Part 3: Reflection on Emerging Database Technologies

In the constant progress of our Student Management System (SMS), Phase by phase we get to see constant possibilities for more improvements on its scalability, performance and overall efficiency. Two trends' standouts to us specifically and these are: **distributed databases** and **machine learning-integrated (ML-integrated) databases.** Both of these trends can offer promising changes to our database and can be considered to do few implementations when possible.

Distributed Databases like **Google Cloud Spanner**, **Cockroach DB**, and **Amazon Aurora Global Databases.** For a SMS that constantly develops and evolves especially in a multi campus (like our school BU) with online tools, distributed databases ensure that student records, attendance logs, and course information are available anytime, anywhere. This is very useful specifically on real-time dashboards, mobile access for students, and faculty accessing academic records across different departments or campuses. Horizontal scaling also helps support high traffic during enrollment or exam periods which results on the database being less laggy or more functional during these periods But implementing this can result on **increased complexity**, especially in managing **data consistency**, **conflict resolution**, and **latency control** between nodes. It may also raise expenses and require a team with expertise in distributed systems to maintain reliability and security.

Now for ML-Integrated Databases like **Big Query ML** and **Oracle Autonomous Database,** these platforms allows us system developers to run machine learning Our SMS might use this to predict student performance if they are at risk or recommend future learning resources based on their previous education behaviors. The system, for example, could analyze attendance, quiz scores and submission patterns, generate proactive alerts for students in need of intervention, and help drive retention and success.

Although these systems typically operate with clean and well-annotated data and then require some sort of restructuring of the existing database schema. However,

to create such models, data science knowledge is required, and it may not be easily accessible by non-technical employees. And there are challenges to meaningful interpretation of model outputs for students and educators. All in all both models can be integrated but limitation should also be considered when trying to implement these new models.

References:

https://cloud.google.com/spanner

https://www.cockroachlabs.com/product/cockroachdb/

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html

https://cloud.google.com/bigquery-ml/docs

https://www.oracle.com/autonomous-database/