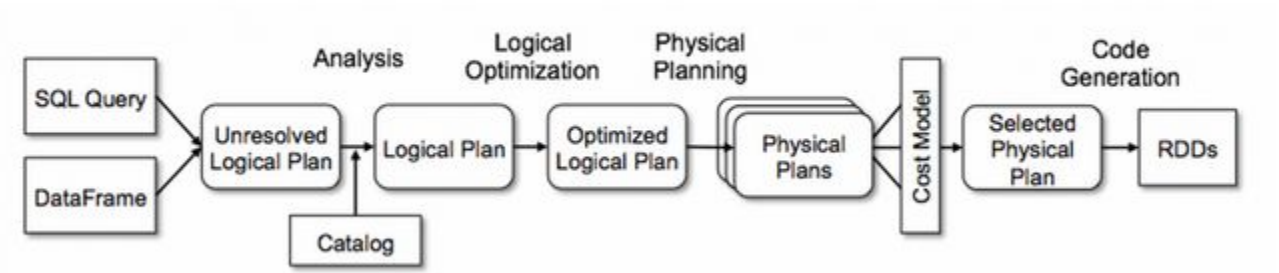
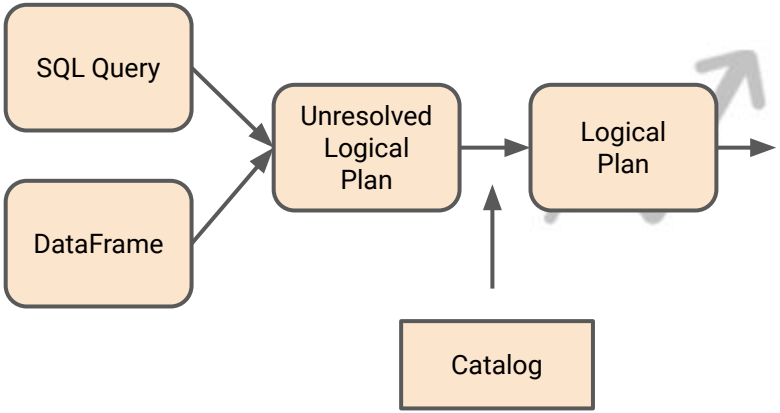
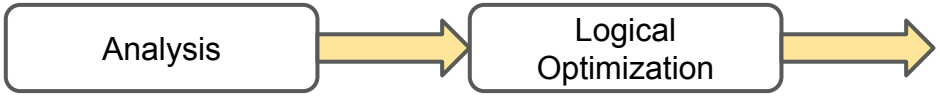
Catalyst Optimizer

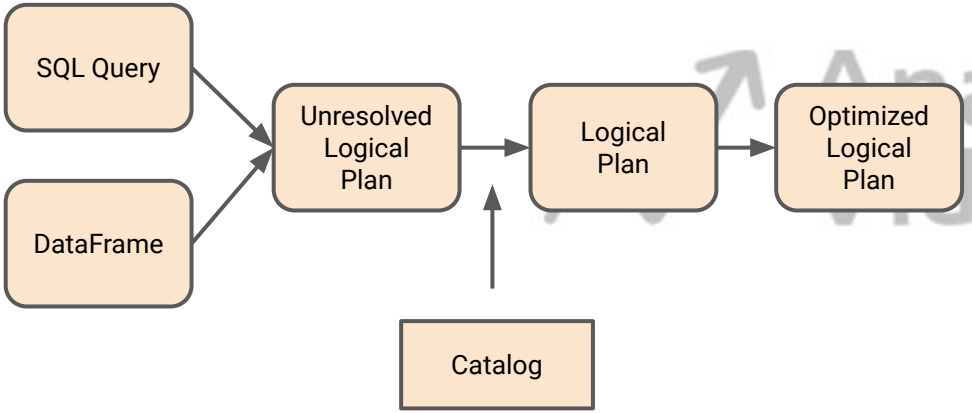
Catalyst Optimizer

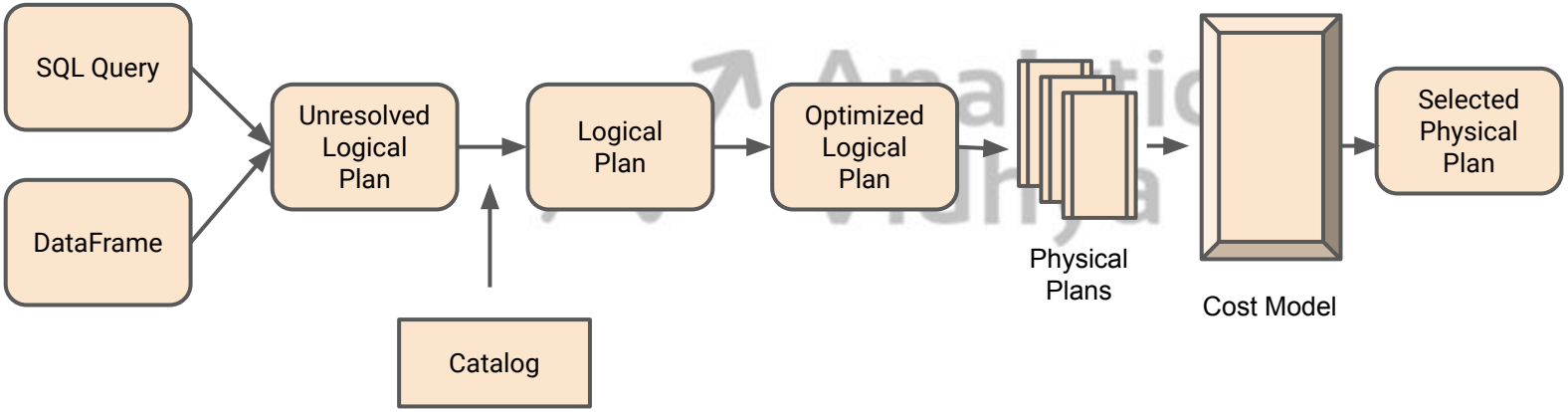
- Optimiser library within Spark SQL API.
- Optimises SQL queries as well as DataFrame code.
- Consists of 4 stages: **Analysis, Logical Optimization, Physical Planning, Code Generation**

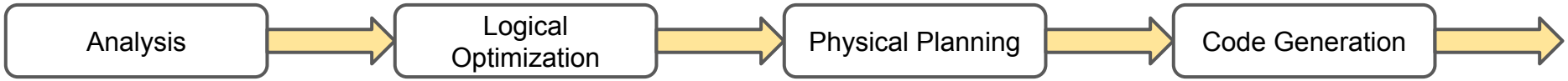


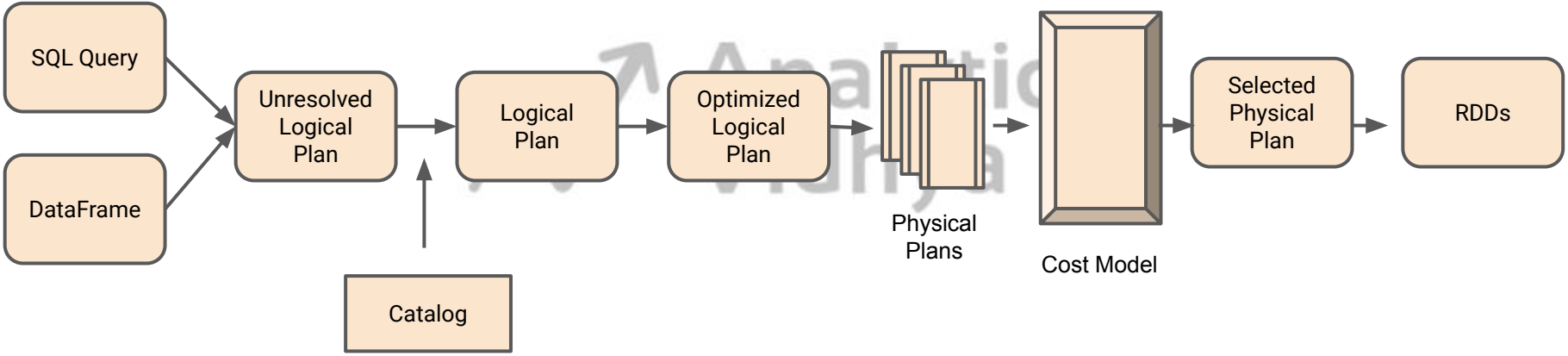












```
df_sql = spark.sql("select * from mobile limit 10")
```

```
df_sql.explain(extended = True)
```

```
== Parsed Logical Plan ==
'GlobalLimit 10
+- 'LocalLimit 10
   +- 'Project [*]
      +- 'UnresolvedRelation [mobile]

== Analyzed Logical Plan ==
Date: string, Country: string, City: string, Region: string, Segment: string, Sales: string, Profit: string
GlobalLimit 10
+- LocalLimit 10
   +- Project [Date#46, Country#47, City#48, Region#49, Segment#50, Sales#51, Profit#52]
      +- SubqueryAlias mobile
         +- Relation[Date#46,Country#47,City#48,Region#49,Segment#50,Sales#51,Profit#52] csv

== Optimized Logical Plan ==
GlobalLimit 10
+- LocalLimit 10
   +- Relation[Date#46,Country#47,City#48,Region#49,Segment#50,Sales#51,Profit#52] csv

== Physical Plan ==
CollectLimit 10
+- FileScan csv [Date#46,Country#47,City#48,Region#49,Segment#50,Sales#51,Profit#52] Batched: false, DataFilters:
[], Format: CSV, Location: InMemoryFileIndex[file:/home/siddharth/Documents/CDP/Datasets/African_Mobile_Data.csv],
PartitionFilters: [], PushedFilters: [], ReadSchema: struct<Date:string,Country:string,City:string,Region:string,S
egment:string,Sales:string,Profit:st...
```

Thank You!!

Analytics Vidhya