**Overview**

This documentation outlines the SQL view and stored procedure used to extract and manage student account data for Active Directory provisioning. The main objective is to identify newly joined students each day, ensuring that the latest student data is captured and processed for Active Directory provisioning.

**Objective**

The primary objective of this implementation is to automate the daily extraction of student account data for students at INTO centers in the UK (specifically MER, LON, and WEC). This process ensures that all newly joined students are identified, and their account details are updated in the system for Active Directory provisioning.

**Data Sources**

**Tables Involved:**

* **SFUK.dbo.Opportunity**: Core table containing student and opportunity data.
* **SFUK.dbo.Courses\_\_c**: Table linking opportunities to courses.
* **SFUK.dbo.Product2**: Table storing course details like title, start, and end dates.
* **SFUK.dbo.Account**: Contains account-level details such as the student's first and last names.
* **SFUK.dbo.Caf\_Plus\_\_c**: Additional information related to courses.
* **SF\_STG.dbo.STG\_IDX\_Account\_In**: External staging table for account data.
* **SF\_STG.dbo.STG\_IDX\_Ebook\_In**: External staging table for ebook data.
* **IDX\_Account\_In\_Audit**: Audit table for account data.

**View Creation: v\_IDX\_Account\_ActiveDir\_V4\_MER**

The view v\_IDX\_Account\_ActiveDir\_V4\_MER is designed to aggregate and format the necessary data into a single, easily accessible structure. This view is crucial for daily monitoring of newly joined students and updating the Active Directory with the latest student information.

**Key Columns Extracted:**

* **Sits\_Number**: Salesforce ‘SITS No’ – reference for the application.
* **sAMAccountName**: Combination of "INTO" and "SITS No" with leading zeroes removed.
* **userPrincipalName**: Email address, either existing or generated based on student names.
* **mail**: Same as userPrincipalName, used as the official email.
* **SuggestedEmail**: Generated email address based on the student's name and SITS number.
* **givenName**: Student's first name.
* **sn**: Student's surname.
* **middleName**: Student's middle name(s).
* **displayName**: Combination of givenName and sn.
* **description**: Course title from Product2.Course\_Title\_\_c.
* **Start\_Date**: Course start date from Product2.start\_date\_\_c.
* **End\_Date**: Course end date from Product2.End\_Date\_\_c.
* **accountExpires**: Account expiry date (end date + 90 days).
* **department**: Text description of account expiry date.
* **Password**: Default password based on the student's date of birth in the format INTOpass!ddMMyyyy.
* **passwordNeverExpires**: Flag indicating the password never expires.
* **O365Mailbox**: Default value ‘TRUE’.
* **c**: Country code (‘GB’ for Great Britain).
* **mailNickname**: Similar to sAMAccountName, used as the mail nickname.
* **Registration\_Status\_Code\_\_c**: Registration status of the student.
* **Centre**: Center code, filtered to include LON, MER, and WEC.

**SQL View Script:**

sql

USE [SF\_STG]

GO

-- Drop the existing view if it exists

IF OBJECT\_ID('dbo.v\_IDX\_Account\_ActiveDir\_V4\_MER', 'V') IS NOT NULL

BEGIN

DROP VIEW dbo.v\_IDX\_Account\_ActiveDir\_V4\_MER;

END

GO

-- Create or recreate the view

CREATE VIEW [dbo].[v\_IDX\_Account\_ActiveDir\_V4\_MER] AS

SELECT DISTINCT

-- Columns definitions...

-- <Include full view script here>

FROM SFUK.dbo.Opportunity AS o

LEFT JOIN SFUK.dbo.Account AS a ON o.accountid = a.id

LEFT JOIN SFUK.dbo.Courses\_\_c AS c ON o.id = c.Study\_Plan\_\_c

LEFT JOIN SFUK.dbo.Caf\_Plus\_\_c as cp on c.id = cp.Course\_Applied\_For\_\_c

LEFT JOIN SFUK.dbo.Product2 p ON c.course\_name\_\_c = p.id

FULL OUTER JOIN SF\_STG.dbo.STG\_IDX\_Account\_In AS ia ON ia.university\_student\_code = o.University\_Student\_Code\_\_c

FULL OUTER JOIN SF\_STG.dbo.STG\_IDX\_Ebook\_In AS eb ON eb.Sits\_Number = o.SITS\_Number\_\_c

LEFT JOIN IDX\_Account\_In\_Audit aud ON o.AccountId = aud.sf\_Account\_Id

WHERE

(o.INTO\_Centre\_Country\_\_c = 'UK'

AND o.Centre\_MIS\_Code\_\_c IN ('LON', 'MER','WEC')) -- Adjusted filter for Centre\_MIS\_Code\_\_c

AND o.recordtypeid != '01230000001aE33AAE';

GO

**Stored Procedure: GETREP\_Active\_Directory\_Passwords\_V4\_MER**

The stored procedure GETREP\_Active\_Directory\_Passwords\_V4\_MER is designed to run daily, extracting the latest student data from the view. This procedure identifies newly joined students and updates their account information in Active Directory.

**Parameters:**

* **@pCentre**: Filter by center code (e.g., 'MER', 'LON', 'WEC').
* **@pStartDate**: Start date for filtering courses.
* **@pEndDate**: End date for filtering courses.
* **@pRegStatus**: Registration statuses to include (defaults to 'P,C,CR').

**Output:**

The stored procedure returns a set of columns derived from the view, with data formatted and filtered according to the parameters provided. It ensures that the most recent student data is captured daily.

**SQL Stored Procedure Script:**

sql

Copy code

USE [SF\_STG]

GO

ALTER PROCEDURE [dbo].[GETREP\_Active\_Directory\_Passwords\_V4\_MER] (

@pCentre VARCHAR(5) = NULL, -- Default to NULL, can be 'MER' or 'LON'

@pStartDate DATETIME = NULL,

@pEndDate DATETIME = NULL,

@pRegStatus VARCHAR(10) = 'P,C,CR' -- Default to 'P,C,CR'

)

AS

BEGIN

-- Procedure logic...

-- <Include full stored procedure script here>

END

GO

**Requirements:**

1. **Registration Status**: The procedure filters students based on specific registration statuses ('P', 'C', 'CR').
2. **Date Range**: It includes students whose course start date is between 14 days before and 30 days after the current date, unless specified otherwise.
3. **Center**: It filters the data to include only students from the MER, LON, or WEC centers.

**Purpose:**

This stored procedure ensures that the system is updated daily with the latest information about newly joined students. By doing so, it allows the institution to keep track of new students joining the institution and ensures that their Active Directory accounts are created and maintained accurately.