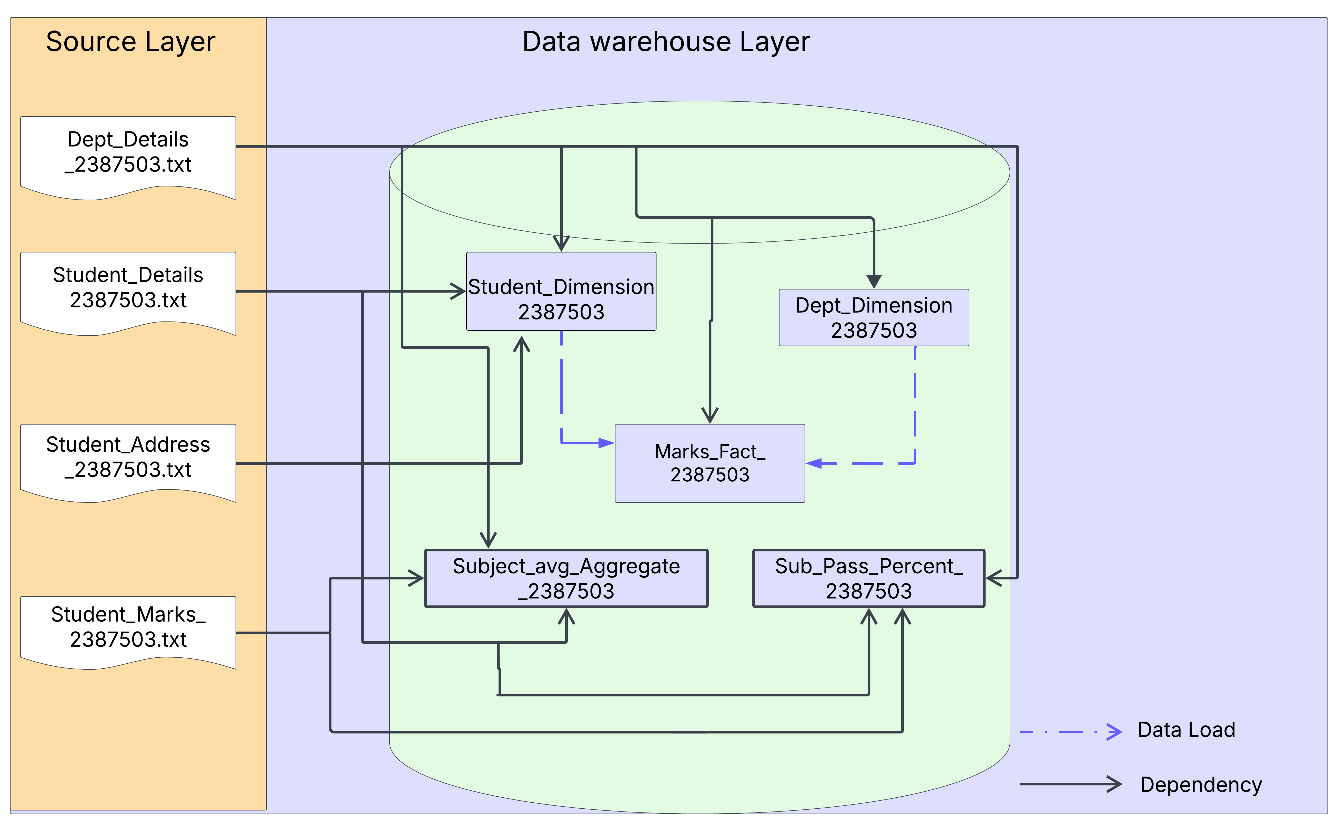


Fig. 1b) Logical Architecture Diagram

**FLOW CHART**

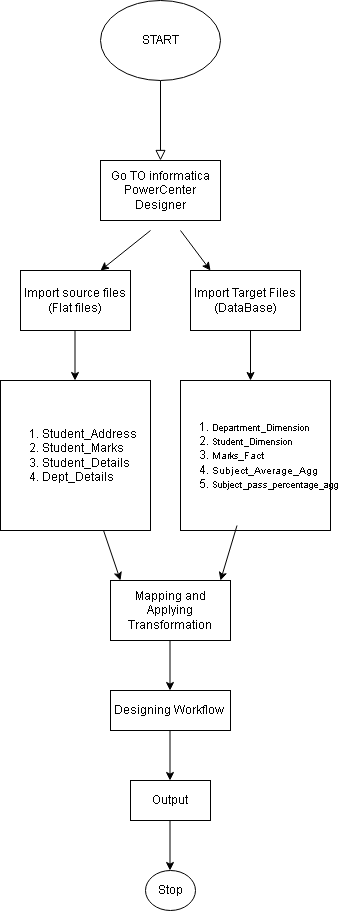


Fig. 2: Flow Chart

**Functional Requirement 1**

**DEPARTMENT DIMENSION**

**SOURCE**

The first functional requirement is to load the data into the Department Table from a source flat file as a one-to-one load.

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| Dept\_Details\_2387460.txt | This is a tab-delimited file with 2 input fields.  Columns:   1. Dept Code 2. Dept Name |  |

Table 1: Department Dimension Source Description

**TARGET**

The target table has been generated by creating a schema from the data provided in the requirements document.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET SCHEMA** |
| ***DEPARTMENT\_DIMENSION*** | It is a dimension table maintaining department information. | Dimension Table |  |

Table 2: Department Dimension Target Description

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 3: Department Dimension Target Definition

**TRANSFORMATIONS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Mapping Logic** | **Transformation Name** | **Function Used** |
| DEPT\_ID | Direct Mapping | Source Qualifier |  |
| DEPT\_CODE | Direct Mapping | Source Qualifier |  |

Table 3: Department Dimension Transformations

**MAPPING**

The figure below illustrates the data flow from the source to the target table, applying the one-to-one load to meet the business requirements.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 4: Department Dimension Mapping

**WORKFLOW**

The figure below illustrates the workflow that executes the mappings created in the designer. This workflow ensures that the data is extracted from the source files, transformed according to the specified business logic, and loaded into the target tables in the data warehouse.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 5: Department Dimension Workflow

**TARGET POST EXECUTION:**

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 6: Department Dimension Target Table

**Functional Requirement 2**

**STUDENT DIMENSION**

**SOURCE**

The second functional requirement is to load the data into the Student Table from two source flat files i.e;

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| Student\_Details\_2387460.txt | This is a tab-delimited file with 3 input fields.  Columns:  Student\_Id  Student\_Name  Dept\_Code |  |
| Student\_Address\_2387460.txt | This is a tab-delimited file with 4 input fields.  Columns:  Student\_Id  Address  Postal\_Code  Phone\_Number |  |

Table 4: Student Dimension Source Description

**TARGET**

The target table has been generated by creating a schema from the data provided in the requirements document.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET SCHEMA** |
| ***STUDENT\_DIMENSION*** | It is a dimension table maintaining detailed student information. | Dimension Table |  |

Table 5: Student Dimension Target Description

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 7: Student Dimension Target Definition

**TRANSFORMATIONS:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Target Table Column** | **Source Column Name (Student Details)** | **Source Column Name (Student Address)** | **Mapping Logic** | **Transformation Name** | **Function Used** |
| Student\_ID | Student\_ID | Student\_ID | Join on Student\_ID | Joiner |  |
| Student\_name | Student\_name |  | Direct Mapping |  |  |
| Dept\_ID | Dept\_ID |  | Direct Mapping |  |  |
| Address |  | Address | Converted into uppercase | Expression Transformation | UPPER |
| Postal\_code |  | Postal\_code | Cleansed by removing extra space and special characters | Expression Transformation | REG\_REPLACE, LTRIM, RTRIM |
| Phone\_Number |  | Phone\_Number | validated for valid 10-digit phone number | Expression Transformation | IIF, LENGTH |

Table 6: Student Dimension Transformations

**MAPPING**

The figure below illustrates the data flow from the source to the target table, applying the necessary transformations to meet the business requirements.

A computer screen shot of a computer

AI-generated content may be incorrect.

Fig. 8: Student Dimension Mapping

**WORKFLOW**

The figure below illustrates the workflow that executes the mappings created in the designer. This workflow ensures that the data is extracted from the source files, transformed according to the specified business logic, and loaded into the target tables in the data warehouse.

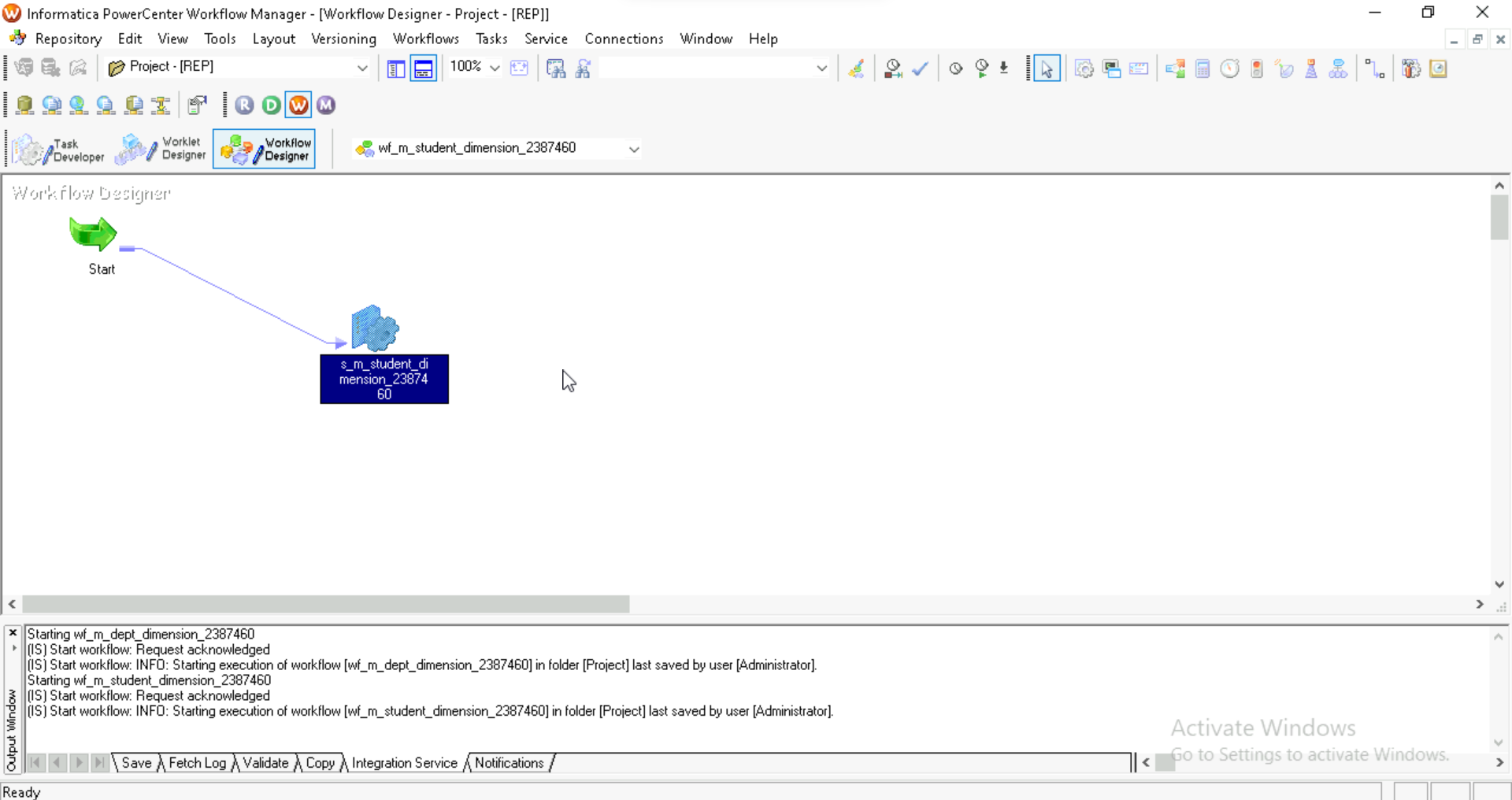


Fig. 9: Student Dimension Workflow

**TARGET POST EXECUTION:**

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 10: Student Dimension Target Table

**Functional Requirement 3**

**MARKS FACT**

**SOURCE**

The third functional requirement is to load the data into the Marks Table from three source flat files i.e;

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| Dept\_Details\_2387460.txt | This is a tab-delimited file with 2 input fields. |  |
| Student\_Details\_2387460.txt | This is a tab-delimited file with 3 input fields. |  |
| Student\_Marks\_2387460.txt | This is a tab-delimited file with 7 input fields |  |

Table 7: Marks Fact Source Description

**TARGET**

The target table has been generated by creating a schema from the data provided in the requirements document.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET SCHEMA** |
| ***MARKS\_FACT*** | It is a fact table maintaining detailed student information, including Student ID, Student name, Department details, marks in subjects, etc. | Fact Table |  |

Table 8: Marks Fact Target Description

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 11: Marks Fact Target Definition

**TRANSFORMATIONS:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Target**  **Table**  **Column** | **Source Column Name**  **(Student Details)** | **Source**  **Column Name**  **(Dept\_Detail)** | **Source**  **Column Name**  **(Student\_Marks)** | **Mapping Logic** | **Transformation Name** | **Function Used** |
| Marks\_Id |  |  |  |  | Sequence  generator |  |
| Student\_ID | Student\_ID |  |  | Direct Mapping |  |  |
| Student\_name | Student\_name |  |  | Direct Mapping |  |  |
| Dept\_ID  Dep\_Name | Dept\_ID | Dept\_ID  Dep\_Name | Dept\_ID | Joiner on Dept\_id  Direct Mapping | Joiner |  |
| Subject1 |  |  | Mark1 | Direct Mapping |  |  |
| Subject2 |  |  | Mark2 | Direct Mapping |  |  |
| Subject3 |  |  | Mark3 | Direct Mapping |  |  |
| Subject4 |  |  | Mark4 | Direct Mapping |  |  |
| Subject5 |  |  | Mark5 | Direct Mapping |  |  |
| Total |  |  |  | Sum of All subjects | Expression |  |
| Average |  |  |  | Sum of subjects / no of subjects | Expression |  |
| Result |  |  |  | Marks in each subject > 50 | Expression | IIF |

Table 9: Marks Fact Transformations

**MAPPING**

The figure below illustrates the data flow from the source to the target table, applying the necessary transformations to meet the business requirements.

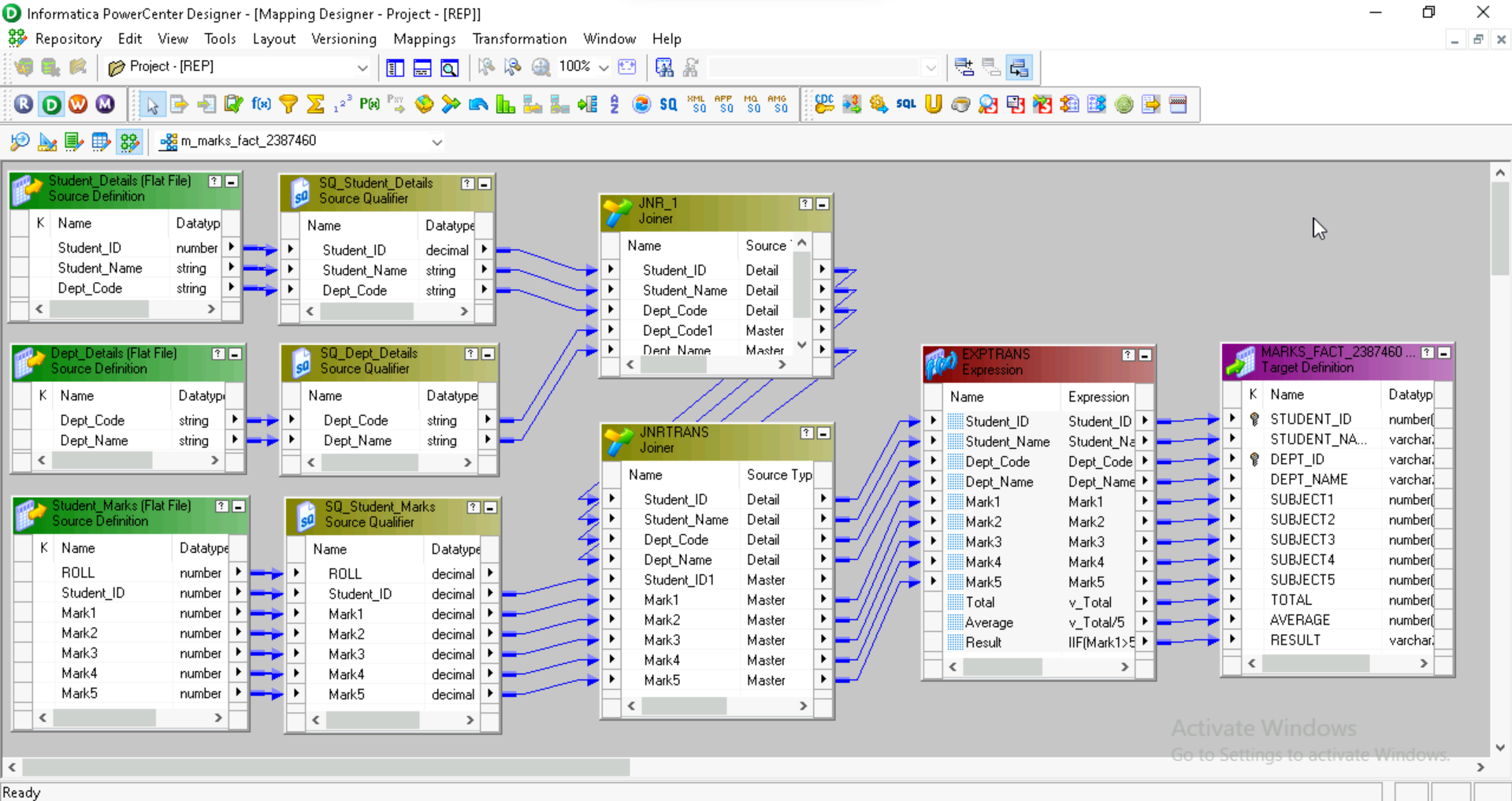


Fig. 12: Marks Fact Mapping

**WORKFLOW**

The figure below illustrates the workflow that executes the mappings created in the designer. This workflow ensures that the data is extracted from the source files, transformed according to the specified business logic, and loaded into the target tables in the data warehouse.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 13: Marks Fact Workflow

**TARGET POST EXECUTION:**

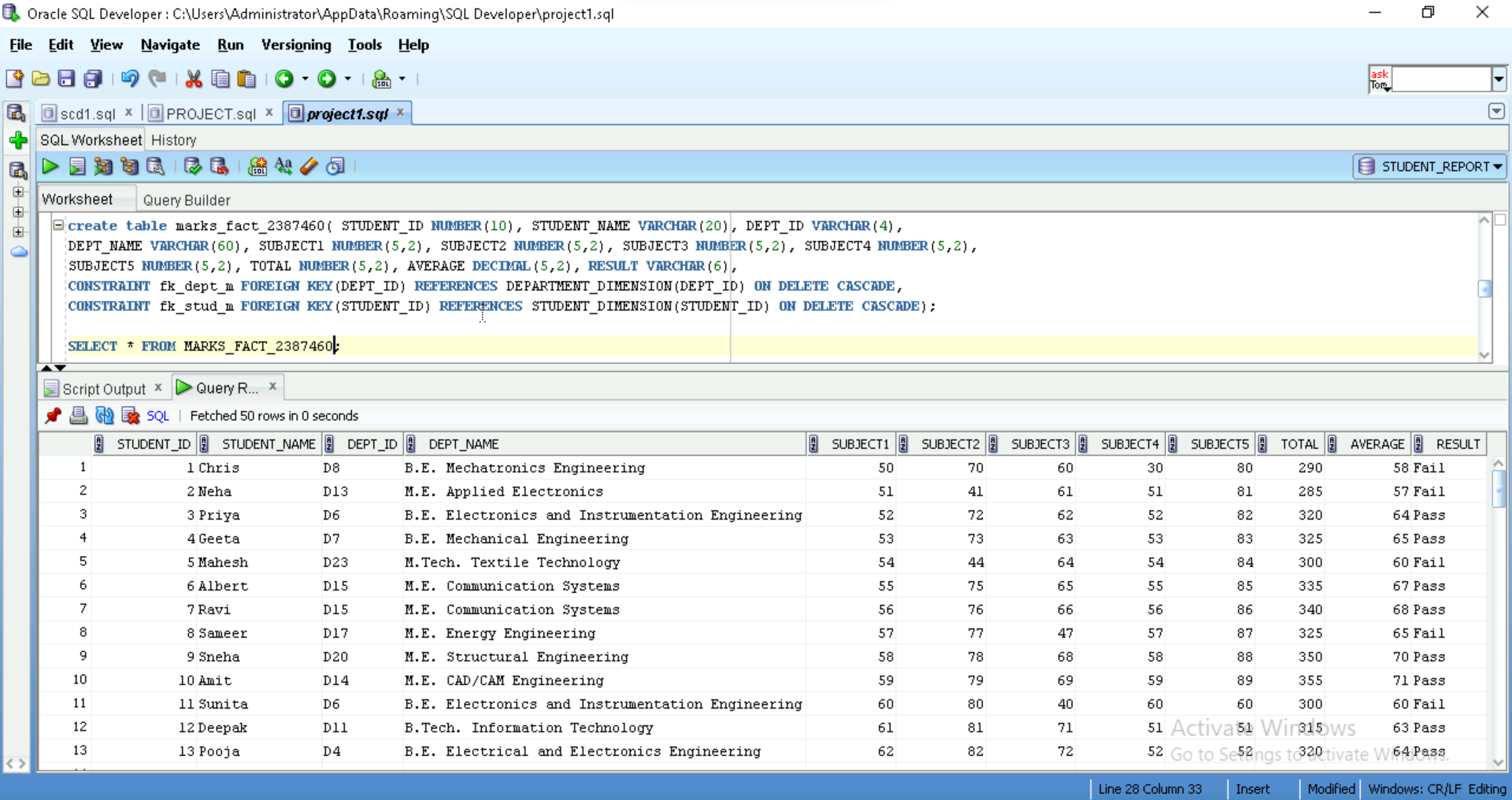


Fig. 14: Marks Fact Target Table

**Functional Requirement 4**

**SUBJECT\_AVERAGE\_AGGREGATE \_TABLE**

**SOURCE**

The fourth functional requirement is to load the data into the Subject Average Aggregate Table from a source flat file i.e;

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| Dept\_Details\_2387460.txt | This is a tab-delimited file with 2 input fields. |  |
| Student\_Marks\_2387460.txt | This is a tab-delimited file with 7 input fields. |  |
| Student\_Details\_2387460.txt | This is a tab-delimited file with 3 input fields. |  |

Table 10: Subject Average Aggregate Source Description

**TARGET**

The target table has been generated by creating a schema from the data provided in the requirements document.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET SCHEMA** |
| ***SUBJECT\_AVERAGE\_AGGREGATE*** | It is a fact table maintaining detailed department information, including Department ID, Department Name  And average marks in each subject. | Aggregate Table |  |

Table 11: Subject Average Aggregate Target Description

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 15: Subject Average Aggregate Target Definition

**TRANSFORMATIONS:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Target Column Name** | **Column Name**  **(Dept\_Details)** | **Column Name**  **(std\_Details)** | **Column Name**  **(Std\_marks)** | **Mapping Logic**  **(std\_Marks)** | **Transformation Name** | **Function Used** |
| Dept\_ID | Dept\_ID | Dept\_ID |  | Joiner on Dept\_Id | Joiner |  |
| Dept\_Name | Dept\_Name |  |  | Direct Mapping |  |  |
| Subject1 Avg |  |  | marks1 | Avg of marks 1  Group by Dept\_ID | AggTrans | Avg |
| Subject2 Avg |  |  | marks2 | Avg of marks 2  Group by Dept\_ID | AggTrans | Avg |
| Subject3 Avg |  |  | marks3 | Avg of marks 3  Group by Dept\_ID | AggTrans | Avg |
| Subject4 Avg |  |  | marks4 | Avg of marks 4  Group by Dept\_ID | AggTrans | Avg |
| Subject5 Avg |  |  | marks5 | Avg of marks 5  Group by Dept\_ID | AggTrans | Avg |

Table 12: Subject Average Aggregate Transformations

**MAPPING**

The figure below illustrates the data flow from the source to the target table, applying the necessary transformations to meet the business requirements.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 16: Subject Average Aggregate Mapping

**WORKFLOW**

The figure below illustrates the workflow that executes the mappings created in the designer. This workflow ensures that the data is extracted from the source files, transformed according to the specified business logic, and loaded into the target tables in the data warehouse.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 17: Subject Average Aggregate Workflow

**TARGET POST EXECUTION:**



Fig. 18: Subject Average Aggregate Target Table

**Functional Requirement 5**

**SUBJECT\_PASS\_PERCENTAGE\_AGGREGATE \_TABLE**

**SOURCE**

The fourth functional requirement is to load the data into the Subject Average Aggregate Table from a source flat file i.e;

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| Dept\_Details\_2387460.txt | This is a tab-delimited file with 2 input fields. |  |
| Student\_Marks\_2387460.txt | This is a tab-delimited file with 7 input fields. |  |
| Student\_Details\_2387460.txt | This is a tab-delimited file with 3 input fields. |  |

Table 13: Subject Pass Percentage Aggregate Source Description

**TARGET**

The target table has been generated by creating a schema from the data provided in the requirements document.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET SCHEMA** |
| ***SUBJECT\_PASS\_PERCENT\_AGGREGATE*** | It is a fact table maintaining detailed department information, including Department ID, Department Name  And pass percentage in each subject. | Aggregate Table |  |

Table 14: Subject Pass Percentage Aggregate Target Description

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 19: Subject Pass Percentage Aggregate Target Definition

**TRANSFORMATIONS:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Target Column** | **Column Name**  **(Dept\_Details)** | **Column Name**  **(Stud\_details)** | **Column Name**  **(std\_marks)** | **Mapping Logic** | **Transformation Name** | **Function Used** |
| Dept\_Id | Dept\_Id | Dept\_Id |  | Joiner on Dept\_Id | Joiner |  |
| Dept\_Name | Dept\_Name |  |  | Direct Mapping |  |  |
| Subject1\_pass  \_Percentage |  |  | Mark1 |  | Agg\_Trans | Sum,count |
| Subject2\_pass  \_Percentage |  |  | Mark2 |  | Agg\_Trans | Sum,count |
| Subject3\_pass  \_Percentage |  |  | Mark3 |  | Agg\_Trans | Sum,count |
| Subject4\_pass  \_Percentage |  |  | Mark4 |  | Agg\_Trans | Sum,count |
| Subject5\_pass\_Percentage |  |  | Mark5 |  | Agg\_Trans | Sum,count |

Table 15: Subject Pass Percentage Aggregate Transformations

**MAPPING**

The figure below illustrates the data flow from the source to the target table, applying the necessary transformations to meet the business requirements.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 20: Subject Pass Percentage Aggregate Mapping

**WORKFLOW**

The figure below illustrates the workflow that executes the mappings created in the designer. This workflow ensures that the data is extracted from the source files, transformed according to the specified business logic, and loaded into the target tables in the data warehouse.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 21: Subject Pass Percentage Aggregate Workflow

**TARGET POST EXECUTION:**

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 22: Subject Pass Percentage Aggregate Target Table

**TASK VIEW AND SESSION STATISTICS**

**A screenshot of a computer

AI-generated content may be incorrect.**

Fig. 23: Final Workflow

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 24: Task Details

**DESIGN CONSTRAINTS**

|  |  |  |
| --- | --- | --- |
| **CONSTRAINT** | **SUCCESS** | **IMPLEMENTATION** |
| Department Dimension table should be loaded from Department Details source file as a one-to-one load | Yes | The Department Details has been loaded into Department Dimension table successfully. |
| Student Dimension table should be loaded from Student Details and Student Address source files | Yes | Student Details and Student Address are connected using Student\_id and loaded into the target table. |
| Postal code column should be cleansed to remove any extra space and special characters | Yes | The extra spaces and special characters were removed from the Postal code. |
| Phone Number column should be validated for valid 10-digit phone number, if there are any invalid number found then phone number must be loaded as NULL. | Yes | The phone number validation logic functions effectively by retaining the valid data as it is, while assigning NULL to any data that fails validation. |
| All characters in address column must be converted to upper case. | Yes | The data in address field is changed successfully to title case. |
| Marks Fact table should be loaded from Student Details, Student Address and Student marks source files. | Yes | Student Details, Student Address and Student Marks are connected using the common column Student\_id and loaded into the target table. |
| Total must be calculated as sum of all subject marks. | Yes | Applied transformation logic using SUM() function. |
| Average column must be calculated as average of all marks. | Yes | Applied transformation logic using AVG() function. |
| Result must be calculated as Pass or Fail, Students scoring 50 and above marks in all subjects are termed as Pass. | Yes | The Pass/Fail validation is implemented in Marks Fact table successfully. |
| Subject Average Aggregate Table should be loaded from Department Details, Student Details and Student Marks table. | Yes | Student Details, Student Marks and Department Details are connected using the common columns Student\_id and Dept\_id. |
| Table should contain average marks scored by students in all subjects by department. | Yes | Applied Transformation logic to load average marks scored by students in all subjects by department wise in Subject Average Aggregate Table. |
| Subject Pass Percentage Aggregate Table should be loaded from Department Details, Student Details and Student Marks table. | Yes | Student Details, Student Marks and Department Details are connected using the common columns Student\_id and Dept\_id. |
| Table should contain pass percentage for each subject by department. | Yes | Applied Transformation logic to load pass percentage of each subject by department in Subject Pass Percentage  Aggregate Table. |
| All sessions should be connected in a sequential manner in one single workflow. If there is any failure to one of the sessions, then following sessions should not run. | Yes | Implemented validation checks and abort logic in the ETL process. |

Table 16: Design Constraints

**CONCLUSION AND FUTURE SCOPE**

All business requirements were considered while developing the "**Student Report System**." By creating three target tables—Dept\_Dimension, Student\_Dimension, and Marks\_Fact—the data was accurately loaded into the dimensions and fact table. The system successfully implemented all the necessary transformations and validations to ensure data integrity and quality. By successfully loading and testing the data, the project's main goals have been accomplished. The system has been operated in accordance with the functional specifications outlined in the system requirements document. The "**Student Report System**" is thus successful in implementing all the requirements and can be executed to manage and analyze customer data effectively.

The future scope of the "**Student Report System**" includes several enhancements leveraging Informatica PowerCenter. These enhancements involve integrating additional data sources to provide a more comprehensive view of student interactions, implementing advanced analytics and reporting capabilities for deeper insights, and optimizing the system's scalability and performance to handle large data volumes. Additionally, enhancing data quality and governance frameworks will ensure ongoing data accuracy and compliance. By pursuing these improvements, the "**Student Report System**" can continue to evolve, providing greater value and supporting strategic decision-making within the organization.