**In UMA, the Operational Data Store (ODS) is used as an intermediary system that processes and consolidates data from multiple sources before it is utilized for reporting and analysis. Here’s a detailed step-by-step overview of how ODS is used in UMA:**

**Data Sources:**

**D2L: This system is crucial for grade submissions from students. Data from D2L is brought into LMS\_Reporting twice daily at 3:30 PM and 11:30 PM.**

**Anthology: Contains comprehensive student information such as addresses, social security numbers, contact information, and other personal and registrar-related data.**

**Data Integration:**

**The data from D2L and Anthology are combined in the LMS\_Reporting system. This integration ensures that all relevant student information and grade submissions are centralized.**

**Processing in ODS:**

**After the data is brought into LMS\_Reporting, the ODS runs a process post the 11:30 PM data pull from D2L.**

**The ODS applies business logic to the integrated data, which has been defined by the business stakeholders. This business logic transforms the raw data into a more structured and meaningful format, suitable for reporting and analysis.**

**Final Data Push:**

**The processed data from ODS is then pushed to the LS\_ODS schema tables. These tables contain the final, processed data that has been validated and formatted according to the business requirements.**

**Utilization:**

**The data in the LS\_ODS schema tables is utilized by various reports and the IDASH system. These reports and dashboards provide insights and analytics based on the processed data, supporting decision-making and operational activities within UMA.**

**In summary, the ODS in UMA acts as a critical layer that ensures data from D2L and Anthology is processed accurately and efficiently, applying necessary business rules before making it available for reporting and analysis.**

**What is ODS in Ultimate Medical Academy and explain in detail the process**

**ChatGPT**

**Memory updated**

**Operational Data Store (ODS) in Ultimate Medical Academy (UMA)**

**What is ODS?**

**At Ultimate Medical Academy (UMA), the Operational Data Store (ODS) serves as a centralized repository that consolidates data from multiple systems. It is a critical component for integrating and processing data from various sources to support reporting and analysis.**

**Data Sources:**

**D2L (Brightspace):**

**Purpose: Key system for grade submissions from students.**

**Process: Data from D2L is differentially extracted using Azure Data Factory (ADF) twice a day (at 3:30 PM and 11:30 PM). This data is then brought into the LMS\_Reporting database.**

**Anthology:**

**Purpose: Contains comprehensive student information such as addresses, social security numbers, contact details, and other personal and registrar information.**

**Data Integration and Processing:**

**LMS\_Reporting:**

**Role: Serves as the intermediary database where data from D2L and Anthology is combined.**

**Process: The data from D2L (grades) and Anthology (student information) is integrated into LMS\_Reporting.**

**ODS Creation:**

**Timing: The ODS is only updated after the 11:30 PM run of data extraction from D2L.**

**Role: Implements business logic defined by UMA for student and staff data. The ODS processes the integrated data from LMS\_Reporting, applying business rules and transformations.**

**Final Data Storage and Usage:**

**LS\_ODS Schema:**

**Purpose: The processed data is stored in tables within the LS\_ODS schema.**

**Usage: These tables are used by various reports and the IDASH system, providing insights and data analysis for decision-making.**

**Process Summary:**

**Data Extraction: D2L extracts grade data twice daily.**

**Data Integration: Combined with Anthology student data in LMS\_Reporting.**

**Business Logic Application: ODS processes the integrated data post 11:30 PM, applying business rules.**

**Data Storage: Final data is stored in LS\_ODS schema tables.**

**Data Utilization: Used for reporting and IDASH.**

**This detailed process ensures that UMA has a reliable and integrated data source for accurate reporting and analysis, supporting the needs of students, staff, and administration.**

**ODS Modification Script - ODS\_Process\_New\_v1**

**Author and Revision History**

**Author:** Mimi Pierce-Byrd  
**Create Date:** 03/14/2023

**Change Control**

| **Version** | **Author** | **Date** | **Description** |
| --- | --- | --- | --- |
| 1.0 | MPB | 03/14/2023 | Initial Code Deployed |
| 1.1 | MPB | 03/15/2023 | Removed Blackboard code |
| 1.2 | MPB | 03/20/2023 | Added the Audit.GradeExtractImport\_d2l table to save the data in case we need to reprocess grade records. |
| 1.3 | PP | 05/01/2023 | Added audit.grades table and error handling for merge process, also catch block added |
| 1.3.1 | RR | 01/25/2024 | Edited for Ticket VA Attendance Report issue SR#3322624 |
| 1.4 | CML | 02/28/2024 | Updated course code logic to capture EMT course data and week number code for 16-week courses. |
| 1.4.1 | MM | 07/05/2024 | Temporary fix for a duplicate data issue by adding DISTINCT to line 5384 |

**Description**

This document outlines the modifications made to the ODS (Operational Data Store) process. The changes include updates to handle various scenarios such as deduplication, data translation, and specific handling for EMT course data.

**Detailed Change Log**

**Initial Code Deployment (03/14/2023)**

* Deployed the initial version of the stored procedure.

**Blackboard Code Removal (03/15/2023)**

* Removed specific Blackboard code from the procedure.

**Addition of Grade Extract Import Audit Table (03/20/2023)**

* Added the Audit.GradeExtractImport\_d2l table to save the data for potential reprocessing of grade records.

**Error Handling and Merge Process Update (05/01/2023)**

* Added the audit.grades table.
* Implemented error handling for the merge process.
* Added a catch block to handle exceptions.

**VA Attendance Report Issue (01/25/2024)**

* Edited by Rogan Richeart to address the VA Attendance Report issue (SR#3322624).
* Uncommented the VA Report execution lines.

**EMT Course Data Handling (02/28/2024)**

* Updated course code logic to capture EMT course data, which follows a different naming convention (e.g., AAA1111[xx] or AAA-1111[xx]).
* Updated week number code to account for EMT courses running 16 weeks.
* Noted that EMT only performs weekly grades in week 6, currently captured in week 5.

**Temporary Fix for Duplicate Data Issue (07/05/2024)**

* Added DISTINCT to line 5384 to address a duplicate data issue.

**Processing Steps**

**Grade Extract Import**

**Step 1: Clean and Deduplicate Data**

* + Drop Indexes: Remove specific indexes (idx\_GEI\_0001, idx\_GEI\_0002, idx\_ODS\_019) from the stage.GradeExtractImport table to optimize performance for subsequent operations.
  + Identify and Log Duplicates: Use a Common Table Expression (CTE) to find duplicate rows based on specific columns and insert these duplicates into the Stage.ODS\_Duplicates table for logging.
  + Delete Duplicates: Use another CTE to identify the primary keys of the duplicate rows and delete them from the stage.GradeExtractImport\_d2l table.
  + Merge Cleaned Data into Destination Table
  + Prepare Data for Merge: Select the necessary columns from stage.GradeExtractImport\_d2l where SourceSystem is 'D2L' using a CTE.
  + Perform Merge Operation: Use the MERGE statement to:
  + Update existing rows in the target table (GEICTE) that match on the primary key fields but have different data.
  + Insert new rows from the source table that do not exist in the target table.
  + Delete rows in the target table that do not have a corresponding row in the source table
  + .

1. **Audit Table Backup:**
   * Save data from the GradeExtractImport\_d2l to an audit table in case reruns are necessary.

**Data Translation and Cleanup**

* Translate new student-facing reporting values to internal-facing values.
* Clean up bad values in Gen 3 Courses.
* Clean up incorrect max points values found in the GradeExtract file.
* Compare assignments to determine and store those with multiple points possible values.
* Update the GradeExtractImport table to correct assignment points possible values.
* Clean up missing assignment status values found in the GradeExtract file.
* Fix bad display score values by replacing double quotes with commas.

**Stage Table Population**

* Truncate necessary stage tables for data processing.
* Populate stage.Students, stage.Courses, and stage.Assignments tables with values from the raw import table.
* Populate stage.Grades table with values from the raw import table.

**Updates to Stage Tables**

1. **IEHR Assignment Statuses:**
   * Update the stage.Grades table with IEHR assignment statuses.
2. **Student Information:**
   * Update the stage.Students table with calculated values such as last logins and time in class.
   * Update the stage.Students table with I3 interaction information.
3. **Weekly Grades:**
   * Update the stage.Students table with weekly grades, noting special handling for EMT courses.
4. **Assignment and Discussion Post Counts:**
   * Update the stage.Students table with counts of practice exercises, tests, assignments, and discussion posts.

**Course Table Updates**

* Update the stage.Courses table with section start/end dates and course week numbers.
* Update the stage.Grades table with Cengage values.
* Adjust stage.Courses table to account for holiday schedules defined by CampusVue.

**Additional Updates**

1. **Update IsReportable and CountsAsSubmission Values:**
   * Update the stage.Assignments table with IsReportable and CountsAsSubmission values.
2. **Update Weekly Assignment Counts:**
   * Update the stage.Courses table with the weekly assignment counts.
   * Update the stage.Students table with weekly completed assignment counts and submission rates.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Student Counts Of Completed Assignments And Submission Rates', 0;.
3. **Completion/Submission Rates by Assignment Type:**
   * Update completion/submission rates by assignment type using EXEC LS\_ODS.ProcessStudentRatesByAssignmentType;.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Completion/Submission Rates By Assignment Type', 0;.
4. **Update Weekly LDAs:**
   * Update the stage.Students table with weekly LDAs for each week from 1 to 5.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Student Weekly LDAs', 0;.
5. **Grade Counts of Attempts:**
   * Update the stage.Grades table with the number of attempts.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Grade Counts Of Attempts', 0;.
6. **Update Time in Module Table:**
   * Add new records to the TimeInModule table.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Time In Module Table', 0;.
7. **Update Current Course and Posted Grades:**
   * Update the stage.Students table with the current course grade and posted grades.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Student Current Course Grade', 0; and EXEC LS\_ODS.AddODSLoadLog 'Updated Student Posted Grades', 0;.
8. **Update Changed Student Records:**
   * Identify and update changed student records.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Students Records That Have Changed', 0;.
9. **Update Changed Course Records:**
   * Identify and update changed course records.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Course Records That Have Changed', 0;.
10. **Update Changed Assignment Records:**
    * Identify and update changed assignment records.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Assignment Records That Have Changed', 0;.
11. **Update Changed Grade Records:**
    * Identify and update changed grade records.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Grades Records That Have Changed', 0;.
12. **Add New Records:**
    * Add new student, course, and assignment records.
    * Log the addition with EXEC LS\_ODS.AddODSLoadLog 'Added New Students Records', 0;, EXEC LS\_ODS.AddODSLoadLog 'Added New Course Records', 0;, and EXEC LS\_ODS.AddODSLoadLog 'Added New Assignment Records', 0;.
13. **Add New Grade Records:**
    * Insert new grade records into the Grades table.
    * Log the addition with EXEC LS\_ODS.AddODSLoadLog 'Added New Grade Records', 0;.
14. **Remove Invalid Student Records:**
    * Remove records in the Students table with no StudentCourseUserKey.
    * Log the removal with EXEC LS\_ODS.AddODSLoadLog 'Removed Student Records With No Valid StudentCourseUserKey Value', 0;.
15. **Remove Negative Primary Key Grade Records:**
    * Handle grade records with negative primary keys by deleting them.
    * Log the removal with EXEC LS\_ODS.AddODSLoadLog 'Removed Grade Records With Negative Primary Keys', 0;.
16. **Process Course Activity Counts:**
    * Process course activity counts for BI reporting.
    * Log the processing with EXEC LS\_ODS.AddODSLoadLog 'Processed Course Activity Counts', 0;.
17. **Disable Courses Not in GradeExtract:**
    * Create a distinct list of all courses and disable any course no longer in the GradeExtract.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Removed Disable Courses', 0;.
18. **Disable Old Student/Course Combinations:**
    * Create a distinct list of all student/section combinations and disable any old section.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Removed Disabled Student/Course Combinations', 0;.
19. **Disable Students Without Enrollment Records:**
    * Disable all students with no matching CampusVue enrollment records.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Removed Students With No CampusVue Enrollment Records', 0;.
20. **Archive Old Records:**
    * Move old student, course, assignment, and grade records to audit tables.
    * Log the archiving with EXEC LS\_ODS.AddODSLoadLog 'Moved Old Student Records To Archive Table', 0;, EXEC LS\_ODS.AddODSLoadLog 'Moved Old Course Records To Archive Table', 0;, EXEC LS\_ODS.AddODSLoadLog 'Moved Old Assignments Records To Archive Table', 0;, and EXEC LS\_ODS.AddODSLoadLog 'Moved Old Grades Records To Archive Table', 0;.
21. **Remove Duplicate Records:**
    * Set active flags, update the most recent grade record to active, and delete any remaining duplicates.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Grade Duplicate Check And Deletion Complete', 0;.
22. **Remove Orphaned Student Records:**
    * Identify and move orphaned student records to the archive table.
    * Log the removal with EXEC LS\_ODS.AddODSLoadLog 'Removed Orphaned Student Records', 0;.
23. **Update Tables for iDash Reporting:**
    * Update CourseWeeklyGradedActivity and CourseWeeklyGrades tables.
    * Log the processing with EXEC LS\_ODS.AddODSLoadLog 'Processed Course Weekly Graded Activity', 0; and EXEC LS\_ODS.AddODSLoadLog 'Processed Course Weekly Grades', 0;.
24. **Process ActiveSubmissionSummary Table:**
    * Process the ActiveSubmissionSummary table.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Active Submission Summary Procesing Complete', 0;.
25. **Update Total Course Points Earned:**
    * Process and update the Total Course Points Earned table.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Total Course Points Earned Procesing Complete', 0;.
26. **ICD-10 Reporting Table Update:**
    * Populate tables needed for ICD-10 reporting.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'ICD Reporting Table Update Complete', 0;.
27. **Sandbox Reporting Data Update:**
    * Process sandbox reporting data.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Sandbox Reporting Table Update Complete', 0;.
28. **RHIT Reporting Data Update:**
    * Process RHIT reporting data.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'RHIT Table Update Complete', 0;.
29. **PTCE Reporting Data Update:**
    * Process PTCE reporting data.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'PTCE Table Update Complete', 0;.
30. **Program Certification Data Update:**
    * Process program certification data.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Program Certification Tables Update Complete', 0;.
31. **Course Aggregate Data Update:**
    * Process course aggregate data.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Course Aggregate Tables Update Complete', 0;.
32. **Assignment Attempt Counts Update:**
    * Merge assignment attempt counts.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Assignment Attempt Counts Update Complete', 0;.
33. **GAR Data Processing:**
    * Process GAR data.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'GAR Data Processing Complete', 0;.
34. **VA Report Data Processing:**
    * Process VA report data according to the October 2015 policy.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'VA Report Data Processing Complete', 0;.
35. **SSRS Reports Creation:**
    * Generate SSRS reports.
    * Log the start with EXEC LS\_ODS.AddODSLoadLog 'SSRS Reports Creation Started', 0;.
    * Execute SSRS report jobs:
      + EXEC [MLK-REP-P-SQ02].msdb.dbo.sp\_start\_job N'E5401A80-B99C-4840-83DE-57DDFDCD6C81';
      + EXEC [MLK-REP-P-SQ02].msdb.dbo.sp\_start\_job N'F98F0617-E4F1-4F1F-A384-B6EE78BA9EF5';
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'SSRS Reports Creation Complete', 0;.
36. **Update the UpdateLog Table:**
    * Finalize the process and update the UpdateLog table for status reporting.
    * Log the final step with EXEC LS\_ODS.AddODSLoadLog 'ODS Load Process Complete', 0;.

**Conclusion**

This document provides a detailed overview of the ODS modification process, including all updates, processing steps, and logging procedures. It ensures transparency and traceability for all changes made to the ODS.

Let me know if you need further modifications or details added!

4o

Basin on this can you create a ODS documents in UMA and the process

ChatGPT

Sure, I can help create an ODS (Operational Data Store) document for UMA, outlining the process in a structured format. Below is a draft of the document, following the detailed steps and structure from the previous ODS modification script documentation.

**UMA ODS Documentation**

**Author and Revision History**

**Author:** [Your Name]  
**Create Date:** [Date]

**Change Control**

| **Version** | **Author** | **Date** | **Description** |
| --- | --- | --- | --- |
| 1.0 | [Your Initials] | [Date] | Initial Document Creation |

**Description**

This document outlines the process for the UMA (University of Management and Administration) Operational Data Store (ODS). The ODS process involves handling various scenarios such as deduplication, data translation, and specific handling for course data.

**Detailed Change Log**

**Initial Document Creation ([Date])**

* Created the initial version of the ODS process documentation for UMA.

**Processing Steps**

**Grade Extract Import**

1. **Duplication Handling:**
   * Insert duplicates into an error table using a CTE.
   * Delete duplicates from the source table using a new CTE.
   * Merge deduplicated data into the destination table.
2. **Audit Table Backup:**
   * Save data from the GradeExtractImport\_d2l to an audit table in case reruns are necessary.

**Data Translation and Cleanup**

* Translate new student-facing reporting values to internal-facing values.
* Clean up bad values in Gen 3 Courses.
* Clean up incorrect max points values found in the GradeExtract file.
* Compare assignments to determine and store those with multiple points possible values.
* Update the GradeExtractImport table to correct assignment points possible values.
* Clean up missing assignment status values found in the GradeExtract file.
* Fix bad display score values by replacing double quotes with commas.

**Stage Table Population**

* Truncate necessary stage tables for data processing.
* Populate stage.Students, stage.Courses, and stage.Assignments tables with values from the raw import table.
* Populate stage.Grades table with values from the raw import table.

**Updates to Stage Tables**

1. **IEHR Assignment Statuses:**
   * Update the stage.Grades table with IEHR assignment statuses.
2. **Student Information:**
   * Update the stage.Students table with calculated values such as last logins and time in class.
   * Update the stage.Students table with I3 interaction information.
3. **Weekly Grades:**
   * Update the stage.Students table with weekly grades, noting special handling for EMT courses.
4. **Assignment and Discussion Post Counts:**
   * Update the stage.Students table with counts of practice exercises, tests, assignments, and discussion posts.

**Course Table Updates**

* Update the stage.Courses table with section start/end dates and course week numbers.
* Update the stage.Grades table with Cengage values.
* Adjust stage.Courses table to account for holiday schedules defined by CampusVue.

**Additional Updates**

1. **Update IsReportable and CountsAsSubmission Values:**
   * Update the stage.Assignments table with IsReportable and CountsAsSubmission values.
2. **Update Weekly Assignment Counts:**
   * Update the stage.Courses table with the weekly assignment counts.
   * Update the stage.Students table with weekly completed assignment counts and submission rates.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Student Counts Of Completed Assignments And Submission Rates', 0;.
3. **Completion/Submission Rates by Assignment Type:**
   * Update completion/submission rates by assignment type using EXEC LS\_ODS.ProcessStudentRatesByAssignmentType;.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Completion/Submission Rates By Assignment Type', 0;.
4. **Update Weekly LDAs:**
   * Update the stage.Students table with weekly LDAs for each week from 1 to 5.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Student Weekly LDAs', 0;.
5. **Grade Counts of Attempts:**
   * Update the stage.Grades table with the number of attempts.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Grade Counts Of Attempts', 0;.
6. **Update Time in Module Table:**
   * Add new records to the TimeInModule table.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Time In Module Table', 0;.
7. **Update Current Course and Posted Grades:**
   * Update the stage.Students table with the current course grade and posted grades.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Student Current Course Grade', 0; and EXEC LS\_ODS.AddODSLoadLog 'Updated Student Posted Grades', 0;.
8. **Update Changed Student Records:**
   * Identify and update changed student records.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Students Records That Have Changed', 0;.
9. **Update Changed Course Records:**
   * Identify and update changed course records.
   * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Course Records That Have Changed', 0;.
10. **Update Changed Assignment Records:**
    * Identify and update changed assignment records.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Assignment Records That Have Changed', 0;.
11. **Update Changed Grade Records:**
    * Identify and update changed grade records.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Updated Grades Records That Have Changed', 0;.
12. **Add New Records:**
    * Add new student, course, and assignment records.
    * Log the addition with EXEC LS\_ODS.AddODSLoadLog 'Added New Students Records', 0;, EXEC LS\_ODS.AddODSLoadLog 'Added New Course Records', 0;, and EXEC LS\_ODS.AddODSLoadLog 'Added New Assignment Records', 0;.
13. **Add New Grade Records:**
    * Insert new grade records into the Grades table.
    * Log the addition with EXEC LS\_ODS.AddODSLoadLog 'Added New Grade Records', 0;.
14. **Remove Invalid Student Records:**
    * Remove records in the Students table with no StudentCourseUserKey.
    * Log the removal with EXEC LS\_ODS.AddODSLoadLog 'Removed Student Records With No Valid StudentCourseUserKey Value', 0;.
15. **Remove Negative Primary Key Grade Records:**
    * Handle grade records with negative primary keys by deleting them.
    * Log the removal with EXEC LS\_ODS.AddODSLoadLog 'Removed Grade Records With Negative Primary Keys', 0;.
16. **Process Course Activity Counts:**
    * Process course activity counts for BI reporting.
    * Log the processing with EXEC LS\_ODS.AddODSLoadLog 'Processed Course Activity Counts', 0;.
17. **Disable Courses Not in GradeExtract:**
    * Create a distinct list of all courses and disable any course no longer in the GradeExtract.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Removed Disable Courses', 0;.
18. **Disable Old Student/Course Combinations:**
    * Create a distinct list of all student/section combinations and disable any old section.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Removed Disabled Student/Course Combinations', 0;.
19. **Disable Students Without Enrollment Records:**
    * Disable all students with no matching CampusVue enrollment records.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Removed Students With No CampusVue Enrollment Records', 0;.
20. **Archive Old Records:**
    * Move old student, course, assignment, and grade records to audit tables.
    * Log the archiving with EXEC LS\_ODS.AddODSLoadLog 'Moved Old Student Records To Archive Table', 0;, EXEC LS\_ODS.AddODSLoadLog 'Moved Old Course Records To Archive Table', 0;, EXEC LS\_ODS.AddODSLoadLog 'Moved Old Assignments Records To Archive Table', 0;, and EXEC LS\_ODS.AddODSLoadLog 'Moved Old Grades Records To Archive Table', 0;.
21. **Remove Duplicate Records:**
    * Set active flags, update the most recent grade record to active, and delete any remaining duplicates.
    * Log the completion with EXEC LS\_ODS.AddODSLoadLog 'Grade Duplicate Check And Deletion Complete', 0;.
22. **Remove Orphaned Student Records:**
    * Identify and move orphaned student records to the archive table.
    * Log the removal with EXEC LS\_ODS.AddODSLoadLog 'Removed Orphaned Student Records', 0;.
23. **Update Tables for iDash Reporting:**
    * Update CourseWeeklyGradedActivity and CourseWeeklyGrades tables.
    * Log the processing with EXEC LS\_ODS.AddODSLoadLog 'Processed Course Weekly Graded Activity', 0; and EXEC LS\_ODS.AddODSLoadLog 'Processed Course Weekly Grades', 0;.
24. **Process ActiveSubmissionSummary Table:**
    * Process the ActiveSubmissionSummary table.
    * Log the completion with `EXEC LS\_ODS.AddODSLoadLog 'Active Submission Summary Procesing Complete', 0;

4o

can you make a user story document

ChatGPT

Certainly! Below is a user story document based on the ODS (Operational Data Store) process for UMA (University of Management and Administration). The document is structured to describe user stories for the different roles involved in the process, such as database administrators, data analysts, and business intelligence teams.

**User Story Document: UMA ODS Process**

**Introduction**

This document outlines user stories for the UMA Operational Data Store (ODS) process. These user stories describe the needs and expectations of various roles involved in managing and using the ODS.

**User Stories**

**1. As a Database Administrator**

**Title:** Ensure Data Integrity  
**Description:** As a Database Administrator, I need to ensure that the data in the ODS is accurate, deduplicated, and cleaned to maintain data integrity for reporting and analysis.  
**Acceptance Criteria:**

* All duplicate records are identified and removed.
* Data translation and cleanup scripts are executed successfully.
* Error logs are reviewed and no unresolved errors are present.

**Title:** Maintain Historical Data  
**Description:** As a Database Administrator, I need to archive old records in the audit tables to keep the ODS tables manageable and maintain historical data for reference.  
**Acceptance Criteria:**

* Old student, course, assignment, and grade records are moved to audit tables.
* Archive logs confirm the successful transfer of records.

**Title:** Log Process Steps  
**Description:** As a Database Administrator, I need to log each step of the ODS process to provide a clear audit trail and facilitate troubleshooting.  
**Acceptance Criteria:**

* Each major step in the ODS process logs a message to the load log table.
* Logs include timestamps and descriptions of completed actions.

**2. As a Data Analyst**

**Title:** Access Cleaned and Updated Data  
**Description:** As a Data Analyst, I need access to cleaned and updated data in the ODS to perform accurate data analysis and generate reports.  
**Acceptance Criteria:**

* Data in the stage.Students, stage.Courses, stage.Assignments, and stage.Grades tables is up-to-date and free of duplicates and errors.
* The ODS load process runs successfully without any data inconsistencies.

**Title:** Track Data Changes  
**Description:** As a Data Analyst, I need to track changes in student, course, assignment, and grade records to analyze trends and provide insights.  
**Acceptance Criteria:**

* Changed records are identified and logged.
* Reports on changes are generated and reviewed.

**3. As a Business Intelligence (BI) Team Member**

**Title:** Generate Accurate Reports  
**Description:** As a BI Team Member, I need to generate accurate reports based on the ODS data to provide stakeholders with reliable information for decision-making.  
**Acceptance Criteria:**

* Reports are generated from the latest ODS data.
* Reports are free of discrepancies and reflect the current state of the data.

**Title:** Process Specialized Data  
**Description:** As a BI Team Member, I need to process specific data sets, such as ICD-10, RHIT, PTCE, and program certification data, for specialized reporting needs.  
**Acceptance Criteria:**

* Specialized data processing scripts are executed successfully.
* Logs confirm the completion of data processing for each specialized data set.

**Title:** Ensure Timely Data Availability  
**Description:** As a BI Team Member, I need to ensure that the data is available in the ODS promptly to meet reporting deadlines.  
**Acceptance Criteria:**

* The ODS load process completes within the scheduled time frame.
* Reports are generated and delivered on schedule.

**4. As a Quality Assurance (QA) Tester**

**Title:** Verify Data Quality  
**Description:** As a QA Tester, I need to verify the quality of the data in the ODS to ensure that it meets the required standards for accuracy and completeness.  
**Acceptance Criteria:**

* Data quality checks are performed after the ODS load process.
* Any data quality issues are documented and addressed before reports are generated.

**Title:** Test Data Cleanup and Deduplication Scripts  
**Description:** As a QA Tester, I need to test the data cleanup and deduplication scripts to ensure they are functioning correctly and not removing valid data.  
**Acceptance Criteria:**

* Test cases for data cleanup and deduplication are executed.
* Results are reviewed, and any issues are reported and resolved.

**Conclusion**

These user stories outline the expectations and requirements of different roles involved in the UMA ODS process. By addressing these user stories, the ODS process can be improved to meet the needs of database administrators, data analysts, BI team members, and QA testers effectively.

|  |
| --- |
| EventDetails |
| ODS Load Process Started |
| Finished Loading GradeExtract Data from D2L |
| Cleaned Up Bad Gen 3 Course Values |
| Cleaned Up Bad Assignment Max Points Values |
| Cleaned Up Missing Assignment Status Values |
| Truncated Working Tables |
| Loaded Students Working Table |
| Loaded Courses Working Table |
| Loaded Assignments Working Table |
| Loaded Grades Working Table |
| Updated IEHR Assignment Statuses |
| Updated Student Last Logins |
| Updated Student Times In Class |
| Updated Student Last I3 Interactions |
| Updated Student Weekly Grades |
| Updated Student Counts Of Practice Exercises, Tests And Assignments |
| Updated Student Counts Of Discussion Posts |
| Updated Course Start Dates And Week Numbers |
| Updated Course Week X Start Dates And Extension Week Start Date |
| Updated Assignments IsReportable And CountsAsSubmission Flags |
| Updated Course Weekly Assignment Counts |
| Updated Student Counts Of Completed Assignments And Submission Rates |
| Updated Completion/Submission Rates By Assignment Type |
| Updated Student Weekly LDAs |
| Updated Grade Counts Of Attempts |
| Updated Time In Module Table |
| Updated Student Current Course Grade |
| Updated Student Posted Grades |
| Updated Students Records That Have Changed |
| Updated Course Records That Have Changed |
| Updated Assignment Records That Have Changed |
| Updated Grades Records That Have Changed |
| Added New Students Records |
| Added New Course Records |
| Added New Assignment Records |
| Added New Grade Records |
| Removed Student Records With No Valid StudentCourseUserKey Value |
| Removed Grade Records With Negative Primary Keys |
| Processed Course Activity Counts |
| Removed Disable Courses |
| Removed Disabled Student/Course Combinations |
| Removed Students With No CampusVue Enrollment Records |
| Moved Old Student Records To Archive Table |
| Moved Old Course Records To Archive Table |
| Moved Old Assignments Records To Archive Table |
| Moved Old Grades Records To Archive Table |
| Student Duplicate Check And Deletion Complete |
| Course Duplicate Check And Deletion Complete |
| Assignment Duplicate Check And Deletion Complete |
| Grade Duplicate Check And Deletion Complete |
| Removed Orphaned Student Records |
| LDA Counts Calculation Complete |
| Processed Course Weekly Graded Activity |
| Processed Course Weekly Grades |
| Active Submission Summary Procesing Complete |
| Total Course Points Earned Procesing Complete |
| ICD Reporting Table Update Complete |
| Sandbox Reporting Table Update Complete |
| RHIT Table Update Complete |
| PTCE Table Update Complete |
| Program Certification Tables Update Complete |
| Course Aggregate Tables Update Complete |
| Assignment Attempt Counts Update Complete |
| GAR Data Processing Complete |
| VA Report Data Processing Complete |
| SSRS Reports Creation Started |
| SSRS Reports Creation Complete |
| ODS Load Process Complete |