Data Warehouse and Hive

Presented By: Shalva Gelenidze

Supervisor: Nodar Momtselidze

Decision support systems

- Decision Support Systems allowed managers, supervisors, and executives to once again see the clipboard with all its information.
- By 1992, Windows® 3.1 was in the stores and Ralph Kimball and Bill Inmon were figuring out how to gather data from two business areas and figuring out how to warehouse the data of an enterprise.

Data warehouse philosophy

A data warehouse is an asset of an enterprise and exists for the benefit of an entire enterprise. It does not exist for the benefit of a single entity (e.g., business unit, individual customer, etc.) to the exclusion of all others in the enterprise. As such, data in a data warehouse does not conform specifically to the preferences of any single enterprise entity. Instead, a data warehouse is intended to provide data to the entire enterprise in such a way that all members can use the data in the warehouse throughout its lifespan.

In the 1990s, Kimball and Inmon created and documented the concepts and principles of data warehouses, which today are the foundation of all data warehouses.

Decision Support Systems

- allowed managers, supervisors, and executives to once again see the clipboard with all its information

Principles:

- Subject Orientation: Data will be grouped by subject, rather than author, department, or physical location.
- Data Integration: Even though data comes from separate applications, departments, etc., differences should be smoothed out so they have the same look and feel.
- Nonvolatility: Unlike the data in operational applications, which is discarded once the company is finished using it, the data in a data warehouse will remain in the warehouse.
- Time Variant: All data has a context at a moment in time. A data warehouse will keep that context. So, all data from 1995 will retain its context within 1995.
- One Version of the truth: The proliferation of data in the 1980s and 1990s yielded many copies of the same data. Only the one, true gold, standard copy of each data element would be included in a data warehouse.
- Long-Term Investment: A data warehouse should be flexible enough to absorb changes in the company and the world, and scalable enough to grow with the company. By doing so, a data warehouse can add value to the company for a long time.

Ralph Kimball & Bill Inmon

- Ralph Kimball was a co-creator of the Xerox Star Workstation, the world's first commercially viable GUI application. Ralph was the founder and CEO of Red Brick Systems, the group which created an extremely fast RDBMS targeted specifically for data warehousing. When he authored The Data Warehouse Lifecycle Toolkit, Ralph introduced the Dimension-al Data Model.
- Bill Inmon was the creator of the Corporate Information Factory and Government Information Factory. In so doing, Bill also established many of the principles of Data Warehousing.

DW principles

Ralph Kimball & Bill Inmon Working separately arrived at a common set of guidelines and principles:

- **Subject Orientation** - Data will be grouped by subject, rather than author, department, or physical location. So, all manufacturing data goes together, and the sales data, and the promotions data, etc., regardless of where it came from.
- **Data Integration** - Even though data comes from separate applications, departments, etc., differences should be smoothed out so they have the same look and feel.
  - Form: When two data elements have different layouts, for eg. Telephone. One should be imposed on both of them.
  - Function: When two data elements identify the same thing with different names, it should be changed into one name
  - Grain: When two data elements apply different hierarchies (e.g., region and district) to the same thing, or different levels of detail (e.g., miles and feet), the two data elements will be resolved to the same level of hierarchy or detail.

DW principles

- **Nonvolatility:** Unlike the data in operational applications, which is discarded once the company is finished using it, the data in a data warehouse will remain in the warehouse.
- **Time Variant:** All data has a context at a moment in time. A data warehouse will keep that context. So, all data from 1995 will retain its context within 1995.
- **One Version of the Truth:** The proliferation of data in the 1980s and 1990s yielded many copies of the same data. Only the one, true gold, standard copy of each data element would be included in a data warehouse.
- **Long-Term Investment:** A data warehouse should be flexible enough to absorb changes in the company and the world, and scalable enough to grow with the company. By doing so, a data warehouse can add value to the company for a long time.

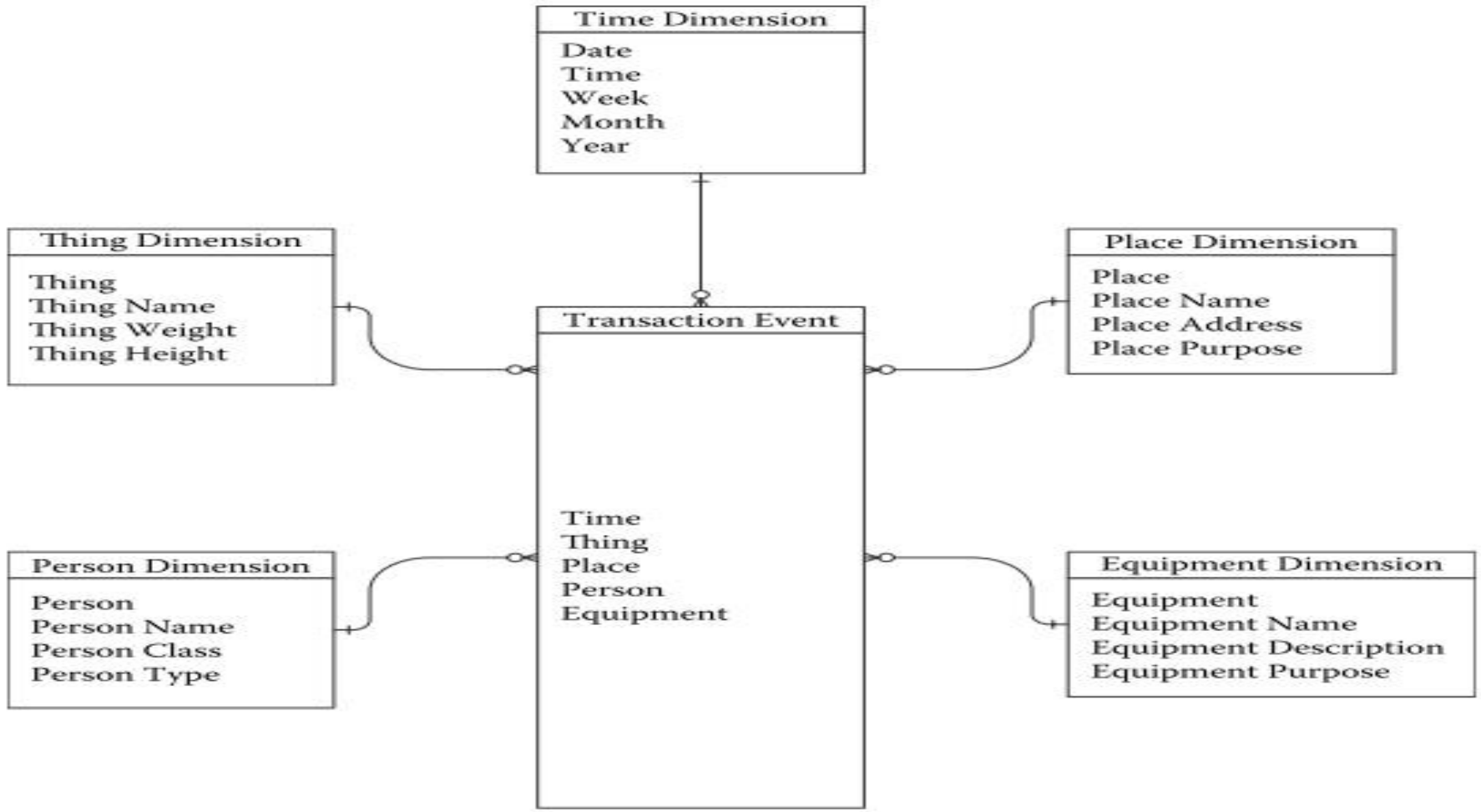
Dimensional and Third Normal Form Data Models

Kimball and Inmon arrived at the same set of principles, yet each used completely different designs.

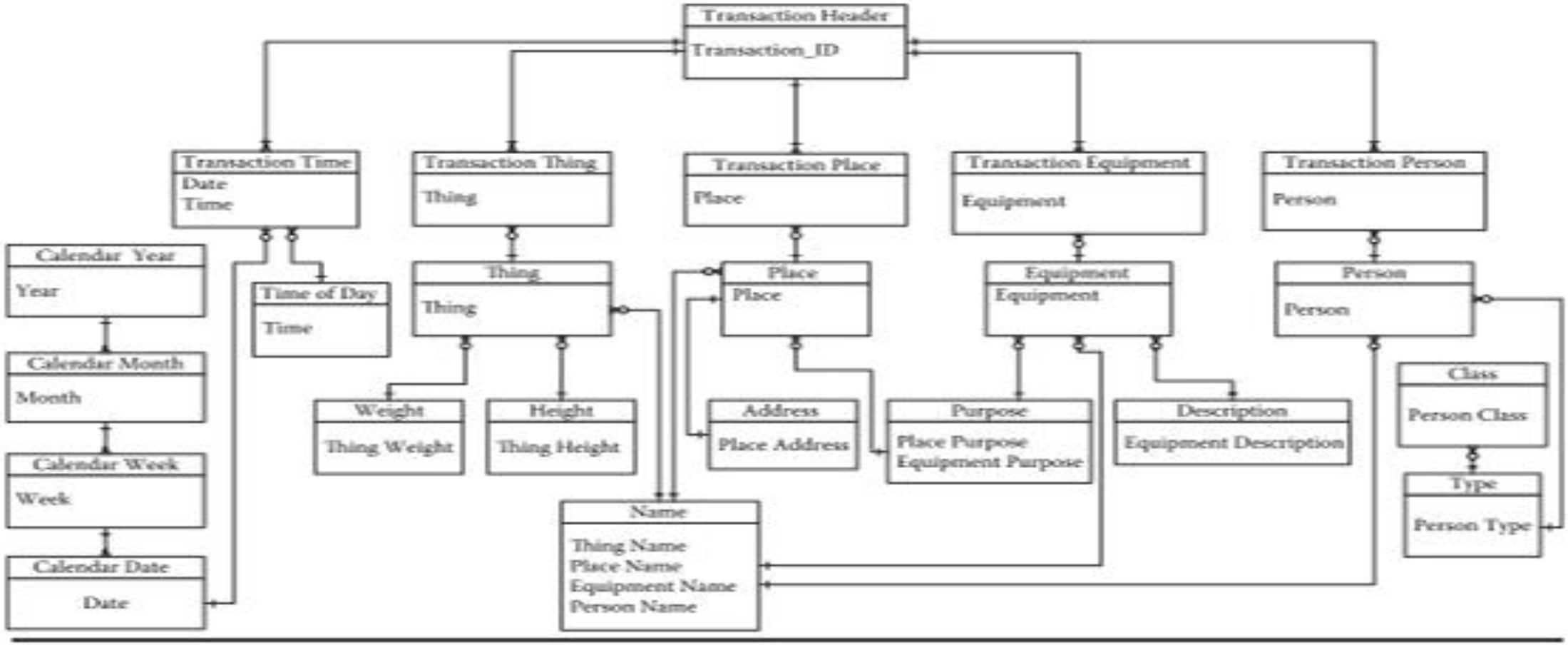
- Kimball created the Dimensional Data Model also known as a Star Schema because it looks like star. In the middle is a Fact table surrounding tables are dimensional tables
- Bill Inmon Created the Third Normal Form Data Model

Within the data warehousing community, a debate emerged. Which was better, the Dimensional Data Model or the Third Normal Form data model? By the twenty-first century, the answer was clear — both. Both designs had their strengths and their weaknesses. Rather than apply a “one size fits all” mindset, data warehouse designers learned to apply the strengths and avoid the weaknesses of both in each situation.

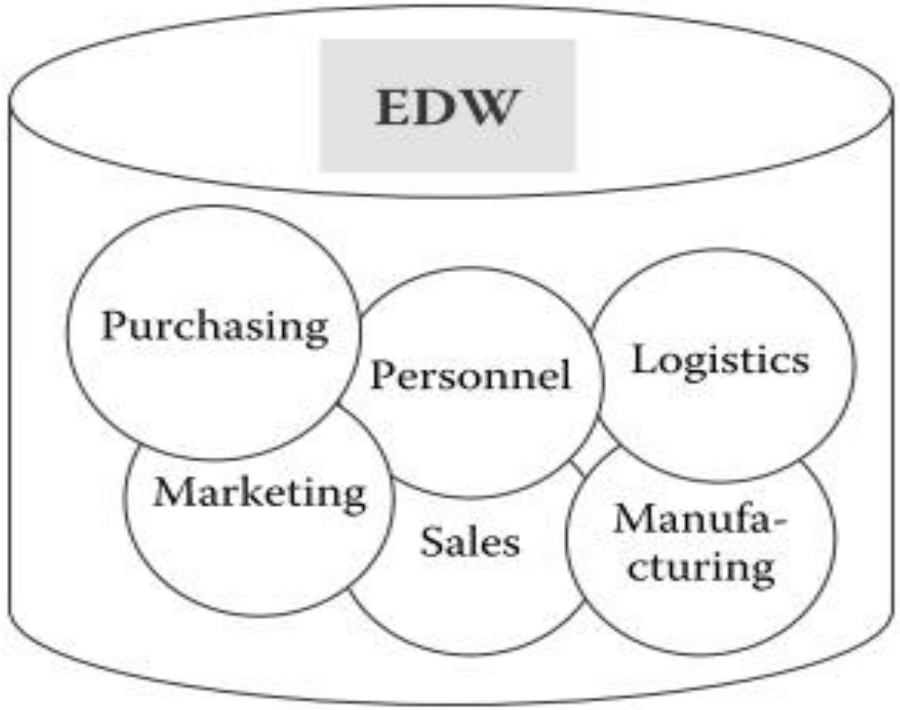
Dimensional model



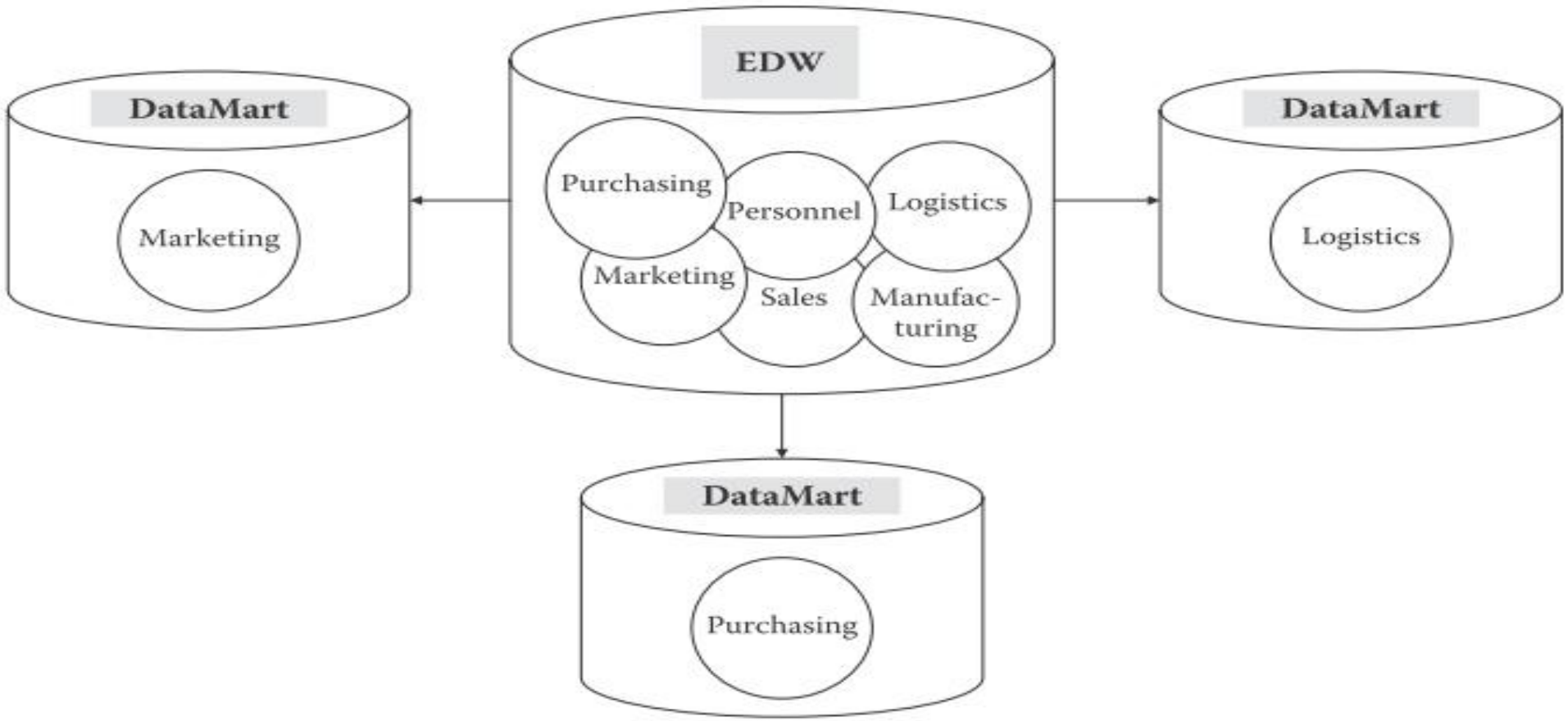
Third normal form data model



EDW model


EDW
Purchasing
Personnel
Logistics
Marketing
Sales
Manufa-
cturing

EDW and Data Marts



Data Warehouse

- **Data warehouse** (**DW** or **DWH**), also known as an **enterprise data warehouse** (**EDW**), is a system used for reporting and data analysis
- DWs are central repositories of integrated data from one or more disparate sources. They store current and historical data and are used for creating trending reports for senior management reporting such as annual and quarterly comparisons.
- The data stored in the warehouse is uploaded from the operational systems
- The data may pass through an operational data store for additional operations before it is used in the DW for reporting.

Source: Wikipedia

Data Mart

A data mart is a simple form of a data warehouse that is focused on a single subject (or functional area), such as sales, finance or marketing. Data marts are often built and controlled by a single department within an organization. Given their single-subject focus, data marts usually draw data from only a few sources. The sources could be internal operational systems, a central data warehouse, or external data.

Source: wikipedia

Data Warehouse and Hadoop

- Identify all possible enterprise data assets
- Select those that have actionable content and can accessed
- Bring assets into logically centralized data warehouse
- Expose data warehouse for decision making most effectively

RDBS

- Pros
  - Places data in well defined structure
  - SQL language
- Cons
  - Not capable of handling unstructured data
  - Can handle an data in MBs and GBs, performance goes down when data increases
  - RDBS cannot be scaled out (if data size increases you cannot add extra database server)

Big data

Big Data is a term applied to data sets whose size is beyond the ability of commonly used software tools to capture, manage, and process within a tolerable elapsed time. Big Data sizes are a constantly moving target currently ranging from a few dozen terabytes to many petabytes in a single data set.

Examples include Web logs; RFID; sensor networks; social networks; Internet text and documents; Internet search indexing; call detail records; astronomy, atmospheric science, biological, genomics, biochemical, medical records; scientific research; military surveillance; photography archives; video archives; and large scale eCommerce

Types of big data

- Structured data
  - Data is organized into entities that have a defined format, they conform to a particular schema.
- Semi structured data
- Unstructured data

Nosql summary

- 32bit MongoDB handles only 2GB of data and has 12 node limitations.

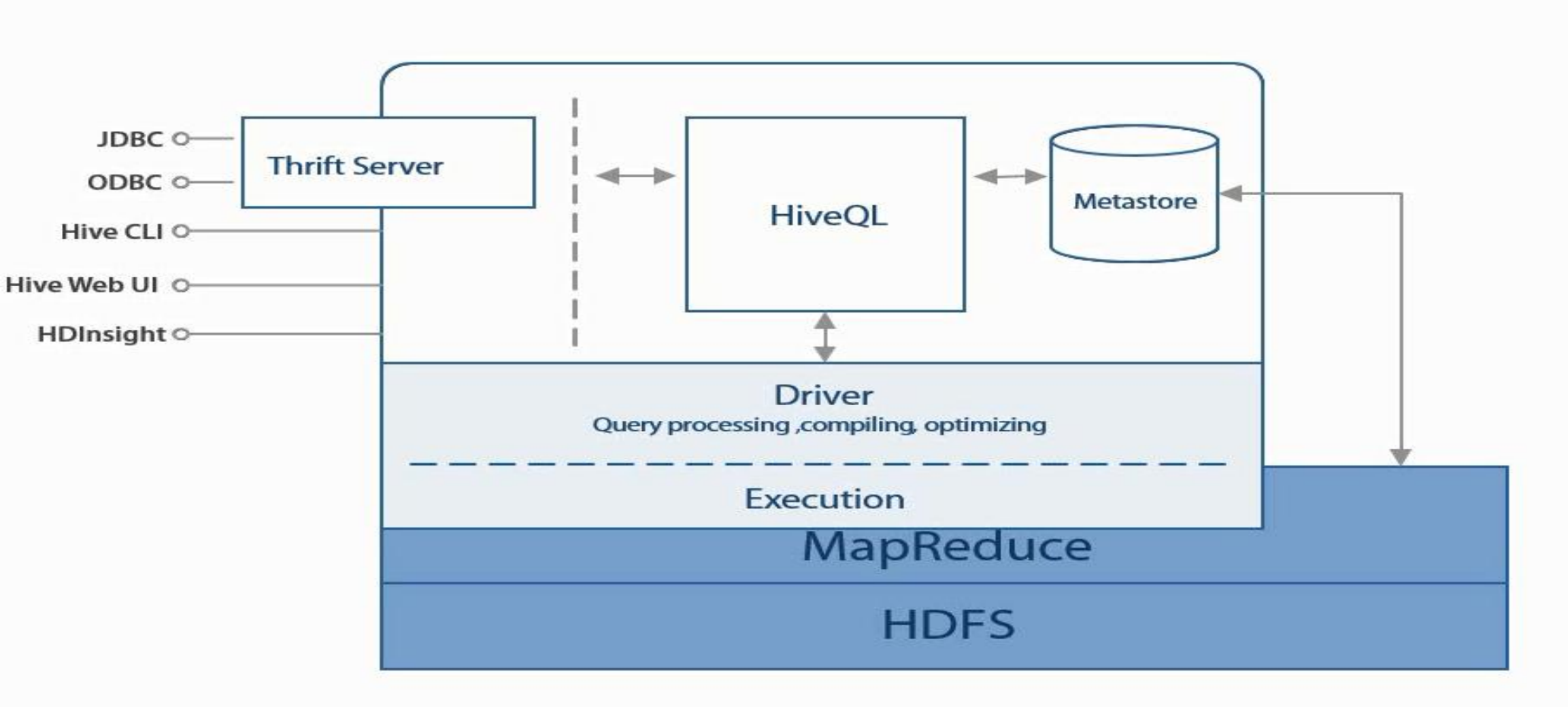
Is Hadoop a Data Warehouse???

- Mapreduce
- Hive
- Environment
- Distributed file system

Can we create data warehouse on Hadoop??

- Generating data
- Capture data
- Store data
- Analyze data
- Present results

Hive’s Architecture



Metastore

Hive’s architecture consists of a Relational Metastore which store data about (table definitions, location of data, data type info, how tables are partitioned …) it is relational database, by default hive installs derby database however it has limitations and that’s why hives metastore is installed in sql db.

Hive Query Language

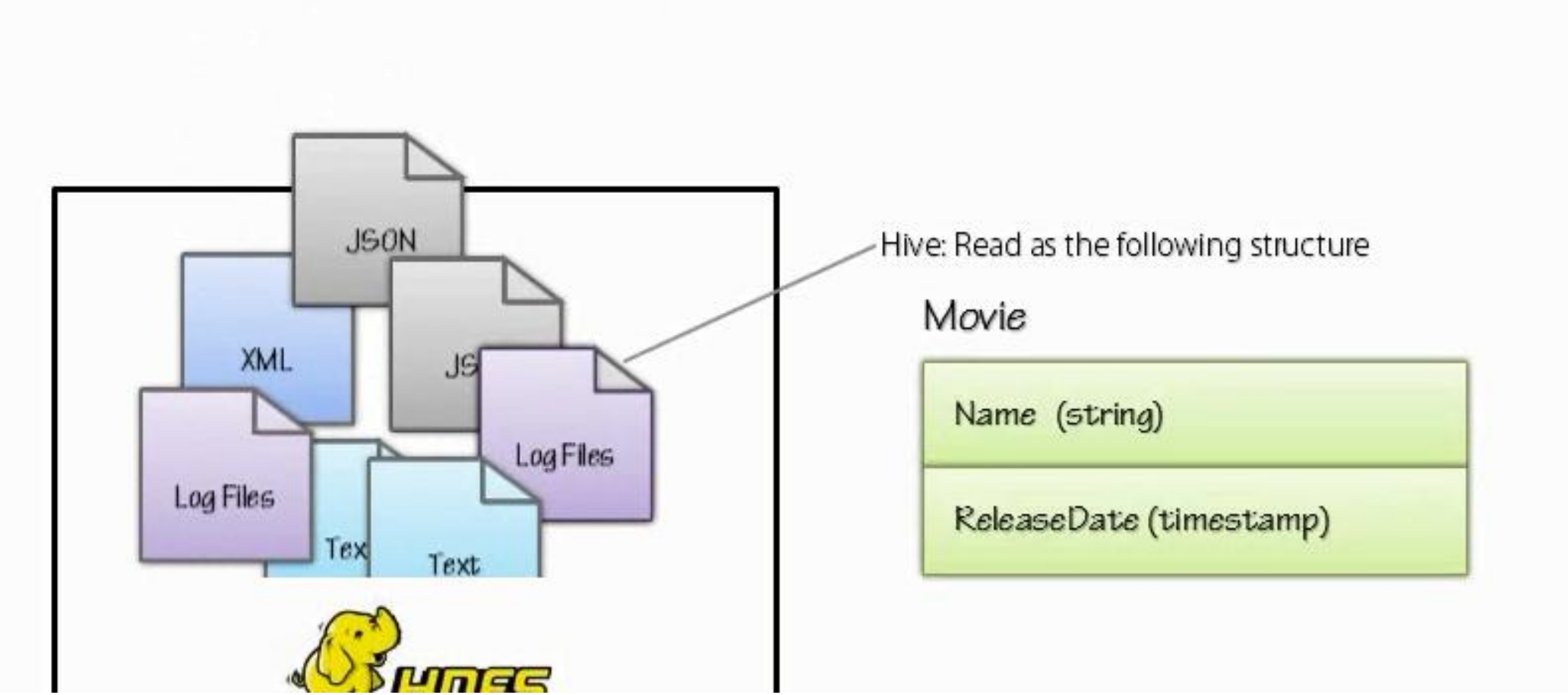
Interacts with Driver – responsible for Query processing, compiling, optimizing …

So hive is just an interface which allows you to write SQL like queries which hive than translates to mapreduce jobs and executes them.

Hive interpreter sits on user's machine and compiles HiveQL into MapReduce jobs then submits to the Hadoop cluster.

You can access hive using command line interface or any other interfaces as JDBC or ODBC, you can also use a web UI which provide web interface for hiveQL

Hive Principles



Hive Principles

In HDFS data is unstructured and hive adds some structure to it but this is assign when the data is read not as its written.

Hive warehouse

Consists of

- Meta data about all the objects known to Hive, persisted in meta store.
- Data consists of:
  - Databases
  - Tables
  - Partitions (tables are split into) based on the value of the column which determines where the data is actually stored. Helps to officially query the tables.
  - Buckets/ Clusters, partitions are divided into, based on the value of the hash function of column or set of columns. They have performance benefits.
- Local Hive warehouse – hive defines a location in HDFS which is marked as local Hive Warehouse, this is how hive distinguishes between locally managed tables and unmanaged external tables.
  - Managed by hive
  - Dropping the table will drop table as well as meta data
- External Tables –
  - Hive manages the meta data only
  - Hive can create tables and make some changes in it , but it does not assume the ownership of external tables.
  - Dropping the table will drop only a table definition the data remains in untouched.

HiveQL

SELECT

- Select *exp1, exp2, exp3 …* FROM *table* WHETE *condition* LIMIT *limitation*
- SELECT DISTINCT *col1, col2,col3 …* FROM *table;*
- SELECT *col1 + col2 AS col3* FROM *table;*
- SELECT ‘(ID|NAME)?+.+’ FROM *table ; - not regular java regex*
- FROM *table* Select *exp1, exp2, exp3 …* WHETE *condition; interchangeable select*
- *Interchangeable constructs*
- *Hive is not case sensitive*
- *Needs semicolon at the end of each expression*
- **Select** *col (***select** *cola + colb* **as** *col* **from** *some_table)* **subq***;*

HiveQL

- Hive lets as look at the data stored on hdfs as table perspective.
- Database in Hive is simple an abstraction to group tables together.
- /hive/warehouse – here is placed data if we define table and load data into it.
- /hive/warehouse/marketing.db – database is just metadata that defines a logical unit and it is another directory under hive directory. With same name of database, and ‘.db’ extension suffix to database name to differentiate that it is a database.
- /somewhere/on/hdfs – if you don’t want to save files in default location you can save them anywhere under hdfs folder.

HiveQL Create Database

CREATE (DATABASE|SCHEMA) [IF NOT EXISTS] database_name
[COMMENT some_comment]
[LOCATION some_location]
[WITH DBPROPERTIES (property_name = property_value)];
USE db_name;
DROP (DATABASE|SCHEMA) [IF EXISTS] database_name;

CREATE [EXTERNAL] TABLE (IF NOT EXISTS) [db_name] table_name
[(col_name data type]

Hive Create table

CREATE [EXTERNAL] TABLE (IF NOT EXISTS) [db_name]table_name

[(col_name data type [COMMENT col_comment], …)]

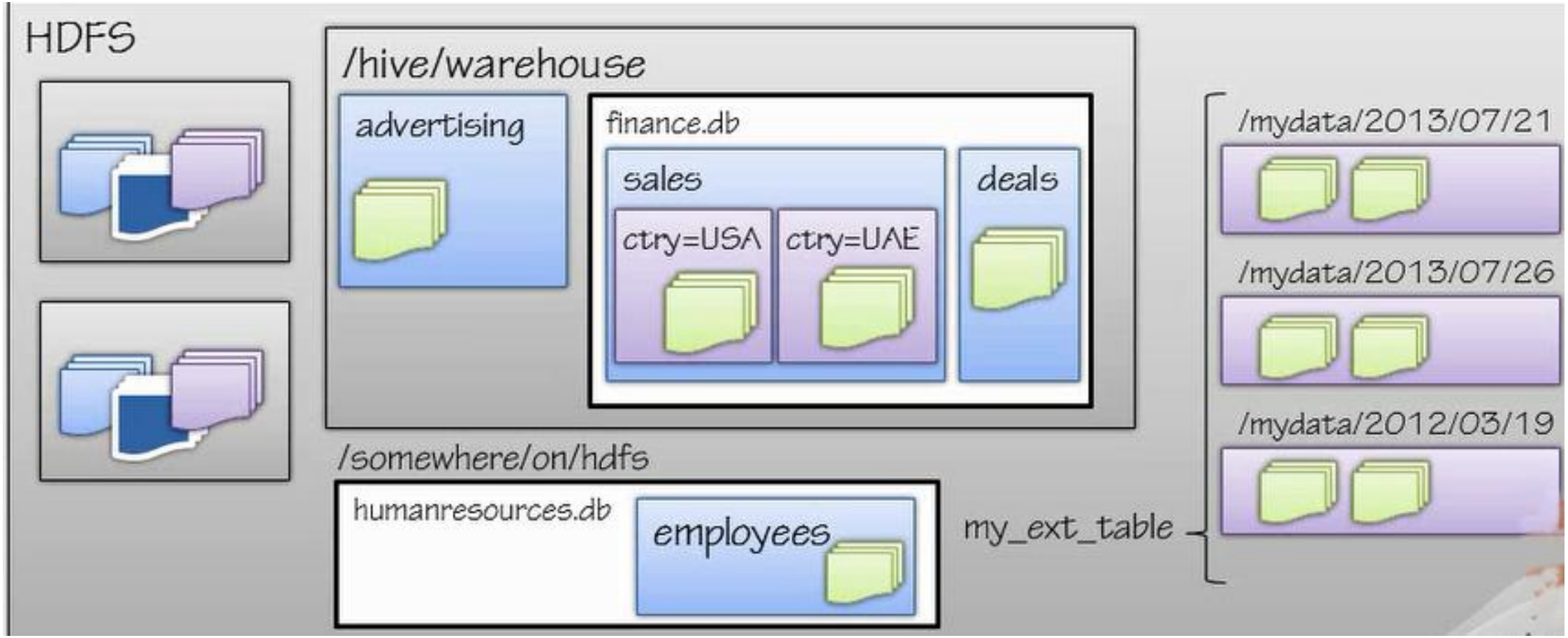
[PARTITIONED BY (col_name data type [COMMENT col_comment], …)]

[ROW FORMAT row_format][STORED AS file_format]

[LOCATION hdfs_location]

[TBLPROPERTIES (property_name = property_value, …)];

Tables - How they are stored



Primitive data types

- Numeric
  - TINYINT, SMALLING, INT, BIGINT
  - FLOAT
  - DOUBLE
  - DECIMAL
- DATE/TIME
  - TIMESTAMP
  - YYYY-MM-DD HH:MM:SS.FFFFFFFF
  - DATE
- MISC
  - BOOLEAN
  - STRING
  - BINARY

COMPLEX DATA TYPES

- ARRAY
  - ARRAY <DATA_TYPE>
- MAPS
  - MAP <PRIMITIVE_TYPE, DATA_TYPE>
- STRUCT
  - STRUCT <COL_NAEM:DATA_TYPE[COMMENT COL_COMMENT], …>
- UNION TYPE
  - UNIONTYPE <DATA_TYPE, DATA_TYPE>

Thank you!