## Introduction

### Project Overview

The HireSmith-Salesforce Integration for Student Data (Phase 6) project, launched on September 28, 2023, marks a significant step towards enhancing the data integration process between HireSmith and Salesforce platforms. This project focuses on one-way data flow from Salesforce to HireSmith, prioritizing data consistency, efficient python code development, and effective use of MySQL for database management. The integration ensures system performance and usability. The initiative substantially reduces manual effort by automating data entry, supporting efficient job search activities within the Smith School's career portal, and representing a strategic collaboration with Smith IT to ensure a smooth and automated data flow.

### Context and Integration Process

Historically, the HireSmith platform relied on manual data updates from SIS (Student Information System) and ERx (External Reporting System) processes managed by the Office of Career Services (OCS). This project revolutionizes this approach by introducing an automated data flow system. The necessity for automation stems from the inefficiencies inherent in the previous manual data update processes, which this project aims to address. Under this new regime, data originally coming in from SIS and ERx will first flow into the Salesforce platform, managed by Smith IT, and then transferred into the MySQL database, populating the student profile and student degree tables. This in turn, gets transformed into appropriate data frames which will be utilized for REST API calls.

Our collaboration with Smith IT has been pivotal in aligning the Salesforce student database with the requirements of our project. Multiple meetings were held to ensure a mutual understanding of the fields to be included and how they would integrate into our system. This new automated process aims to enhance accuracy and efficiency in data integration, replacing previous manual procedures.

## 

## Technical Documentation

### Database Design

#### Initial Design and Evolution

The HireSmith-Salesforce Integration project initially began with the development of a single SQL table designed to encapsulate all student details. However, to more effectively manage data structure and relationships, the team decided to split the information into two distinct tables: student\_degree and student\_profile. The student\_degree table was uniquely designed to accommodate multiple degrees per student, leveraging a composite primary key that combines sf\_contact\_id and program.

#### Lookup Tables

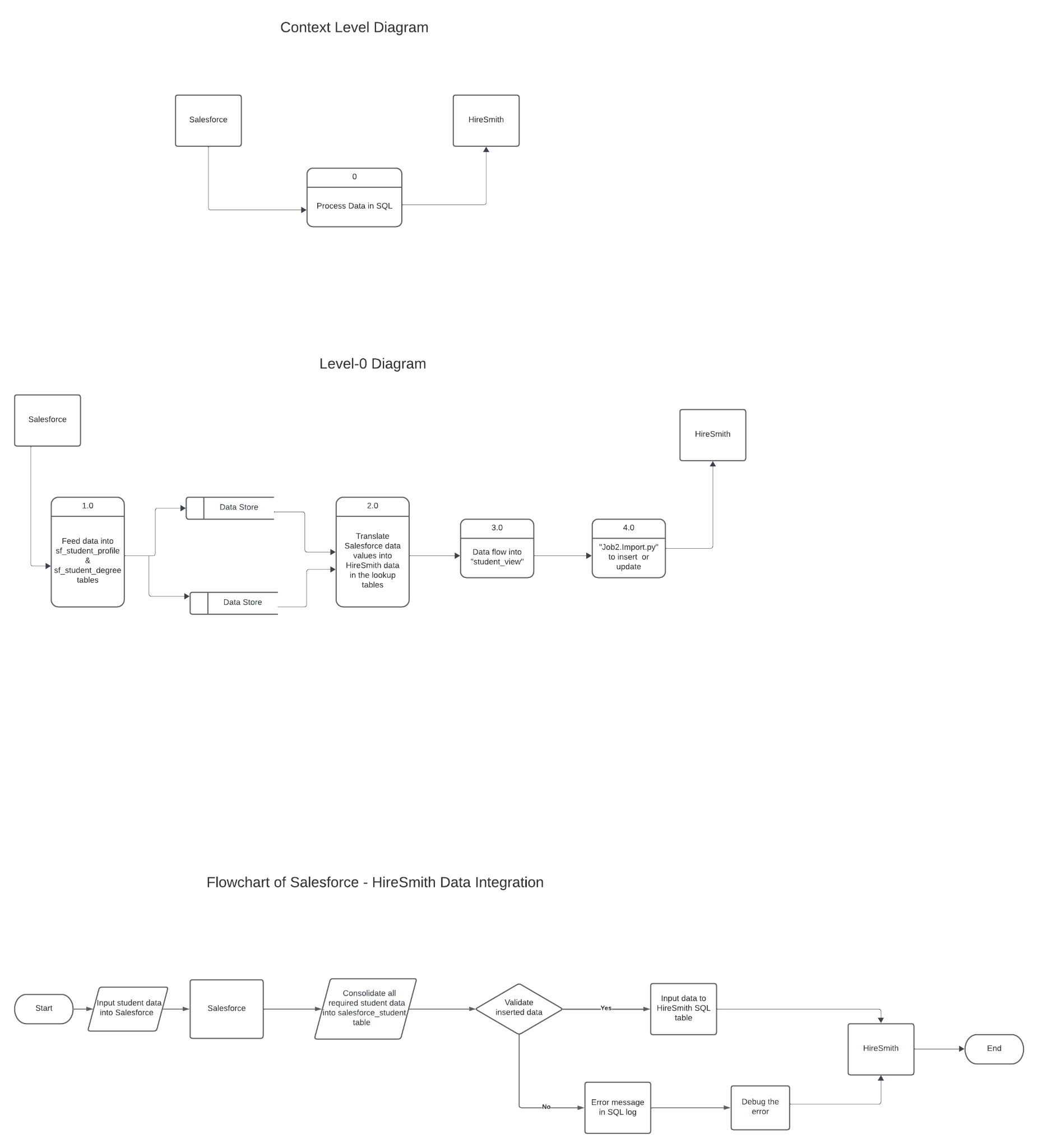
In addition to these tables, the project saw the creation of lookup tables, a critical component in managing and streamlining data processes. This initiative was led by a Python code developer from the OCS tech team, who developed a script to automatically generate all required lookup tables from the HireSmith API. A notable addition to the database was the hs\_lookup table, comprising fields like hs\_lookup\_id, hs\_lookup\_name, table\_name, and is\_used. This particular table plays a significant role in tracking and managing the application of various lookup tables within the system. The is\_used field, acting as a flag, indicates the active usage of a lookup table, with a value of 1 signifying active use and 0 for non-use. This systematic approach ensured the automatic creation and efficient management of these lookup tables upon running the script, significantly enhancing the database's functionality and organization.

#### Implementation of “Student View” in SQL

A pivotal feature we've implemented is a view composed of Common Table Expressions and Inner Joins called "student\_view." This view is designed to merge data from two student tables: “sf\_student\_profile” and “sf\_student\_degree.” By doing so, it creates a singular, efficient point for extracting data for API calls. The student view plays a crucial role in providing a unified data structure, offering a comprehensive perspective of each student’s records. It simplifies data retrieval processes significantly, eliminating the need for complex SQL joins and maintaining a stable schema. This stability is vital for ensuring consistent and reliable access to data, even when there are changes in the underlying tables.

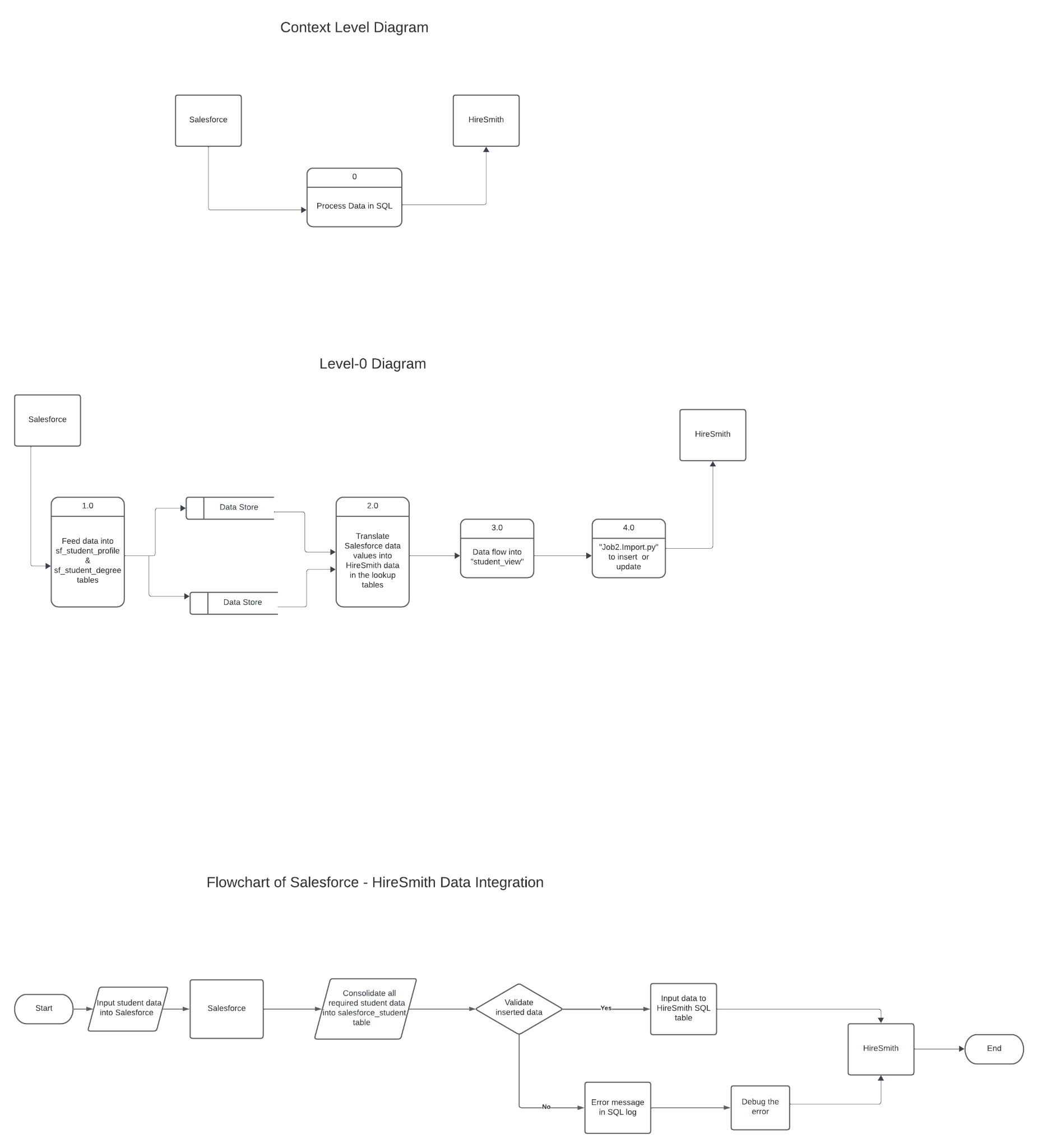
Another benefit of the student view is its optimized performance for common data requests, which enhances efficiency, particularly in reporting and analysis tasks. Additionally, it acts as an abstraction layer, adding a layer of data security and offering controlled access to sensitive student information. Finally, its design supports and facilitates reporting and analytics by streamlining data extraction, thereby enabling more insightful decision-making based on the data. This integration of the student view into our SQL framework is a testament to our commitment to efficient, secure, and effective data management in the HireSmith-Salesforce Integration project.

#### Data Flow Chart



*Figure-1: End-to-end dataflow from Salesforce to Hiresmith via Phase-6 Project*

#### ER Diagram



*Figure-2: Entity Relationship Diagram*

#### Referential Integrity with Lookup Tables:

### 

### Code Development

#### Code Structure and Adaptation

This Python script handles data extraction from the SQL database (student\_view) and prepares the data for insertion or update into a system via 12Twenty - based REST APIs (a student information system or a CRM tool). Here’s a breakdown of the script:

1. Imports and Initializations: The script starts with importing necessary modules and initializing variables required for connections and data processing.
2. Database Connection: Establishes a connection to the database server.
3. Data Loading: Loads various CSV files containing mapping data required for data transformation and processing.
4. Creating Lookup Tables: Functions are used to create lookup tables from the database.
5. Data Preparation: Defines a large dictionary (default\_df\_student) containing multiple attributes and sub-attributes. This dictionary serves as a template for the student data to be processed.
6. Looping Through SQL Results: Fetches data from the SQL database (student\_view), maps SQL fields to an API schema, prepares data according to specified logic, and constructs a payload for API insertion.
7. API Data Insertion: Sends HTTP POST requests to an API endpoint (https://url/12twenty.com/api/v2/students) with the constructed payload data (final\_student\_post).
8. Error Handling: Contains conditional blocks to handle potential errors during API requests.
9. Logging: Handle logging or recording events/errors into a database table (job\_log and potentially contact\_error\_log).

## Testing Documentation

### Overview

Our testing approach was tailored to address the challenge of one-way data integration from Salesforce. Given the absence of actual Salesforce data for testing, the project adopted a mockup data strategy. This enabled the simulation of the data integration process and validation of the system's functionality.

### Data Preparation and Collaboration

#### Mockup Data Creation:

In the absence of real-time data flow from Salesforce, we developed a comprehensive set of mockup data. This was essential for simulating real-world scenarios and validating the data integration process.

The mockup data was meticulously crafted to align with the fields and data types outlined in our database schema, ensuring a robust testing environment.

#### Collaboration with Smith IT:

We engaged in multiple discussions with the Smith IT department to understand the extent of data integration required on their end. These conversations were crucial in ensuring our system's compatibility with their data feeding processes into our SQL database.

### Testing Approach

#### Field and Data Type Listing

As a preliminary step, we compiled a complete list of all fields, their corresponding data types as defined in our database tables, and unique values for each field. This served as a reference point for generating accurate and representative mockup data.

#### Utilizing AI for Mockup Data Generation

To generate diverse and realistic student data, we employed an innovative approach by inputting our field specifications into ChatGPT. This AI tool was instrumental in creating a dataset of 5 unique student mockup profiles, each tailored to our project's specifications.

Test Results

#### Next Steps

The Smith IT team is actively configuring Informatica scripts for data loading into the student\_degree and student\_profile tables. This setup is earmarked for completion by the upcoming semester's end. Once deployed, these scripts will merge with existing employer and contact scripts, slated to execute every 5 minutes. This integration aims to streamline the data flow, ensuring comprehensive and regular updates across systems. The planned synchronization of these scripts promises a robust and timely data handling system, pivotal for maintaining updated student records and facilitating smoother operational workflows.

## Data Dictionary

The data dictionary for the HireSmith-Salesforce Integration For Student Data (Phase 6) project meticulously catalogs all fields relevant to student data drawn from both Salesforce and HireSmith. This comprehensive list includes, but is not limited to, fields such as "Field Actual Name - HireSmith", "ETL SF Table Name", "ETL SF Column Name", "ETL HS Column Name", "Prior Source", "Original ERx Field Name", and "Original SIS Field Name". A critical aspect of this process involved determining which fields were essential for inclusion in our SQL database. Each field entry in the dictionary is detailed with its data type and the source from which it originates, ensuring clarity and consistency in data management.

### Resource Links:

(+gitlab link)

[HireSmith API Documentation](https://docs.google.com/document/d/1FNloEz37YHlMWiKFMNK7QVbsY-bNFLxotcQ6fK2CHjs/edit)