

Case Study

Building a Reproducible SIS and LMS Test Environment with Standards-Conformant Synthetic Data



The Client

The client is a data and AI organization developing higher education analytics to help institutions identify and support students who may be at risk of not completing their programs. Its product depended on data from student information and learning management systems. However, obtaining real institutional records for development, integration testing, and demonstrations introduced privacy reviews, institutional approvals, data-sharing requirements, and long provisioning cycles.

The Challenge

The client's engineering teams needed source-system-aligned data to build ingestion pipelines, test cross-system identity matching, reproduce defects, and demonstrate the product. Real institutional data depended on agreements, approvals, and campus IT provisioning. Generic mock data was easier to obtain, but it did not reliably reproduce referential integrity, rare retention outcomes, cross-system identifier mismatches, or the schemas used by the client's SIS and LMS integrations. The team needed a reusable non-production environment that could test those conditions without using real student records.

Critical Success Parameters

- ✓ Represent students, courses, sections, enrollments, grades, programs, and engagement data across connected SIS and LMS models.
- ✓ Preserve referential integrity and cross-system identity relationships.
- ✓ Generate different institution profiles, academic calendars, edge cases, and rare retention outcomes.
- ✓ Support API-based integration testing and bulk warehouse or lakehouse ingestion.
- ✓ Recreate the same dataset and defect from a recorded seed.
- ✓ Fail builds when schemas, relationships, or exported files did not validate.
- ✓ Keep real student records outside the development and demonstration environment.

Our Approach

Magic EdTech structured the engagement around three steps: specify, generate, and prove.

- ✓ Magic EdTech first converted the client's product and reporting requirements into a written synthetic data specification. The specification defined the entities, relationships, institution profiles, historical depth, distributions, edge cases, and validation criteria the data had to support.
- ✓ Magic then built a deterministic, multi-institution test-data platform with separate SIS and LMS interfaces generated from one connected data model. The environment supported transactional API testing as well as bulk columnar snapshots for analytics ingestion.
- ✓ Automated checks validated responses against official source-system schemas, tested cross-table and cross-system relationships, and compared exported files with the underlying database. These checks ran as build gates, allowing defects to be caught before they reached downstream analytics.

Build failures were raised when data did not conform, rather than being left for downstream teams to discover.

For more details, visit: www.magicedtech.com



Key Result Highlights

0 student records processed in the synthetic generation environment.

A **10,000**-student institution with **500K+** records provisioned in approximately 2 mins on modest infrastructure.

100s of SIS interfaces and dozens of LMS tables supported within one reusable environment.

Automated schema and relationship validation applied to every build.

20+ unresolvable reference fields corrected and verified against freshly generated data.

Dozens of additional latent defects detected by a generalized regression control before downstream reporting.

Every downstream integrity alert raised during the engagement was traced with reproducible evidence, allowing engineers to focus on the true cause.

A repeatable foundation established for product development, integration testing, demonstrations, and future data-fidelity work.