Section 1: Week 2: Organizational Data Management Problems

Nate Bachmeier

TIM-7020: Database and Business Intelligence

December 14th, 2019

North Central University

EdTech Data Management

Educational institutions need to maintain student rostering information, such as the class enrollments and the instructional hierarchy of their courses. In some instances, these institutions must associate their rosters with parent organizations, such as county and state districts. Specific aspects of these hierarchical structures are made available to various third-party providers that use these feeds to expose an ecosystem of personalized student learning, teacher dashboarding, and classroom analytical tooling.

# Challenges of Data Sharing

Innovating within the Education Technology (EdTech) ecosystem is challenging because of non-uniform exchange protocols, insufficient information governance, and inefficient synchronization strategies. These issues complicate data management solutions and force businesses to design solutions that operate in a heterogeneous environment that cannot adequately address the needs of its audience.

## Use of Proprietary Protocols

Student enrollments are not static, and mechanisms need to exist for notifying partners of these changes. These notifications started life as proprietary unstructured messages that lacked a focus on interoperability. The Instructional Management Solutions (IMS), an international standards organization, has proposed OneRoster to standardize the communication protocol (IMS, 2019). However, these efforts need broader adoption by the tool ecosystem and more strict enforcement of data values. Many schools degrade the protocol to semi-structured messages because they lack sophisticated data quality processes (Herald, 2016). Herald provides an example that different systems might know the same student as Jim, Jimmy Smith, and James. Another challenge arises from the reuse of student numbers (primary key) that might be unique within a given school (local context) but overlap at the district level (global context).

## Insufficient Privacy and Security Controls

Parents, administrators, and legislatures have concerns around their privacy of student’s information (Regan & Jesse, 2019). Regan and Jesse state that federal regulators oversee sharing between organizations using powers granted under the Family Educational Rights and Privacy Act of 1974 (FERPA), Children’s Online Privacy Protection Act of 1998 (COPPA), and related laws. “The central tension in Edtech is between the need to protect student data privacy on the one hand, and Edtech companies’ ability to innovate on the other (Peterson, 2016, p. 962).” Peterson’s statement is missing the critical point that students have entrusted their data to their schools explicitly, and the EdTech implicitly. These additional protections necessitate an information governance policy that is natively part of the roster sync protocols, at both a coarse (e.g., entity filters) and fine-grained (e.g., attribute filters) level of control.

## Inefficient Synchronization Protocols

A third aspect of the problem is an inability to perform data integrity checks and efficiently remediate in real-time. Many third-party consumers handle this drift through a full periodic synchronization with the source of truth. However, this can be challenging to scale across large customer bases with millions of students. The delays also reduce the customer experience, as students cannot immediately enjoy the third-party products (ClassLink, 2018). Specific sources of truth providers expose proprietary protocols for streaming update deltas (Clever, 2019). These solutions introduce external system dependencies, such as Lightweight Directory Access Protocol (LDAP), that can be clunky for specific workloads.

# Evaluating Solutions

Similar to other brick and mortar establishments, school districts are slow to adopt change, and EdTech businesses need to find solutions that minimize these data management constraints.

## Data Store Selection

There are several different Storage as a Service (StaaS) solutions available to host the student data, each with a unique workload optimization point (Mansouri, Nadjaran, & Buyya, 2017). Due to a lack of schema standardization between organizations, a NoSQL store would be more appropriate than a traditional Relational store. In a relational system, such as Postgres or Microsoft SQL Server, rely on ‘schema on write’ protocols, which would work for a subset of organizations (Zambetti, Pinto, & Pezzotta, 2019). Instead, key-value and document stores can accept the opaque third-party payloads and leverage ‘schema on reading’ strategies at a later stage in the pipeline.

Different abstraction layers exist across NoSQL stores for simplification of data modeling and access patterns. If the business questions start from *known* entities*,* then a graph database might be the right choice (Kronmueller, Chang, Hu, & and Desoky, 2018). However, a more exploratory system that begins from *unknown* objects might select an in-memory store instead (Knabke & Olbrich, 2018). Consider the difference between (a) find all students in Ms. Allen’s math class and (b) all third-grader girls with a low grade in history. **Scenario (a)** begins at Ms. Allen (vertex) and then traverses a relationship (edge) to their students (vertices). While a relational store would need to use globalized table indexes, a graph database can efficiently use local vertex indexes to find the related objects (Patil, Kiran, Kavya, & Naresh, 2018). These optimizations enable the system to scale to higher degrees of complexity and more massive data sets. However, these local indexes can introduce additional costs when the starting point is unknown, and many teachers (initial vertices) considered. **Scenario (b)** addresses these challenges by reversing the index and mapping keyword terms to documents. Now, the query ‘all third-grade girls’ decomposes into the union of record identifiers that are associated with the terms ‘gender=female’ and ‘grade=3.’ Then each item can be efficiently retrieved using their primary-key value.

## Information Governance

Administrators, parents, and students expect their data remains private and secure on third-party systems. These assurances require information governance policies that focus on security and access control (Taleb et al., 2018) (McKendrick, 2019). Taleb et al. propose process-driven strategies that clean incoming data and remove sensitive details. Consider the scenario where XYZ school includes student email and physical addresses in their roster. If ABC EdTech does not support physical mail, then those attributes should be truncated. Governance should also consider data retention periods and remove data that not needed, to further reduce the attack surface.

McKendrick expands on these ideas by demarcating the technology stack into a Data Lake and Data Warehouse. They propose leaving the source data inside of the Data Lake and then constructing Extract Transform and Load (ETL) workflows to move normalized structure into the Online Analytical Processing (OLAP) Data Warehouse. Afterward, industry-standard patterns exist for employing explicit row and column level security across OLAP technologies (Amazon, 2019) (Microsoft, 2019) (Oracle, 2017).

While McKendrick’s strategy provides a clear separation of *internal* and *external* data, it trades explicit security decisions for an increase in complexity. Exposing new attributes through the OLAP system requires changes to multiple aspects of the data pipeline. It can be problematic to scale the access management policy as the (a) number of partners and (b) the level of granularity increases. Another critical concern is this approach requires moving to a relational model that does not align with the business use cases. Instead, a security solution needs to exist across the NoSQL store directly. Network administrators on the Amazon Web Service (AWS) cloud can configure such a model using DynamoDB and IAM Security Roles for similar document and attribute level protections (Amazon, 2019). Microsoft Azure offers the same experience through its CosmoDB and DocumentDb products. Both Public Cloud Solution Providers (CSP) also have integrated solutions for encryption and access auditing.

## Heterogeneous Replication

An interesting problem arises with the need to keep disjoined organizations in sync across various technology stacks that partially trust each other. When a remote replica is missing a segment of student data, then those students are impaired and unaccounted in reporting platforms. The naïve solution is to perform a full synchronization on a configured cadence. However, for specific scenarios, such as freemium models, these discrepancies are unacceptable delays to the business model (Eaton, 2012).

An alternative mechanism could pull data during a lifecycle event, such as when a user authenticates into the system (Freckle, 2019). This approach can lead to questions from administrators why aggregate entity counts are inconsistent and fluctuate across the school year. A more efficient strategy can send push notifications between the systems when a change has occurred. Hierarchical structures should align these announcements with the branch structure (Kvet & Matiasko, 2019). For example, when a student enrolls in a course, then subscribers should receive notifications for that object in addition to the associated school. This strategy allows for third-parties to register for events at the level of granularity that is relevant to their product. The message to the receiver should contain object pointers, instead of the value directly to conserve network resources between organizations (Steen & Tanenbaum, 2016).

An assumption with notification strategies is the subscriber is highly-available and highly-reliable. If (a) an extended outage occurs or (b) an event that resulted in data corruption, then the two organizations will be out of sync. After restoring to an arbitrary point in time, a process needs to verify each segment of the local replicas is consistent. A novel management strategy could rely on Merkle trees to provide checksums over the hierarchical rostering data, similar to solutions found in peer-to-peer networks (Kan & Kim, 2019) (Furtado, 2005). For instance, the validation checksum for a school equals hash(hash(courses)), with the hash(course) equal to hash(hash(associated users)). This protocol has very low overhead and is capable of scaling to enormous datasets, allowing for fine-grained detection of errors.

# References

Amazon. (2019). *Restricting Access to a Data Set by Using Row-Level Security*. Retrieved from AWS Documentation: https://docs.aws.amazon.com/quicksight/latest/user/restrict-access-to-a-data-set-using-row-level-security.html

Amazon. (2019). *Using IAM Policy Conditions for Fine-Grained Access Control*. Retrieved from Amazon DynamoDB: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/specifying-conditions.html

ClassLink. (2018, May 14). *ClassLink OneSync*. Retrieved from YouTube: https://www.youtube.com/watch?v=HdxfEgqaS-w

Clever. (2019, December). *Clever Secure Sync*. Retrieved from Clever: https://clever.com/products/rostering

Eaton, K. (2012, March 15). *How One Second Could Cost Amazon $1.6B In Sales*. Retrieved from Fast Company: https://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales

Freckle. (2019). *Freckle Plans*. Retrieved from Freckle EdTech: https://www.freckle.com/plans/

Furtado, P. (2005). Flexible Networked Data Management on P2P. *16th International Workshop on Database and Expert Systems Applications (DEXA'05)*.

Herald, B. (2016). New Tech Standard Aims to Ease Sharing Of Digital Roster Data. *Education Week Volume 35, Issue 19*, 8.

IMS. (2019, December 14). *Introduction to OneRoster 1.1*. Retrieved from IMS Global: https://www.imsglobal.org/oneroster-11-introduction

Kan, J., & Kim, K. (2019). MTFS: Merkle-Tree-Based File System. *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) Blockchain and Cryptocurrency (ICBC), 2019 IEEE International*.

Knabke, T., & Olbrich, S. (2018). Building novel capabilities to enable business intelligence agility.

Kronmueller, M., Chang, D., Hu, H., & and Desoky, A. (2018). A Graph Database of Yelp Dataset Challenge and Using Cypher for Basic Statistics and Graph Pattern Exploration.

Kvet, M., & Matiasko, K. (2019). Data Management in Hierarchical Database - Branches Notification. *PROCEEDING OF THE 24TH CONFERENCE OF FRUCT ASSOCIATION*, 206-214.

Mansouri, Y., Nadjaran, T., & Buyya, R. (2017). Data Storage Management in Cloud Environments: Taxonomy, Survey, and Future Directions.

McKendrick, J. (2019). Data Lakes and Data Warehouses, Working Tandom. *Database Trends and Applications, October/November*, 4-8.

Microsoft. (2019). *Dynamic row-level security with Analysis services tabular model*. Retrieved from Microsoft Documentation: https://docs.microsoft.com/en-us/power-bi/desktop-tutorial-row-level-security-onprem-ssas-tabular

Oracle. (2017). *Enforcing Row-Level Security with Oracle Label Security*. Retrieved from Oracle Help Center: https://docs.oracle.com/database/121/TDPSG/GUID-72D524FF-5A86-495A-9D12-14CB13819D42.htm#TDPSG94446

Patil, N., Kiran, P., Kavya, P., & Naresh, K. (2018). A Survey on Graph Database Management Techniques for Huge Unstructured Data.

Peterson, D. (2016). EdTech and Student Privacy: California Law As a Model. *Berkeley Technology Law Journal Volume 31*, 961-997.

Regan, P., & Jesse, J. (2019). Ethical challenges of Edtech, big data, and personalized learning. *Ethics and Information Technology Volume 21*, 167-179.

Steen, M., & Tanenbaum, A. (2016). A brief introduction to distributed systems. *Computing. Oct, Vol. 98, Issue 10*, 967-1009.

Taleb et al. (2018). Big Data Quality: A Survey.

Zambetti, M., Pinto, R., & Pezzotta, G. (2019). Data lifecycle and technology-based opportunities in the new Product Service System.