

Integrations:

Cassandra

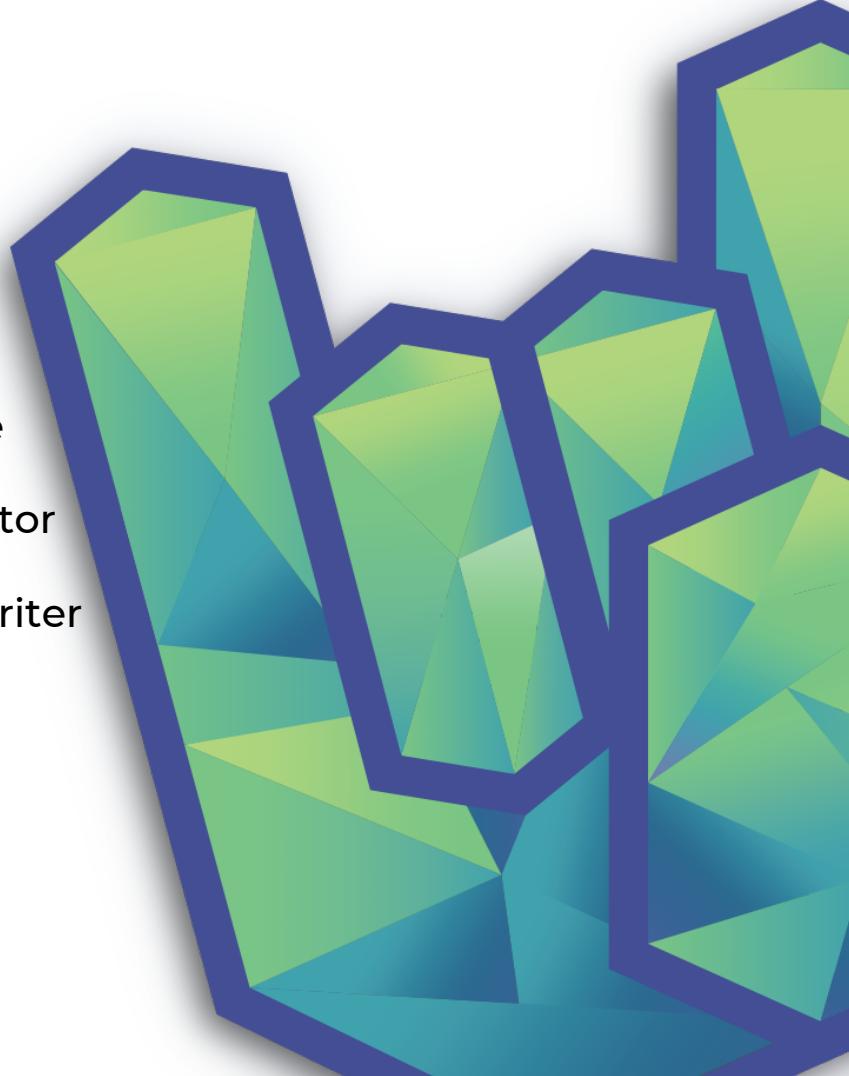


Objective

Integrate a Cassandra NoSQL distributed store

Learn how to use the Cassandra Spark connector

Learn advanced technique: custom ForeachWriter



Takeaways

```
class CassandraForeachWriter extends ForeachWriter[Car] {  
    // define private fields and methods  
    val connector = CassandraConnector(spark.sparkContext.getConf)  
  
    override def open(partitionId: Long, epochId: Long): Boolean = true  
  
    override def process(car: Car): Unit = {  
        connector.withSessionDo(session =>  
            session.execute(s"""insert into $keyspace.$table(name, horsepower)  
                values('${car.Name}', ${car.Horsepower.orNull})""")  
    }  
    }  
    override def close(errorOrNull: Throwable): Unit = {}  
}
```

```
carsDS.writeStream.foreach(new CassandraForeachWriter).start().awaitTermination()
```

For each partition (partitionId), for each batch (epochId):

- call the open method & skip the batch if returns false
- for each entry in the batch call the process method
- call the close method once done or when error is thrown

Spark rocks

