## Module 1

1. What’s true when it comes to the business value of big data?
   1. The size of the data businesses collect is growing. (As the technology improves, businesses are collecting more and more data.)
   2. Businesses are increasingly making data-driven decisions. (More and more, businesses are seeing the value of driving decision-making using data.)
2. Spark uses…
   1. A driver node to distribute work across a number of executor nodes.
   2. A distributed cluster of networked computers made of a driver node and many executor nodes.
3. How does Spark execute code backed by DataFrames?
   1. It optimises your query by figuring out the best “how” to execute what you want. (Since Spark knows what you want to accomplish, it’s able to figure out the best way to do it.)
   2. It separates the “logical” plan of what you want to accomplish from the “physical plan” of how to do it so it can optimise the query. (Spark generates code on the fly to provide the most optimal way of serving your query.)
4. What are the properties of Spark DataFrames?
   1. Resilient: Fault-tolerant. (If you lose a worker, only recompute work that worker was responsible for.)
   2. Distributed: Computed across multiple nodes. (Each node computes on its own data.)
   3. Dataset: Collection of partitioned data. (The collection of data is partitioned so it can be distributed across the cluster.)
5. What’s the difference between Spark and database technologies?
   1. Spark is a highly optimised compute engine and isn’t a database. (Spark is a robust unified analytics engine and doesn’t act like a database.)
   2. Spark is a computation engine and isn’t for data storage. (Spark is a computation engine, whereas database technology is meant for data storage.)
6. What is Amdahl’s law of scalability?
   1. A formula that gives the theoretical speedup as a function of the percentage of a computation that can be parallelised.
   2. Amdahl’s law states that the speedup of a task is a function of how much of that task can be parallelised.
7. Spark offers a unified approach to analytics. What does this include?
   1. Spark code can be written in SQL, Scala, Java, Python and R. (Spark code can be written in a number of languages that are executed in the same way regardless of the language.)
   2. Spark unifies applications such as SQL queries, streaming and machine learning. (Spark works seamlessly with streaming, SQL, machine learning and graph processing.)
   3. Spark is able to connect to data where it lives in any number of sources, unifying the components of a data application. (Spark has a diverse set of connectors that can connect to data where it lives.)
   4. Spark allows analysts, data scientists and data engineers to all use the same core technology. (Spark provides a common framework for data analysts, data scientists and data engineers to all use the same technology and design patterns.)
8. What is a Databricks notebook?
   1. A collaborative, interactive workspace that allows you to execute Spark queries at scale. (A notebook is an interactive way of interacting with Spark code.)
9. How can you get data into Databricks?
   1. By registering the data as a table. (Data in Spark can be registered as its own table.)
   2. By uploading it through the user interface. (Uploading data through the user interface works well for small datasets.)
   3. By “mounting” the data backed by cloud storage. (Mounting data makes it appear in Spark as though the data were sitting on the cluster itself.)
10. What are the qualities of big data?
    1. Volume: the amount of data. (The amount of data is growing exponentially.)
    2. Variety: the diversity of data. (More and more different kinds of data are being processed by data applications.)
    3. Veracity: the reliability of data. (Data is not always reliable as it is sometimes user generated, poorly processed or with other problems.)
    4. Velocity: the speed of data. (The speed at which data arrives in architectures is growing exponentially.)

## Module 2

1. What are the different units of parallelism?
   1. Task. (A job can be divided into many tasks.)
   2. Executor. (An executor is one worker node in a cluster.)
   3. Core. (A processor has many cores.)
   4. Partition. (A partition is a subset of data.)
2. What’s a partition?
   1. A portion of a large distributed set of data. (Data distributed across the cluster is divided into different partitions.)
3. What’s the difference between in-memory computing and other technologies?
   1. In-memory operates from RAM while other technologies operate from disk.
   2. In-memory operations were not realistic in older technologies when memory was more expensive. (The price of memory has come down drastically, enabling Spark to rely on in-memory calculations.)
   3. Computation not done in-memory (such as Hadoop) reads and writes from disk in between each step. (Hadoop, the precursor to Spark, was much slower because it had to read from and write to disk between every step.)
4. Why is caching important?
   1. It stores data on the cluster to improve query performance. (By storing data we know we’ll see again, caching improves query performance.)
5. What are examples of wide transformations?
   1. Order by. (An order by transfers data across the network, and is therefore a wide transformation.)
   2. Group by. (A group by transfers data across the network, and is therefore a wide transformation.)
6. Broadcast joins…
   1. Transfer the smaller of the two tables to the larger, minimising data transfer.
7. Adaptive Query Execution uses runtime statistics to:
   1. Dynamically coalesce shuffle partitions.
   2. Dynamically switch join strategies.
   3. Dynamically optimise skew joins.
8. What are bottlenecks that you can detect with the Spark UI?
   1. Data skew. (Data skew is when partitions are not of similar sizes and can be detected by the Spark UI.)
   2. Shuffle reads. (The Spark UI can show shuffles triggered by Spark actions.)
   3. Shuffle writes. (The Spark UI can show shuffles triggered by Spark actions.)
9. What is a stage boundary?
   1. When all the slots or available units of processing have to sync with one another. (A stage boundary is when all Spark tasks must come together to exchange a result.)
10. What happens when Spark code is executed in local mode?
    1. The executor and driver are on the same machine. (Local mode refers to when the executor and driver are on the same machine, such as when prototyping Spark code on your laptop.)

## Module 3

1. Decoupling storage and compute means storing data in one location and processing it using a separate resource. What are the benefits of this design principle?
   1. Resources are isolated and therefore more manageable and debuggable. (New database and computation versions can be installed on new hardware due to the ephemeral nature of the underlying data.)
   2. It allows for elastic resources so larger storage or compute resources are used only when needed. (Decoupled resources that aren’t utilised can easily be shut down.)
   3. It makes updates to new software versions easier. (With each component of the architecture responsible for specific tasks, debugging is significantly easier.)
2. You want to run a report entailing summary statistics on a large dataset sitting in a database. What is the main resource limitation of this task?
   1. IO: The transfer of data is more demanding than the computation. (The main bottleneck here is the transfer of data across the network.)
3. Processing virtual shopping cart orders in real time is an example of…
   1. Online Transaction Processing (OLTP). (Processing real time information involves transactional processing.)
4. When are BLOB stores an appropriate place to store data?
   1. For cheap storage. (BLOB stores are significantly cheaper than databases.)
   2. For storing large files. (BLOB stores scale effectively infinitely.)
   3. For a “data lake” of largely unstructured data. (BLOB stores are the backbone of most data lakes.)
5. JDBC is the standard protocol for interacting with databases in the Java environment. How do parallel connections work between Spark and a database using JDBC?
   1. Specify a column, number of partitions, and the column’s minimum and maximum values. Spark then divides that range of values between parallel connections. (Spark uses the max and min of a range of values to know which connection should receive which data.)
6. What are some of the advantages of the file format Parquet over CSV?
   1. Compression. (Parquet is compressed by default and has many additional compression options.)
   2. Parallelism. (Parquet is column-based rather than a row-based format.)
   3. Columnar. (Parquet easily parallelised so one file is written per Spark connection.)
7. SQL is normally used to query tabular (or “structured”) data. Semi-structured data like JSON is common in big data environments. Why?
   1. It allows for data change over time. (JSON allows for schema evolution over time.)
   2. It does not need a formal structure. (No formal structure is needed to be declared in advance like with relational tables.)
   3. It allows for complex data types. (Complex types like arrays are allowed in JSON.)
   4. It allows for missing data. (JSON doesn’t require all keys to appear in a dataset.)
8. Data writes in Spark can happen in serial or in parallel. What controls this parallelism?
   1. The number of data partitions in a DataFrame. (Controlling the data partitions controls the parallelism of data writes.)
9. Fill in the blanks with the appropriate response below: A \_\_\_ table manages \_\_\_and a DROP TABLE command will result in data loss.
   1. Managed, both the data and metadata such as the schema and data location. (When dropping a managed table, the underlying data will be deleted too.)

## Module 4

1. What are the ACID properties?
   1. Atomicity, Consistency, Isolation and Durability. (Taken together, these properties guarantee data validity despite any errors or mishaps.)
2. What’s true about data warehouses?
   1. They provide the structure needed for BI applications. (Repeated BI applications like reports are one of the primary focuses of data warehouses.)
   2. They use closed protocols and proprietary software. (Many data warehouses use closed protocols and proprietary software rather than open source standards.)
3. What features does Delta Lake support?
   1. Delete. (Delete unwanted records.)
   2. Time travel. (You can travel back to an earlier state of the Delta table.)
   3. Schema evolution. (Allow the schema to evolve over time.)
4. What’s true about data lakes?
   1. They enable machine learning workloads. (Data lakes support a wide variety of datasets used in machine learning.)
   2. They have a high degree of flexibility. (Data lakes are flexible given that they allow for most file types and any schema.)
5. What are valid data models?
   1. Relational. (Relational models use normalisation to organise data in a database.)
   2. Non-relational. (Non-relational models allow for flexible storage of items such as documents.)
   3. Query-oriented. (Query-oriented modelling optimises for the speed of queries.)
   4. Star. (Star schemas organise data into fact and dimension tables.)
   5. Medallion. (Medallion architectures use bronze/silver/gold tables of increasingly valuable data.)
6. What are the benefits a lakehouse architecture provides?
   1. Combine scalability and low-cost storage of data lakes with the speed and ACID transactional guarantees of data warehouses. (With a lakehouse, you’re able to get the best of both worlds.)
7. Machine learning is suited to solve what?
   1. Image recognition. (Neural networks are effective for operating on image data.)
   2. Fraud detection. (Web traffic data can be used to predict fraudulent transactions.)
   3. Natural Language Processing. (NLP applies statistical methods to language.)
   4. Financial forecasting. (Time series analyses are commonly applied to financial data.)
   5. A/B Testing. (ML optimises between different versions of websites or emails.)
   6. Churn analysis. (Churn analysis predicts customer engagement.)
8. What’s machine learning?
   1. Learning patterns in your data without being explicitly programmed. (ML uses linear algebra and calculus to learn patterns in data without being explicitly programmed.)
   2. A function that maps features to an output. (ML maps input features to an output.)
9. Predicting whether a website user is fraudulent or not is an example of supervised machine learning. It’s a classification task. (In this case, whether the user is fraudulent is the dependent variable and we are classifying fraudulent from non-fraudulent users.)
10. Linear regression is one algorithm used for machine learning. What is this algorithm learning.
    1. It learns the line of best fit through the data. (Linear regression learns the line that best fits the relationship between input features and the label.)