inovex

Big Data MDX with Mondrian and Apache Kylin

Sébastien Jelsch

London, 7-11-2015

Agenda

- OLAP-on-Hadoop with Apache Kylin
- Features
- Apache Kylin & Mondrian
- Conclusion & Discussion

Agenda

- OLAP-on-Hadoop with Apache Kylin
- Features
- Apache Kylin & Mondrian
- Conclusion & Discussion

Big Data

Situation

- More and more data becoming available on Hadoop
- Limitations in existing Business Intelligence Tools
  - Limited support for Hadoop
  - Data size growing exponentially
  - High latency of interactive queries
- Challenges to adapt Hadoop for interactive analysis
  - OLAP capability on Hadoop ecosystem not ready yet

OLAP and Big Data

Goals

- Full OLAP capability and advanced functionality
- Interactive analysis in subseconds
- ANSI SQL or MDX for analysts and engineers
- Seamless integration with BI Tools
- High concurrency with thousands of end users
- Distributed and scale out architecture for large data volume

What is Apache Kylin?

Solution: Apache Kylin

Extreme OLAP Engine for Big Data

- Distributed Analytics Engine from eBay
- OLAP-on-Hadoop
- Provides SQL interface for multidimensional analysis
- Based on Hadoop ecosystem

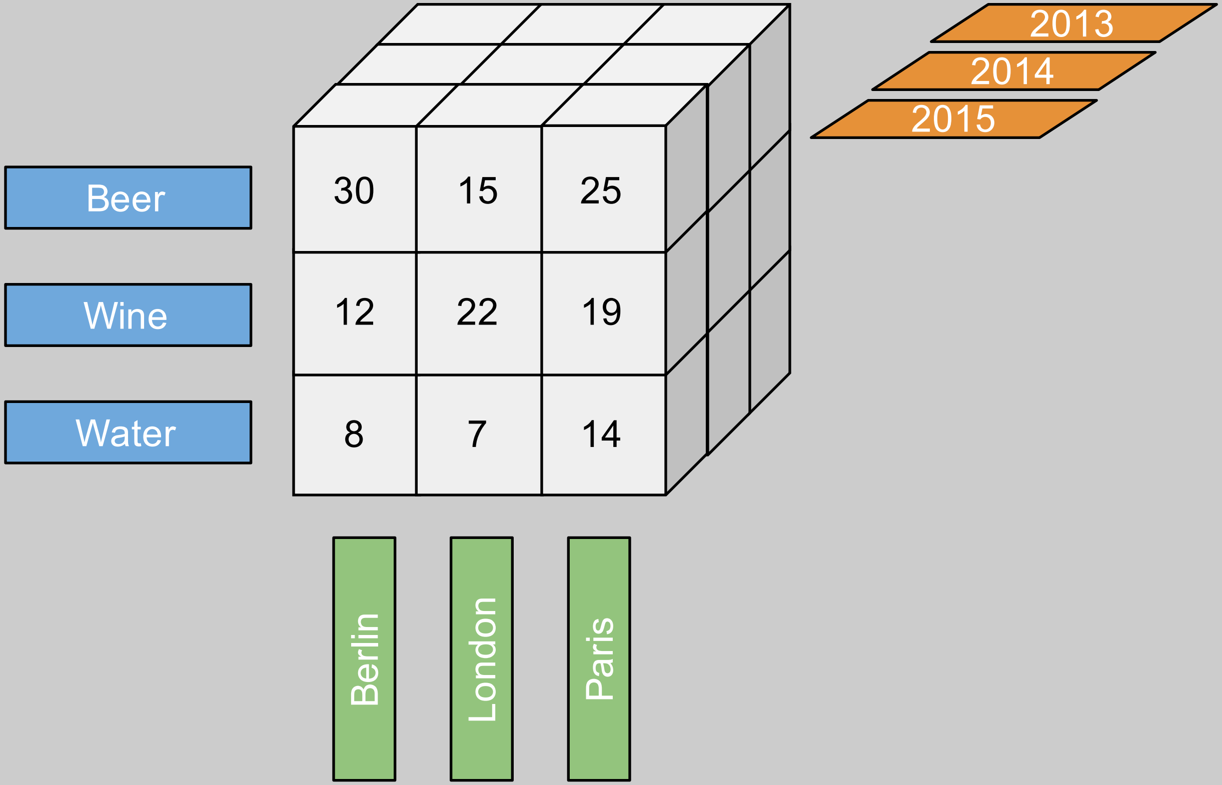
Open Source on: 1. October 2014

Accepted into incubation: 25. November 2014

Current version: 1.1 (25. October 2015)

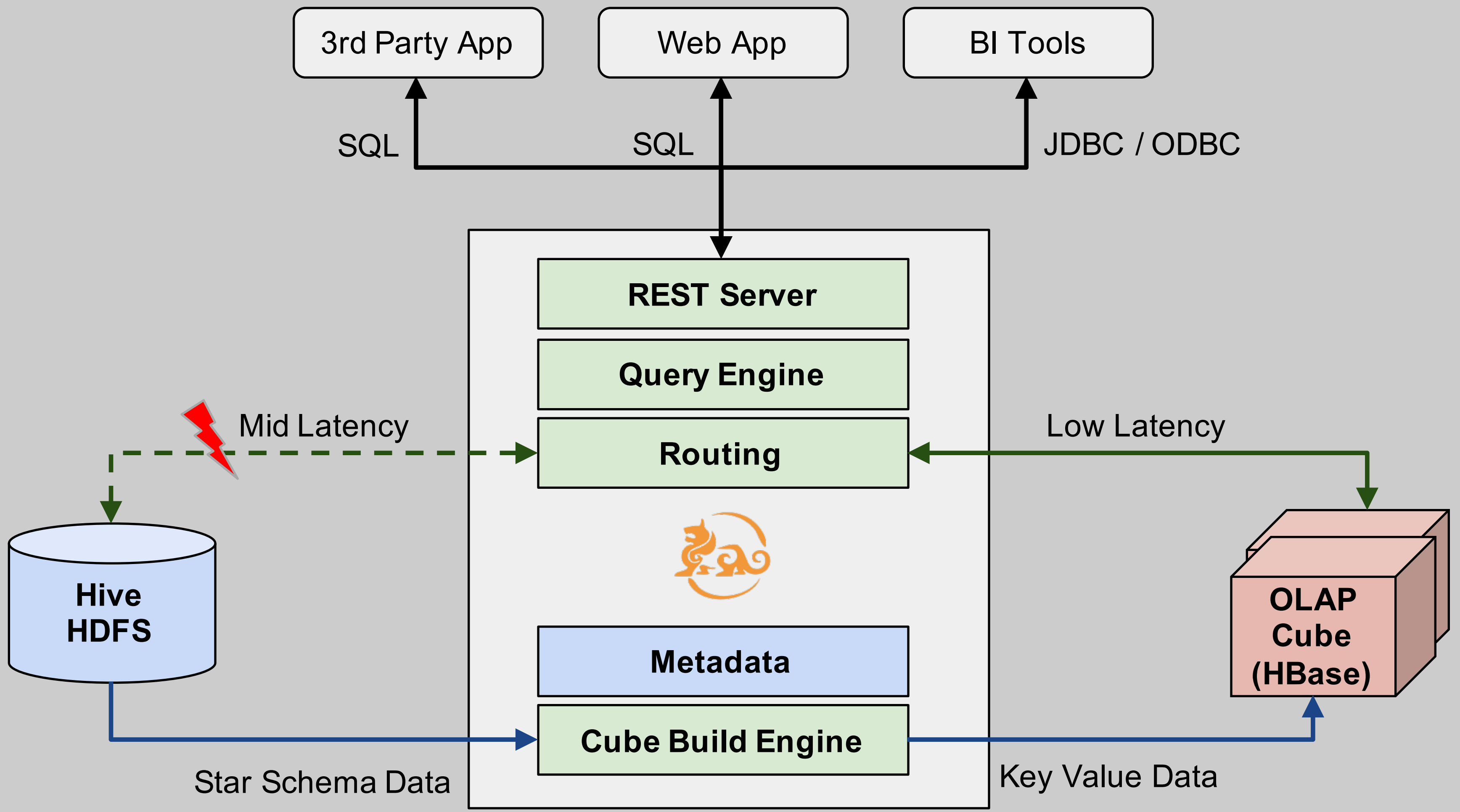
Short introduction into OLAP

OLAP Cube



	Berlin	London	Paris
Beer	30	15	25
Wine	12	22	19
Water	8	7	14

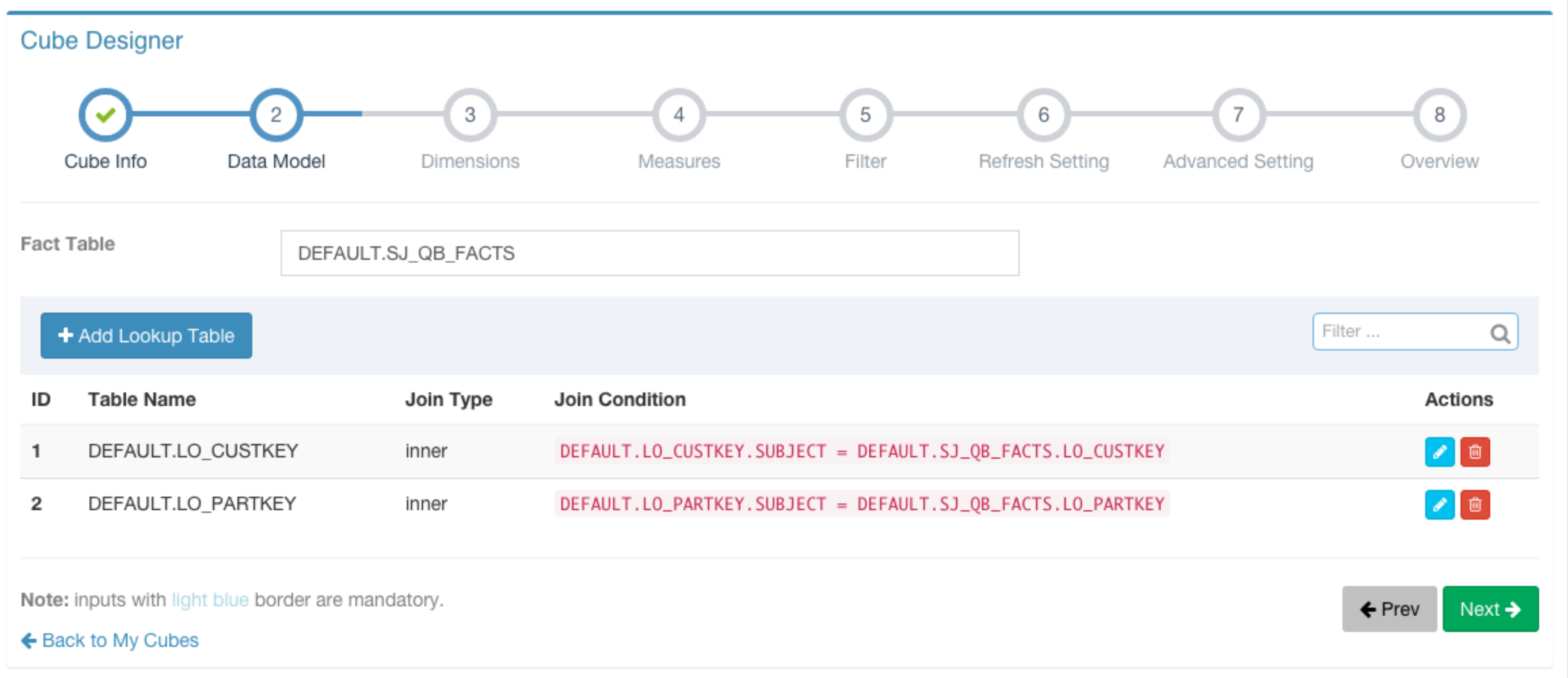
Apache Kylin: Architecture



Agenda

- OLAP-on-Hadoop with Apache Kylin
- Features
- Apache Kylin & Mondrian
- Conclusion & Discussion

Apache Kylin: Cube Designer



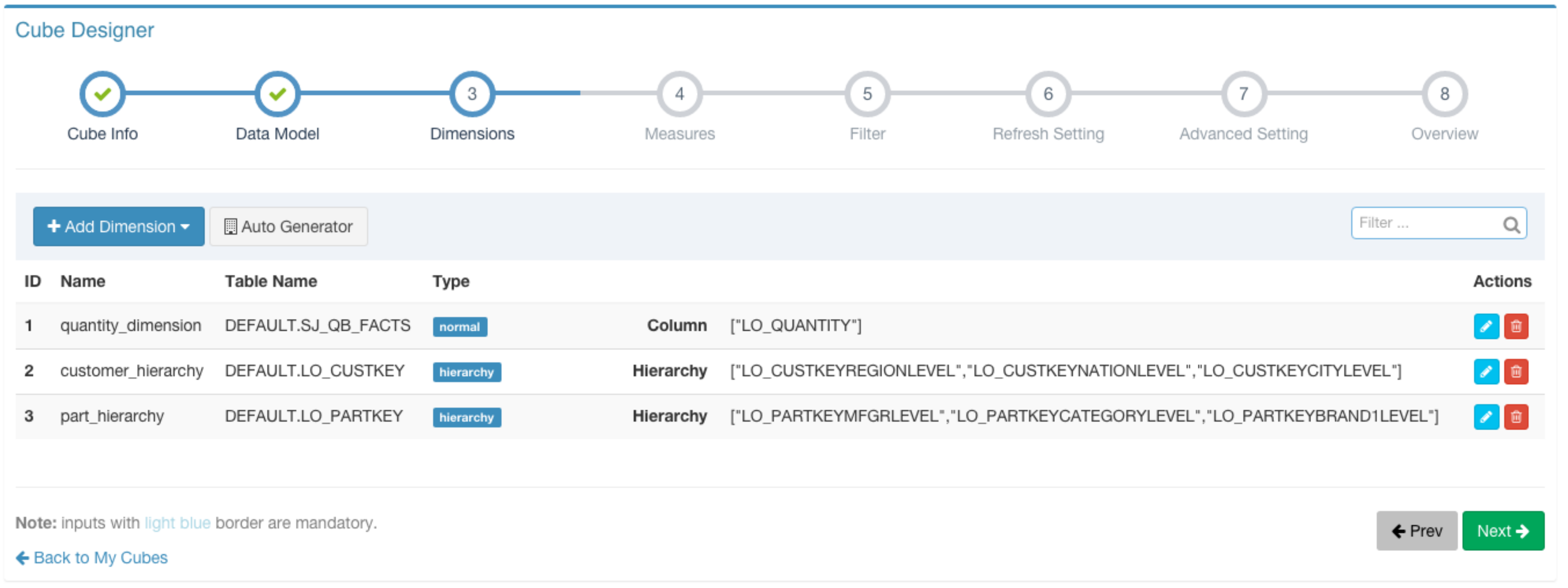
Cube Designer

Fact Table: DEFAULT.SJ_QB_FACTS

ID	Table Name	Join Type	Join Condition	Actions
1	DEFAULT.LO_CUSTKEY	inner	`DEFAULT.LO_CUSTKEY.SUBJECT = DEFAULT.SJ_QB_FACTS.LO_CUSTKEY`	
2	DEFAULT.LO_PARTKEY	inner	`DEFAULT.LO_PARTKEY.SUBJECT = DEFAULT.SJ_QB_FACTS.LO_PARTKEY`	

Note: inputs with light blue border are mandatory.

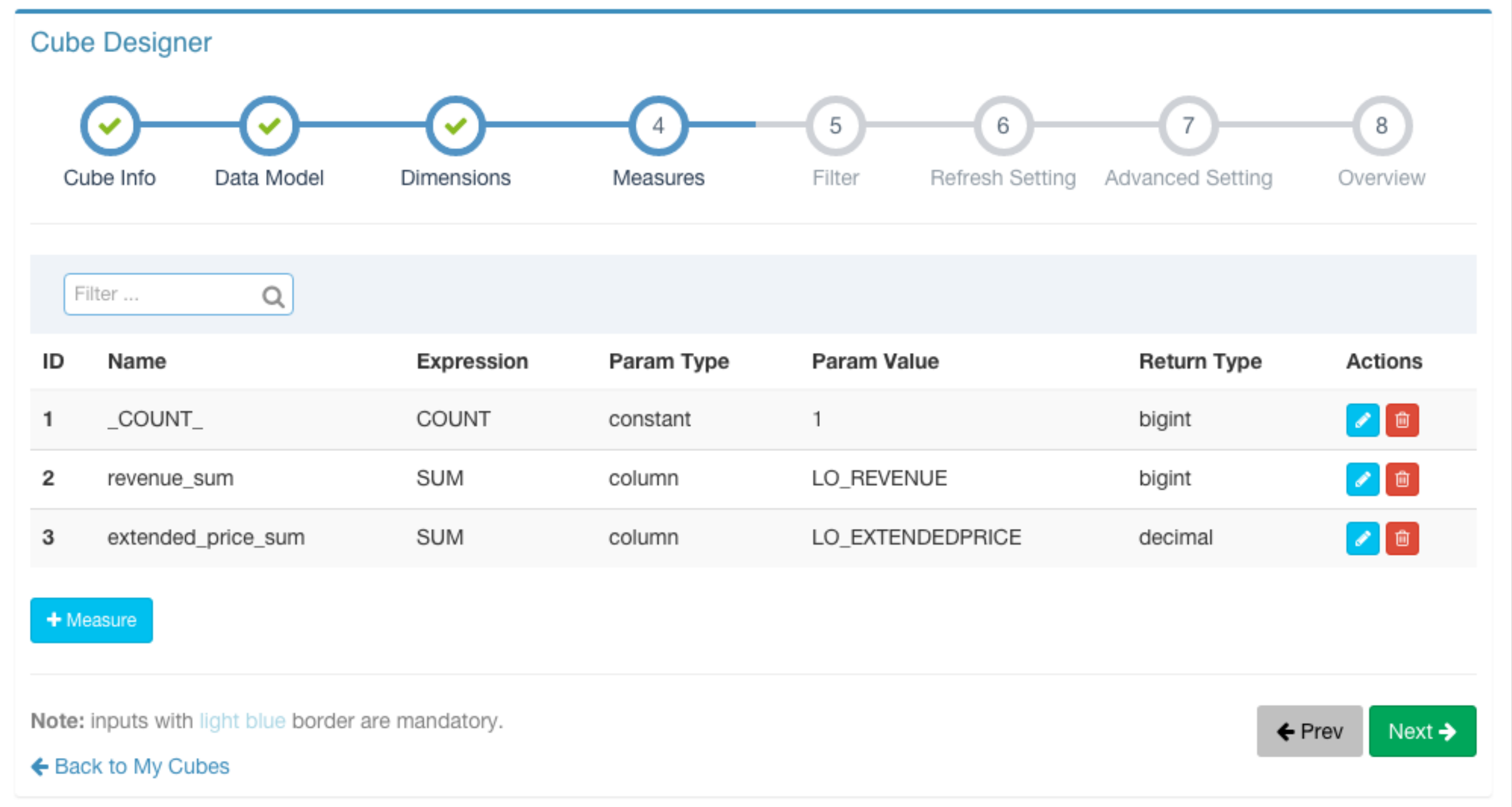
Apache Kylin: Cube Designer



ID	Name	Table Name	Type			Actions
1	quantity_dimension	DEFAULT.SJ_QB_FACTS	normal	**Column**	["LO_QUANTITY"]	✎ 🗑
2	customer_hierarchy	DEFAULT.LO_CUSTKEY	hierarchy	**Hierarchy**	["LO_CUSTKEYREGIONLEVEL","LO_CUSTKEYNATIONLEVEL","LO_CUSTKEYCITYLEVEL"]	✎ 🗑
3	part_hierarchy	DEFAULT.LO_PARTKEY	hierarchy	**Hierarchy**	["LO_PARTKEYMFGRLEVEL","LO_PARTKEYCATEGORYLEVEL","LO_PARTKEYBRAND1LEVEL"]	✎ 🗑

Note: inputs with light blue border are mandatory.

Apache Kylin: Cube Designer

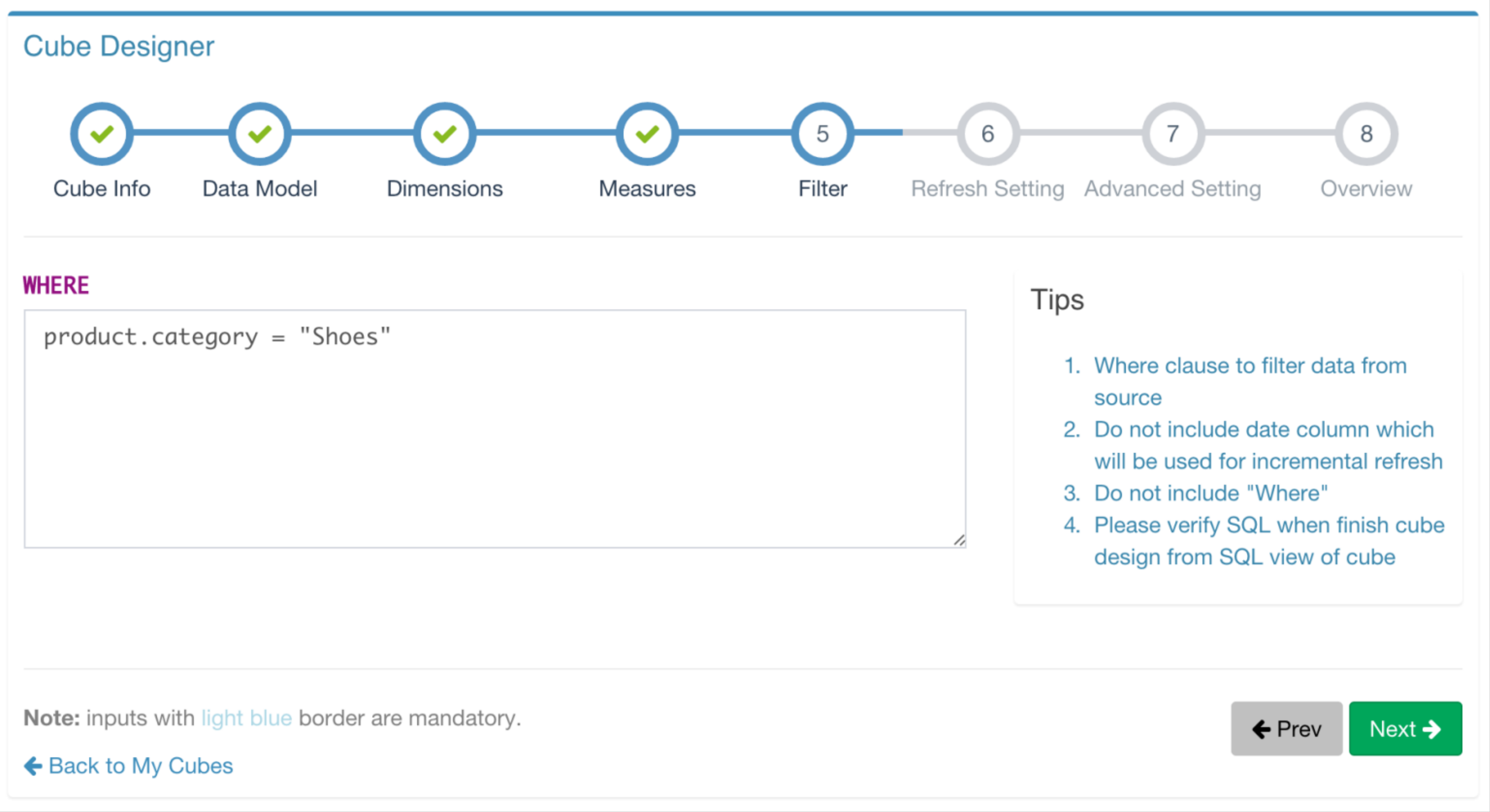


Cube Designer

ID	Name	Expression	Param Type	Param Value	Return Type	Actions
1	_COUNT_	COUNT	constant	1	bigint	✎ 🗑
2	revenue_sum	SUM	column	LO_REVENUE	bigint	✎ 🗑
3	extended_price_sum	SUM	column	LO_EXTENDEDPRICE	decimal	✎ 🗑

Note: inputs with light blue border are mandatory.

Apache Kylin: Cube Designer



Cube Designer

WHERE

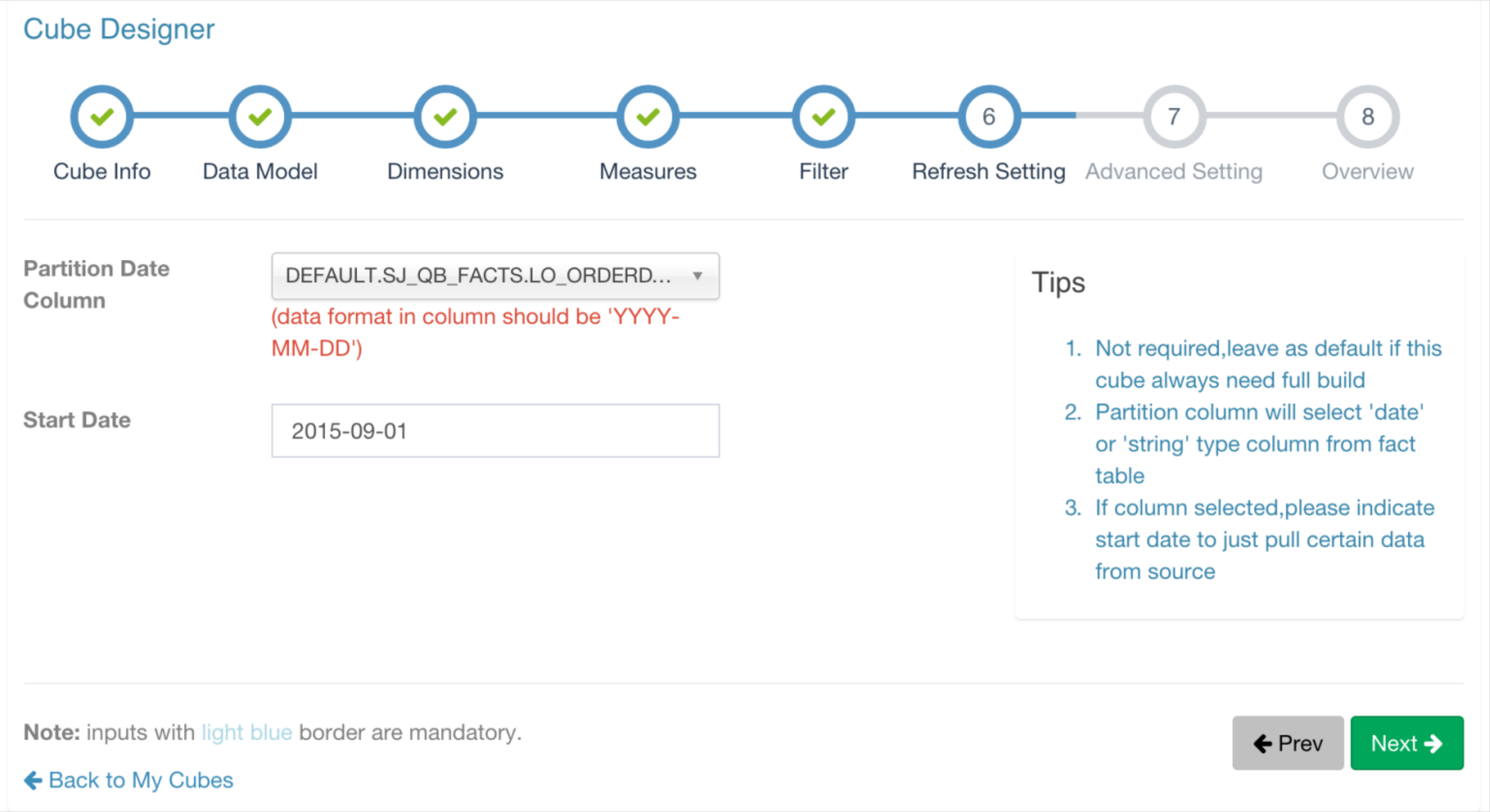
```
product.category = "Shoes"
```

Tips

1. Where clause to filter data from source
2. Do not include date column which will be used for incremental refresh
3. Do not include "Where"
4. Please verify SQL when finish cube design from SQL view of cube

Note: inputs with light blue border are mandatory.

Apache Kylin: Cube Designer



Cube Designer

Partition Date Column: DEFAULT.SJ_QB_FACTS.LO_ORDERD... ▾

(data format in column should be 'YYYY-MM-DD')

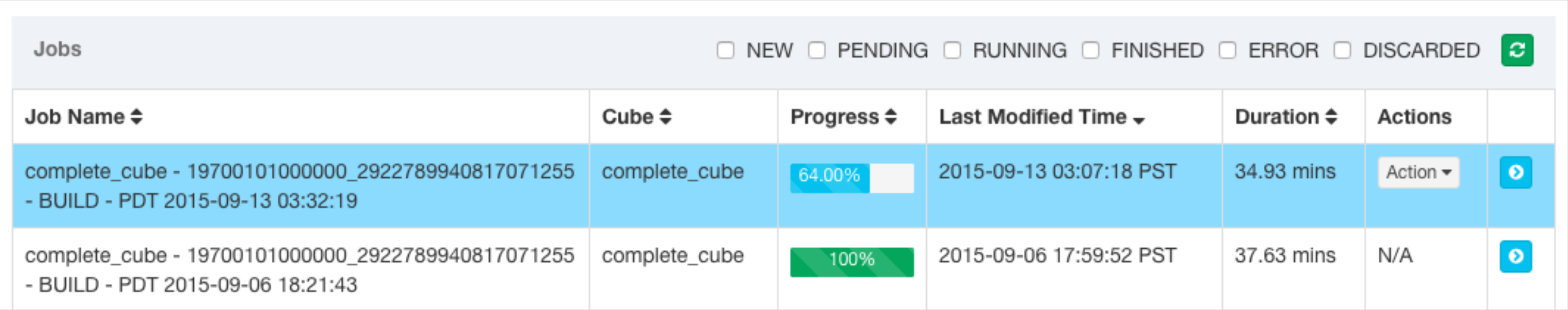
Start Date: 2015-09-01

Tips

1. Not required,leave as default if this cube always need full build
2. Partition column will select 'date' or 'string' type column from fact table
3. If column selected,please indicate start date to just pull certain data from source

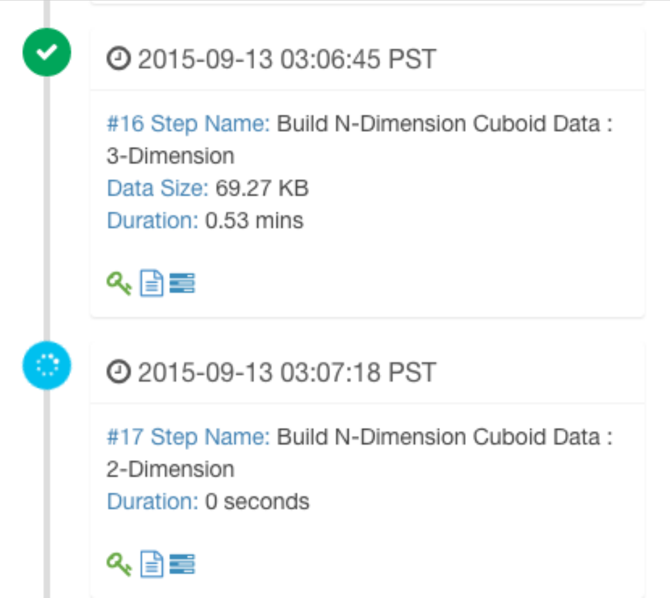
Note: inputs with light blue border are mandatory.

Apache Kylin: Monitoring



Jobs □ NEW □ PENDING □ RUNNING □ FINISHED □ ERROR □ DISCARDED

Job Name ⇕	Cube ⇕	Progress ⇕	Last Modified Time ▾	Duration ⇕	Actions	
complete_cube - 19700101000000_2922789940817071255 - BUILD - PDT 2015-09-13 03:32:19	complete_cube	64.00%	2015-09-13 03:07:18 PST	34.93 mins	Action ▾	❯
complete_cube - 19700101000000_2922789940817071255 - BUILD - PDT 2015-09-06 18:21:43	complete_cube	100%	2015-09-06 17:59:52 PST	37.63 mins	N/A	❯

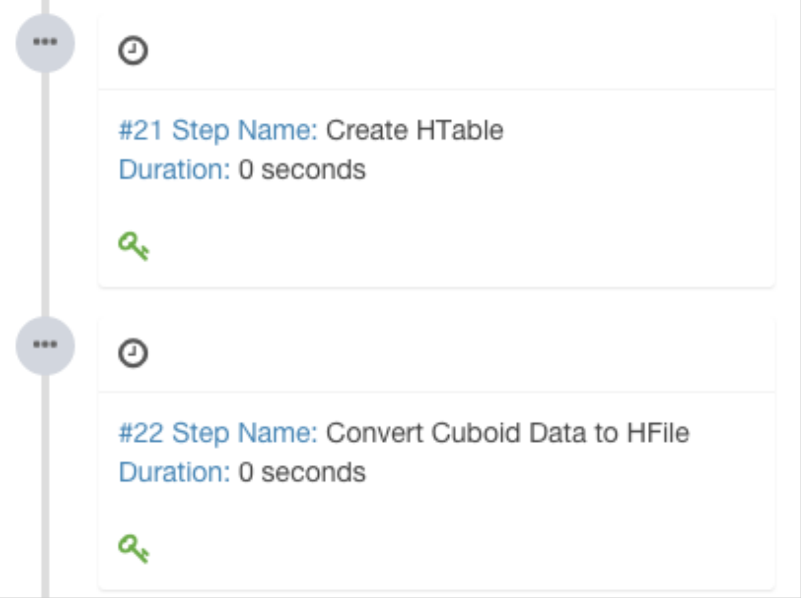


✓ 2015-09-13 03:06:45 PST

#16 Step Name: Build N-Dimension Cuboid Data : 3-Dimension
Data Size: 69.27 KB
Duration: 0.53 mins

2015-09-13 03:07:18 PST

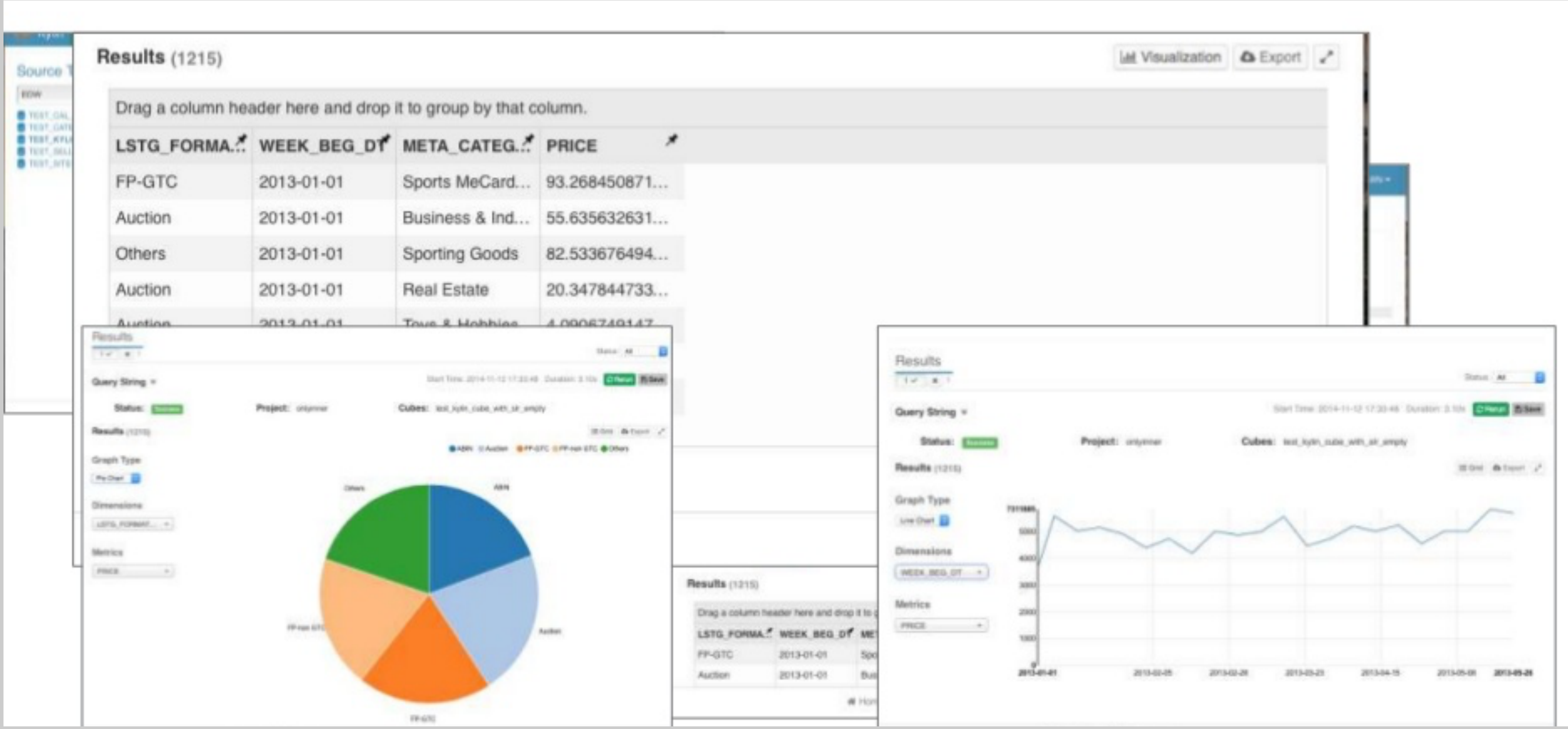
#17 Step Name: Build N-Dimension Cuboid Data : 2-Dimension
Duration: 0 seconds



#21 Step Name: Create HTable
Duration: 0 seconds

#22 Step Name: Convert Cuboid Data to HFile
Duration: 0 seconds

Apache Kylin: SQL Interface



Results (1215)

Visualization · Export

Drag a column header here and drop it to group by that column.

LSTG_FORMA...	WEEK_BEG_DT	META_CATEG...	PRICE
FP-GTC	2013-01-01	Sports MeCard...	93.268450871...
Auction	2013-01-01	Business & Ind...	55.635632631...
Others	2013-01-01	Sporting Goods	82.533676494...
Auction	2013-01-01	Real Estate	20.347844733...
Auction	2013-01-01	Toys & Hobbies	4.0906749147...

Query String

Status: Success · Project: onlyinner · Cubes: test_kylin_cube_with_slr_empty

Results (1215)

Graph Type: Pie Chart · Dimensions: LSTG_FORMAT... · Metrics: PRICE

Graph Type: Line Chart · Dimensions: WEEK_BEG_DT · Metrics: PRICE

Agenda

- OLAP-on-Hadoop with Apache Kylin
- Features
- Apache Kylin & Mondrian
- Conclusion & Discussion

Apache Kylin and MDX

SQL returns 2-dimensional result set

For more dimensions SQL was not designed

Wish:

- Multidimensional result set
- Consider hierarchies and levels in the data

⇨ Query Language: MDX

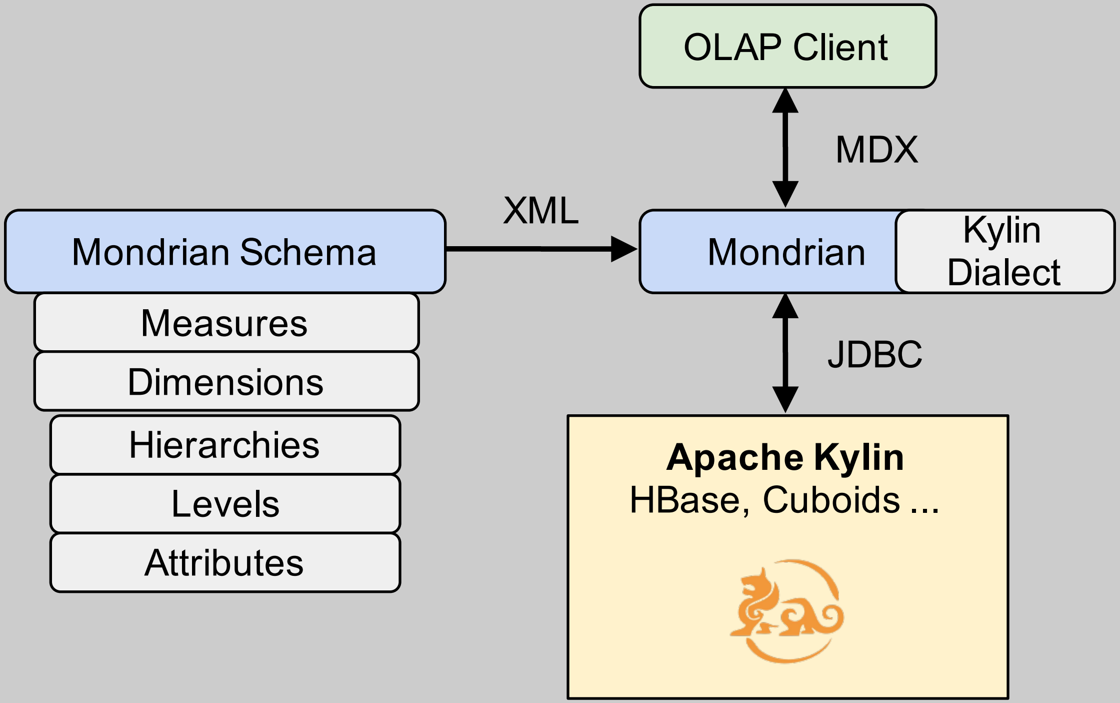
Pentahos Mondrian

Mondrian

- OLAP Engine
- Transforms MDX queries into SQL
- Multidimensional representation of data
- Integrated into Saiku / Pentahos Business Analytics Platform
- Expandable through SQL dialects

  e.g. MySQL, Postgres, Hive, Impala, …

Apache Kylin + Mondrian: Idea



Apache Kylin + Mondrian: Implementation

Work done:

- Kylin dialect created
- Optimized Kylins JDBC driver
- Bugs fixed to get Mondrian working with Kylin

TBD:

- Integrate Kylin dialect into Mondrians official code*
- Make every MDX query executable

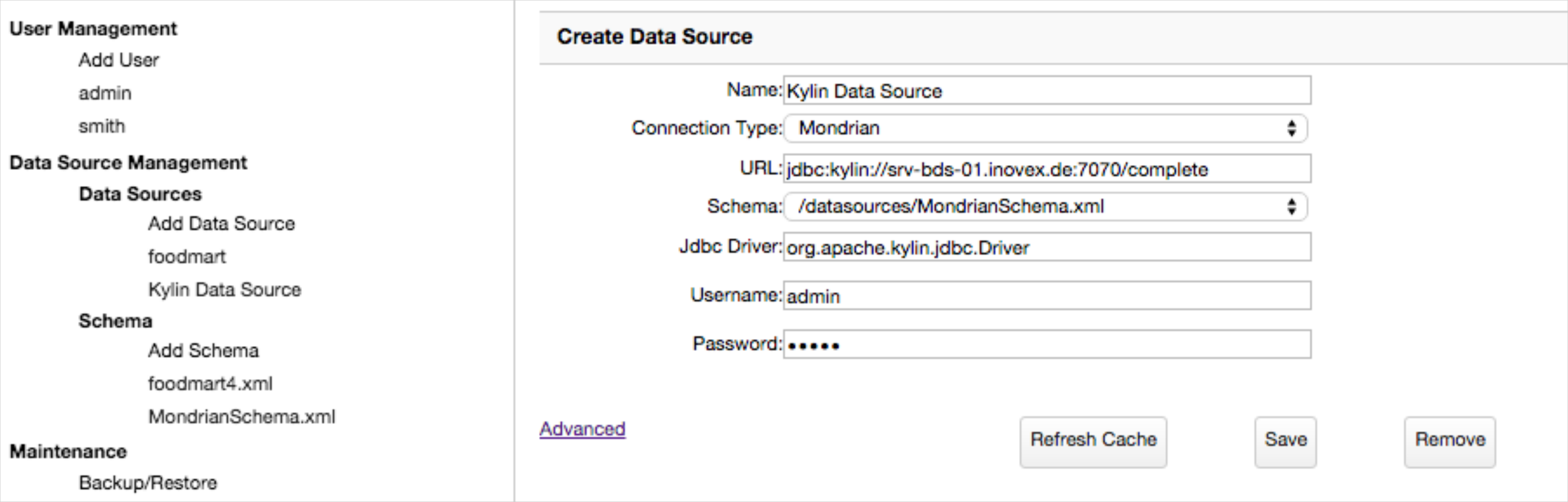
Successful tests**:**

- Current Saiku and Mondrian 4.4
- Current Saiku and Mondrian 3.x (not tested very well)

* Pull Request: https://github.com/pentaho/mondrian/pull/480

** Github Project: https://github.com/mustangore/kylin-mondrian-interaction

Apache Kylin + Mondrian: Examples



Create Data Source

Name: Kylin Data Source

Connection Type: Mondrian

URL: jdbc:kylin://srv-bds-01.inovex.de:7070/complete

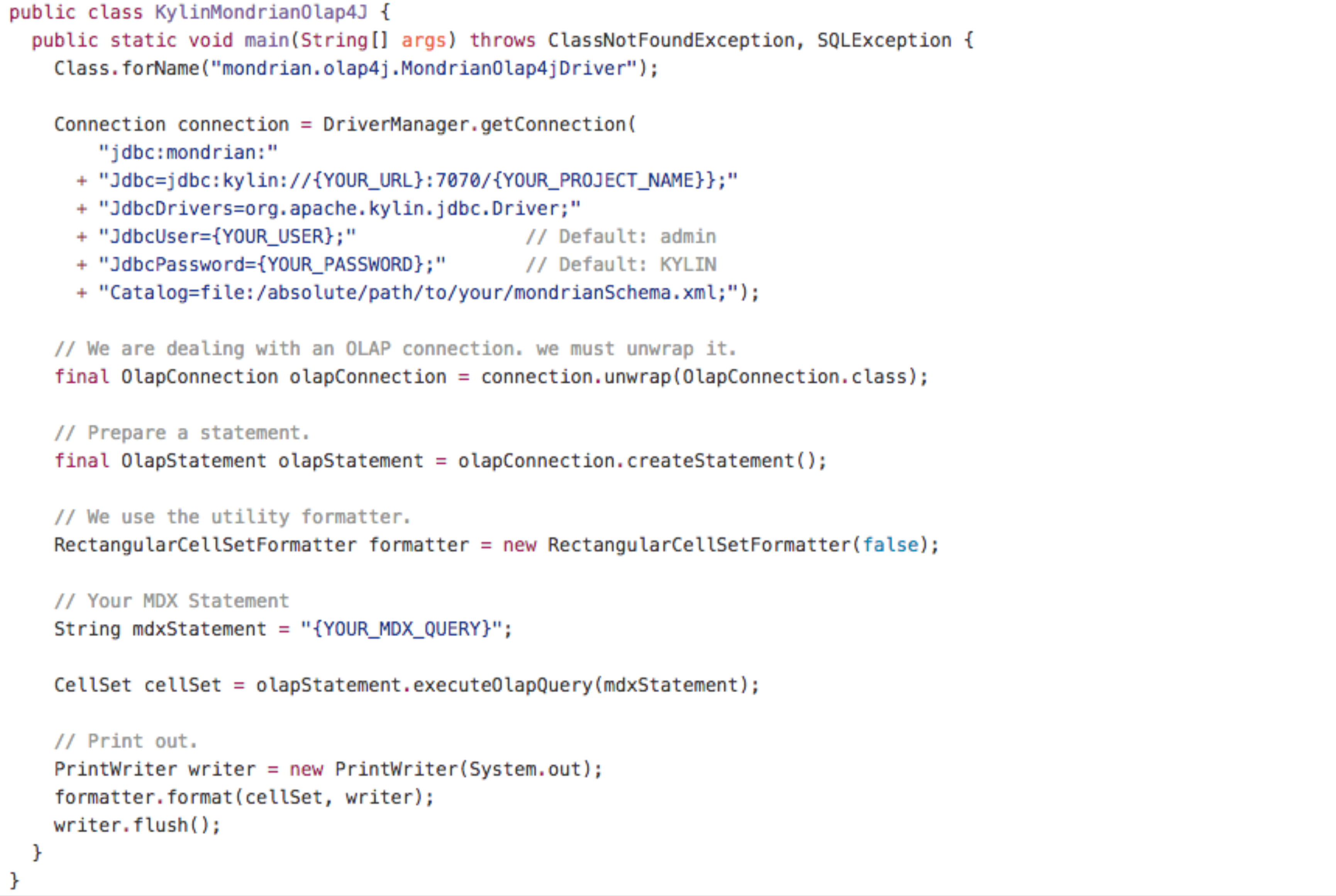
Schema: /datasources/MondrianSchema.xml

Jdbc Driver: org.apache.kylin.jdbc.Driver

Username: admin

Password: •••••

Apache Kylin + Mondrian: Examples



```java
public class KylinMondrianOlap4J {
  public static void main(String[] args) throws ClassNotFoundException, SQLException {
    Class.forName("mondrian.olap4j.MondrianOlap4jDriver");

    Connection connection = DriverManager.getConnection(
        "jdbc:mondrian:"
      + "Jdbc=jdbc:kylin://{YOUR_URL}:7070/{YOUR_PROJECT_NAME}};"
      + "JdbcDrivers=org.apache.kylin.jdbc.Driver;"
      + "JdbcUser={YOUR_USER};"               // Default: admin
      + "JdbcPassword={YOUR_PASSWORD};"       // Default: KYLIN
      + "Catalog=file:/absolute/path/to/your/mondrianSchema.xml;");

    // We are dealing with an OLAP connection. we must unwrap it.
    final OlapConnection olapConnection = connection.unwrap(OlapConnection.class);

    // Prepare a statement.
    final OlapStatement olapStatement = olapConnection.createStatement();

    // We use the utility formatter.
    RectangularCellSetFormatter formatter = new RectangularCellSetFormatter(false);

    // Your MDX Statement
    String mdxStatement = "{YOUR_MDX_QUERY}";

    CellSet cellSet = olapStatement.executeOlapQuery(mdxStatement);

    // Print out.
    PrintWriter writer = new PrintWriter(System.out);
    formatter.format(cellSet, writer);
    writer.flush();
  }
}
```

Agenda

- OLAP-on-Hadoop with Apache Kylin
- Features
- Apache Kylin & Mondrian
- Conclusion & Discussion

Apache Kylin: Conclusion

- Extremely fast and scalable OLAP Engine
- OLAP-on-Hadoop
- Depends on Apache Hadoop infrastructure
- MOLAP Cube
- Incremental refresh of cubes
- Integration into existing BI Tools
- MDX queries with Mondrian possible (ongoing work)

Thank you for your attention

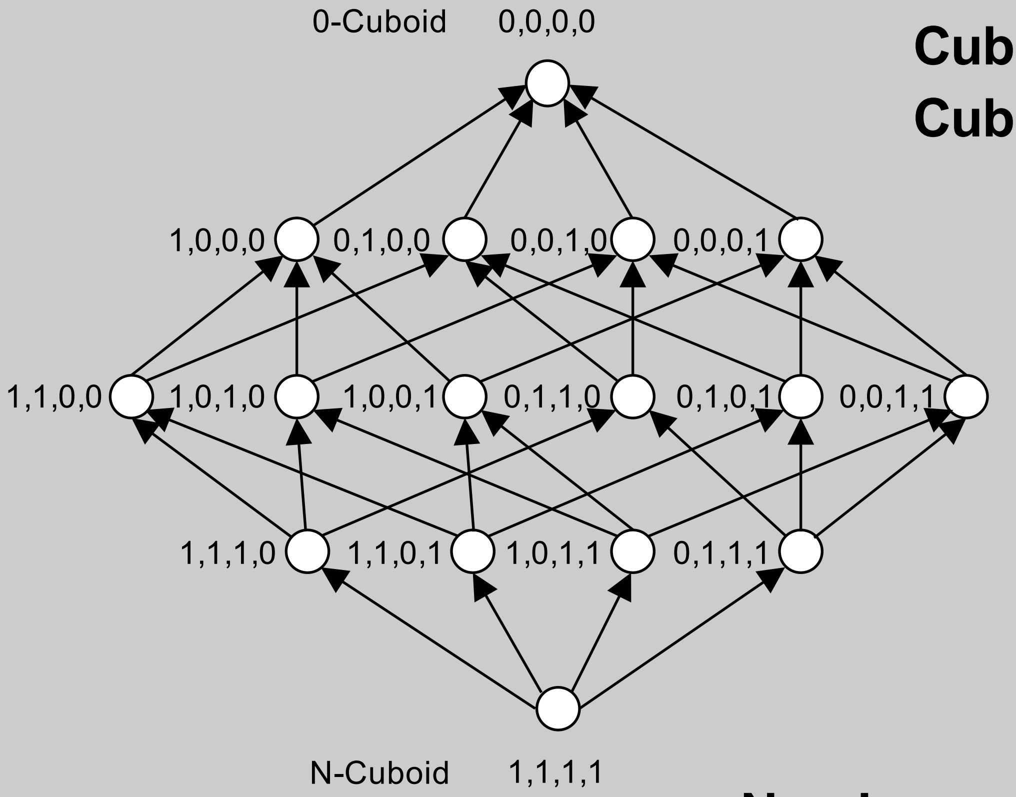
Contact

Sébastien Jelsch
Big Data Scientist

inovex GmbH
Office Karlsruhe
Ludwig-Erhard-Allee 6
76131 Karlsruhe

Tel: +49 176 - 45786280
E-Mail: sjelsch@inovex.de
Twitter: @inovexgmbh | @Mustangore

Introduction into OLAP



Cube: All combinations

Cuboid: One single combination

Number cuboids growing exponentially

Apache Kylin: Aggregation Groups

Problem: Number of Cuboids grows exponentially

Example:

Cube with 30 dimensions

Number of Cuboids: $2^{30} > 1$ billion

Solution: Partial Cube

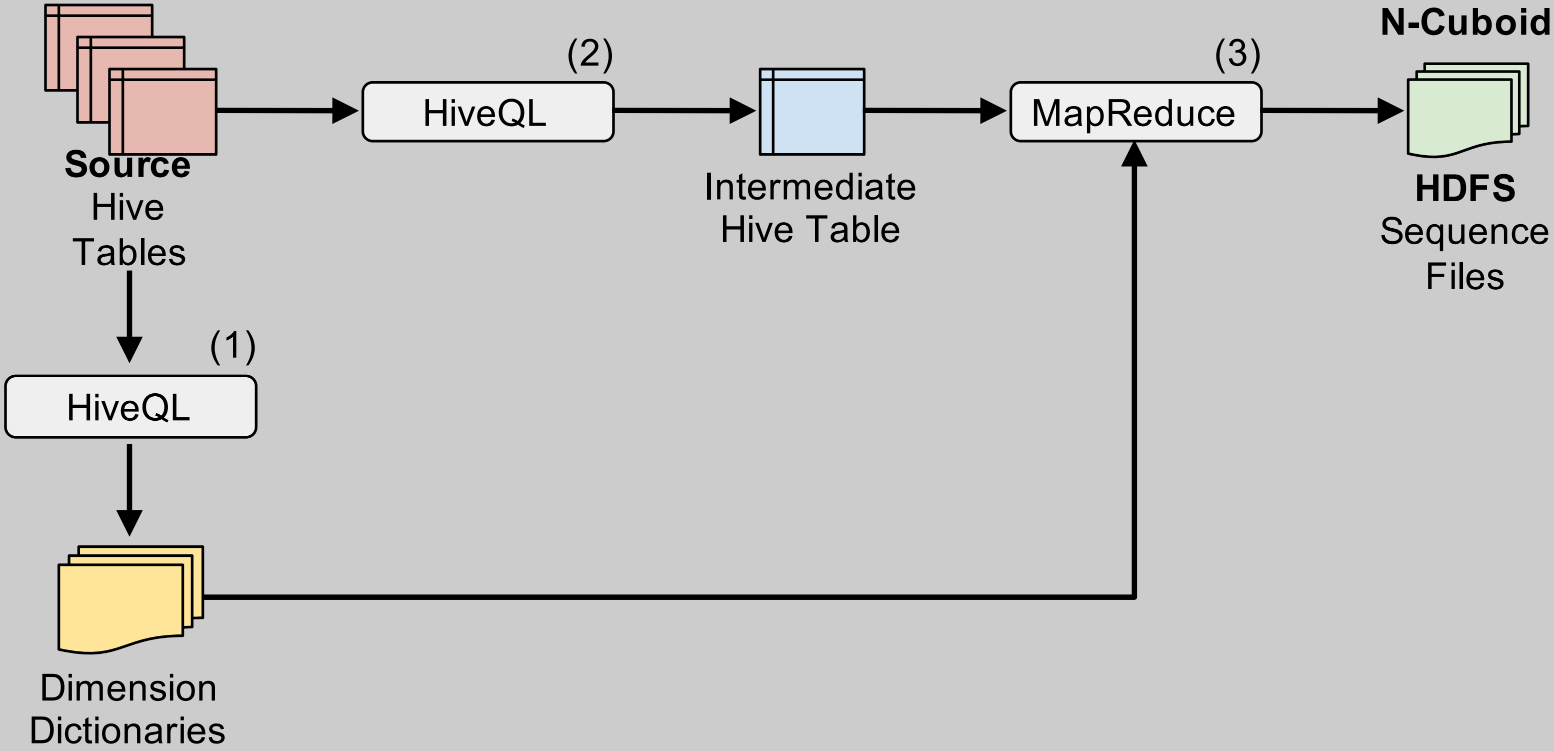
Classificate the OLAP Cube in Aggregation Groups

Example:

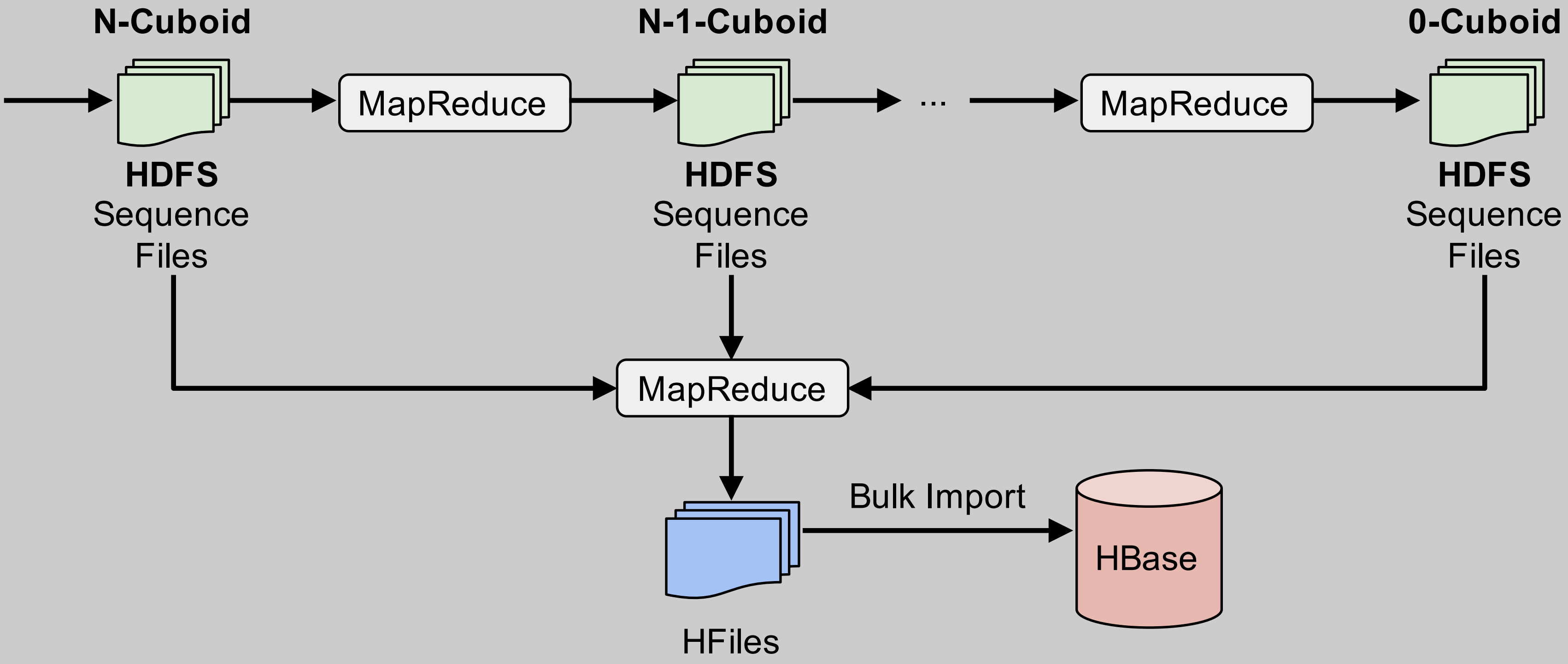
30 dimensions splitted into 3 groups of 10 dimensions

Number of Cuboids: $2^{10} + 2^{10} + 2^{10} = 3072 << 1$ billion

Apache Kylin: Cube Build Process



Apache Kylin: Cube Build Process


N-Cuboid
MapReduce
N-1-Cuboid
...
MapReduce
0-Cuboid
HDFS Sequence Files
HDFS Sequence Files
HDFS Sequence Files
MapReduce
Bulk Import
HFiles
HBase