HDFS-S3-Schema Design

> **Bemerkung**
>
> Struktur bezieht sich noch auf den Cloudera Cluster onpremis und muss im Detail auf die S3 Struktur angepasst werden

HDFS-S3-Schema Design

Hier wird das HDFS-Schema Design des CDP Hadoop Clusters beschrieben. Grundsätzlich hat das Hadoop Schema-On-Read Modell keine Anforderungen, um Daten in das System zu laden. Da der Hadoop Cluster jedoch als Data Hub für „**C**orporate**D**igital**P**latform" und deren betreuten Units dienen soll, ist eine minimale Strukturierung der Daten sinnvoll.
Vorteile:

- Einfacher Datenaustausch zwischen den Teams
- Einfache Rechte- und Quota-Verwaltung
- Einfache Wiederverwendung von Code

Location der HDFS Dateien

Grundsätzliches

/data	/dip	/<Unit>	/app	/<Prozessgruppe>	/<Datenbank>	/<HiveQL>
			/data	/<Prozessgruppe>	/	
			/rawdata	/<Quelle>	/<Datei>	/<Partition>
			/tmp	/<Prozessgruppe>		
			/warehouse	/<Datenbank>	/<Tabelle>	/<Partition>

Unit

Alle Rohdaten und Skripte, die zu einer bestimmten Unit gehören, z.B. „global", „welt", „bild", … für CDP Analysten, die das Globale Profil bearbeiten.

app

Das Verzeichnis enthält alles, was benötigt wird, um die Hadoop Applikation für die Unit laufen lassen zu können, außer Daten. Dies sind z.B. JAR-Dateien, Oozie Workflow Definitionen, Hive HQL-Dateien und mehr. Das Verzeichnis sollte für jede Ausführungsgruppe in Hadoop ein eigenes Unterverzeichnis haben, z.B.:

- /user/dip/global/**app**/dataflows/dip_stage_tl_global/tl_df_dip_tl_gl_code_stage.sql

Es sind folgende Verzeichnisse angelegt:

• Bin	hier sind u.a. .sh-Skripte abgelegt	invalidate_metadata.sh

• Dataflows	hier sind Dataflow-Verzeichnisse abgelegt	stage, cleanse, core, work, mart, mining, export .sql Skripte abgelegt (df_co_tl_visitor_s_event_dip_variabel.sql)
	Beispiel:	CREATE TEMPORARY FUNCTION from_json AS 'brickhouse.udf.json.FromJsonUDF'; SET hive.exec.dynamic.partition.mode=nonstrict; INSERT OVERWRITE TABLE dip_global_cleanse.cls_tl_visitor_h SELECT a.visitor_id AS DWH_ID, '${LOAD_ID}' AS DWH_CR_LOAD_ID, '${JOB_ID}' AS DWH_CR_JOB_ID, CURRENT_TIMESTAMP AS DWH_LOAD_DATE, MIN(a.FILE_NAME) AS DWH_LOAD_SRC, a.visitor_id AS DWH_BK_VISITOR_ID FROM dip_global_stage.stg_tl_eventstore a GROUP BY a.visitor_id,'${LOAD_ID}','${JOB_ID}',CURRENT_TIMESTAMP;
• Workflow	hier sind Workflow-Verzeichnisse abgelegt	wf_dip_global_stage, …_cleanse, …core, …mart01, …mining, …export job.properties, workflow.xml

rawdata

Das Verzeichnis enthält die Rohdaten, welche durch die Quellsysteme als CSV- oder Json-Datei zur Verfügung gestellt werden. Hier bestehen nur Leserechte, damit können diese Daten nicht verändert und/oder gelöscht werden. Der Zugriff erfolgt hier üblicherweise mit Hive Externe Tabellen.
Für jede Quelle wird ein eigenes Unterverzeichnis angelegt, sowie weitere Unterverzeichnisse als Partitionen je Struktur der Dateien aus dem Quellsystem, z.B.:

- /user/dip/global/**rawdata**/tealium/eventstore/file_date=2017-01-01

warehouse

Location der HIVE-Datenbanken. Eine HIVE-Datenbank ist nur ein Namensraum von Tabellen. Sie wird verwendet, um die Tabellen in einer logischen Verzeichnisstruktur zu organisieren. Damit können Konflikte mit Tabellennamen bei mehreren Teams vermieden werden. Man kann Tabellen mit demselben Namen in der HIVE-Datenbank erstellen, solange es verschiedene HIVE-Datenbanken sind. Auch für die Verwaltung der Sicherheit sind sie sehr nützlich.
Für jede Datenbank wird ein eigener Ordner als Unterverzeichnis angelegt. Dieser wiederum enthält weitere Unterordner für die Tabellen, z. B.:

- /user/dip/global/**warehouse**/dip_stage_tl_global/stg_tl_webmetrics_ext/file_date=2017-01-01

tmp

Temporäre Daten, die von Tools generiert oder von Benutzern gemeinsam genutzt werden. Dieses Verzeichnis wird typischerweise von einem automatisierten Prozess gereinigt und speichert keine Langzeitdaten. Dieses Verzeichnis ist in der Regel lesbar und beschreibbar, z.B.:

- /user/dip/global/**tmp**/project01

data

Daten in verschiedenen Stadien der Verarbeitung von Analysten. Das Verzeichnis ist für jeden Analysten les- und beschreibbar. Es hat für jede Benutzergruppe ein eigenes Unterverzeichnis, z.B.:

- /user/dip/global/**data**/project01