Pig, Hive, and Jaql have much in common. They all translate high-level languages into MapReduce jobs so that the programmer can work at a higher level than he or she would when writing MapReduce jobs in Java or other lower-level languages supported by Hadoop using Hadoop streaming. The high level languages offered by Pig, Hive and Jaql let you write programs that are much smaller than the equivalent Java code. When you find that you need to work at a lower level to accomplish something these high-level languages do not support themselves, you have the option to extend these languages, often by writing user-defined functions in Java. Interoperability can work both ways since programs written in these high-level languages can be imbedded inside other languages as well. Finally, since all these technologies run on top of Hadoop, when they do so, they have the same limitations with respect to random reads and writes and low-latency queries as Hadoop does.
Now, let us examine what is unique about each technology, starting with Pig. Pig was developed at Yahoo Research around 2006 and moved into the Apache Software Foundation in 2007. Pig's language, called PigLatin, is a data flow language - this is the kind of language in which you program by connecting things together. Pig can operate on complex data structures, even those that can have levels of nesting. Unlike SQL, Pig does not require that the data have a schema, so it is well suited to processing unstructured data. However, Pig can still leverage the value of a schema if you choose to supply one. Like SQL, PigLatin is relationally complete, which means it is at least as powerful as relational algebra. Turing completeness requires looping constructs, an infinite memory model, and conditional constructs. PigLatin is not Turing complete on its own, but is Turing complete when extended with User-Defined Functions.
Hive is a technology developed at Facebook that turns Hadoop into a data warehouse complete with a dialect of SQL for querying. Being an SQL dialect, HiveQL is a declarative language. Unlike in PigLatin, you do not specify the data flow, but instead describe the result you want and Hive figures out how to build a data flow to achieve it. Also unlike Pig, a schema is required, but you are not limited

to one schema. Like PigLatin and SQL, HiveQL on its own is a relationally complete language but not a Turing complete language. It can be extended through UDFs just like Pig to be Turing complete.

The final technology is Jaql. Jaql was developed at IBM. It is a data flow language like PigLatin but its native data structure format is JavaScript Object Notation, or JSON. Schemas are optional and the Jaql language itself is Turing complete on its own without the need for extension through UDFs.

Let us examine Pig in detail. Pig consists of a language and an execution environment. The language is called PigLatin. There are two choices of execution environment: a local environment and distributed environment. A local environment is good for testing when you do not have a full distributed Hadoop environment deployed. You tell Pig to run in the local environment when you start Pig's command line interpreter by passing it the -x local option. You tell Pig to run in a distributed environment by passing -x mapreduce instead. Alternatively, you can start the Pig command line interpreter without any arguments and it will start it in the distributed environment.

There are three different ways to run Pig. You can run your PigLatin code as a script, just by passing the name of your script file to the pig command. You can run it interactively through the grunt command line launched using pig with no script argument. Finally, you can call into Pig from within Java using Pig's embedded form.

As mentioned in the overview, Hive is a technology for turning Hadoop into a data warehouse, complete with an SQL dialect for querying it.

There are three ways to run Hive. You can run it interactively by launching the hive shell using the hive command with no arguments. You can run a Hive script by passing the -f option to the hive command along with the path to your script file. Finally, you can execute a Hive program as one command by passing the -e option to the hive command followed by your Hive program in quotes.

Jaql is a JSON-based query language that, like PigLatin and HiveQL, translates into

Hadoop MapReduce jobs. JSON is the data interchange standard that is human-readable like XML but is designed to be lighter-weight. You run Jaql programs using the Jaql shell. You start the Jaql shell using the jaqlshell command. If you pass it no arguments, you start it in interactive mode. If you pass the -b argument and the path to a file, you will execute the contents of that file as a Jaql script. Finally, if you pass the -e argument, the Jaql shell will execute the Jaql statement that follows the -e. There are two modes that the Jaql shell can run in: The first is cluster mode, specified with a -c argument. It uses your Hadoop cluster if you have one configured. The other option is minicluster mode, which starts a minicluster that is useful for quick tests.

This lesson is continued in the next video.