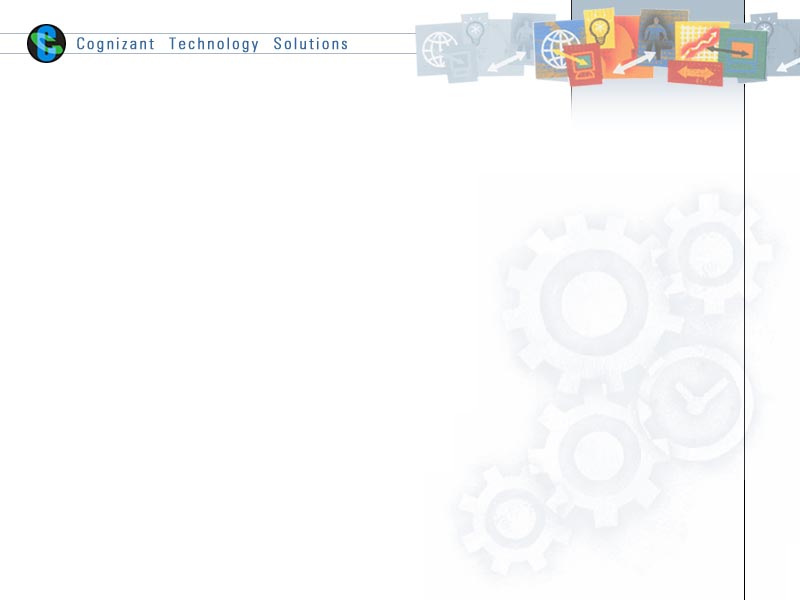
**Final Project Report**

**UNIVERSITY ANALYTICS SYSTEM**



**PREPARED BY:**

|  |  |
| --- | --- |
| **ASSOCIATE ID** | **ASSOCIATE NAME** |
| 2387333 | Praveen Annadurai |

**GUIDED BY:**

**Mentor & Trainer Coach**

Kavin Kumar G Akash Nair

Associate CPO L&D GenC Training

AIA - Cloud Data Integration Contractor – GenC

**CONTENTS**

* **Introduction**
* **Purpose of the Project**
* **About the Software System**
* **Scope of the System**
* **Architecture Diagram**
* **Flowchart**
* **Functional Requirement 1: Student Dimension**
* **Functional Requirement 2: College Dimension**
* **Functional Requirement 3: Faculty Dimension**
* **Functional Requirement 4: Department Dimension**
* **Functional Requirement 5: Course Dimension**
* **Functional Requirement 6: Time Dimension**
* **Functional Requirement 7: Result Staging**
* **Functional Requirement 8: Result Fact**
* **Functional Requirement 9: Result Aggregate**
* **Integrated Taskflow**
* **Integrated Task Details**
* **Design Constraints**
* **Conclusion**

**ABBREVIATIONS**

|  |  |
| --- | --- |
| **TERM** | **DESCRIPTION** |
| ETL | Extraction, Transformation and Loading / Extract, Transform and Load |
| SRC | Source |
| TGT | Target |
| SQ | Source Qualifier |
| WF | Workflow |
| EXP | Expression |
| SEQ | Sequence Generator |
| LKP | Lookup |
| AGG | Aggregator |
| FIL | Filter |
| RTR | Router |
| SRT | Sorter |
| UN | Union |
| JNR | Joiner |

**TABLES**

Table 1: Student Dimension Source Description

Table 2: Student Target Description

Table 3: Student Dimension Transformations

Table 4: College Dimension Source Description

Table 5: College Target Description

Table 6: College Dimension Transformations

Table 7: Faculty Dimension Source Description

Table 8: Faculty Target Description

Table 9: Faculty Dimension Transformations

Table 10: Department Dimension Source Description

Table 11: Department Target Description

Table 12: Department Transformations

Table 13: Course Dimension Source Description

Table 14: Course Target Description

Table 15: Course Transformations

Table 16: Time Dimension Source Description

Table 17: Time Target Description

Table 18: Time Transformations

Table 19: Result Staging Dimension Source Description

Table 20: Result Staging Target Description

Table 21: Result Staging Transformations

Table 22: Result Fact Dimension Source Description

Table 23: Result Fact Target Description

Table 24: Result Fact Transformations

Table 25: Result Aggregate Student Dimension Source Description

Table 26: Result Aggregate Target Description

Table 27: Result Aggregate Transformations

Table 28: Design Constraints

**FIGURES**

Fig. 1: System Flow Diagram

Fig. 2: Architecture Diagram

Fig. 3: Flow Chart

Fig. 4: Student Mapping

Fig. 5: Student Dimension Mapping Task

Fig. 6: Student Dimension Monitor

Fig. 7: Student Dimension Target Table

Fig. 8: College Mapping

Fig. 9: College Dimension Mapping Task

Fig. 10: College Dimension Monitor

Fig. 11: College Dimension Target Table

Fig. 12: Faculty Mapping

Fig. 13: Faculty Dimension Mapping Task

Fig. 14: Faculty Dimension Monitor

Fig. 15: Faculty Dimension Target Table

Fig. 16: Department Mapping

Fig. 17: Department Dimension Mapping Task

Fig. 18: Department Dimension Monitor

Fig. 19: Department Dimension Target Table

Fig. 20: Course Target Definition

Fig. 21: Course Mapping

Fig. 22: Course Dimension Mapping Task

Fig. 23: Course Dimension Monitor

Fig. 24: Time Mapping

Fig. 25: Time Dimension Mapping Task

Fig. 26: Time Dimension Monitor

Fig. 27: Time Dimension Target Table

Fig. 28: Staging Mapping

Fig. 29: Staging Mapping Task

Fig. 30: Staging Result Monitor

Fig. 31: Staging Result Target Table

Fig. 32: Staging Error Target Table

Fig. 33: Result Fact Mapping

Fig. 34: Result Fact Mapping Task

Fig. 35: Result Fact Monitor

Fig. 36: Result Fact Target Table

Fig. 37: Result Aggregate Mapping

Fig. 38: Result Aggregate Mapping Task

Fig. 39: Result Aggregate Monitor

Fig. 40: Result Aggregate Target Table

Fig. 41: Integrated Taskflow

**Introduction**

The project titled University Analytics System aims to streamline and enhance the management of academic data within a university setting. This system focuses on aggregating data from multiple supporting dimension tables to generate a comprehensive Result table, which provides a unified view of critical information. By integrating data from various dimensions such as Student, Department, Time, Faculty, Course, and College, the system ensures efficient data handling and analysis, enabling informed decision-making and improved institutional management.

The dimension tables serve specific purposes: the Student table holds individual student records, while the Department table captures details of departments within the college. The Time dimension table organizes examination schedules, and the Faculty table stores faculty-related information. Similarly, the Course table maintains course-related data, and the College table provides an overview of college attributes, including its location, code, and name. Together, these tables form a robust foundation for data analytics, supporting a seamless flow of information and enabling the generation of accurate and actionable insights.

This project is significant as it addresses the need for a centralized analytical system in academic institutions, ensuring data consistency, accessibility, and enhanced decision-making capabilities. The introduction outlines the scope and structure of the project, highlighting its contribution to improving operational efficiency and educational outcomes.

**Purpose and Scope of the Project**

The primary purpose of the University Analytics System project is to develop an efficient and centralized system for managing and analyzing academic data within a university setting. By aggregating data from various dimension tables—such as Student, Department, Time, Faculty, Course, and College—the system aims to provide accurate and actionable insights that support decision-making and improve institutional operations. This project seeks to enhance the accessibility, consistency, and usability of academic data for administrators, faculty, and other stakeholders within the university.

The scope of this project encompasses the design and development of a Result table that integrates data from all supporting dimension tables, ensuring a comprehensive view of institutional analytics. This includes capturing and managing data related to students, departments, examinations, faculty, courses, and colleges, which collectively form the foundation for effective data analysis. The project is intended to serve as a robust analytics system that addresses business needs and operational challenges faced by universities, enabling data-driven decision-making and facilitating strategic improvements in academic and administrative processes.

**Aim of the Project**

The aim of the University Analytics System project is to create a centralized and efficient platform for managing and analyzing academic data across various dimensions within a university. The system seeks to generate a comprehensive Result table by aggregating data from supporting dimension tables, including Student, Department, Time, Faculty, Course, and College. By integrating these data sources, the project aims to provide accurate insights that support informed decision-making, improve institutional operations, and enhance the overall management of academic and administrative processes.

This project is designed to empower universities with a robust analytical framework, facilitating data-driven decisions that promote strategic improvements and operational excellence in the academic environment

**Modules present in the project**

1. Student dimension
2. College dimension
3. Time dimension
4. Department dimension
5. Course dimension
6. Faculty dimension
7. Result Staging
8. Result fact
9. Result Aggregate

**System diagram**

A green rectangle with black text

AI-generated content may be incorrect.

Fig.1: System Flow diagram

The system follows a structured data flow model comprising three key stages: Source, Staging, and Target. In the Source stage, raw data is collected from various input tables such as Student, Department, Faculty, and others. This data is then transferred to the Staging area, where it undergoes processing, transformation, and validation to ensure consistency and accuracy. Finally, the refined data is moved to the Target stage, where it is consolidated into the Result table, enabling comprehensive analysis and decision-making. This streamlined approach ensures efficient data integration and management across the system.

**Architecture Diagram**

The Logical Architecture defines the Processes (the activities and functions) that are required to provide the required User Services. Many different Processes must work together and share information to provide a User Service. The processes can be implemented via software, hardware, or firmware. Logical Architecture is independent of technologies and implementations.

University Analytics system has 6-dimensional table, 1 staging table and 1 fact table.

A diagram of a data warehouse layer

AI-generated content may be incorrect.

Fig. 2: Architecture Diagram

**Flow Chart**

A screenshot of a computer screen

AI-generated content may be incorrect.

**Stop**

Fig. 3: Flow Chart

**Functional Requirement 1**

**STUDENT**

The Student dimension table CDW\_UAS\_D\_Student is a Type – II SCD table which contains all students’ information. The source file’s name is CDW\_UAS\_STUDENT.txt. The source file is a comma (,) delimited file. The source sends the file on a daily basis. History of the students’ needs to be maintained in the table.

**SOURCE**

The source files provided in the requirement document is a comma separated file described in the table containing the following attributes like FIRST\_NAME, MIDDLE\_NAME, LAST\_NAME, ROLL, STREET\_NAME, APT\_NO, STUD\_CITY, STUD\_STATE, STUD\_COUNTRY, STUD\_ZIP, STUD\_PHONE and STUD\_EMAIL. The source is imported to mapping task by creating a connection to flat file provided the location of the file as shown in fig .

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| CDW\_UAS\_Student.txt | This is a delimited file (comma) with 12 input fields. |  |

Table 1: CDW\_UAS\_D\_STUDENT Source Description

**TARGET**

The target table is created in the SQL server database where the attributes are selected from the provided requirements document and imported to the Informatica mapping data integration. The output table description is explained in the table. The target database need to maintain unlimited history [SCD-2] with date flag.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET FILE** |
| CDW\_UAS\_D\_STUDENT | It is an SQL server database table maintaining the required information. | SQL server Database |  |

Table 2: CDW\_UAS\_D\_STUDENT Target Description

**TRANSFORMATIONS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Mapping Logic** | **Transformation Name** | **Function Used** |
| STUD\_ID | System generated based on Student SSN.   Insert for new Student/ expire the old record and   insert the updated record for existing Students | Sequence Generator |  |
| STUD\_F\_NAME | Convert the Name to Title Case | Expression | InitCap() |
| STUD\_M\_NAME | Convert the middle name in lower case | Expression | Lower() |
| STUD\_L\_NAME | Convert the Last Name in Title Case | Expression | InitCap() |
| STUD\_ROLLNO | Direct move ( Abort the session if Roll Number is invalid. Convert to number ) | Expression | IIF() |
| STUD\_STREET | Concatenate Apartment no and Street name of Student's Residence with comma as a separator . | Expression | IIF() |
| STUD\_CITY | Direct Move |  | Concat() |
| STUD\_STATE | Direct Move |  |  |
| STUD\_COUNTRY | Direct move |  |  |
| STUD\_ZIP | Direct move ( Convert to number ) |  |  |
| STUD\_PHONE | Change the format to xxx-xxx-xxxx |  | ToInteger() |
| STUD\_EMAIL | Direct move |  | Substr() |
| CREATED\_DATE | SYSDATE |  |  |
| END\_DATE | Null on insert / SYSDATE - (1Sec) on update. |  | SYSDATE |

Table 3: CDW\_UAS\_D\_STUDENT 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.

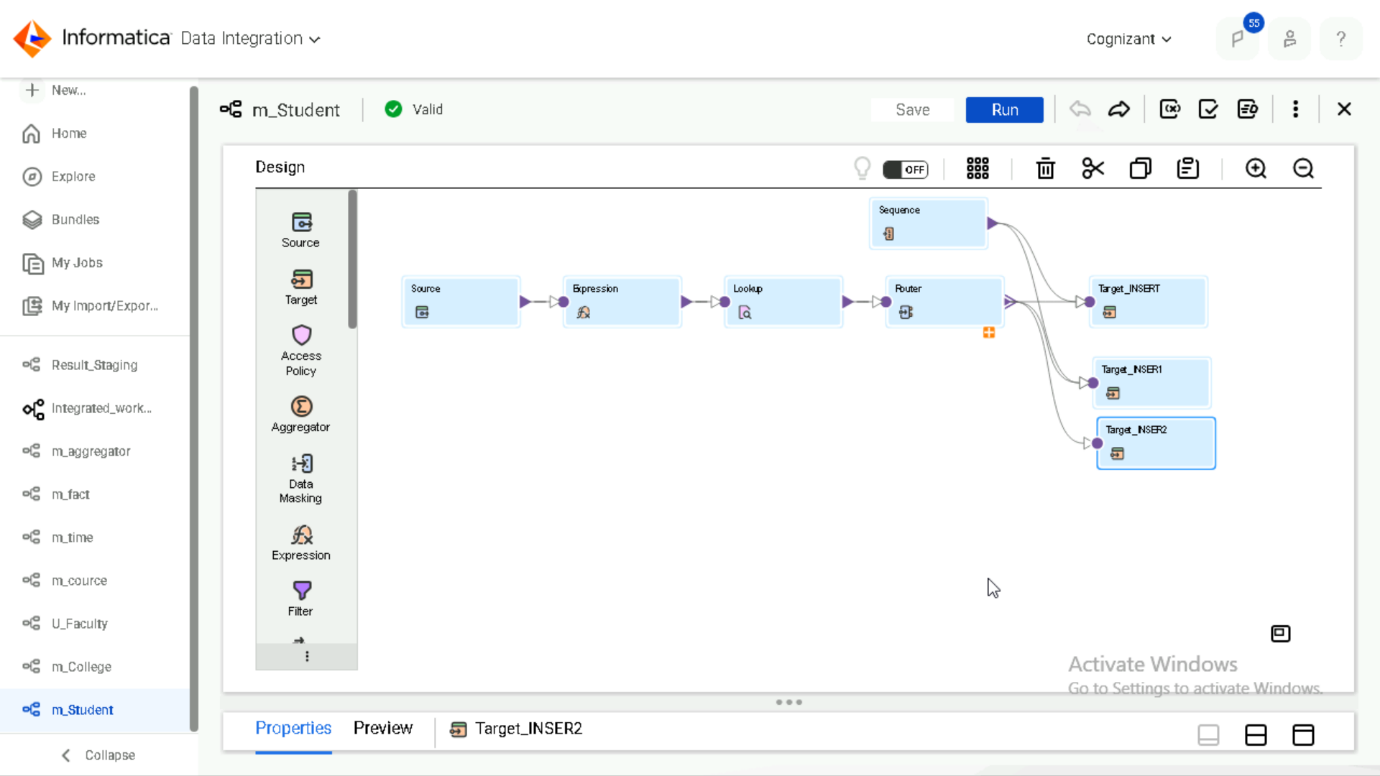


Fig. 4: Student Dimension Mapping

**Mapping Task**

The figure below illustrates the Mapping task that executes the mappings created in the data integration service of IICS. This mapping task ensures that the data is extracted from the source files, transformed according to the specified business logic and loaded into the target table.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 5: Student Dimension Mapping Task

**MONITOR DETAILS:**

The monitor gives the task status, and the number of rows accepted is also verified. The target table STUDENT\_UAS is verified in the SQL server database as shown in the figure.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 6: Student Dimension Monitor

**TARGET POST EXECUTION:**

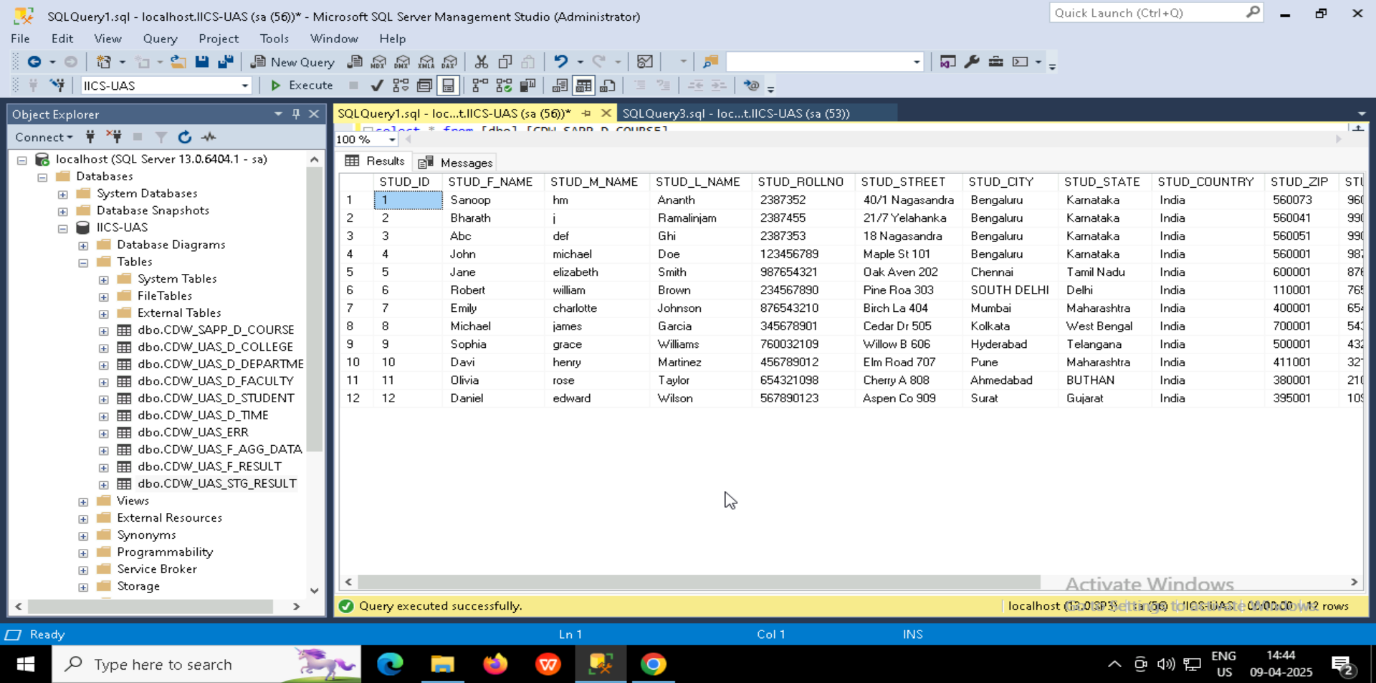


Fig. 7: Student Dimension Target Table

**Functional Requirement 2**

**COLLEGE**

The College dimension table CDW\_UAS\_D\_COLLEGE is a Type – I SCD which contains all college details. Data is added to this table, only when a new college is being opened. So, the load schedule would be on demand. The source file’s name is CDW\_UAS\_COLLEGE.txt and it is pipe (|) delimited. The table is updated/ inserted for old/ new branches respectively and no history is maintained.

**SOURCE**

The source files provided in the requirement document are a text file described in the table containing the following attributes like COLLEGE\_CODE, COLLEGE\_NAME, COLLEGE\_STREET, COLLEGE\_CITY, COLLEGE\_STATE, COLLEGE\_ZIP and COLLEGE\_PHONE. The source is imported to the source analyser as shown in fig .

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| CDW\_UAS\_D\_COLLEGE (flat file) | This is a pipeline separated file. |  |

Table 4: CDW\_UAS\_D\_COLLEGE Source Description

**TARGET**

The target table is created in the SQL server database where the attributes are selected from the provided requirements document and imported to the Informatica mapping data integration service.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET FILE** |
| CDW\_UAS\_D\_COLLEGE | It is an SQL server database table maintaining the required information. | SQL server Database |  |

Table 5: CDW\_UAS\_D\_COLLEGE Target Description

**TRANSFORMATIONS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Mapping Logic** | **Transformation Name** | **Function Used** |
| COLLEGE\_CODE | Insert if new College/update the entire record for existing College |  |  |
| COLLEGE\_NAME | Direct move |  |  |
| COLLEGE\_STREET | Direct move |  |  |
| COLLEGE\_CITY | Direct move |  |  |
| COLLEGE\_STATE | Direct move |  |  |
| COLLEGE\_ZIP | If the source value is null load default value else Direct move |  | IIF() |
| COLLEGE\_PHONE | Change the format of phone number to (XXX)XXX-XXXX |  | SUBSTR()  Concat() |

Table 6: CDW\_UAS\_D\_COLLEGE 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. 8: College Dimension Mapping

**MAPPING TASK**

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

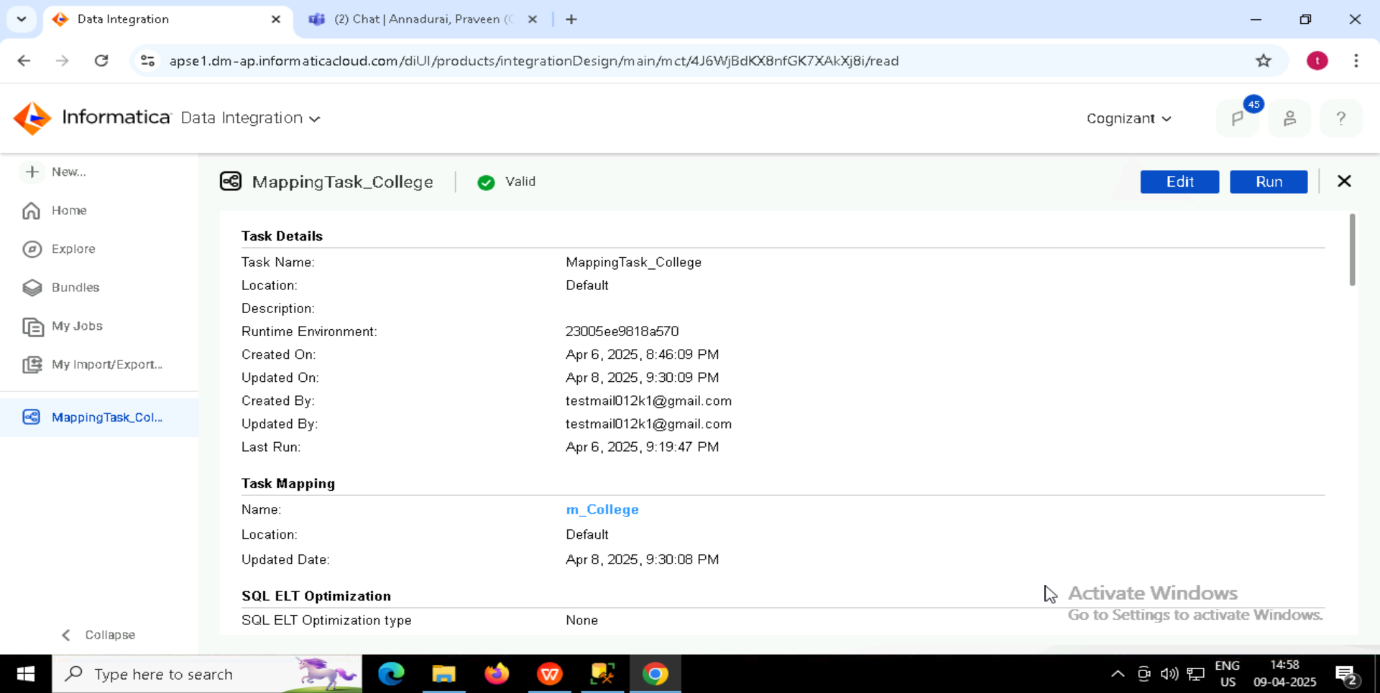


Fig. 9: College Dimension Mapping Task

**MONITOR DETAILS:**

The monitor gives the task status, and the number of rows accepted is also verified. The data sent to the target file CDW\_UAS\_D\_COLLEGE is verified in the SQL server database as shown in the figure.

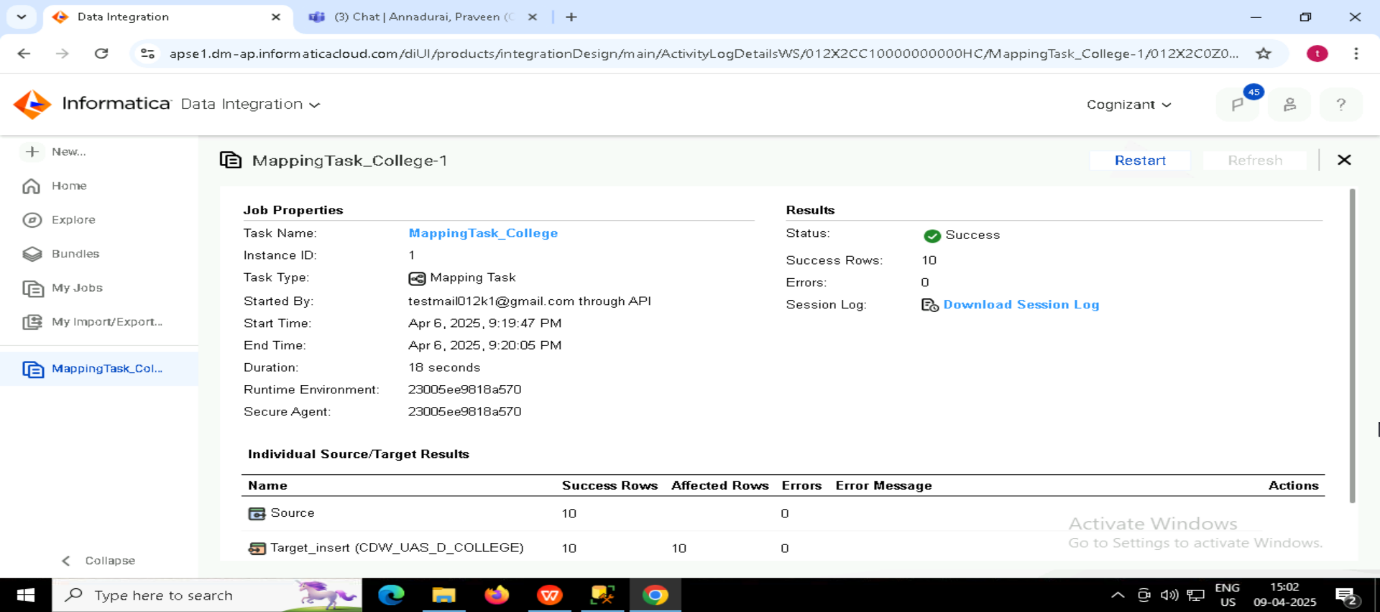


Fig. 10: College Dimension Monitor

**TARGET POST EXECUTION:**

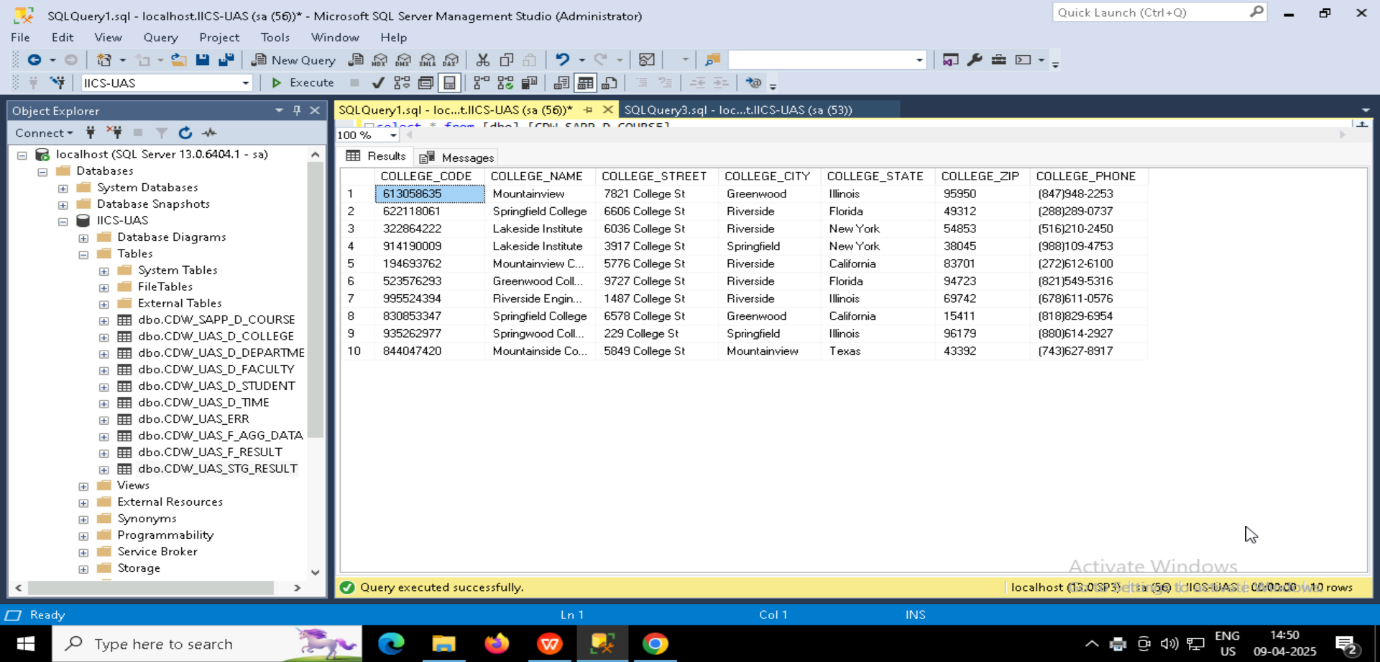


Fig. 11: College Dimension Target Table

**Functional Requirement 3**

**FACULTY**

The faculty dimension table CDW\_UAS\_D\_FACULTY is a Type – I SCD which contains all faculty details. The source file’s name is CDW\_UAS\_FACULTY.txt and it is comma (,) delimited. The table is updated/ inserted for old/ new branches respectively and no history is maintained.

**SOURCE**

As per the requirement, we have been provided with a source comma-separated file. The source file contains field such as a FACULTY\_CODE, FACULTY\_NAME, FACULTY\_DESIGNATION and FACULTY\_QUALIFICATION.

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| CDW\_UAS\_FACULTY.txt (flat file) | This is a comma-separated file. |  |

Table 7: CDW\_UAS\_FACULTY Source Description

**TARGET**

The target table is created in the SQL server database where the attributes are selected from the provided requirements document and imported to the Informatica.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET FILE** |
| CDW\_UAS\_D\_FACULTY | It is a SCD type 1 table | SQL Server Database |  |

Table 8: CDW\_UAS\_D\_FACULTY Target Description

**TRANSFORMATIONS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Mapping Logic** | **Transformation Name** | **Function Used** |
| FACULTY\_CODE | Insert if new College/update the entire record for existing College |  |  |
| FACULTY\_NAME | Direct move |  |  |
| FACULTY\_DESIGNATION | Direct move |  |  |
| FACULTY\_QUALIFICATION | Direct move |  |  |

Table 9: CDW\_UAS\_D\_FACULTY 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. 12: Faculty Dimension Mapping

**MAPPING TASK**

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

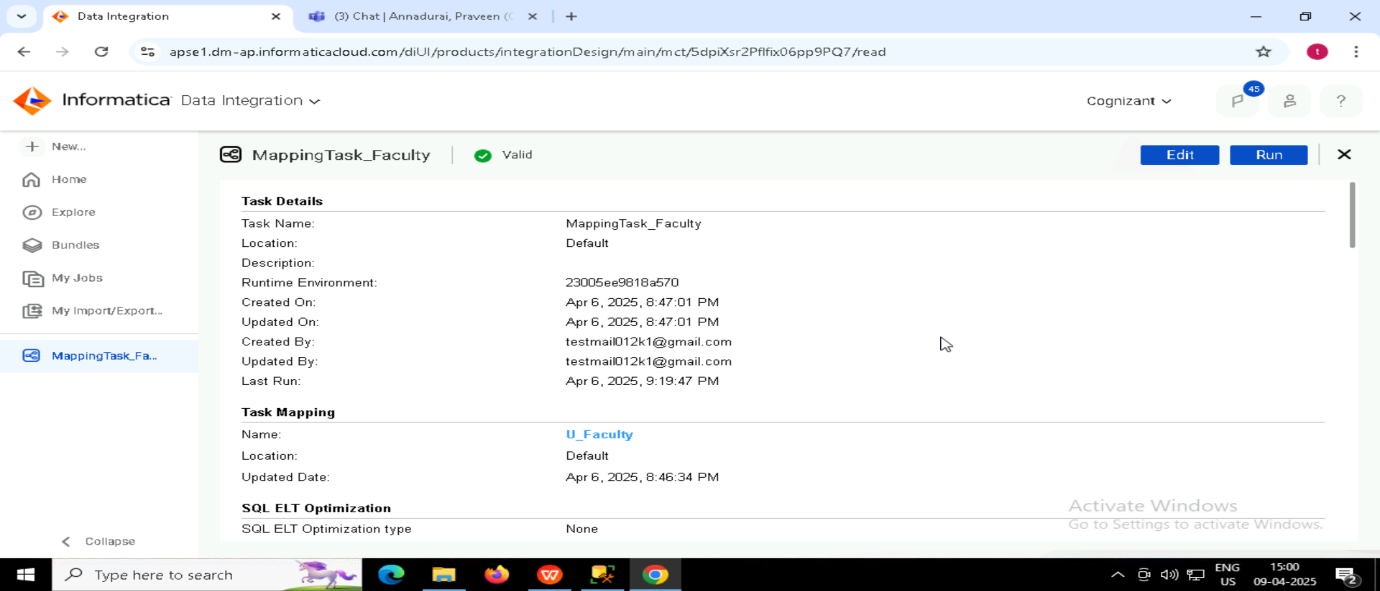


Fig. 13: Faculty Dimension Mapping Task

**MONITOR DETAILS:**

The monitor gives the task status, and the number of rows accepted is also verified. The data IN the target file CDW\_UAS\_D\_FACULTY is verified in the SQL server database as shown in the figure.

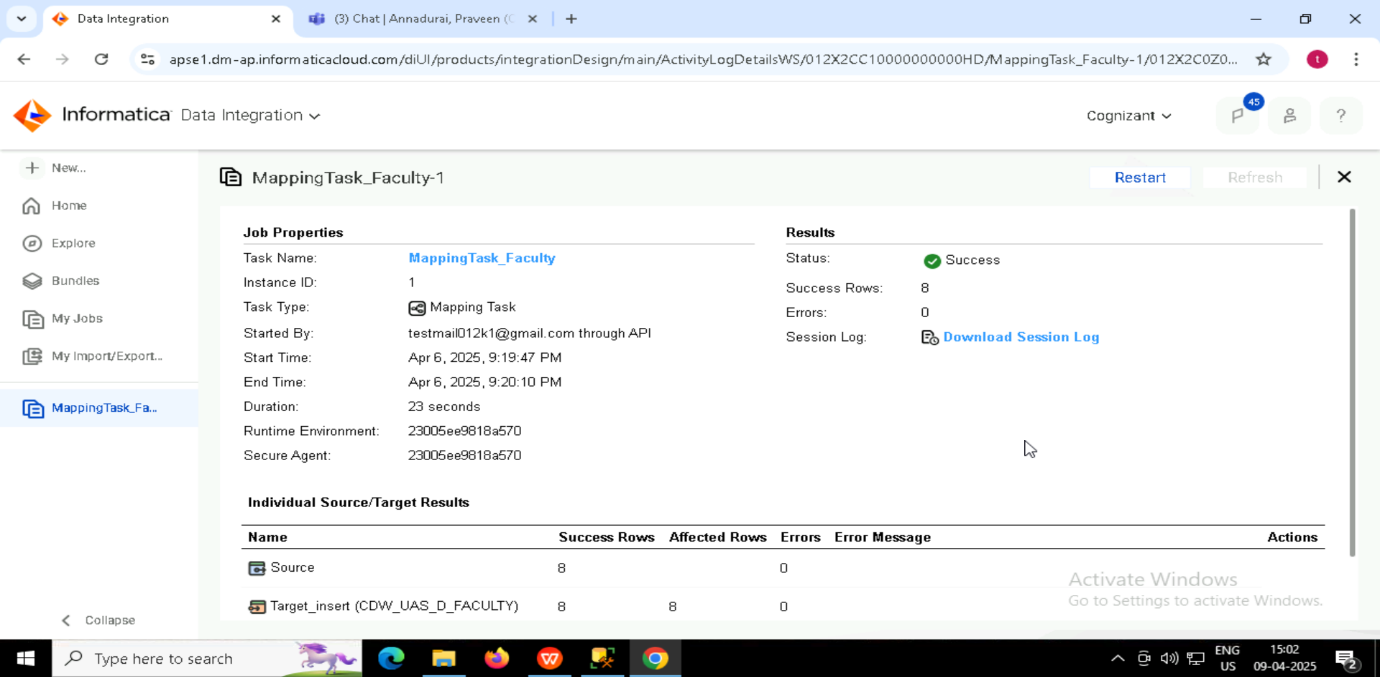


Fig. 14: Faculty Dimension Monitor

**TARGET POST EXECUTION:**

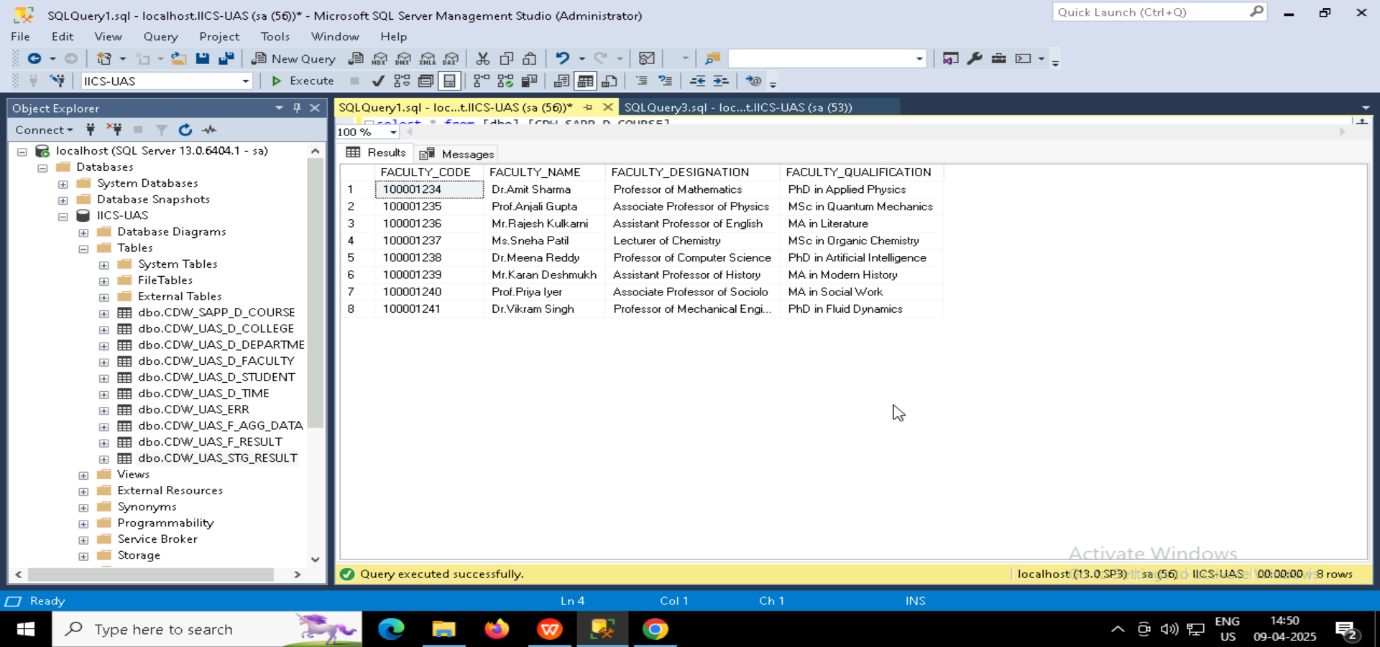


Fig. 15: Faculty Dimension Target Table

**Functional Requirement 4**

**DEPARTMENT**

The Department dimension table CDW\_UAS\_D\_DEPARTMENT is a Type – I SCD table which contains all the departments’ information. The source file’s name is CDW\_UAS\_DEPARTMENT.txt and it is pipe (|) delimited. The table is updated/inserted for old/new suppliers respectively on a daily basis and no history is maintained.

**SOURCE**

As per the requirement, we have been provided with a source comma-separated file. The source file contains field such as a DEPARTMENT\_NAME, DEPARTMENT\_NO, DEPARTMENT\_PHONE and DEPARTMENT\_HOD.

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| CDW\_UAS\_DEPARTMENT.txt | This is an fixed file (pipeline separated) file. |  |

Table 10: CDW\_UAS\_D\_DEPARTMENT source

**TARGET**

The target table is created in the SQL server database where the attributes are selected from the provided requirements document and imported to the Informatica.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET FILE** |
| CDW\_UAS\_D\_DEPARTMENT | It is a SCD type 1 table | SQL Server Database |  |

Table 11: CDW\_UAS\_D\_DEPARTMENT Target Description

**TRANSFORMATIONS:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Mapping Logic** | **Transformation Name** |
| DEPARTMENT\_ID | System generated based on Department No. Insert for new  Department/update the entire record for existing  Departments. |  |
| DEPARTMENT\_NAME | If a percentage symbol is present in the name of Department remove and load else direct move |  |
| DEPARTMENT\_NO | Direct move (Abort the session if Number is invalid) |  |
| DEPARTMENT\_PHONE | Standardize the phone number to XXX-XXX-XXXX | Expression |

Table 12: CDW\_UAS\_D\_DEPARTMENT Transformations

**MAPPING**

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

**A screenshot of a computer

AI-generated content may be incorrect.**

Fig. 16: Department Dimension Mapping

**MAPPING TASK**

The figure below illustrates the mapping task that executes the mappings created in the Informatica. This mapping task ensures that the data is extracted from the source file, transformed according to the specified business logic and loaded into the target files.

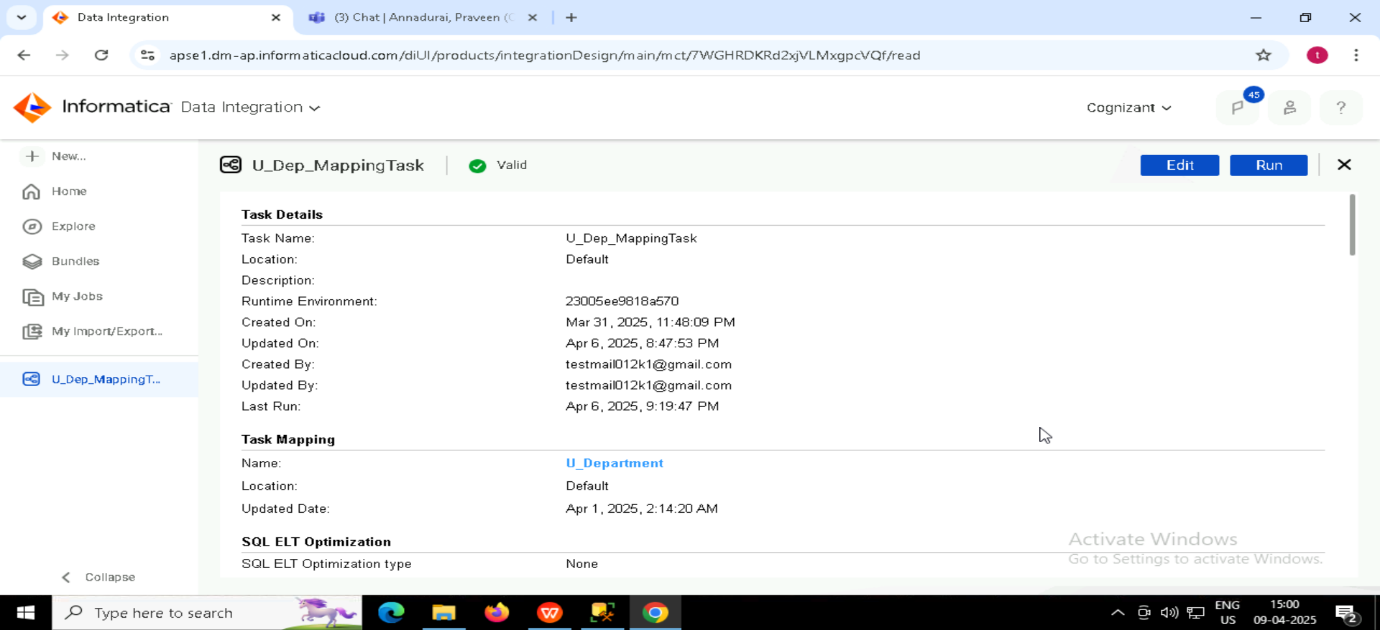


Fig. 17: Department Dimension Mapping Task

**TASK DETAILS:**

TheIICS monitor gives the task status and the number of rows accepted is also verified. The target CDW\_UAS\_DEPARTMENT files are verified by SQL server database as shown in the figure.



Fig. 18: Department Dimension Monitor

**TARGET POST EXECUTION:**

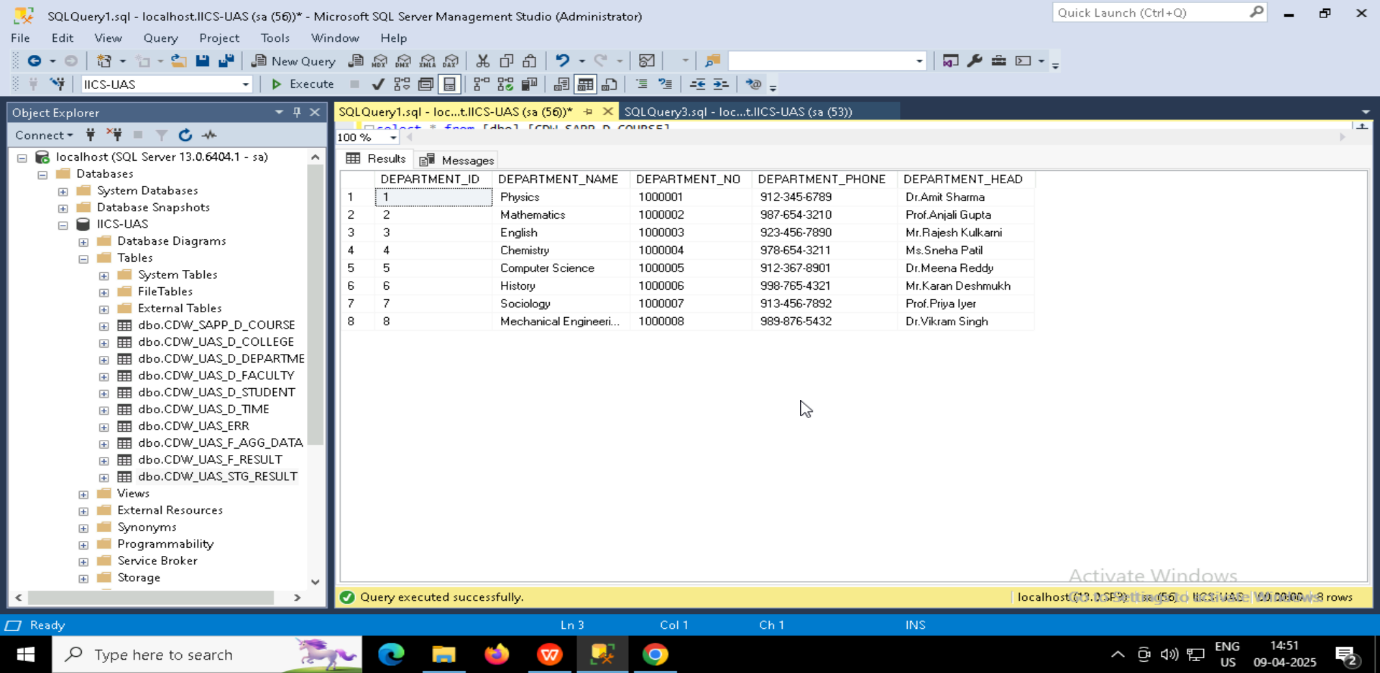


Fig. 19: Department Dimension Target Table

**Functional Requirement 5**

**COURSE**

The course dimension table CDW\_UAS\_D\_COURSE is a Type – I SCD which contains all course information. The source file’s name is CDW\_UAS\_COURSE.txt and it is comma (,) delimited. This table is dependent on the Department table for the Department Id. The department table must contain the respective department information before loading course table. The table is updated/inserted for old/new products respectively on a daily basis and no history is maintained.

**SOURCE**

As per the requirement, we have been provided with a source comma-separated file. The source file contains field such as a COURSE\_CODE, COURSE\_NAME, DEPARTMENT\_NO and MARK\_REQ. Since, the CDW\_UAS\_D\_COURSE table has the field DEPARTMENT\_NO which is dependent on the CDW\_UAS\_D\_DEPARTMENT table as source.

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| CDW\_UAS\_COURSE.txt (flat file) | This is a comma-separated file. |  |
| CDW\_UAS\_D\_DEPARTMENT  (SQL server) | This is a SQL server database. |  |

Table 13: CDW\_UAS\_COURSE Source Description

**TARGET**

The target table is created in the SQL server database where the attributes are selected from the provided requirements document and imported to the Informatica.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET FILE** |
| CDW\_UAS\_D\_COURSE | It is a SCD type 1 table | SQL Server Database |  |

Table 14: CDW\_UAS\_D\_COURSE Target Description

**TRANSFORMATIONS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Mapping Logic** | **Transformation Name** | **Function Used** |
| COURSE\_CODE | Insert for new Course/update the entire record for existing Course( Abort the session if Code is invalid ) | Expression | To\_Decimal( )  Abort( ) |
| COURSE\_NAME | Trim the trailing spaces and load the data | Expression | Ltrim( )  Rtrim( ) |
| DEPARTMENT\_ID | Join with Department table, based on No from Department Table  and load the corresponding Department ID | Lookup |  |
| MARK\_REQ | Direct move |  |  |

Table 15: CDW\_UAS\_D\_COURSE 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: Course Dimension Mapping

**MAPPING TASK**

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

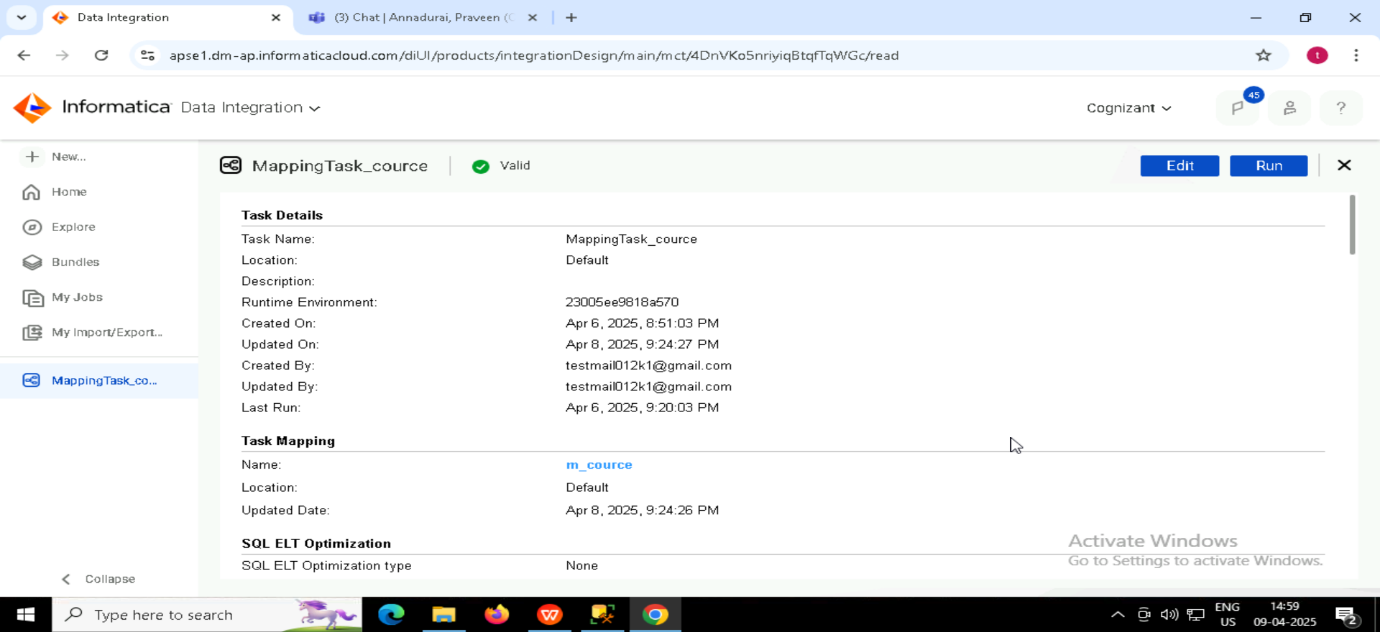


Fig. 21: Course Dimension Mapping Task

**MONITOR DETAILS:**

The monitor gives the task status, and the number of rows accepted is also verified. The data IN the target file CDW\_UAS\_D\_COURSE is verified in the SQL server database as shown in the figure.

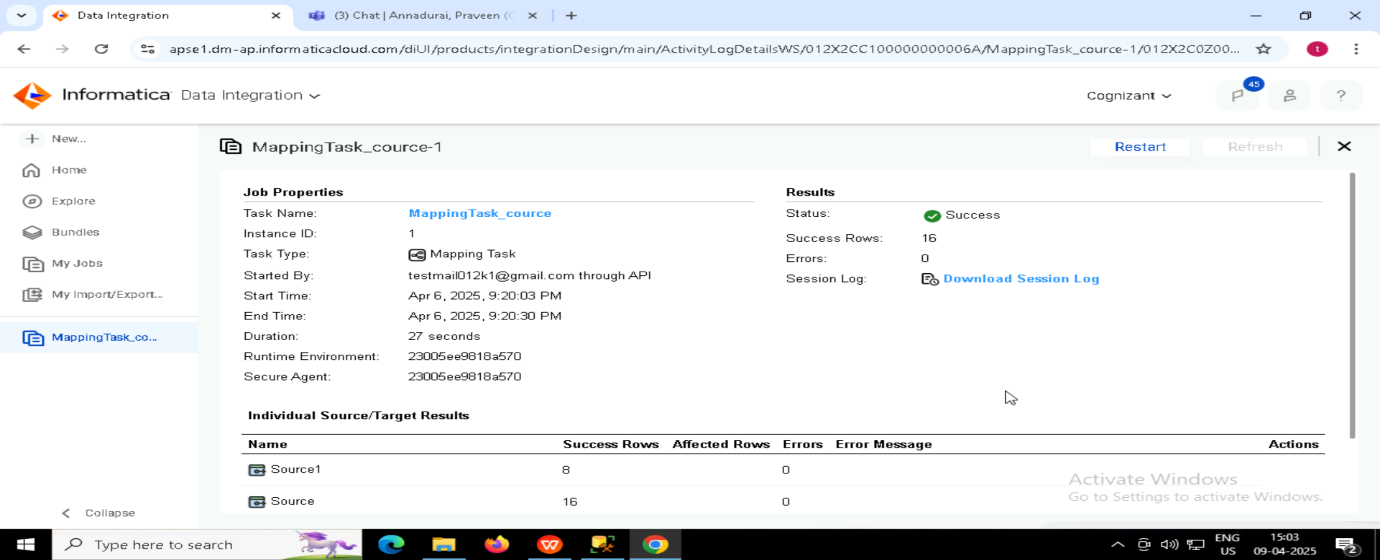


Fig. 22: Course Dimension Monitor

**TARGET POST EXECUTION:**

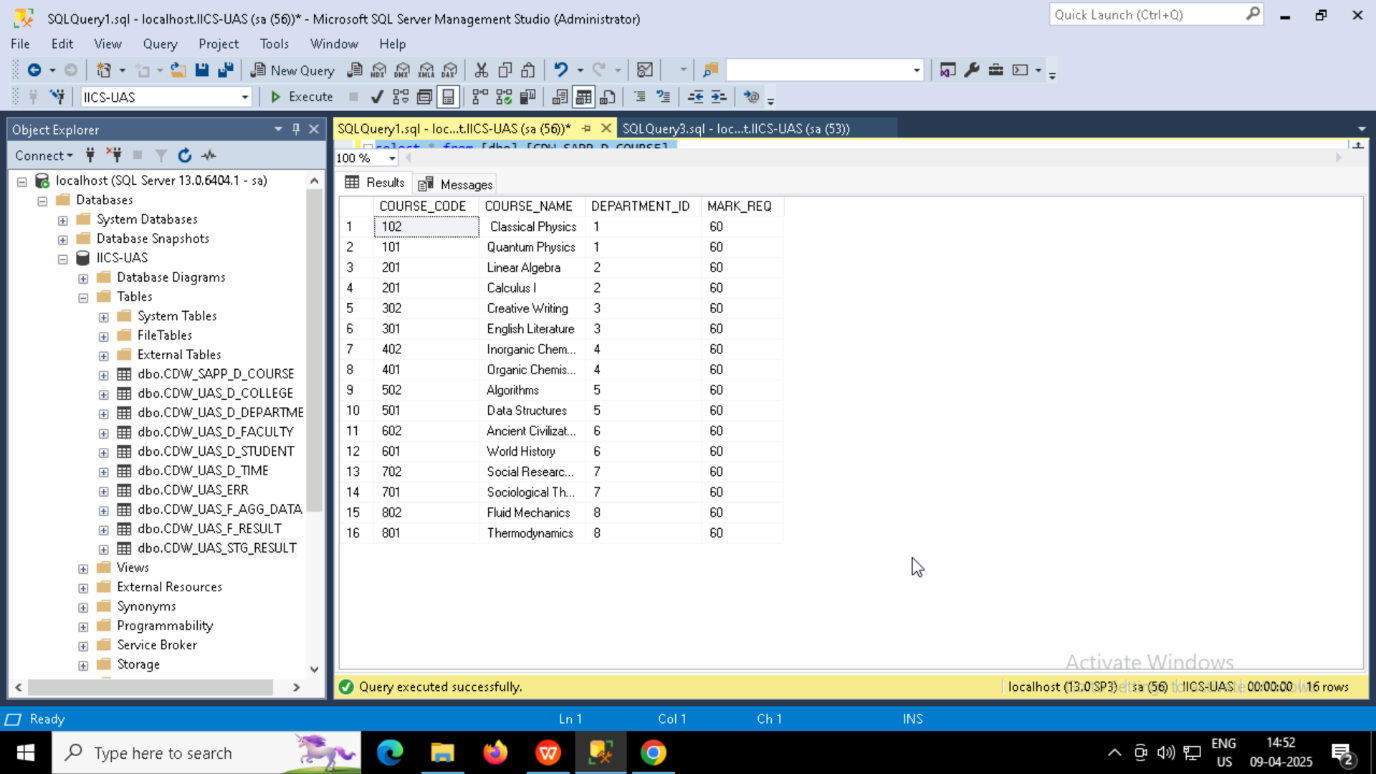


Fig. 23: Course Dimension Target Table

**Functional Requirement 6**

**TIME**

The Time dimension table ***SRC\_CALENDER\_FILE*** is a static table which contains the time key and other relevant information. This table is loaded once in a year from the source file ***SRC\_CALENDER\_FILE.txt*** which is fixed width file.

**SOURCE**

As per the requirement, we have been provided with a source comma-separated file. The source file contains field TIME\_KEY.

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| SRC\_CALENDER\_FILE.txt (fixed width file) | This is a fixed width file |  |

Table 16: CDW\_UAS\_TIME Source Description

**TARGET**

The target table is created in the SQL server database where the attributes are selected from the provided requirements document and imported to the Informatica.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET FILE** |
| CDW\_UAS\_D\_TIME | Static Dimension | SQL Server Database |  |

Table 17: CDW\_UAS\_D\_TIME Target Description

**TRANSFORMATIONS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **MAPPING LOGIC** | **TRANSFORMATION**  **NAME** | **FUNCTION USED** |
| DAY | Substring from TIMEKEY | Expression | SUBSTR( )  TO\_DECIMAL( ) |
| MONTH | Substring from TIMEKEY | Expression | SUBSTR( )  TO\_DECIMAL( ) |
| YEAR | Substring from TIMEKEY | Expression | SUBSTR( )  TO\_DECIMAL( ) |
| QUARTER | Substring from TIMEKEY | Expression | SUBSTR( ) |

Table 18: CDW\_UAS\_D\_TIME 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. 24: Time Dimension Mapping

**MAPPING TASK**

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

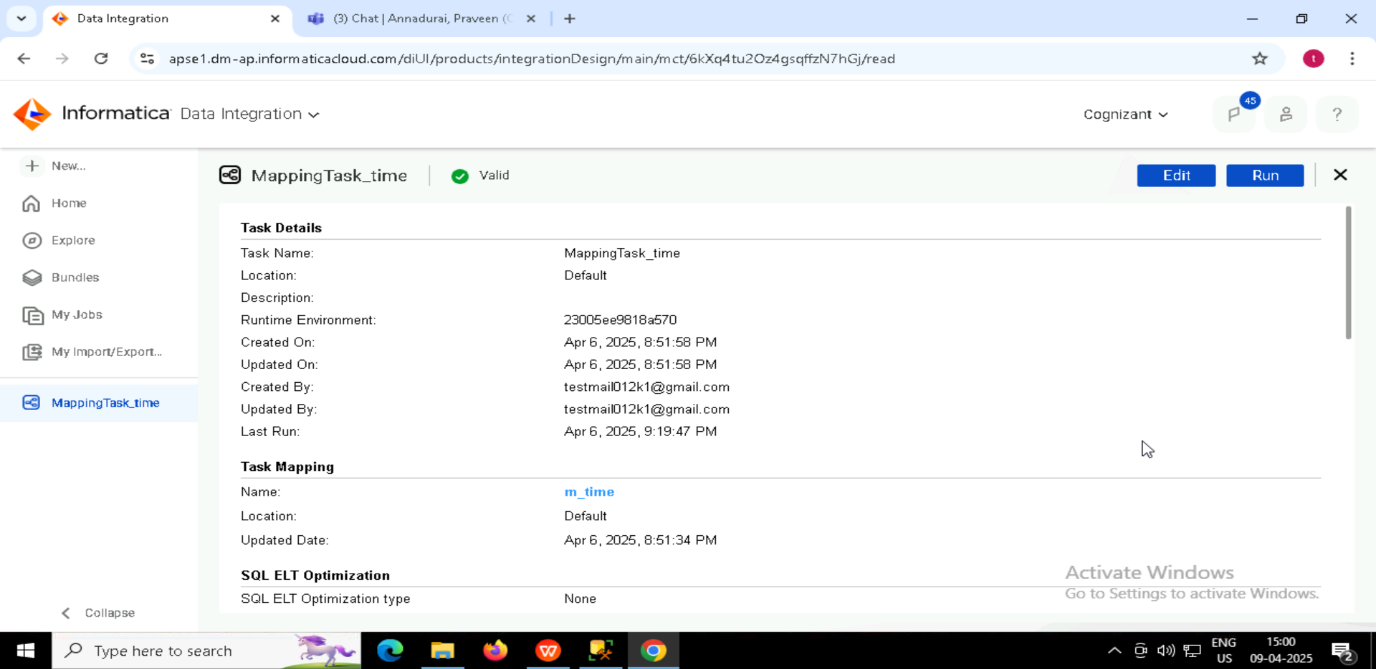


Fig. 25: Time Dimension Mapping Task

**MONITOR DETAILS:**

The monitor gives the task status, and the number of rows accepted is also verified. The data IN the target file CDW\_UAS\_D\_TIME is verified in the SQL server database as shown in the figure.

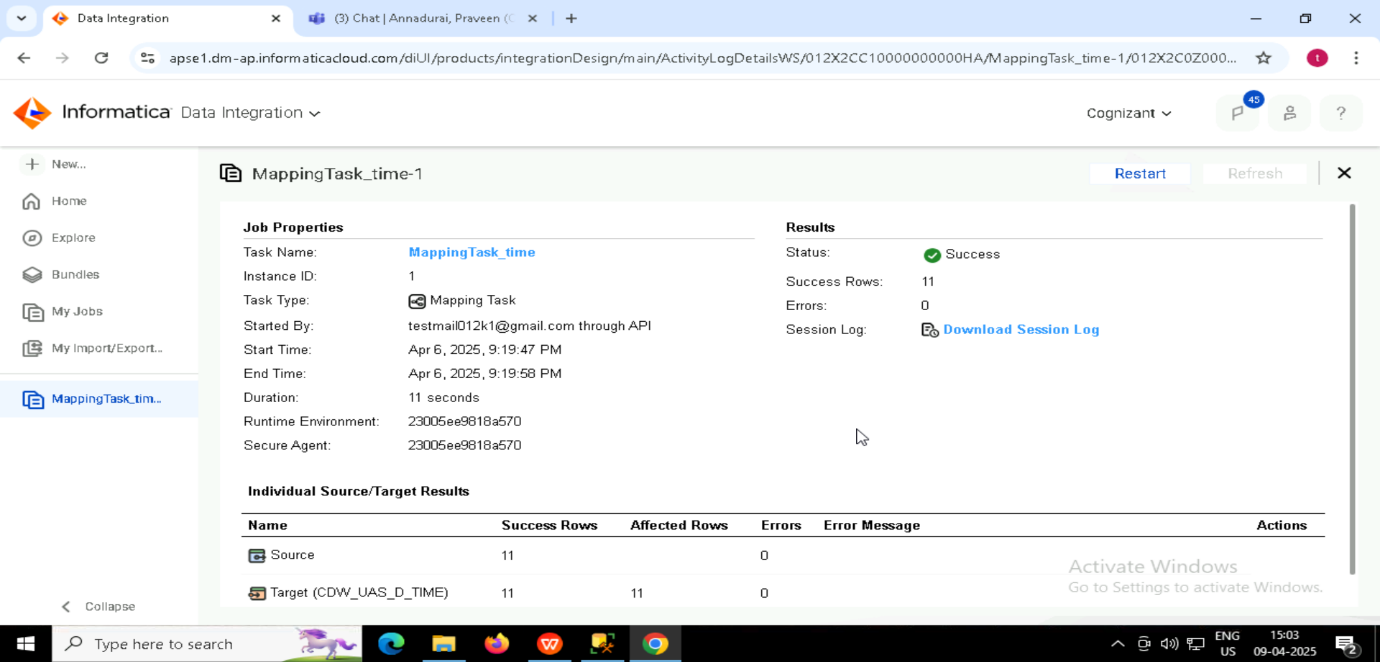


Fig. 26: Time Dimension Monitor

**TARGET POST EXECUTION:**

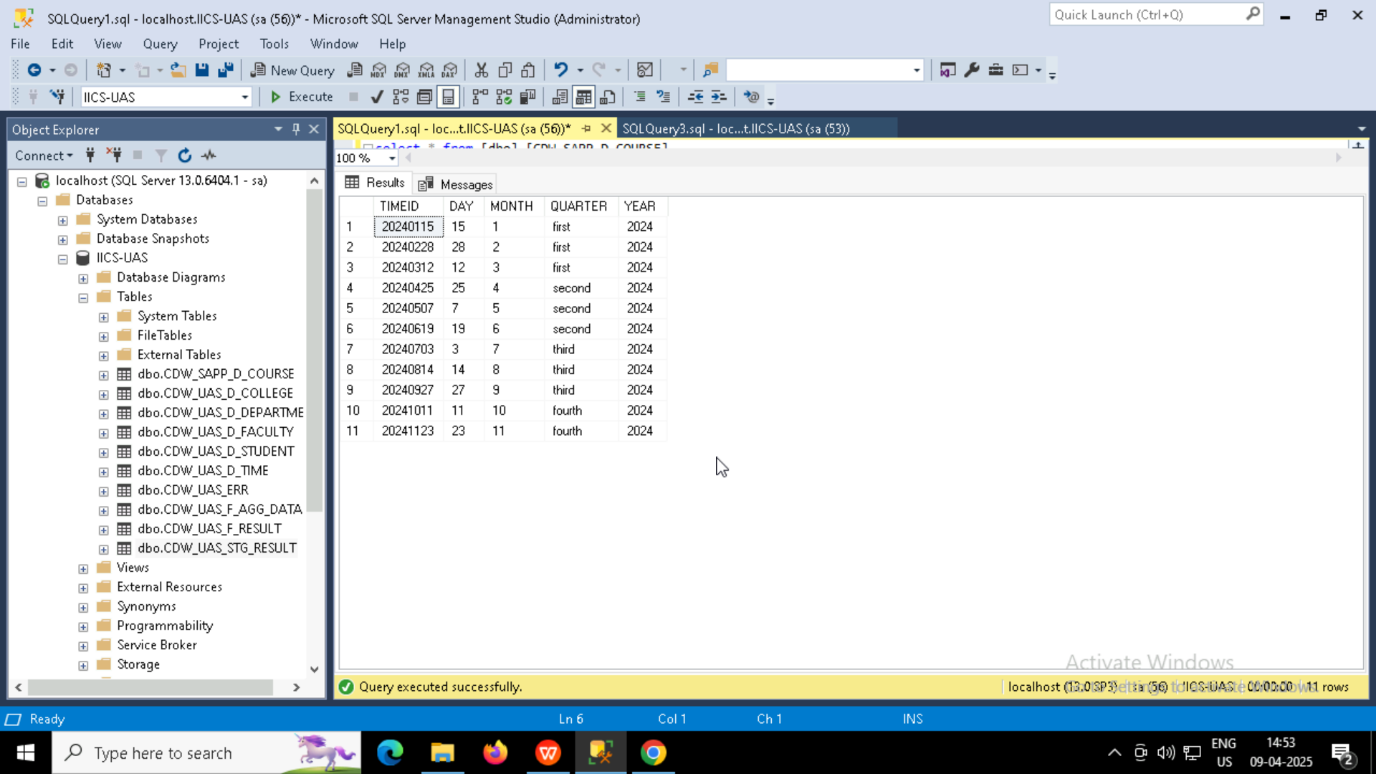


Fig. 27: Time Dimension Target Table

**Functional Requirement 7**

**RESULT STAGING**

The results staging table CDW\_UAS\_D\_STG\_RESULT receives the results data from the different evaluation centers (Two delimited & one fixed width) CDW\_UAS\_F\_RESULTS\_BR\_XX.txt and the delimiter used is comma. The information from different files is standardized in the staging table. This table is refreshed every day. The data is loaded with all related dimension information from the respective dimension tables in the warehouse. Records containing invalid information will be sent to the CDW\_UAS\_ERR error table. The two delimited files are joined before loading in to the staging area.

**SOURCE**

As per the requirement, we have been provided with a source comma-separated file. The source file contains field TIMEKEY, Student\_Roll, DEPARTMENT\_No, COLLEGE\_NAME, COLLEGE\_NAME, COURSE\_NAME, COURSE\_NAME, FACULTY\_NAME, RESULT\_MARKS and RESULT\_MARKS.

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| CDW\_UAS\_D\_RESULT\_CL\_01.txt | This is a flat file delimited with comma. |  |
| CDW\_UAS\_D\_RESULT\_CL\_02.txt | This is a flat file delimited with comma. |  |
| CDW\_UAS\_F\_RESULT\_CL\_01.txt | This is a fixed width file |  |

Table 19: Result staging Source Description

**TARGET**

The target table is created in the SQL server database where the attributes are selected from the provided requirements document and imported to the Informatica.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET FILE** |
| CDW\_UAS\_D\_ STG\_RESULT | It is a Staging table | SQL Server Database |  |
| CDW\_UAS\_ERR | It is a table containing errors while staging | SQL  Server  Database |  |

Table 20: Result staging Target Description

**TRANSFORMATIONS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **MAPPING LOGIC** | **TRANSFORMATION**  **NAME** | **FUNCTION USED** |
| CDW\_P\_RESULT\_DSET\_KEY | Surrogate key of target table using sequence generator | Sequence generator | NEXTVAL( ) |
| RESULT\_F\_PERIOD\_KEY\_D | Lookup upon the target CDW\_UAS\_D\_TIME using TIMEKEY | Lookup |  |
| RESULT\_F\_PERIOD\_KEY\_M | Lookup upon the target CDW\_UAS\_D\_TIME using TIMEKEY | Lookup |  |
| RESULT\_F\_PERIOD\_KEY\_Y | Lookup upon the target CDW\_UAS\_D\_TIME using TIMEKEY | Lookup |  |
| RESULT\_F\_Student\_KEY | Surrogate key of the Student dim table | Lookup |  |
| RESULT\_F\_DEPARTMENT\_KEY | Surrogate key of the Department dim table | Lookup |  |
| RESULT\_F\_COLLEGE\_CODE | Look up College Code using College name from the College Dimension Table | Lookup |  |
| RESULT\_F\_COLLEGE\_NAME | Direct move |  |  |
| RESULT\_F\_COURSE\_CODE | Look up Course Code using Course Name from the Course Dimension Table | Lookup |  |
| RESULT\_F\_COURSE\_NAME | Direct move |  |  |
| RESULT\_F\_FACULTY\_NO | Look up Faculty Code using Faculty Name from the Faculty Dimension Table | Lookup |  |
| RESULT\_MARKS | Direct move from the file/check for decimal values | Source transformation |  |
| RESULT\_GRADE | Need to be calculated from the Course table data (If the Result Marks is greater than the Marks Required then PASS else FAIL). | Expression | IIF( ) |
| CREATED\_DATE | SYSDATE | Expression |  |
| CREATED\_BY | Workflow name | Expression | '$CurrentTaskName' |
| ERROR\_CODE | Generate the Error code based on the error. | Expression | IIF( ) |
| ERR\_MSG | Decode the Error code and set a relavent message | Expression | IIF( ) |
| ERR\_COLUMN\_NM | Load the name of the column in which has error | Expression | IIF( ) |

Table 21: Result staging 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. 28: Result Staging Mapping

**MAPPING TASK**

The figure below illustrates the mapping task that executes the mappings created in the Informatica. This mapping task ensures that the data is extracted from the source files, transformed according to the specified business logic and loaded into the target table and errors occurred while joining to error table.

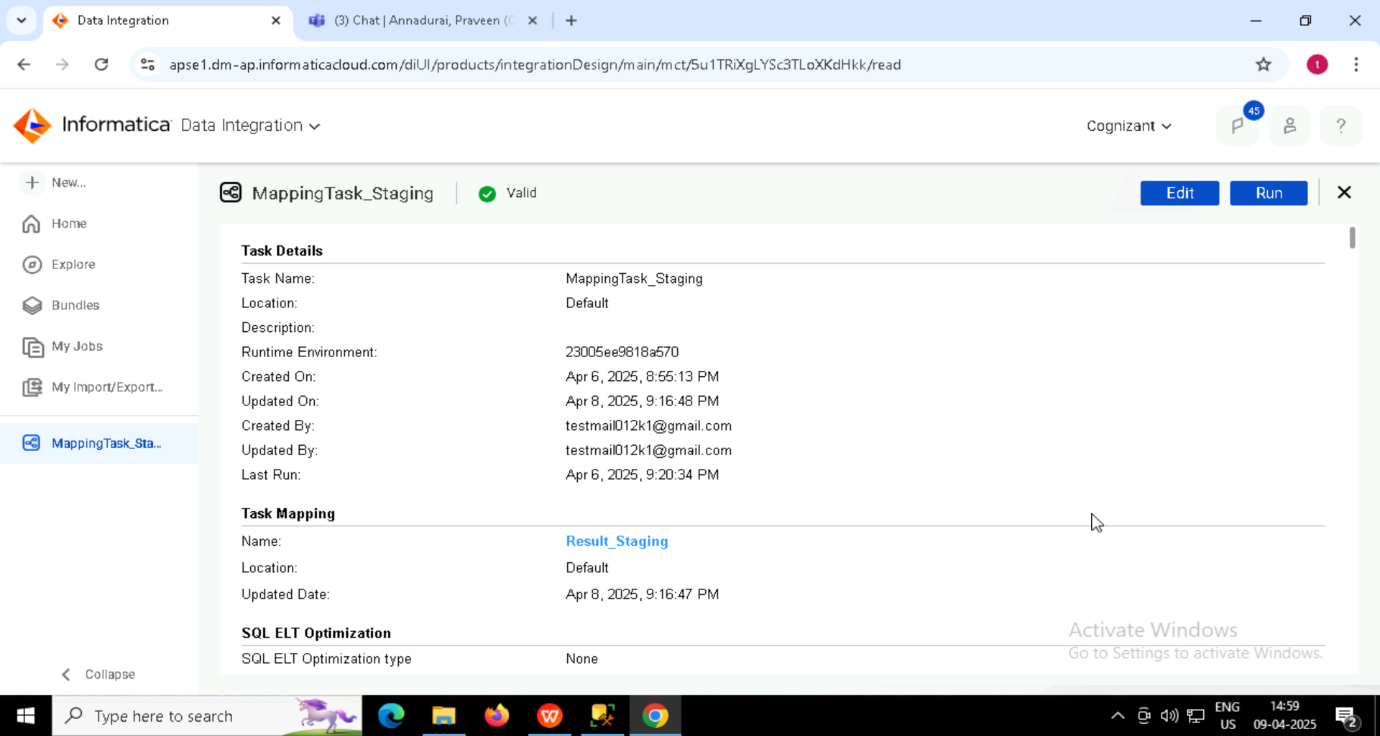


Fig. 29: Result Staging Mapping Task

**MONITOR DETAILS:**

The monitor gives the task status, and the number of rows accepted is also verified. The data IN the target file CDW\_UAS\_STG\_RESULT is verified in the SQL server database and error file to CDW\_UAS\_ERR is verified in SQL server database as shown in the figure.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 30: Result Staging Monitor

**TARGET POST EXECUTION:**

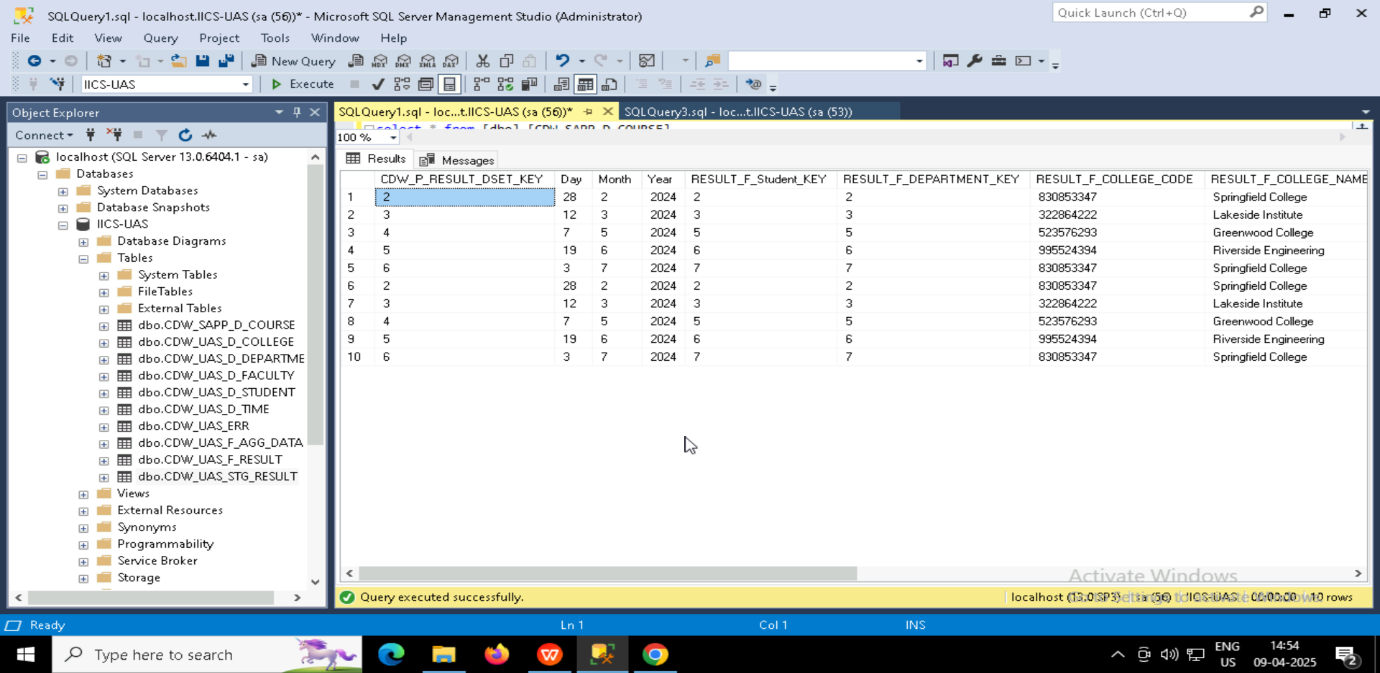


Fig. 31: Result Staging Target Table

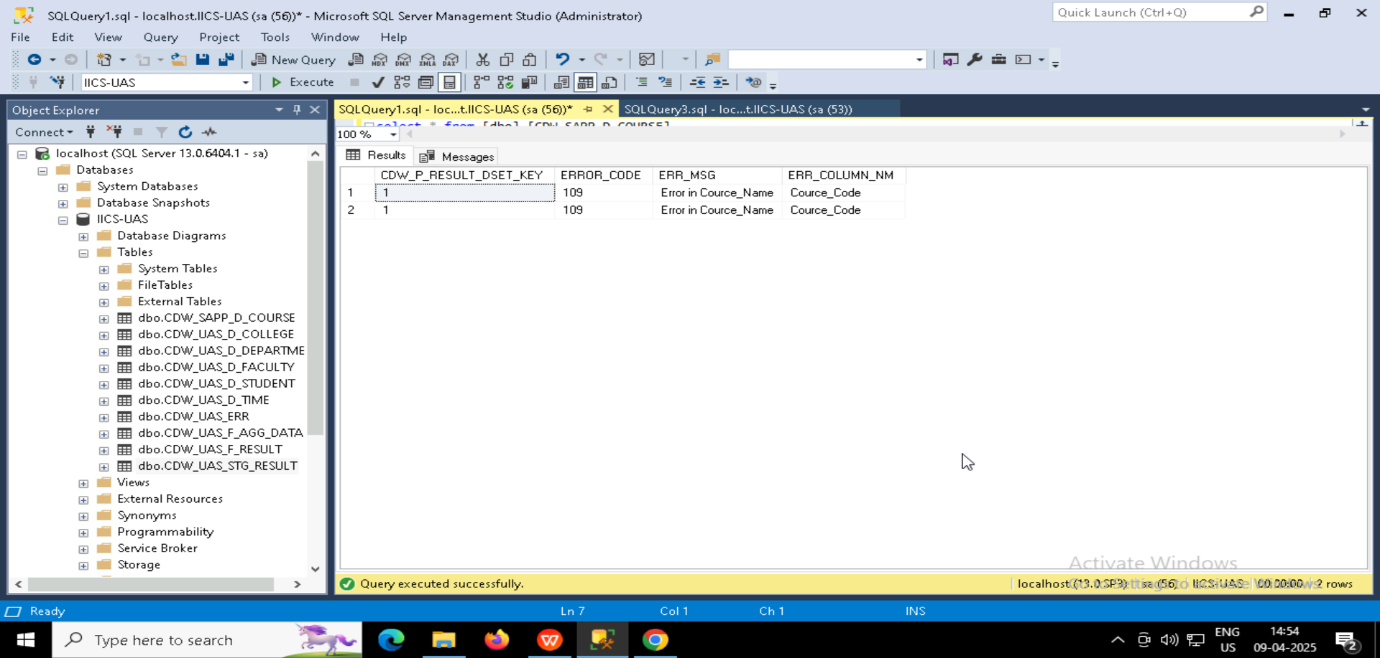


Fig. 32: Staging Error Target Table

**Functional Requirement 8**

**RESULT FACT**

The results fact table CDW\_UAS\_F\_RESULT contains all the transactions loaded from the staging table CDW\_UAS\_STG\_RESULTS. This table is loaded only after the staging table has been loaded completely from the transaction files arriving from all the different centers.

**SOURCE**

As per the requirement, we have been provided with a source database file. The source file contains field CDW\_P\_RESULT\_DSET\_KEY, RESULT\_F\_PERIOD\_KEY, RESULT\_F\_STUDENT\_KEY, etc.

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| CDW\_UAS\_STG\_RESULTS (database) | This is a SQL server database |  |

Table 22: Result Fact Source Description

**TARGET**

The target table is created in the SQL server database where the attributes are selected from the provided requirements document and imported to the Informatica.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET FILE** |
| CDW\_UAS\_F\_RESULT | It is a Fact Table | SQL Server Database |  |

Table 23: Result Fact Target Description

**TRANSFORMATIONS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **MAPPING LOGIC** | **TRANSFORMATION**  **NAME** | **FUNCTION USED** |
| CDW\_XYZ\_F\_RESULT\_  DSET\_KEY | Direct Move |  |  |
| RESULT\_F\_PERIOD\_KEY | Direct Move (Convert to date datatype) | Expression | To\_date( )  To\_char( ) |
| RESULT\_F\_STUDENT\_KEY | Direct Move |  |  |
| RESULT\_F\_DEPARTMENT  \_KEY | Direct Move |  |  |
| RESULT\_F\_COLLEGE\_CODE | Direct Move |  |  |
| RESULT\_F\_COLLEGE\_NAME | Direct Move |  |  |
| RESULT\_F\_COURSE\_CODE | Direct Move |  |  |
| RESULT\_F\_COURSE\_NAME | Direct Move |  |  |
| RESULT\_MARKS | Direct Move |  |  |
| RESULT\_GRADE | Direct Move |  |  |
| CREATED\_DATE | Direct Move |  |  |
| CREATED\_BY | Direct Move |  |  |

Table 24: Result 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.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 33: Result Fact Mapping

**MAPPING TASK**

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

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 34: Result Fact Mapping Task

**MONITOR DETAILS:**

The monitor gives the task status, and the number of rows accepted is also verified. The data IN the target file CDW\_UAS\_F\_RESULT is verified in the SQL server database as shown in the figure.

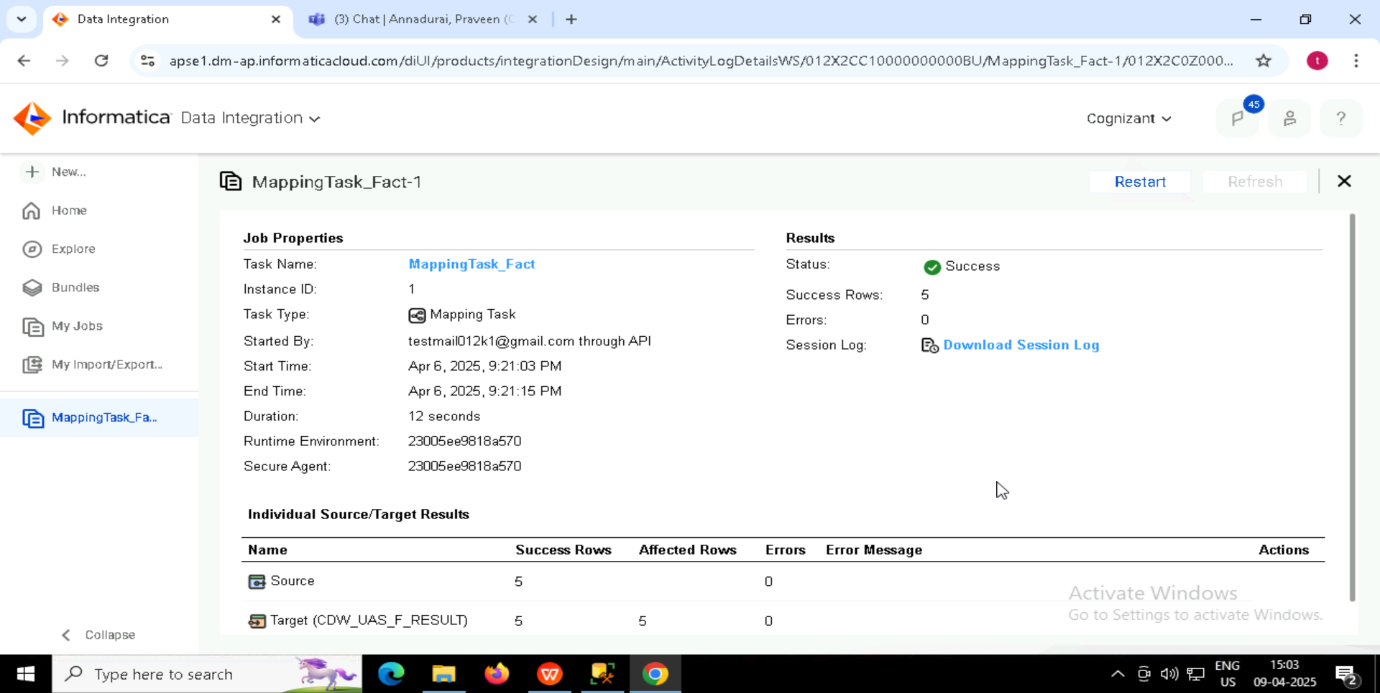


Fig. 35: Result Fact Monitor

**TARGET POST EXECUTION:**

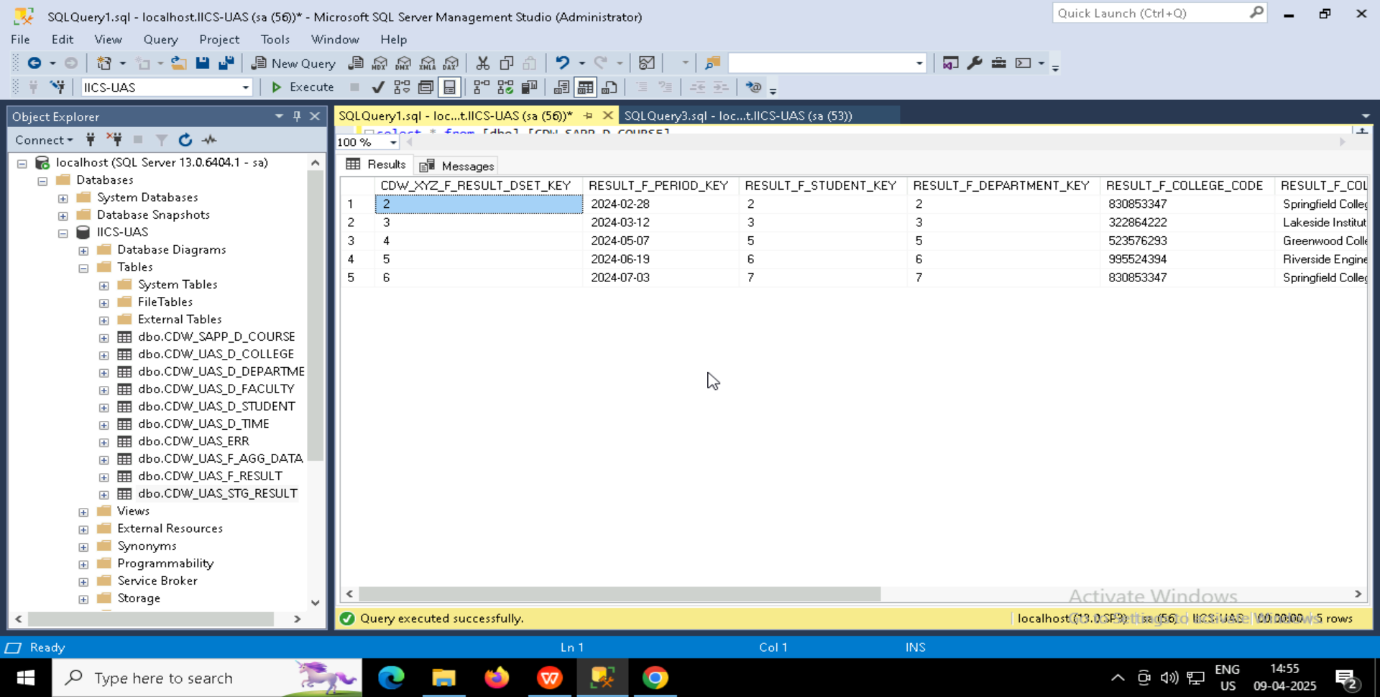


Fig. 36: Result Fact Target Table

**Functional Requirement 9**

**RESULT AGGREGATE**

The Sales Aggregate table CDW\_UAS\_F\_AGG\_DATA pulls data from the Results fact table CDW\_UAS\_F\_RESULTS. The table is aggregated by college\_code & course\_code for the current month.

**SOURCE**

As per the requirement, we have been provided with a source database file. The source file contains field RESULT\_F\_COLLEGE\_CODE, RESULT\_F\_COLLEGE\_NAME, RESULT\_F\_COURSE\_CODE, RESULT\_F\_COURSE\_NAME, RESULT\_GRADE and RESULT\_MARKS.

|  |  |  |
| --- | --- | --- |
| **SOURCE FILE NAME** | **DESCRIPTION** | **SOURCE FILE** |
| CDW\_UAS\_F\_RESULT (database) | This is a SQL server database |  |

Table 25: Result Aggregate Source Description

**TARGET**

The target table is created in the SQL server database where the attributes are selected from the provided requirements document and imported to the Informatica.

|  |  |  |  |
| --- | --- | --- | --- |
| **TARGET NAME** | **DESCRIPTION** | **TARGET TYPE** | **TARGET FILE** |
| CDW\_UAS\_F\_AGG\_DATA | It is an aggregate table | SQL Server Database |  |

Table 26: Result Aggregate Target Description

**TRANSFORMATIONS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **MAPPING LOGIC** | **TRANSFORMATION**  **NAME** | **FUNCTION USED** |
| CDW\_RESULT\_AGG\_  DSET\_KEY | Surrogate key generation | Sequence generator |  |
| COLLEGE\_CODE | College code for the current reporting period sold in each College | Aggregator | GROUP BY |
| COLLEGE\_NAME | College name for the corresponding College | Aggregator | GROUP BY |
| COURSE\_CODE | Course code for the current reporting period evaluated in each College | Aggregator | GROUP BY |
| COURSE\_NAME | Course name for the corresponding Course code | Aggregator | GROUP BY |
| TOTAL PASS | Count of the total students passes in a particular Course evaluated in each College for the current reporting period. | Aggregator | Count( )  IIF( ) |
| MAXIMUM\_MARK | Maximum mark scored in a particular Course in each college for the current reporting period. | Aggregator | Max( ) |
| CREATED\_DATE | SYSDATE | Aggregator | SYSDATE |

Table 27: Result 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. 37: Result Aggregate Mapping

**MAPPING TASK**

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

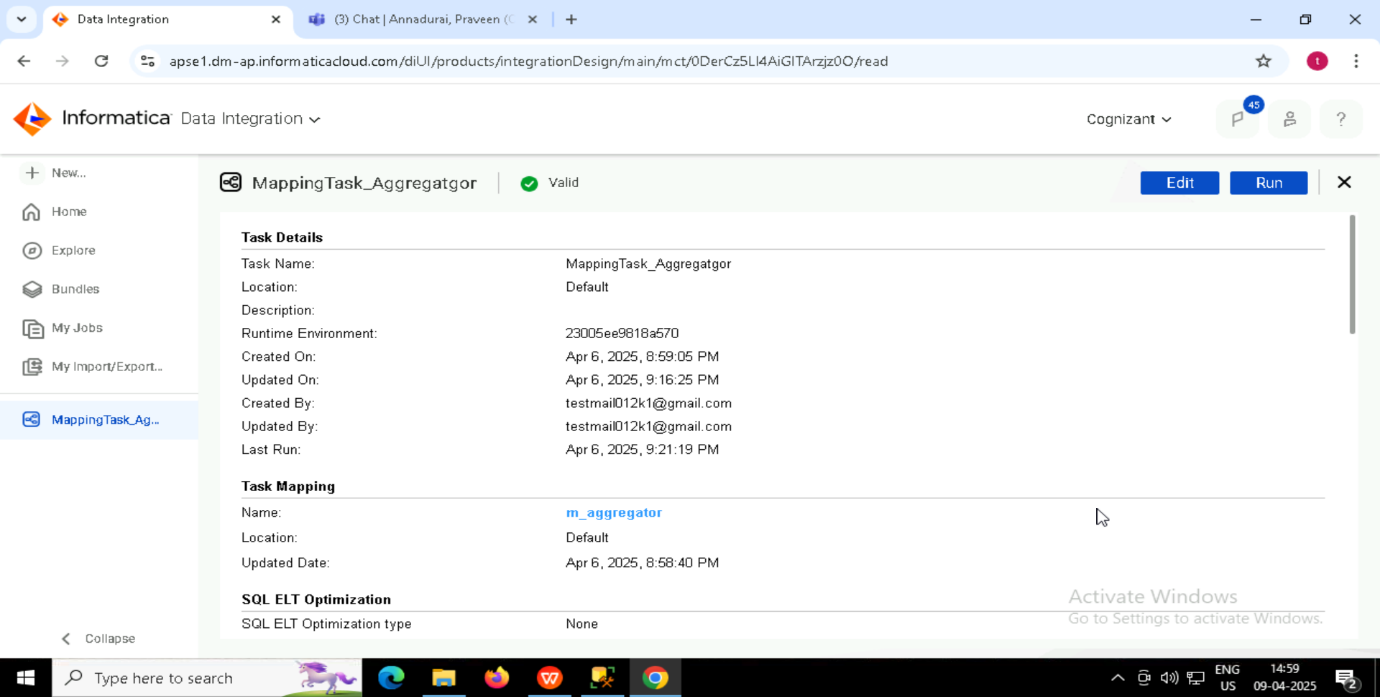


Fig. 38: Result Aggregate Mapping Task

**MONITOR DETAILS:**

The monitor gives the task status, and the number of rows accepted is also verified. The data IN the target file CDW\_UAS\_F\_AGG\_DATA is verified in the SQL server database as shown in the figure.

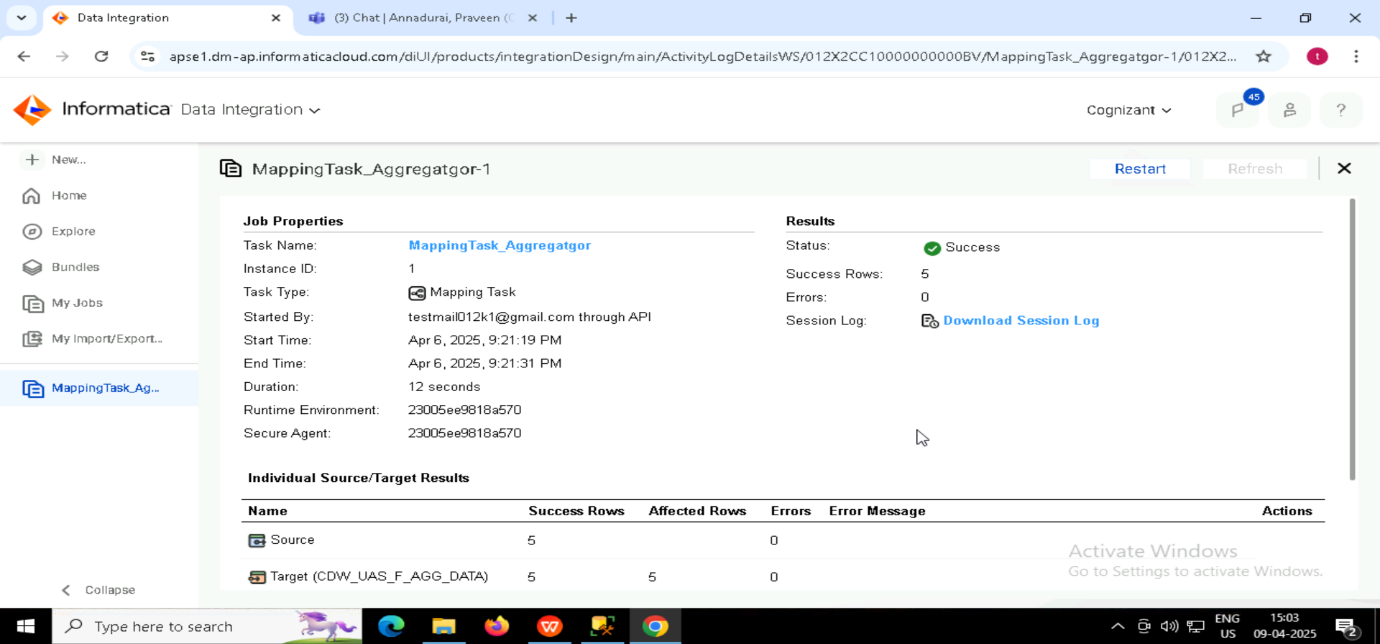


Fig. 39: Result Aggregate Monitor

**TARGET POST EXECUTION:**

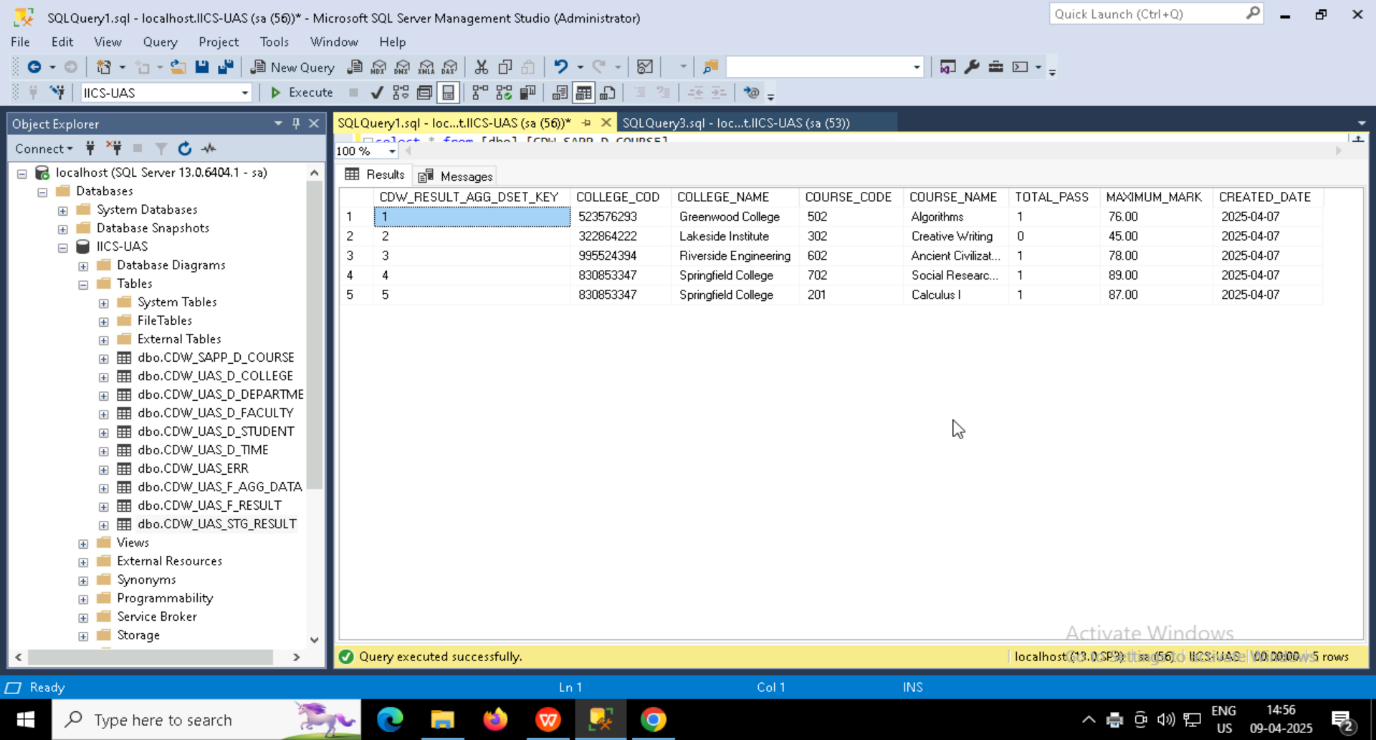


Fig. 40: Result Aggregate Target Table

**Integrated Taskflow**

The project incorporates a comprehensive and streamlined task flow in Informatica Intelligent Cloud Services (ICS) to unify and automate the execution of all mapping tasks. Each dimension Student, College, Department, Course, Time, and Faculty-is managed through its dedicated mapping task, ensuring modularity and data accuracy. These tasks are cohesively integrated into a single task flow, facilitating seamless orchestration of the entire ETI pipeline.  
The staging results table acts as an intermediary, consolidating data from all dimensions to ensure a unified view. This staging layer serves as the foundation for the Result Fact Table, which further feeds into the Result Aggregation Table, enabling multi-level data analysis. The task flow is equipped with scheduling mechanisms, allowing the mappings to run automatically based on predefined triggers, thereby eliminating the need for manual intervention. By integrating all components into a unified workflow, the task flow ensures consistency, reduces error rates, and enhances efficiency in data processing. This design not only supports the current business needs but also provides a scalable framework for future enhancements.

A screenshot of a computer

AI-generated content may be incorrect.

Fig. 41: Integrated Taskflow

**Conclusion and Future Scope**

This project successfully implements a robust data integration and warehousing solution using Informatica Intelligent Cloud Services (IICS). By creating distinct mapping tasks for each dimensional table and seamlessly integrating them into a unified task flow, the architecture guarantees scalability, accuracy, and efficient data management. Employing SCD-2 for the Student table ensures effective historical data tracking, while SCD-1 for other tables provides timely updates with reduced complexity. The staging results layer, Result Fact Table, and Result Aggregation Table collaborate to deliver thorough insights and analytics-ready data.

The future scope of the project includes:

* **Enhanced Automation**: Implement real-time data integration and automation pipelines to reduce latency and improve operational efficiency.
* **Advanced Analytics**: Integrate predictive analytics and machine learning models to derive deeper insights and support decision-making.
* **Scalability Expansion**: Optimize the architecture to handle larger datasets, increasing system capacity to accommodate growing organizational needs.
* **Multi-Cloud Integration**: Extend support to integrate data from multiple cloud platforms for broader data consolidation.
* **Data Quality Enhancements**: Develop robust data cleansing and quality-check mechanisms to ensure the accuracy and reliability of the data.

**Design Constraints**

|  |  |  |
| --- | --- | --- |
| CONSTRAINT | SUCCESS | IMPLEMENTATION |
| Student table should get loaded only when a valid No exists in the source file. | Yes | Same column name (No) having matching values in both source and target tables. |
| An item record getting loaded into the fact table should have a corresponding entry in the item dimension table. | Yes | COURSE\_CODE loaded to the Fact Table. |
| A customer record getting loaded into the fact table should have a corresponding entry in the customer dimension table. | Yes | DEPARTMENT\_ID loaded to the Fact Table. |
| The ETL data flow schedule should accommodate the order of dimension table loading before the fact table load. | Yes | Scheduled the workflows of dimensions in parallel, loading the fact table only when all dimensions are succeeded. |
| The customer-item fact table should calculate the total price based on the number of items and item price. | Yes | Implemented calculation logic in the transformation step. |
| The phone number format should be standardized to xxx-xxx-xxxx. | Yes | Applied transformation logic to format phone numbers. |
| The address fields should be concatenated with a comma separator. | Yes | Used concatenation logic in the transformation step. |
| The names should be converted to title case and lower case as specified. | Yes | Applied transformation logic using InitCap() and Lower() functions. |
| The ROLL NO should be validated and converted to a number. | Yes | Applied validation and conversion logic in the transformation step. |
| The data load process should abort if any critical validation fails. | Yes | Implemented validation checks and abort logic in the ETL process. |

Table 28: Result Aggregate Transformations