



Dr. Hegedűs Péter, Dr. Ferenc Rudolf

## Nagyméretű adatbázisok

Jelen tananyag a Szegedi Tudományegyetemen készült az Európai Unió támogatásával.

Projekt azonosító: EFOP-3.4.3-16-2016-00014

# SQL Hadoop felett

## Összefoglalás

Ez az olvasólecke a Hadoop ökoszisztéma azon eszközeit veszi górcső alá, amelyek SQL alapú lekérdezést és/vagy adatmanipulálást tesznek lehetővé a HDFS klaszteren tárolt adatfájlokon. Az olvasó áttekintést kap a következő eszközökről: Hive, Impala, Drill, Sqoop. Ezek alapvető célja és funkcionalitása mellett az olvasólecke kitér arra is, hogy miben hasonlítanak illetve különböznek ezek az eszközök, és hogy milyen esetben melyiket célszerű használni. Említést teszünk az egyéb olyan eszközökről is, amelyek Hadoop adatok fölötti strukturált lekérdezési támogatást nyújtanak, de nem SQL alapúak, mint például az Apache Pig.

A lecke fejezetei:

- 1. fejezet: **Apache Hive eszköz bemutatása (olvasó)**
- 2. fejezet: **Apache Impala eszköz bemutatása (olvasó)**
- 3. fejezet: **Apache Sqoop és Drill eszközök bemutatása (olvasó)**
- 4. fejezet: **Az egyes lekérdező nyelvet támogató eszközök összehasonlítása (olvasó)**

Téma típusa: **elméleti**

Olvasási idő: **45 perc**

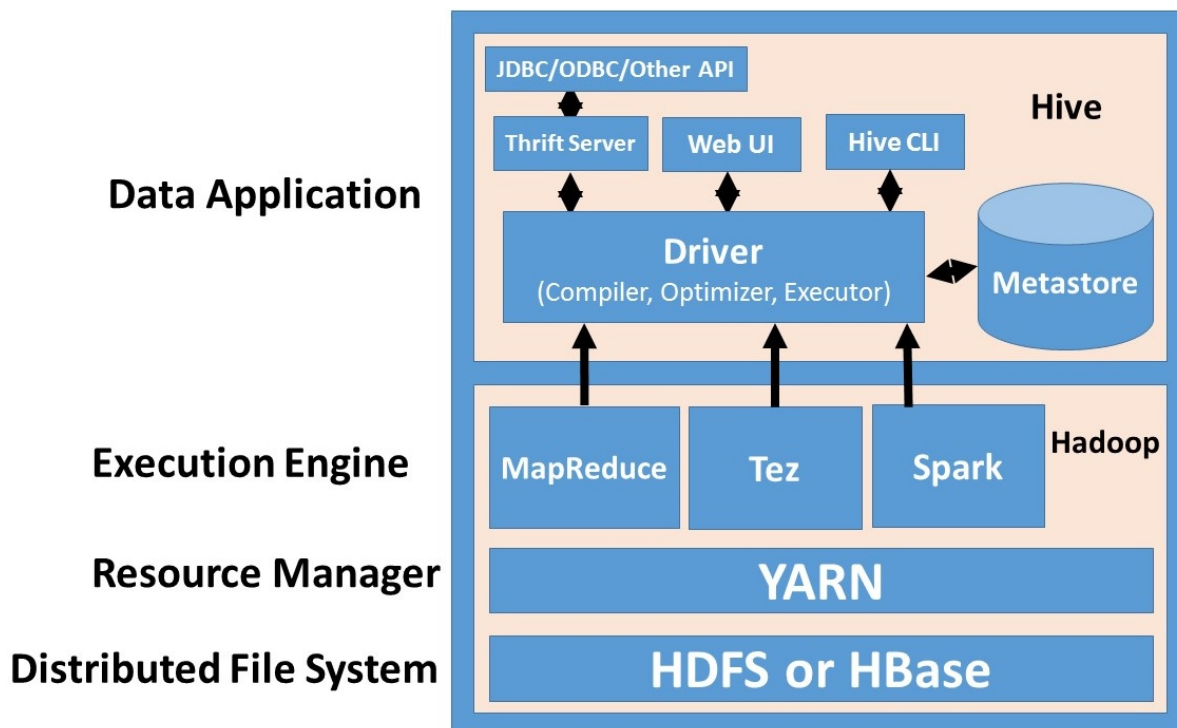
## 1. fejezet

### Apache Hive

Az Apache Hive [1] egy adattárház szoftver, amely elosztottan tárolt hatalmas adathalmazok SQL alapú lekérdezését, írását és kezelését támogatja. Az SQL által használt séma rávetíthető a már tárolt adatokra (tetszőleges strukturált szövegfájl). A felhasználók parancssori kliens vagy JDBC meghajtó segítségével kapcsolódhatnak a Hive-hoz.

A Hive az Apache Hadoop-ra építve (lásd lenti ábra) az alábbiakat nyújtja:

- Eszközök adatok könnyű, SQL-el történő elérése, ideális adattárház támogatáshoz (ETL), jelentések készítése, adat elemzés.
- Módszerek különböző adatformátumok adataira illesztéséhez.
- Fájlok közvetlen elérése a HDFS-en vagy HBase-ben.
- Lekérdezések végrehajtása MapReduce [3], Apache Spark [4] vagy Apache Tez [5] segítségével.
- Procedurális lekérdező nyelv HPL-SQL.
- Másodperc alatti lekérdezés végrehajtás Hive LLAP [6], Apache YARN [7] vagy Apache Slider [8] segítségével.



[https://1.bp.blogspot.com/-6VBHXXI96t4/WlJp8lQ2c6I/AAAAAAAAAPs/Nyfq02y87mEToUtfBDqt\\_8-GhPJgIPfkgCK4B/s1600/hive\\_architecture.jpg](https://1.bp.blogspot.com/-6VBHXXI96t4/WlJp8lQ2c6I/AAAAAAAAAPs/Nyfq02y87mEToUtfBDqt_8-GhPJgIPfkgCK4B/s1600/hive_architecture.jpg)

A **Metastore** egy különálló relációs adatbázis, amely a Hive adattáblák és partíciók leírását tárolja, és a kliensek számára hozzáférést nyújt ezekhez. A Hive két klienst is tartalmaz:

- **Hive CLI** - Hive parancssori interfész, amely közvetlenül kapcsolódik a HDFS-hez és a Metastore adatbázishoz, és csak azon a gépen használható, amely közvetlenül eléri ezeket a szolgáltatásokat
- **Beeline** - a HiveServer2 szerverhez kapcsolódik és egyetlen jar függőség (JDBC) szükséges a futtatásához: `hive-jdbc-<version>-standalone.jar`

A Hive standard SQL támogatást nyújt, a legtöbb újabb SQL szabvány követésével. A Hive által használt SQL nyelv bővíthető felhasználói programkóddal, ami az alábbiak egyike lehet:

- user defined functions (UDF)
- user defined aggregates (UDAF)
- user defined table functions (UDTF).

Nincs egy előre definiált "Hive adatformátum", amibe az adatokat tárolnunk kell, e helyett a Hive ún. csatlakozókat (connector) biztosít különböző adatformátumokhoz, pl. vessző/tab elválasztott érték (CSV/TSV) szövegfájlok, Apache Parquet [9], Apache ORC [10], stb. Részletekért lásd a hivatalos dokumentációt [11, 12], illetve a gyakorlathoz tartozó olvasóleckét.

A Hive-ot nem online tranzakciók feldolgozására (OLTP) tervezték, tradicionális batch alapú adatfeldolgozáshoz ideális. Skálázhatóságra, teljesítményre, bővíthetőségre, hibatűrésre és a bemenő adatformátumoktól való függetlenségre tervezték.

## Példa Hive használatára

Az alábbi példa azt mutatja, hogyan definiálhatunk egy adattábla struktúrát Hive segítségével, és hogyan kapcsolhatjuk azt hozzá egy HDFS-en lévő adatfájlhoz. Ezután standard SQL lekérdezésekkel hozzáférünk a fájl tartalmához.

```
$ /opt/hive/bin/beeline -u jdbc:hive2://localhost:10000
0: jdbc:hive2://localhost:10000> CREATE TABLE pokes (foo INT, bar STRING);
```

```

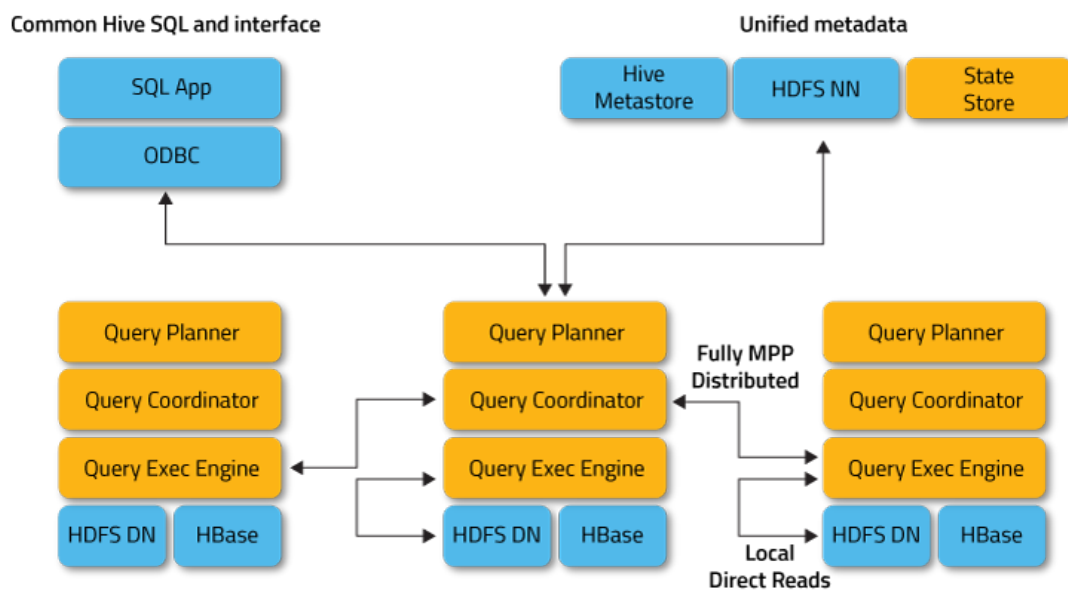
0: jdbc:hive2://localhost:10000> LOAD DATA LOCAL INPATH
'/opt/hive/examples/files/kv1.txt' OVERWRITE INTO TABLE pokes;
0: jdbc:hive2://localhost:10000> select * from pokes where foo>100 and foo<110;
+-----+-----+
| pokes.foo | pokes.bar |
+-----+-----+
| 103      | val_103   |
| 103      | val_103   |
| 104      | val_104   |
| 105      | val_105   |
| 104      | val_104   |
+-----+-----+
5 rows selected (0.12 seconds)

```

## 2. fejezet

### Apache Impala

Az Apache Impala [13] az Apache Hive komponenseit felhasználva (metastore, Hive SQL, ODBC driver, felhasználói felület) nyújt SQL alapú MPP (Massive Parallell Processing) interaktív adat lekérdezési lehetőséget HDFS-en vagy HBase-ben tárolt adatokhoz. A Hive-val ellentétben a valós idejű lekérdezéseket is támogatja. A késleltetés kiküszöbölése érdekében az Impala megkerüli a MapReduce-t, hogy az adatokhoz egy saját elosztott motoron keresztül férjen hozzá. Ezáltal egy nagyságrenddel gyorsabb lekérdezéseket tesz lehetővé, mint a Hive (konfigurációtól függően). Jelenleg az Impala a leghatékonyabb SQL motor a HDFS fölött. Az Impala architektúrája [14] a következő ábrán látható.



<https://impala.apache.org/img/impala.png>

Ez a megközelítés számos előnnyel jár a Hadoop adatok másfajta lekérdezéseivel szemben, többek között:

- Az adat node-okon való lokális feldolgozás miatt a hálózati átviteli korlátok kiiktathatók.
- Egyetlen, nyílt és egységes meta-adat tároló használható.
- A költséges adat formátum konverzió feleslegessé válik.
- Minden adat azonnal rendelkezésre áll, nem szükséges ETL folyamat.
- Az összes elérhető hardware felhasználásra kerül az Impala lekérdezések végrehajtásához, csakúgy, mint MapReduce esetén.

- Könnyű skálázhatóság.

## Példa Impala használatára

A következőkben egy példát mutatunk arra hogyan lehet egy CSV fájlt betölteni az Impala rendszerébe, hogy aztán SQL-lel lekérdezhetővé váljanak a benne tárolt adatok:

```
tab1.csv:
```

```
1,true,123.123,2012-10-24 08:55:00
2,false,1243.5,2012-10-25 13:40:00
3,false,24453.325,2008-08-22 09:33:21.123
4,false,243423.325,2007-05-12 22:32:21.33454
5,true,243.325,1953-04-22 09:11:33
```

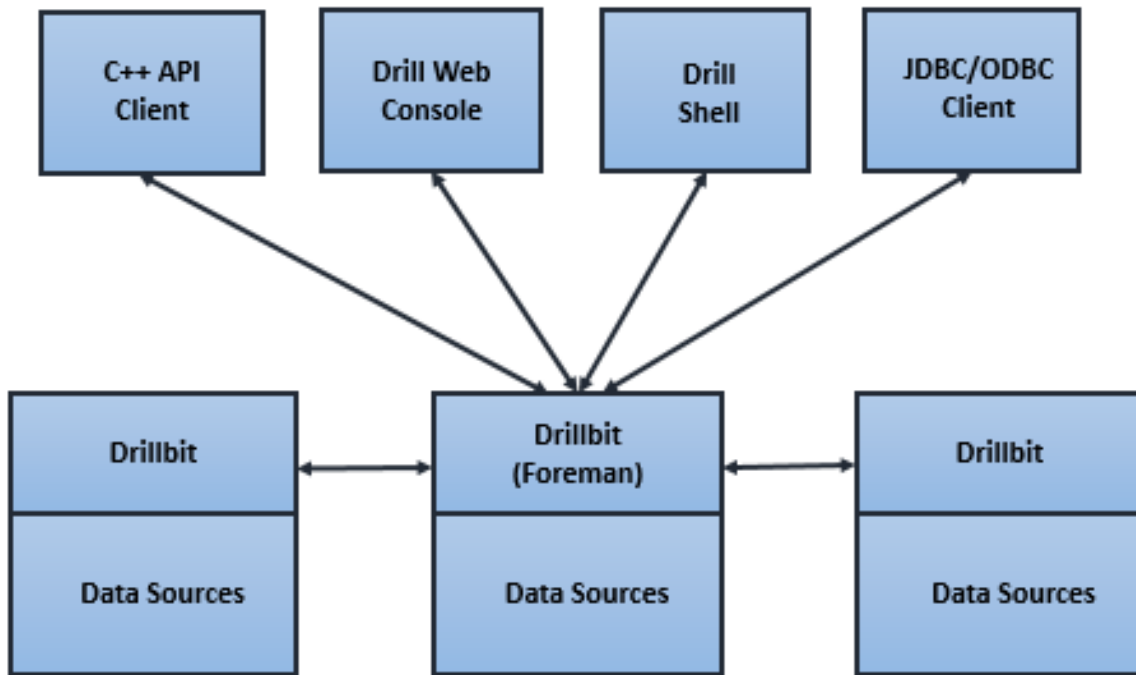
```
$ impala-shell -i localhost --quiet
Starting Impala Shell without Kerberos authentication
welcome to the Impala shell. Press TAB twice to see a list of available
commands.
...
(shell)
      build version: Impala Shell v3.4.x (hash) built on
      date)
[localhost:21000] > CREATE EXTERNAL TABLE tab1
[localhost:21000] > (
[localhost:21000] >   id INT,
[localhost:21000] >   col_1 BOOLEAN,
[localhost:21000] >   col_2 DOUBLE,
[localhost:21000] >   col_3 TIMESTAMP
[localhost:21000] > )
[localhost:21000] > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
[localhost:21000] > LOCATION '/user/username/sample_data/tab1';
[localhost:21000] > SELECT * FROM tab1;
+----+-----+-----+-----+
| id | col_1 | col_2      | col_3      |
+----+-----+-----+-----+
| 1  | true  | 123.123    | 2012-10-24 08:55:00 |
| 2  | false | 1243.5     | 2012-10-25 13:40:00 |
| 3  | false | 24453.325  | 2008-08-22 09:33:21.123000000 |
| 4  | false | 243423.325 | 2007-05-12 22:32:21.334540000 |
| 5  | true  | 243.325    | 1953-04-22 09:11:33 |
+----+-----+-----+-----+
```

## 3. fejezet

### Apache Drill és Sqoop

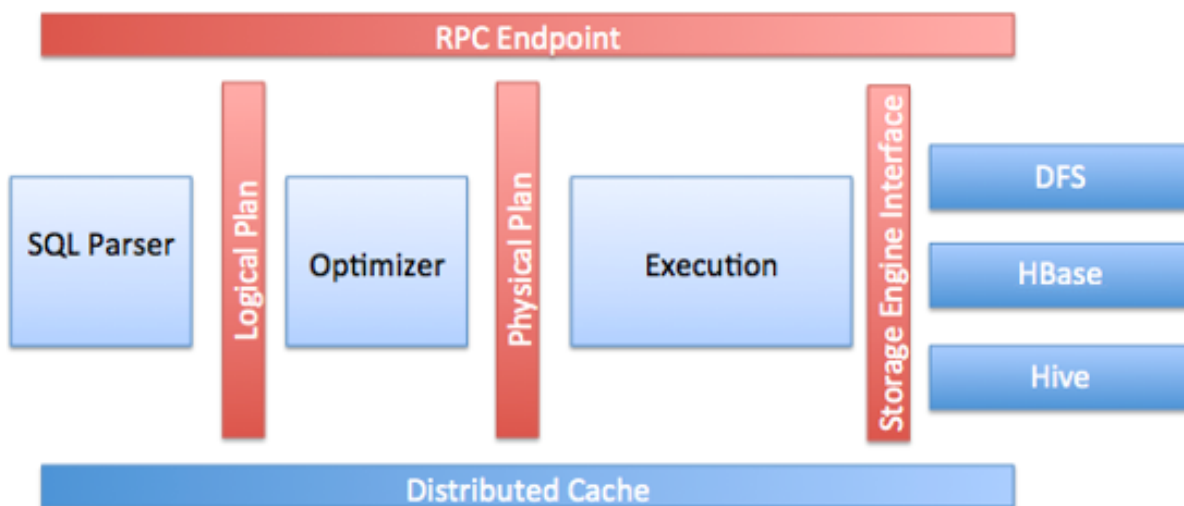
Az Apache Drill [15] egy nyílt forrású SQL lekérdező motor BigData feltárásához. A Drill-t az alapoktól kezdve félig-strukturált és gyorsan változó adatok hatékony analizésére tervezték, amilyeneket tipikusan a BigData alkalmazások állítanak elő. Mindezt a jól ismert és standardizált SQL nyelv segítségével. A Drill könnyen integrálható meglévő Hive illetve HBase környezetekkel. A Drill teljesen független a Hadoop-tól és minden egyéb infrastruktúrától, telepítés után tetszőleges lokális fájl (pl. egy séma nélküli JSON fájl) is használhatjuk. Nincs szükség az adatok sémájának definiálására, amivel az egyetlen ilyen elosztott SQL motor. A MongoDB [16] és Elasticsearch [17]

NoSQL adatbázisok által is használt séma nélküli JSON modellt használja. Lehetővé teszi több adatforrás összekapcsolását is, teljes SQL szabvány támogatással rendelkezik.



<https://drill.apache.org/docs/img/query-flow-client.png>

A Drill elosztott végrehajtási környezet (lásd fenti ábra) lelke az ún. "Drillbit" szolgáltatás, amely fogadja a kliensek felől érkező kéréseket, végrehajtja azokat és visszaküldi az eredményt. A Drillbit szolgáltatás telepíthető pl. egy Hadoop klaszter minden egyes csomópontjára. Ebben az esetben a Drill maximálisan kihasználhatja az adat lokalitást, azaz nem kell az adatokat átmozgatnia a hálózaton. A klaszter menedzsmenethez a ZooKeeper-t [18] használja, amely az egyetlen függősége, egyébként tetszőleges klaszteren képes működni, nem csak Hadoop-on, viszont a YARN-nal is integrálható (így nem kell minden csomópontra külön telepítenünk, a YARN elvégzi a deploy-t). Egy-egy Drillbit belső felépítését a lenti ábra mutatja.



<https://drill.apache.org/docs/img/DrillbitModules.png>

## Példa Drill használatára

Egy lokális JSON fájl lekérdezéséhez egyszerűen le kell tölteni a Drill csomagot, és a beépített klienssel már futtathatunk is lekérdezéseket. A klaszteren való telepítés részleteiről a következő URL ad bővebb információt: <https://drill.apache.org/docs/creating-a-basic-drill-cluster/>

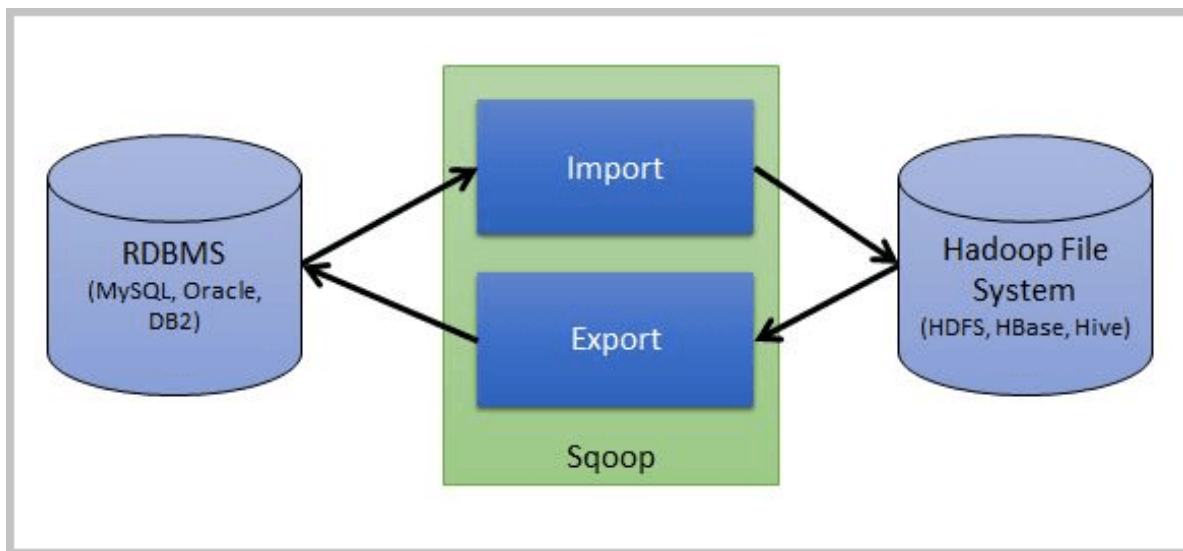
```

$ tar -xvf apache-drill-<version>.tar.gz
$ cp personal_entries.json /tmp
$ <install directory>/bin/drill-embedded
apache drill> select * from dfs.`/tmp/personal_entries.json`;
+-----+-----+-----+-----+-----+-----+
+-----+
|  PID      | name   | gender | last_contacted | birth_year |
useless_info |
+-----+-----+-----+-----+-----+-----+
+-----+
| 1122334455 | John Doe | M      | 2017-03-02T11:43:00 | 1987      | [1,2,3]
|           |         |         |                     |           |
| 2233445566 | Jane Doe | F      | 2019-10-12T15:22:34 | 1965      | [2,3,4]
|           |         |         |                     |           |
| 3344556677 | John Smith |         | 2020-01-05T08:27:12 | 1999      |
[1,2,3,4,5] |
+-----+-----+-----+-----+-----+-----+
+-----+
3 rows selected (2.473 seconds)

```

## Apache Sqoop

Az Apache Sqoop [19] eszköz egy SQL to Hadoop implementáció, azaz lehetővé teszi az adatok relációs adattáblából HDFS-re történő importálását, vagy pedig a HDFS állományok exportálását relációs adattáblába. Egy tipikus ETL eszköz, amely forrása vagy cél tárhelye egy relációs adatbázis illetve egy Hadoop klaszter HDFS fájlrendszere.



<https://www.hdfstutorial.com/wp-content/uploads/2016/07/Sqoop-architecture.png>

## 📖 4. fejezet

### SQL eszközök összehasonlítása

A fentiekben sok hasonló eszközt mutattunk be. Felmerülhet a kérdés, hogy mégis mi a főbb különbség ezen eszközök között, illetve ha lényegében ugyanazt a problémát tudják megoldani, akkor mégis mikor melyiket érdemes használni? Ezek bizony jogos kérdések, ezért ebben a fejezetben összefoglaljuk az egyes eszközök főbb közös vonásait, de leginkább azt, hogy miben térnek el egymástól, mert ez alapján egyértelmű, hogy milyen feladat megoldására melyiket célszerű választani.

- *Sqoop* - egy ETL eszköz, adatok mozgatására relációs adatbázis és HDFS között, nem pedig egy absztrakciós réteg a HDFS adatok fölé, amely SQL alapú lekérdezést tesz lehetővé. A Sqoop és a Hive/Impala közösen használható, pl. az adatok RDBMS-ből HDFS-be emelése után annak lekérdezéséhez Hive/Impala használható.
- *Hive* - egy plusz relációs adat leképező réteg HDFS fölé, amely lehetővé teszi a HDFS-en nem relációs adat sémában tárolt adatok SQL lekérdezését. Batch orientált (a lekérdezéseket MapReduce job-okká konvertálja), azaz elosztott módon, párhuzamosan és hatékonyan futtatja és optimalizálja a lekérdezéseket, de valós idejű/interaktív lekérdezésekhez nem alkalmazható.
- *Impala* - a Hive-val nagyrészt megegyező funkcionalitással bír, számos Hive komponenst maga is felhasznál és teljesen ugyanazt a lekérdező SQL-t használja, mint a Hive. Viszont a lekérdezéseket az Impala nem MapReduce job-okra fordítja, hanem saját motorral natívan hajtja végre, így alkalmas lesz valós idejű/interaktív lekérdezések végrehajtására.
- *Drill* - a Hadoop klasztertől független, ám azzal integrálható SQL alapú adatmanipuláló motor, ám nem használ sémát és alapvetően a félig-strukturált adatok (pl. JSON) lekérdezésére szolgál. Nem kell adat tábla sémát létrehozunk, hanem a félig-strukturált datainkat közvetlenül lekérdezzük vele.

Jól látszik, hogy a fenti eszközök bár hasonlóak, más-más célokat szolgálnak. Egyedül talán a Hive/Impala eszközöknek van nagy közös metszete, de nem real-time környezetben az előbbi, míg valós idejű, interaktív lekérdezések esetén utóbbi az egyértelmű választás.

Azonban ezzel természetesen nem ér véget az eszközök sora, számos másik olyan megoldás létezik, amelyek az adatok strukturált lekérdezését támogatják:

- *Kudu* - oszlop alapú adattárolás Hadoop fölött, amely strukturált adatok gyors lekérdezését támogatja. Használható együtt Impala-val (és hamarosan Hive-val is).
- *Spark* - egy memóriában tárolt gyors adat lekérdező keretrendszer, saját lekérdező nyelvel (SparkQL). Hadoop fölött is működik, de teljesen kiváltja a MapReduce-t (későbbi olvasó leckében bővebben is foglalkozunk vele).
- *Presto* - egy elosztott SQL lekérdező motor, amelyet a Facebook-nál fejlesztettek ki, és számos adatforrásból tud dolgozni. Integrálható Hadoop/Hive klaszterekkel vagy pedig a Cassandra NoSQL adatbázissal.
- *Pig* - szintén Hadoop feletti adat manipulációt nyújt (végrehajtó motorként MapReduce-t, Spark-ot és Tez-t is támogat), ám nem SQL alapú, hanem egy saját nyelvet definiál, a Pig Latin-t.

## ✓ Ellenőrző kérdések

1. Mire szolgál az Apache Hive, milyen fő komponensei vannak?
2. Mely job végrehajtó motorokat támogatja az Apache Hive?
3. Hány fajta kliens nyújt a Hive? Mi az alapvető különbség köztük?
4. Miben különbözik az Apache Impala a Hive-től?
5. Egy web-alkalmazásban, ahol a felhasználói kérések alapján közel valós időben kell HDFS adatokban lekérdezést végrehajtanunk, melyik eszközt használnád és miért?
6. Az olvasóleckében említett eszközök közül melyeket tekinthetjük klasszikus ETL eszköznek?
7. Mi az Apache Drill fő erőssége? Milyen típusú adatokon dolgozik?
8. Milyen nyelvet ad a kezünkbe a Hadoop adatok lekérdezéséhez az Apache Pig?



## Referenciák

- [1] <https://hive.apache.org/>
- [2] <https://cwiki.apache.org/confluence/display/Hive/Home>
- [3] <http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>
- [4] <http://spark.apache.org/>
- [5] <http://tez.apache.org/>
- [6] <https://cwiki.apache.org/confluence/display/Hive/LLAP>
- [7] <https://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/YARN.html>
- [8] <https://slider.incubator.apache.org/>
- [9] <http://parquet.apache.org/>
- [10] <http://orc.apache.org/>
- [11] <https://cwiki.apache.org/confluence/display/Hive/DeveloperGuide#DeveloperGuide-FileFormats>
- [12] <https://cwiki.apache.org/confluence/display/Hive/DeveloperGuide>
- [13] <https://impala.apache.org/overview.html>
- [14] [http://cidrdb.org/cidr2015/Papers/CIDR15\\_Paper28.pdf](http://cidrdb.org/cidr2015/Papers/CIDR15_Paper28.pdf)
- [15] <https://drill.apache.org/>
- [16] <https://www.mongodb.com/>
- [17] <https://www.elastic.co/>
- [18] <https://zookeeper.apache.org/>
- [19] <https://sqoop.apache.org/>