## Week 1 - Introduction to Spark

1. Big data applications address the “5 V’s”. Which of the following qualities of big data does that architecture address?
   1. Volume – the amount of data.
   2. Velocity – the speed of the data.
   3. Variety – the different types of data.
2. Why is Spark important?
   1. It connects to a large amount of databases. (Spark quickly rose in popularity by being able to connect to a large number of different data sources.)
   2. It works with SQL, Python, R and Scala/Java. (Spark has API’s in a number of different languages.)
   3. It distributes computation across a cluster of networked computers. (Spark is able to distribute work across a cluster of machines in order to handle large amounts of data.)
   4. It allows for analytics on big data. (Spark is the standard open source technology for doing data analytics on big data.)
3. How does Spark distribute work across a cluster of networked machines?
   1. The driver node of the cluster distributes data and computation to different executors. (The driver coordinates work for the different nodes on the cluster.)
4. When would you want to use Spark?
   1. When you need to scale out to larger amounts of data. (When you have larger amounts of data, Spark allows you to scale out across a cluster of machines.)
   2. When you need to speed up your queries. (By parallelising your computation, Spark is often able to achieve large improvements to the speed of queries.)
5. What is a Spark DataFrame?
   1. Distributed. (DataFrames store data distributed across a cluster.)
   2. Optimised. (DataFrames benefit from optimised queries.)
   3. What Spark SQL queries execute against. (Spark SQL code is actually executing against DataFrames under the hood.)
6. What’s the benefit of a unified approach to data analytics?
   1. The complexity of deploying open source software yourself is managed for you by Databricks. (Deploying software in distributed environments involves networking together many machines. This complexity is managed for you by Databricks.)
   2. Production jobs like extract, transform, load can be run on the same platform as machine learning, streaming and reporting. (It’s easy to schedule production jobs as well as do reporting, machine learning and other tasks common in big data environments.)
   3. Data analysts can do work on the same platform as data scientists and data engineers. (Since Spark is a standard tool for data analysts, data scientists and data engineers, they can all work together on a single platform.)
7. When you run code on Databricks, where does it run?
   1. On a cluster provisioned with Apache Spark for you. (Databricks hosted notebooks run code that is backed by a cluster that is created for you.)
8. Where can you upload data and view tables in the Databricks user interface?
   1. Under the “Data” tab. (The Data tab allows you to see all the tables you have available to you. You cannot see mounted data here.)

## Week 2 – Core Concepts/Spark Internals

1. The key to working on larger datasets is parallelism. What are units of parallelising computation?
   1. Machine. (Parallelism can happen across machines in a cluster.)
   2. Thread. (A thread refers to a single process running on a core.)
   3. Core. (Modern processors have multiple cores.)
2. Why is caching important?
   1. It reduces data transfer across the network. (Caching reduces how much data is needed to be transferred because that data is cached on the cluster.)
   2. It saved data that has already been seen. (Caching preserves a copy of the data it has already seen.)
3. What is a narrow transformation?
   1. Each executor acts only on the data that is on the partition. (In a narrow transformation, no data is moved across the cluster.)
4. What is a wide transformation?
   1. A transformation requiring data transfer between nodes in the cluster.
5. What is one of the biggest bottlenecks that the Spark UI can identify?
   1. An improper number of partitions resulting in poor query performance.
6. Why is the Spark UI important?
   1. It allows you to go under the hood in how Spark is processing queries. (The Spark UI is a great tool for debugging and otherwise figuring out how Spark is handling queries.)
7. A broadcast join…
   1. Moves only the smaller of the tables being joined. (A broadcast join broadcasts the smaller of the tables to the larger one.)

## Week 3 – Engineering Data Pipelines

1. Why are technologies like Apache Kafka or the managed alternatives Amazon Kinesis or Azure Event Hubs important?
   1. They offer a scalable way of receiving data in a pipeline. (These technologies act as a queue that allows a large volume of data to arrive in a data pipeline.)
2. Before data analysts and data scientists can query data, the ETL process prepares the data for being queried. What does ETL stand for?
   1. Extract, Transform, Load. (This process extracts raw data, transforms it to a state that it can be queried in, and loads it into a target database.)
3. There are two types of bottlenecks on computing tasks. One is IO-bound problems. What does this refer to?
   1. Data transfer across the network. (IO-bound problems refer to input/output problems of transferring data across a network.)
4. There are two types of bottlenecks on computing tasks. The other is CPU-bound problems. What does this refer to?
   1. Complexity of operations. (Since the CPU is performing an operation, the complexity of the operation partially determines how fast that task can be completed.)
5. What are the two main data workloads?
   1. Online Transaction Processing. (OLTP processes live transactions coming into a system.)
   2. Online Analytical Processing. (OLAP handles ad hoc queries, or queries the application wasn’t initially designed to handle.)
6. What is JDBC?
   1. A common protocol for connecting to databases? (Standing for Java Database Connectivity, it is the most common protocol for connecting to databases.)
7. Why do schemas matter?
   1. They allow Spark to use less storage space by using the right data type. (Using the right data type means more efficient encoding. For instance, a number as a string takes up more space than a number as an integer.)
   2. They let Spark know the format of your data so it can optimize your queries. (When Spark knows what your data looks like, it can better optimize your queries.)
   3. They help prevent failures in production because you know the types of your data. (Since Spark knows the type of data it will see, it is able to catch many errors before you go into production.)
8. What command gives you control of the parallelisation of data writes?
   1. SELECT /\*+ COALESCE(n) \*/ (This command creates uneven partitions of data and doesn’t shuffle your data.)
   2. SELECT /\*+ REPARTITION(n) \*/ (This command creates partitions of equal size. For instance, using repartition with a value of 8 gives 8 equal partitions.)
9. A managed table \_\_\_\_\_. An unmanaged table \_\_\_\_\_.
   1. A managed table manages the table itself and the metadata. An unmanaged table only manages the metadata. (A managed table manages the underlying data as well as metadata like the data location and schema information.)

## Week 4 – Data Lakes, Warehouses and Lakehouses

1. What’s true?
   1. Data lakes provide robust support for unstructured data types like images and videos. (Data lakes provide flexible support for a wide variety of different data types.)
2. What are the benefits of a lakehouse architecture?
   1. They integrate with BI, machine learning and other workloads. (Lakehouses support a wide variety of different workloads.)
   2. They support database-like transactions. (Lakehouses support database-like ACID transactions.)
   3. They combine the best aspects of data warehouses and data lakes. (Lakehouses combine the reliability and governance of data warehouses with the flexibility and low cost of data lakes.)
3. What is the purpose of the transaction log?
   1. Central repository that tracks all changes that users make to the table. (This provides a single source of truth for consistent reads from the table.)
   2. Provide atomicity guarantees. (If it’s not recorded in the transaction log, it never happened! This provides atomicity.)
4. Why is time travel useful?
   1. Audit data changes. (Auditing data changes is critical from both a compliance perspective as well as understanding how data has changed over time.)
   2. Rollback to an earlier state in the table. (Data pipelines can sometimes write bad data – we need a way to go back to recover an earlier state.)