

Querying Data Using Spark SQL



Janani Ravi

CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Spark SQL enables querying of DataFrames as database tables

Temporary per-session and global tables

The Catalyst optimizer makes SQL queries fast

Tables schemas can be inferred or explicitly specified

Advanced window operations are also supported

Demo

Introducing SQL operations on Spark

Demo

Explore airline data using Spark SQL

Catalyst Optimizer

Changes Starting Spark 2.0



Easier

Unifying Datasets and
DataFrames, SQL support...



Faster

Optimize like a compiler, not a
DBMS

Catalyst Optimizer

Optimization engine that powers Spark SQL (as well as DataFrame API) since 2015

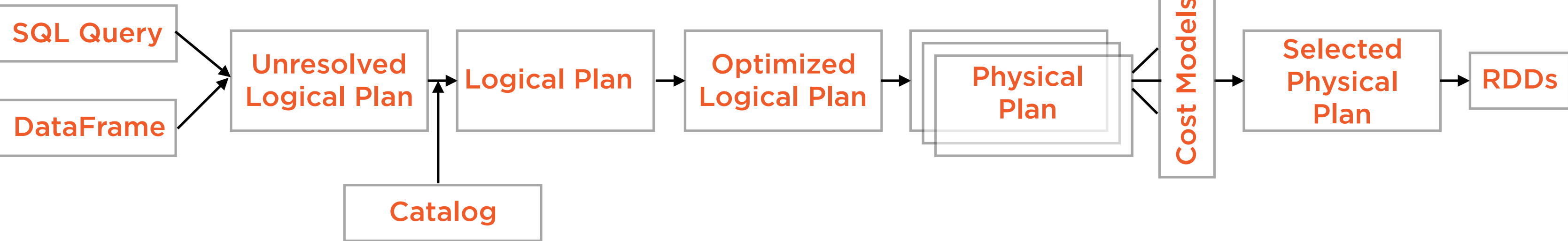
Catalyst Optimizer

Analysis

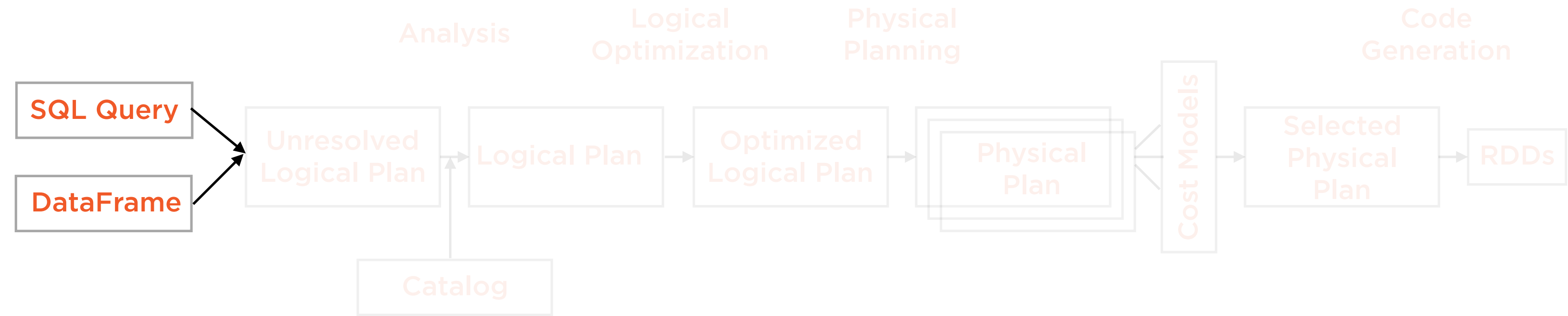
Logical
Optimization

Physical
Planning

Code
Generation

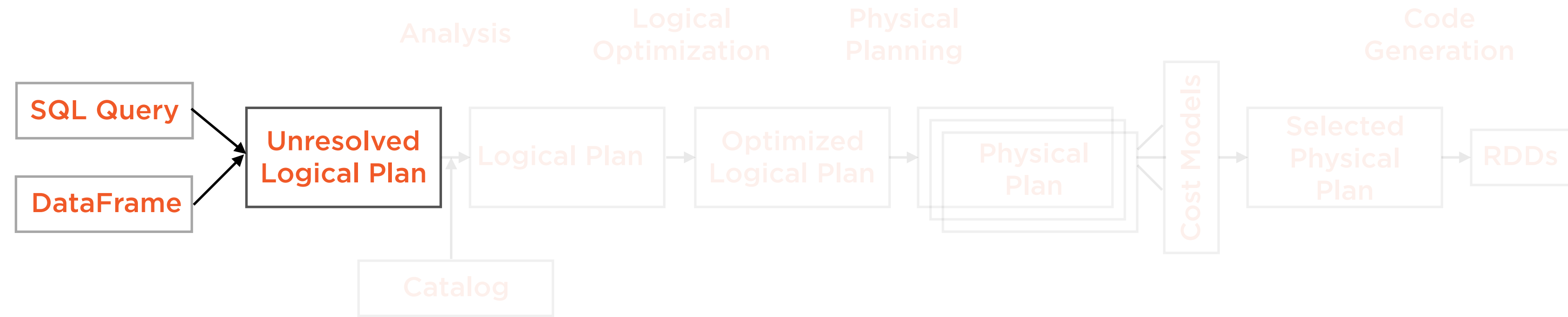


Catalyst Optimizer



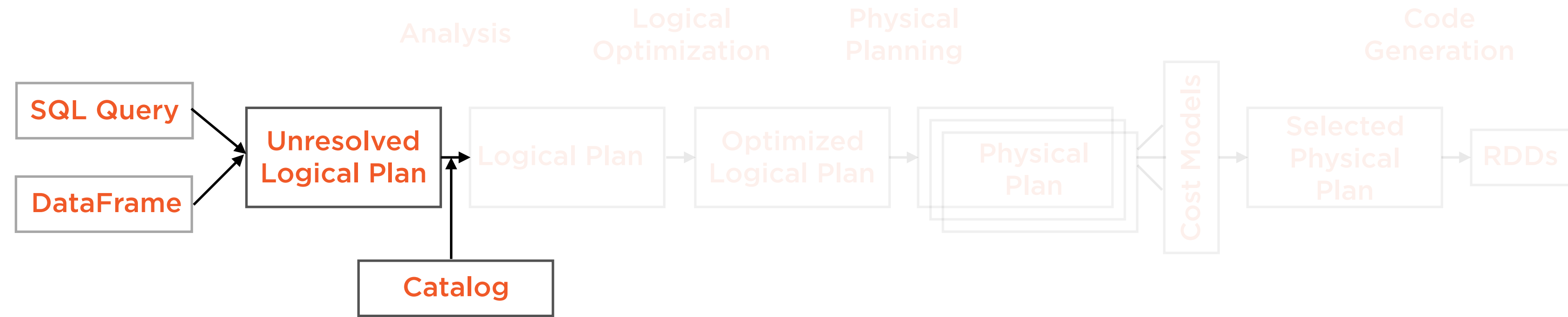
Relations to be
processed

Catalyst Optimizer



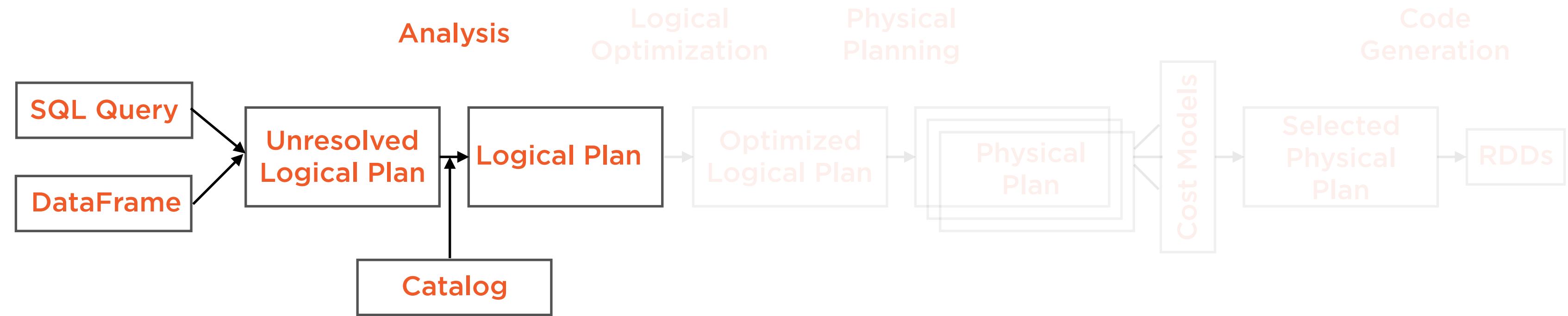
Unresolved as column types and existence yet to be ascertained

Catalyst Optimizer



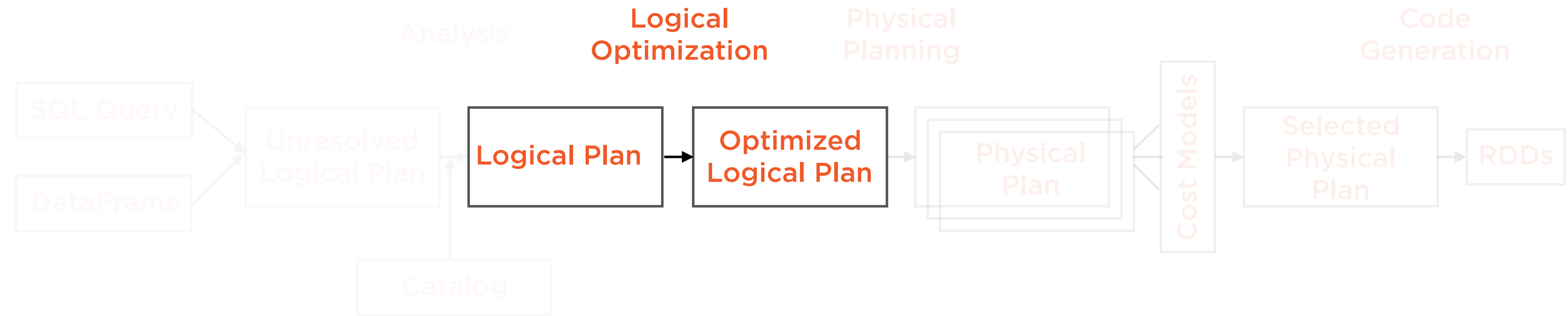
Catalog tracks tables in all data sources to resolve plan

Catalyst Optimizer



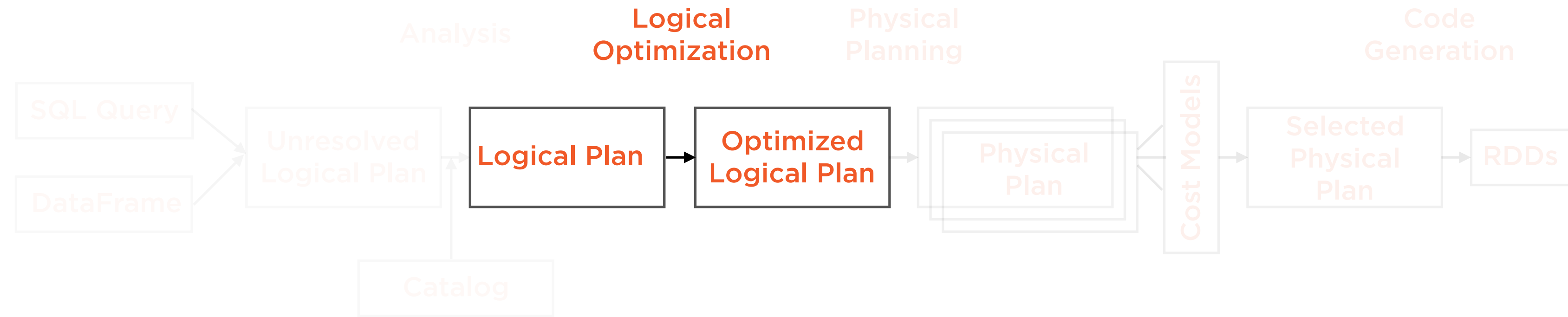
Output of the Analysis phase is a logical plan

Catalyst Optimizer



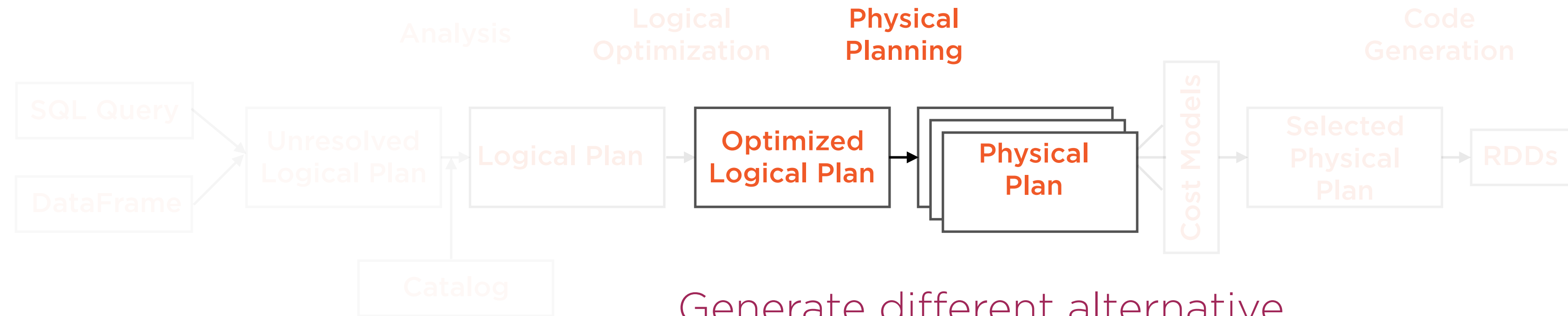
Predicate pushdown, projection pruning, null propagation, expression simplification...

Catalyst Optimizer



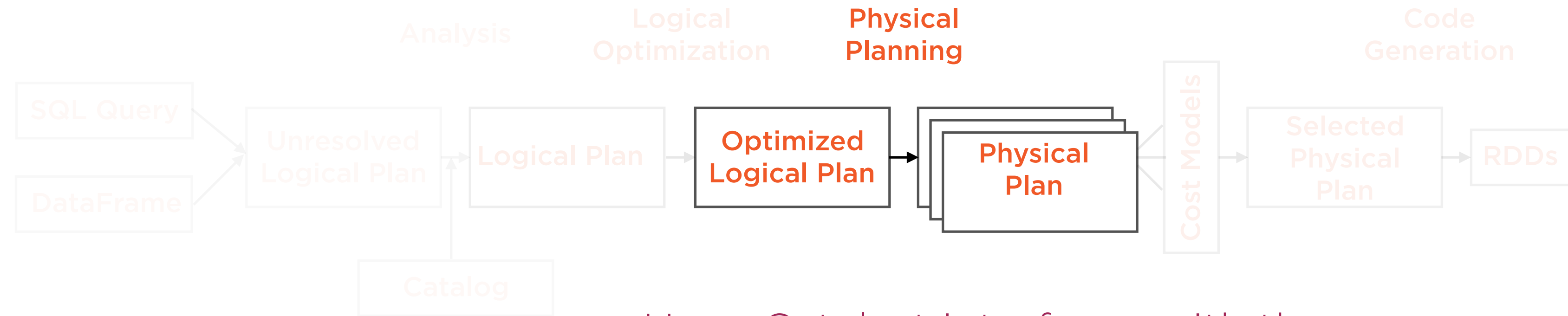
Generate various such logical plans, then pick the lowest-cost (optimized) logical plan

Catalyst Optimizer



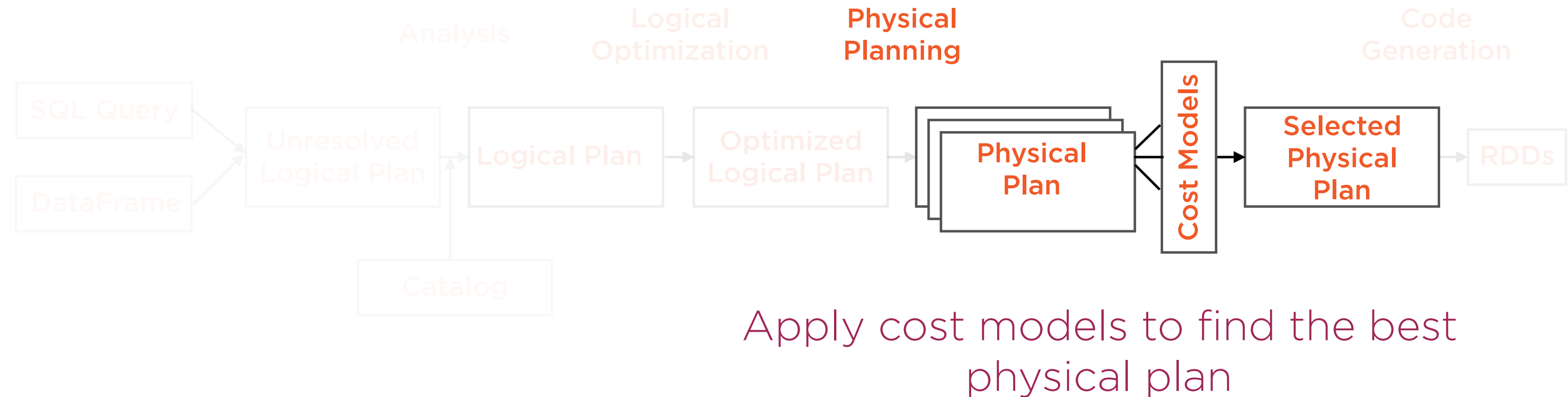
Generate different alternative physical plans for this optimized logical plan

Catalyst Optimizer

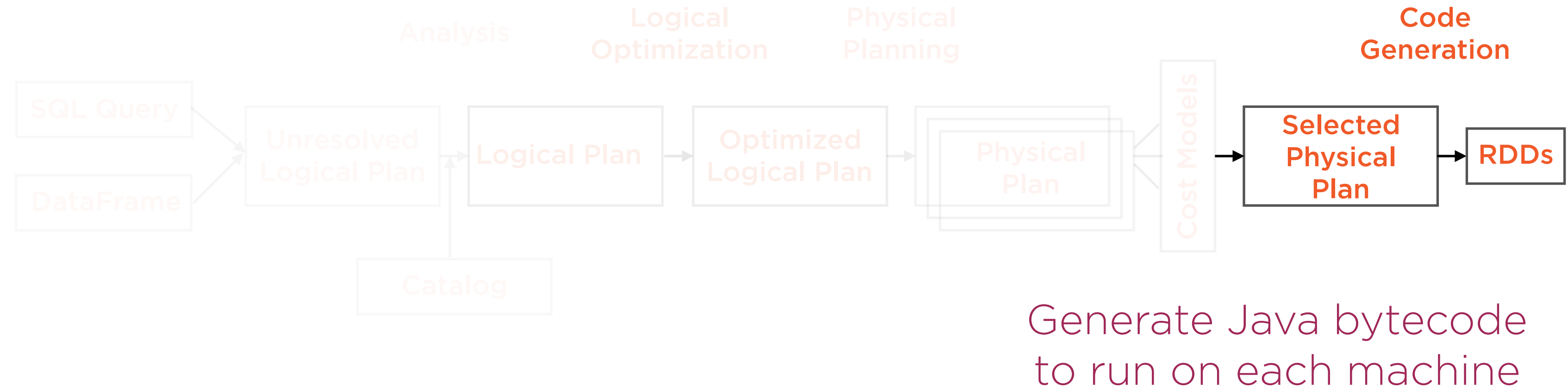


Here, Catalyst interfaces with the Spark execution engine (Tungsten)

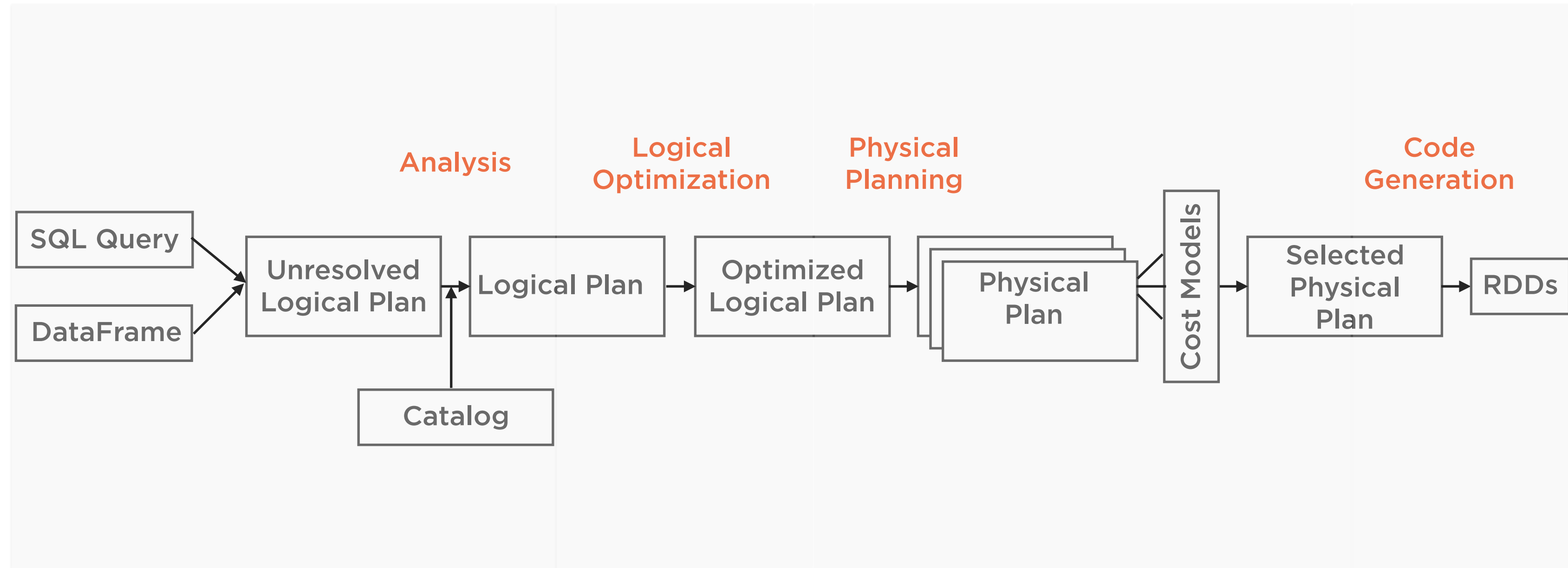
Catalyst Optimizer



Catalyst Optimizer



Catalyst Optimizer



Four phases of query optimization and execution



Catalyst Optimizer

Novel use of advanced Scala constructs

Extensible for new optimizations

Specially designed for big data applications

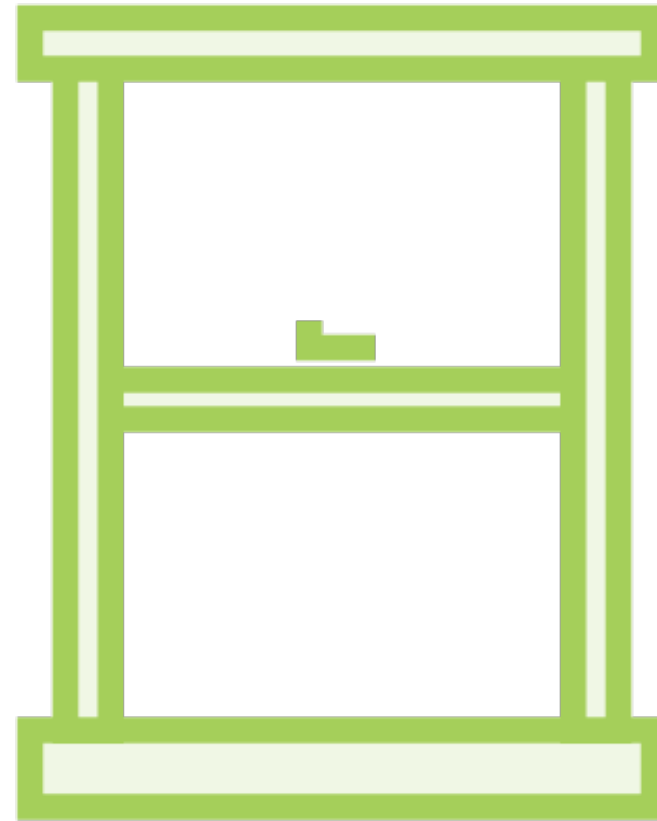
- Semi-structured data

Demo

Inferred and explicit schemas

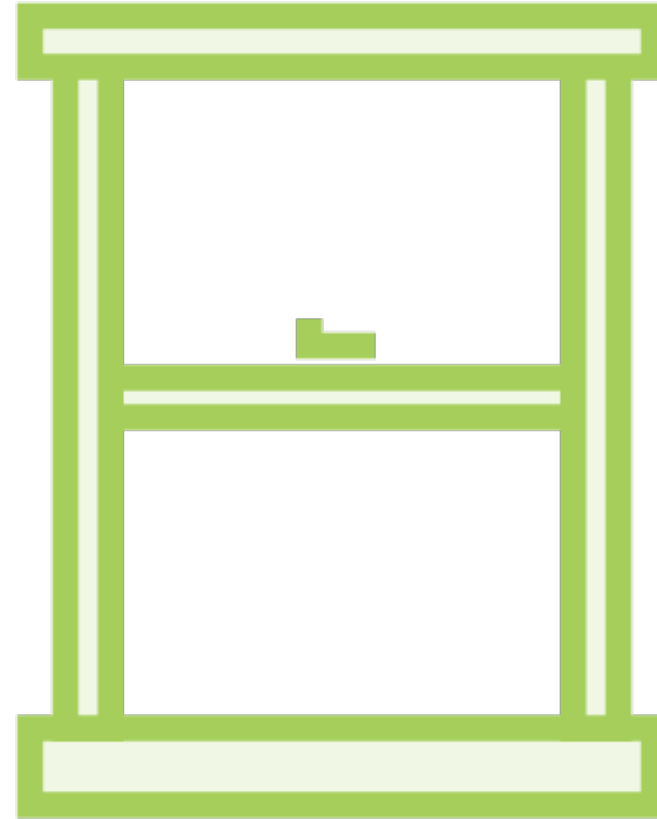
What are Window Functions?

Window Functions



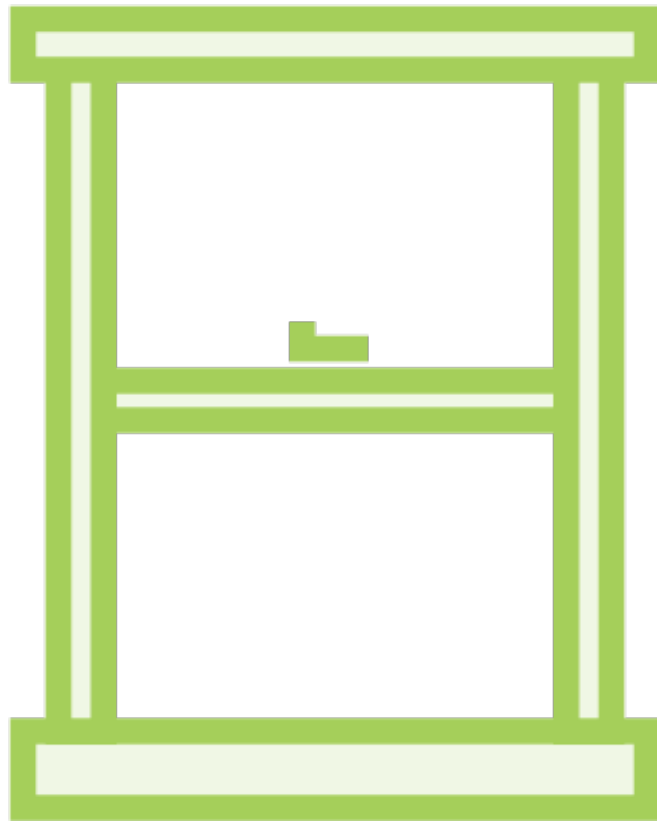
Functions which operate over a range of rows in a DataFrame

Window Functions



A “window”

Window Functions



Find the top selling washing machine in last week's sale

Partition by = home product categories

Order by = number sold

Frame = one week

An Example of a Window Function

Store With the Lowest Product Price

Product	Store	Revenue
Bananas	Seattle	7
Potatoes	Bellevue	5.5
Bananas	Bellevue	6.75
Potatoes	Seattle	6.5
Potatoes	Redmond	4
Bananas	Kent	6.5
Potatoes	Kent	4.5
Bananas	Redmond	6.8

Partition by product

Store With the Lowest Product Price

Product	Store	Revenue
Bananas	Seattle	7
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Potatoes	Seattle	6.5
Potatoes	Bellevue	5.5
Potatoes	Redmond	4
Potatoes	Kent	4.5

Order by price

Store With the Lowest Product Price

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

Find the store with the lowest price

Store With the Lowest Product Price

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

ROW Frame and RANGE Frame

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

partitionBy() product, orderBy() price

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rowsBetween(-1, 0)

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rowsBetween(-1, 0)

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rowsBetween(-1, 0)

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rowsBetween(-1, 0)

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rowsBetween(-1, 0)

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rowsBetween(-1, 1)

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rowsBetween(-1, 1)

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rowsBetween(-1, 1)

ROW Frame Using rowsBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rowsBetween(-1, 1)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(-sys.maxsize, 0)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(-sys.maxsize, 0)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(-sys.maxsize, 0)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(-sys.maxsize, 0)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(-sys.maxsize, 0)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(0, sys.maxsize)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(0, sys.maxsize)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(0, sys.maxsize)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(0, sys.maxsize)

RANGE Frame Using rangeBetween()

Product	Store	Revenue
Bananas	Kent	6.5
Bananas	Bellevue	6.75
Bananas	Redmond	6.8
Bananas	Seattle	7
Potatoes	Redmond	4
Potatoes	Kent	4.5
Potatoes	Bellevue	5.5
Potatoes	Seattle	6.5

rangeBetween(0, sys.maxsize)

Demo

Windowing operations in Spark SQL

Summary

Spark SQL enables querying of DataFrames as database tables

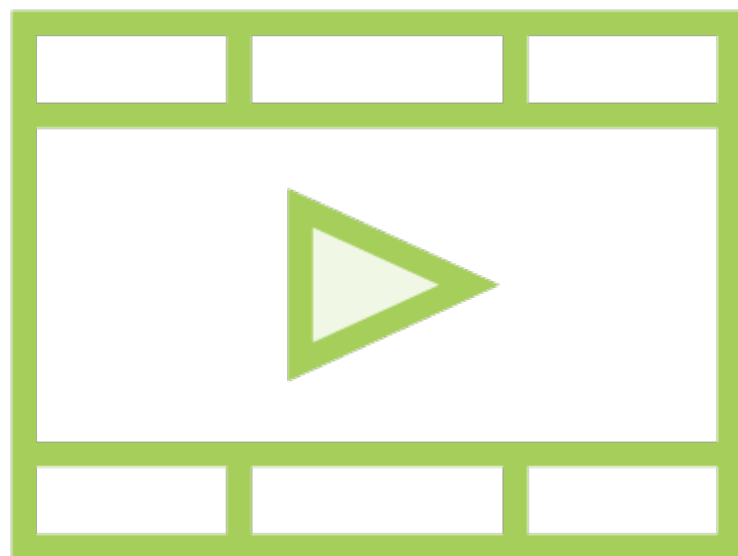
Temporary per-session and global tables

The Catalyst optimizer makes SQL queries fast

Tables schemas can be inferred or explicitly specified

Advanced window operations are also supported

Related Courses



Beginning Data Exploration and Analysis with Apache Spark

- Programming in Spark 1.x using Python

Handling Fast Data with Apache Spark SQL and Streaming

- Programming in Spark 2 using Scala