WHITE PAPER

DATA MODELING CHALLENGES IN CLOUD HOSTED DATABASES

Himanshu Gupta, *Technology Architect, Infosys*

Abstract

The cloud adoption in organizations has picked up the momentum in last few years and the recent COVID-19 pandemic has reinforced the need to accelerate digitization & cloud migration especially in the context of data warehousing and BI and so the importance of data modeling which is often overlooked as a component of cloud computing, has also gained thrust. However, there are associated data modelling challenges with cloud hosted databases and if we follow the best practices supported by data modelling tools, it can overcome these challenges and ease the migration journey.

Infosys®
Navigate your next

Need of Data modelling

Data modeling is one of the most crucial technique in the world of Data management and with more and more organizations migrating to cloud, Data Modelling on Cloud becomes an integral part in this journey.

According to Gartner, *"By 2022, 75% of all databases will be deployed or migrated to a cloud platform, with only 5% ever considered for repatriation to on-premises. New realities in the business environment have transformed data modeling from a luxury to a necessity. By investing in data architecture and data modeling, enterprises can increase their flexibility, lower the cost of data management projects and benefit from greater data consistency across the business through managing their business-relevant data models more effectively."*

Large enterprises deal with terabytes to petabytes of data and often have many diverse sources of structured, semi-structured and unstructured data. If the formats and definitions of these data sources are inconsistent and relationships not defined, the application of the data will produce inconsistent unreliable results. Data modeling transforms such diverse data sources into a common format and defines their relationships and dependencies thus maximizing their usefulness to the business processes to which they are being applied.

Introduction to Data Modelling

Data modeling is a process for defining the properties, logical inter-relationships and flow between different data elements involved in a certain business process. It accelerates data analysis within any enterprise, enhances processes including database design and/or application integration. As more and more organizations are migrating to cloud, an effective data modeling brings its own benefits to this shift. With data modeling, analysis of the data sets and attributes is done first in order to understand all the data in an enterprise. This makes it easier to decide what data should be sent to the cloud and what should be retained on-premise.

Data modelling gives a structure to move data to the cloud effectively.

While loading data onto the cloud hosted databases, organizations are normally focused on the following key features:

1. High-performance and high-availability
2. Data Classification
3. Data Confidentiality
4. On-demand data secure deletion/ shredding
5. Minimal configuration and administration
6. Faster migration with zero or minimal downtime

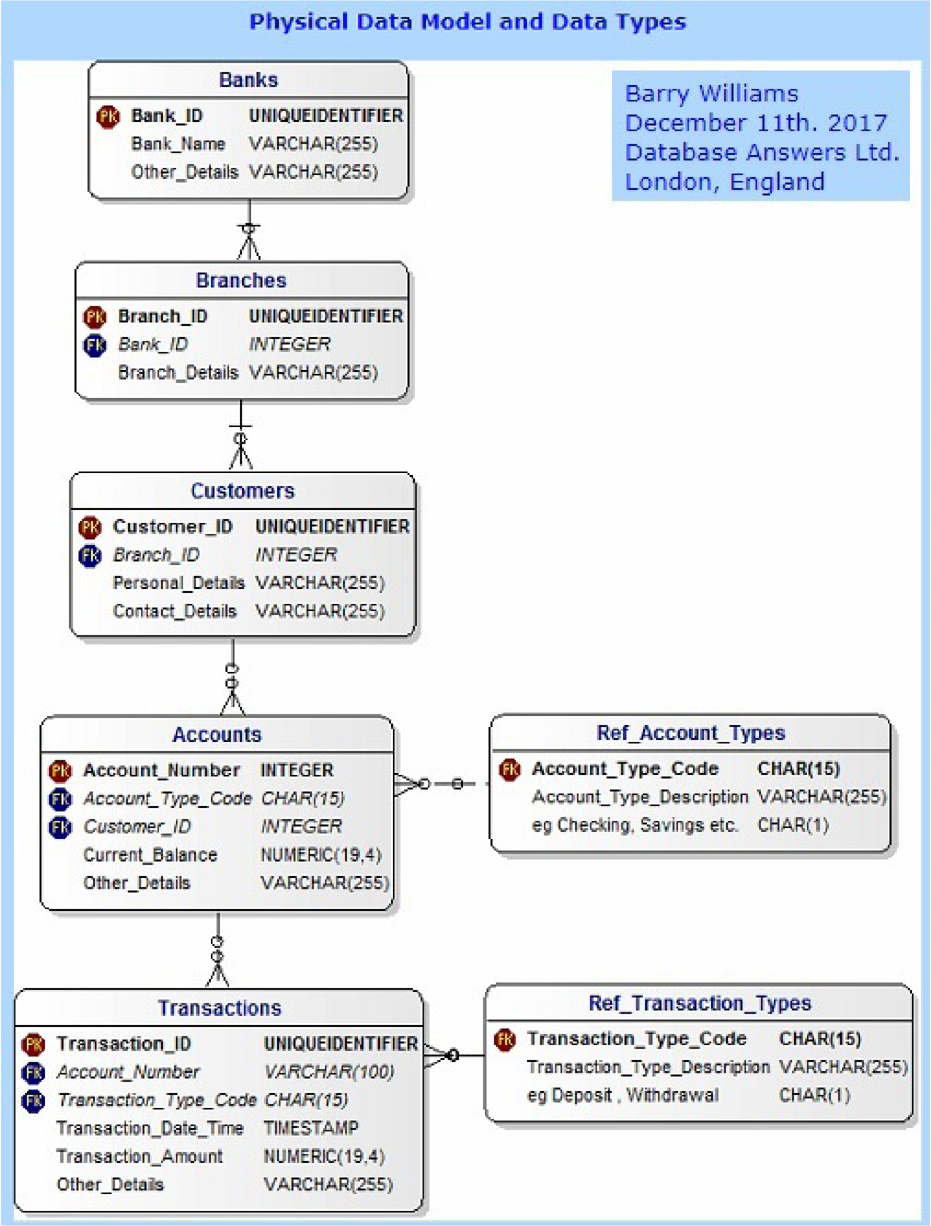
Physical Data Model and Data Types
Barry Williams
December 11th. 2017
Database Answers Ltd.
London, England
Banks
PK Bank_ID UNIQUEIDENTIFIER
Bank_Name VARCHAR(255)
Other_Details VARCHAR(255)
Branches
PK Branch_ID UNIQUEIDENTIFIER
FK Bank_ID INTEGER
Branch_Details VARCHAR(255)
Customers
PK Customer_ID UNIQUEIDENTIFIER
FK Branch_ID INTEGER
Personal_Details VARCHAR(255)
Contact_Details VARCHAR(255)
Accounts
PK Account_Number INTEGER
FK Account_Type_Code CHAR(15)
FK Customer_ID INTEGER
Current_Balance NUMERIC(19,4)
Other_Details VARCHAR(255)
Ref_Account_Types
PK Account_Type_Code CHAR(15)
Account_Type_Description VARCHAR(255)
eg Checking, Savings etc. CHAR(1)
Transactions
PK Transaction_ID UNIQUEIDENTIFIER
FK Account_Number VARCHAR(100)
FK Transaction_Type_Code CHAR(15)
Transaction_Date_Time TIMESTAMP
Transaction_Amount NUMERIC(19,4)
Other_Details VARCHAR(255)
Ref_Transaction_Types
PK Transaction_Type_Code CHAR(15)
Transaction_Type_Description VARCHAR(255)
eg Deposit , Withdrawal CHAR(1)

Figure 1 An example of a data model from databaseanswers.org

Components of an effective data model

1. **Entity-Relationship diagram:** An ER diagram is the blueprint of an application's foundations, showing a visual representation and connections between data sets and attributes. When designing a data model, it is important to think in terms of most common queries and data representation.
2. **Objects metadata:** Object properties and constraints are defined for all entities, attributes, identifiers and relationships describing the data model. This also plays an important role in measuring data quality.
3. **Business rules:** A good data model minimizes inconsistencies and inaccuracies in the data hence increasing the value when it comes to mine, gain insights and report on the data.

Data Modelling Benefits

- **Improved quality of application:** As explained above, data modeling provides a visual representation of the data sets and the relationships between data elements involved in a certain business process thus leading to better understanding of the business and its associated rules. Due to its visual nature, it enables efficient communication and collaboration between business users and subject matter experts ultimately leading to higher application quality.
- **Faster Time to Market:** With an effective data modeling, developers do not discover unknown requirements later in the lifecycle and can rather focus on development with minimal errors. This in turn leads to quick delivery of high-quality software.
- **Minimize cost of development and maintenance:** With good data modeling, we can find errors and inconsistencies early in the process and the earlier we find such issues, the easier and cheaper it is to fix them.
- **High Quality Data:** An effective data model always defines the metadata for the data which ensures that the developers and users have a better and common understanding of data. This enables the data to be effectively queried and reported on and provides the developers with a roadmap and checklist that allows them to identify and fix data quality issues and use high-quality data for business processes.
- **Improved performance:** Data modeling provides ways to better understand the database and improve its performance without the need to scan the code to discover the physical schema of the underlying objects.
- **Compliance to GDPR and Privacy of PII:** Organizations are required to prove compliance with privacy regulations on personally identifiable information (PII). To do so, they need to document the proper handling of associated attributes and maintain compliance. A good data model can support them in meeting such compliance.
- **Effective Business intelligence:** A good data model with strong foundations of data structure, query and reporting requirements, is an enabler for data mining. It identifies trends and patterns in the data and makes predictions to help a business make effective and informed decisions.

Data Modelling Challenges

1. **Data Classification:** While moving data to cloud, one of the major challenges that the organizations face in data modeling is to decide what data to move to cloud and this needs proper data classification rules to be defined and approved during the data modelling exercise. If not done early in the data model, it can really be a cumbersome process to be taken care at a later stage.
2. **Data Confidentiality:** Along with data classification, another important factor to be considered while moving data to cloud is data confidentiality and this becomes more critical if you are storing confidential data on the cloud.
3. **Data Quality:** Maintaining acceptable data quality by addressing data quality issues is the next big challenge. If the data originating at a source is of bad quality, everything in the rest of the data modeling process that is based on that data will have the cascading effect. Any decision or insights generated using that data will be inaccurate and will negatively impact the business. Hence, defining proper metadata in the data model is really important and it helps in understanding, cleaning, querying and reporting the underlying data correctly.
4. **Ensuring fitness for purpose:** Fitness for purpose means the data is correct and trustworthy for its intended uses according to the rules that govern it though the challenge is correct and trustworthy are very contextual terms. They vary based on the data and how it is being used. What seems to be bad data for one person could be good data for another. For example, bad sales data from a customer accuracy perspective is clearly bad for sales. But this same data might be ideal for implementing AI to identify challenges in the current sales process.
5. **Bringing legacy and modern application data together:** Another challenge to the data modeling process is to bring together data originating from legacy and modern applications. While migrating to cloud, organizations often start with the newer modern applications and leave the legacy ones for the last tranche. This becomes a challenge for data modelling exercise to bring together these two sets of data efficiently.
6. **Data Swamp rather than Data Lake:** As cloud storage becomes cheaper and reliable, organizations collect increasing amounts of data and store it on cloud. Without proper modelling of this data, they risk creating data swamp rather than data lake and can’t figure out how this data fits into their business process and how is it supposed to be used. Without understanding of this massive data, they tend to bring more data for each use case rather than cleaning the existing data and utilizing it.
7. **Understanding what the business cares about:**

   If a business cannot see the value of data, then it will not be motivated to curate it. Excel still remains the number 1 BI tool used by the business as well as IT teams in most financial institutions, which is a fundamental demonstration that quality is secondary to immediacy.

Types of Databases and associated data modelling tools

Databases are broadly classified into relational and non-relational (No-SQL) and No-SQL databases are further classified as Key-Value, Column oriented, Document and Graph databases. There are data modelling tools that work for specific database(s) and also generic ones that cover a larger variety of databases. With cloud computing, customers get a wide range of choice for relational as well as No-SQL databases. Data Modelling tools can play an important role to overcome many of the data modelling challenges on databases hosted on premise or on cloud. The Table 1 below covers a summary of these databases with examples of few data modelling tools that can be used for effective data modelling.

Table 1: Types of databases and associated Data modelling tools

Database Group	Characteristics	Examples of databases	Example of Data Modelling Tools	Few unique Features of tools
Relational	A relational database represents the data in the form of a table with rows and columns having a pre-defined schema. There are four properties that are associated with relational database transactions: Atomicity, Consistency, Isolation and Durability and are collectively referred to as ACID	Oracle, SQL Server, Teradata, Sybase, Azure SQL Database, Azure SQL Managed Instance, PostgreSQL, Google Cloud SQL, Amazon Relational Database Service, Azure MariaDB, Amazon Redshift, Snowflake	Erwin, Xplenty, Oracle SQL Developer Data Modeler, DBSchema	Visualization of any data from anywhere, Automated data model & database schema generation, Centralized model development & management, Increased data quality, Successful cloud adoption, Data literacy, collaboration & accountability
Non-Relational: Key-Value	As the name suggests, Key-Value databases store the data as key/value pairs. They are used as collection, dictionaries, associative arrays etc.	Redis, AWS (Amazon Web Services) DynamoDB, Riak, Azure Cosmos DB Table API (Application Programming Interface)	Hackolade	Visual data model navigation, Reverse- and forward-engineering of data models, Generate target-specific artifacts: schemas, DDLs, scripts, Produce human-readable documentation in HTML, Markdown, or PDF format
Non-Relational: Column oriented	Column-oriented databases work on columns and are based on Big Table paper by Google. Every column is treated separately. Values of single column databases are stored contiguously. These databases can provide high performance on aggregation queries like SUM, AVG, COUNT, MAX etc. as the data is readily available in a column. These databases are widely used to manage data warehouses, business intelligence, CRM, Library card catalogs etc.	HBase, Cassandra, Hypertable, Azure Cosmos DB Cassandra API, GCP (Google Cloud Platform) Bigtable	Hackolade	Visual data model navigation, Reverse- and forward-engineering of data models, Generate target-specific artifacts: schemas, DDLs, scripts, Produce human-readable documentation in HTML, Markdown, or PDF format

Non-Relational: Document	Document-Oriented NoSQL databases store and retrieve data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats. They are mainly used for CMS systems, blogging platforms, real-time analytics & e-commerce applications. They are not recommended for complex transactions requiring multiple operations or queries against varying aggregate structures.	Amazon SimpleDB, CouchDB, MongoDB, Riak, Lotus Notes, MongoDB, Azure Cosmos DB Mongo API, GCP Firestore in Datastore Mode	Moon Modeler, Hackolade, Adminer	Database modeling & schema design, Visualization of JSON structures and nested types, Reverse engineering, Support for database-specific settings, Three display modes: metadata, sample data or descriptions, Default values for newly created objects, Export to PDF, SQL script generation
Non-Relational: Graph	A graph type database stores entity as well as the relationship between those entities. The entity is stored as a node and the relationship as edge. Every node and edge have unique identifiers. Traversing relationship is fast as they are already captured into the database and there is no need to calculate them. They are a good fit for use cases like Fraud detection, Recommendation engines, Social networks etc.	Neo4J, Infinite Graph, OrientDB, FlockDB, Azure Cosmos DB Gremlin API	Hackolade Neo4j Data Modeling Tool	Reverse-engineer an existing Neo4j instance to derive the schema, Generates HTML documentation of the database schema, Supports several use cases to help enterprises manage their databases

How to tackle Data Modelling challenges?

Within the cloud database infrastructure, data models have an integral role to play and sometimes they can be referred to as "Model as a Service". Customers can use these models along with few guiding principles explained below to come up with effective data models for their business case.

1. **Shared Standard database models:** Implement standard data models early in the lifecycle of the project and share across teams using similar data so as to have common data understanding, definitions, metadata and models which facilitates data partnership and consistency across teams. When IT and Business teams have common understanding of data and work together, the data can be very efficiently curated and used for business processes. This collaboration with business along with maintenance of data quality parameters can be used to identify which data is fit for which purpose.
2. **Defining database model according to private and public Cloud requirements:** A data model defines features and properties for the cloud database services. For example, partitions, accesses, location, data tier and sensitivity are usually included into the data model. Use the properties in the models to easily classify the data as per the applicable rules and regulations and segregate between private and public cloud databases. This approach followed by regular data discovery, classification and model updates as per the upcoming changes is the way to keep the model live and useful. This also helps in complying with various regulations like GDPR and Data Privacy.
3. **Data Access Tier:** An efficient way to manage huge amount of data is to implement data access tier which segregates the data into Hot (frequently accessed), Cold (older data not used frequently but expected to be available immediately when accessed) and Archive (data that is rarely accessed) tiers based on the frequency of use. The data is then stored in different storage infrastructure as per the tier e.g. Hot data is stored in active premium disks, cold in separate commodity disks and archive in magnetic tapes or cloud blob storage archive tier. The data tier rules must be agreed and clearly defined in the data model itself so that the same concept is carried over to all phases of development and implemented consistently. Azure storage provides Hot, Cool and Archive tiers. AWS S3 Intelligent-Tiering delivers automatic cost savings by moving data between two access tiers — frequent access and infrequent access.
4. **Design the data storage model considering efficient query processing:** Create the data model with the aim of enabling efficient query processing against databases and minimizing cross database queries especially avoiding the need to bring together public and private cloud data in queries. Don't leave the legacy data migration to the last tranche rather try to bring it forward to avoid queries joining cloud and on-premise data together and leading to latency and performance issues. Databases like Azure SQL Database do not inherently support cross database queries though there are alternative ways to do that e.g. using elastic queries but that may have performance issues while processing huge amount of data. On the other hand, databases like Amazon Redshift, Azure SQL Managed Instance support such queries.
5. **Maintaining Data Quality:** The end user reports or insights are as good as the quality of the underlying data. Use the data discovery and analysis tools to identify, define and maintain the metadata in the data model and define rules for specifying the data quality parameters. Report the bad quality data with the supporting metadata/ quality parameters to the source systems as early as possible and request them to fix on priority. This will not only help in gaining better understanding of the data and maintaining data quality but will also save the system from turning into a data swamp. Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytic have in built Data Discovery & Classification features. Amazon Macie uses machine learning to automatically discover, classify, and protect sensitive data in AWS. Dataprep by Trifacta has data quality rules that suggest data quality indicators in GCP.
6. **Data Model to predict usage and optimize accordingly:** Performance and availability are two of the major reasons for organizations migrating to cloud. Availability requirement can be judged by factors like data usage frequency, usage duration, low and peak usage etc. which requires understanding of the data application life cycle followed by optimization of database service properties.
7. **Maintain Data Integrity:** Integrity (Completeness, Accuracy and Prevention of unauthorized changes to data) defined at the data model level should be maintained through the development lifecycle. Data classification in a data model must consider confidentiality, availability, and integrity parameters especially for cloud hosted databases. Services like Azure SQL Database, AWS S3 etc. provide integrity features which can be utilized to maintain integrity of data as defined in the model.

Conclusion

With many of the organizations moving to cloud and consuming ever-increasing volume of data, implementing an effective data model is a challenge but if planned properly and supported by appropriate data modelling tools on-time, effective data modelling can boost their migration journey and support the business with getting the right insights from the data that they collect and store.

Data Modelling started much before cloud era, continue to play an integral role in the world of cloud and is expected to be present and widely used even in future technology landscape

References

- https://rickscloud.com/data-modeling-in-cloud-computing/
- https://hackolade.com/benefits.html
- https://hackolade.com/nosqldb/neo4j-data-modeling.html
- https://searchdatamanagement.techtarget.com/feature/How-to-navigate-the-challenges-of-the-data-modeling-process
- https://www.softwaretestinghelp.com/data-modeling-tools/
- https://www.guru99.com/nosql-tutorial.html
- (PDF) Data Model for Cloud Computing Environment (researchgate.net)
- Retail Banking in the Clouds Data Model (databaseanswers.org)
- Data Modeling: A Necessary and Rewarding Aspect of Data Management (gartner.com)
- Data Modeling and Data Architecture; A Required Strategy for Enterprise Information Architecture (gartner.com)
- Gartner Says the Future of the Database Market Is the Cloud
- Moon Modeler - MariaDB Knowledge Base
- erwin Data Modeler | Industry-Leading Data Modeling Tool | erwin, Inc.
- An example of a data model from databaseanswers.org

For more information, contact askus@infosys.com

Infosys®
Navigate your next