### [https://docs.google.com/document/d/1r4IR4FowwqAecjUMRaBJJbghhGtUtstTuximt1QIH8w/edit#](https://docs.google.com/document/d/1r4IR4FowwqAecjUMRaBJJbghhGtUtstTuximt1QIH8w/edit)

### Java

1. What is Encapsulation?
   1. Encapsulation is defined as the wrapping up of data under a single unit.
   2. The variables or data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared. As in encapsulation, the data in a class is hidden from other classes, so it is also known as data-hiding.
   3. Encapsulation can be achieved by: Declaring all the variables in the class as private and writing public methods in the class to set and get the values of variables.
2. What is Abstraction?
   1. Abstraction is a process of hiding the implementation details from the user. Оnly the functionality will be provided to the user. In Java, abstraction is achieved using abstract classes and interfaces.
3. What is Inheritance?
   1. It is the mechanism in java by which one class is allowed to inherit the features(fields and methods) of another class. To inherit from a class, use the extends keyword.
      1. subclass (child) - the class that inherits from another class
      2. superclass (parent) - the class being inherited from
   2. Represents the IS-A relationship
4. What is Polymorphism?
   1. Any Java object that can pass more than one IS-A test is considered to be polymorphic. In Java, all Java objects are polymorphic since any object will pass the IS-A test for their own type and for the class Object.
5. What is Casting?
   1. Casting tells the JVM to treat an Object like another Object
   2. Casting can be widened and narrowed
      1. Widening casting is done automatically and is converting a smaller type to a larger type.
      2. Narrowing casting is done manually and converts a larger type to a smaller type.
   3. Objects that share an IS-A relationship with another class or interface can be casted to each other’s type
   4. Some rules of Casting:
      1. Casting an object from a subclass to a superclass doesn’t require an explicit cast.
      2. Casting an object from a super class to a sub class requires an explicit cast.
      3. The compiler will not allow casts to unrelated types.
      4. Even when the code compiles without issue, an exception may be thrown at run time if the object being cast is not actually an instance of that class. This will result in the run time exception `ClassCastException`.
6. How many objects on the heap if I cast?
   1. Unless the new keyword is used, only one object will be on the heap.
   2. Casting only tells the JVM to treat the Object like another Object, not that it is of that other Object’s type
7. What is an Interface?
   1. An interface can have methods and variables, but the methods declared in the interface are by default abstract
   2. Interfaces specify what a class must do, not how. It is a blueprint for a class
   3. If a class implements an interface but does not provide method bodies for all functions specified in the interface, then the class must be declared abstract
8. Can I instantiate an Abstract class? Constructor?
   1. Abstract classes cannot be instantiated, but they can be subclassed. When an abstract class is subclassed, the subclass usually provides implementations for all of the abstract methods in its parent class. However, if it does not, then the subclass must also be declared abstract.
   2. Abstract classes can, however, have constructors. When we create an object of any subclass all the constructors in the corresponding inheritance tree are invoked in the top to bottom approach. The same case applies to abstract classes. Though we cannot create an object of an abstract class, when we create an object of a class which is concrete and subclass of the abstract class, the constructor of the abstract class is automatically invoked.
9. Why would you use an Abstract class over an Interface?
   1. An abstract class allows you to create functionality that subclasses can implement or override. An interface only allows you to define functionality, not implement it. And whereas a class can extend only one abstract class, it can take advantage of multiple interfaces.
10. equals() vs ==?
    1. equals() is an object comparison ( method)
    2. == is a reference comparison (operator)
11. Can I force garbage collection?
    1. No, you can only encourage it with System.gc()
12. Which method does the garbage collector call?
    1. The Garbage Collector calls finalize() on an Object
13. What is the finally block?
    1. Finally is a block of code that executes after the try block exits.
14. Is a catch block needed?
    1. A catch block is not needed. You can have try and try-finally
    2. While you don't need a catch block, it means you can't handle the exceptions thrown in the try block
15. What are Generics for?
    1. Generics are for when you are not certain of the type being received
    2. Also known as Parametric Polymorphism
    3. Compile time safety
16. Comparable vs Comparator
    1. Comparable
       1. A comparable object is capable of comparing itself with another object. The class itself must implements the java.lang.Comparable interface to compare its instances.
       2. To use, we can implement the Comparable interface with the Object class, and we override the method compareTo() of Comparable interface.
       3. If sorting of objects needs to be based on natural order then use Comparable.
    2. Comparator
       1. Comparator is external to the element type we are comparing. We create multiple separate classes (that implement Comparator) to compare by different members. Collections class has a second sort() method and it takes Comparator. The sort() method invokes the compare() to sort objects.
       2. To use, create a class that implements Comparator (and thus the compare() method that does the work previously done by compareTo())
       3. If your sorting needs to be done on attributes of different objects, then use Comparator.
17. Hashtable vs Hashmap
    1. HashMap and Hashtable store key/value pairs in a hash table. When using a Hashtable or HashMap, we specify an object that is used as a key, and the value that you want linked to that key. The key is then hashed, and the resulting hash code is used as the index at which the value is stored within the table. But...
    2. Hashtable
       1. is synchronized. It is thread-safe and can be shared with many threads.
       2. doesn’t allow any null key or value. To successfully store and retrieve objects from a HashTable, the objects used as keys must implement the hashCode method and the equals method. Since null is not an object, it can’t implement these methods.
    3. HashMap
       1. is non synchronized. It is not-thread safe and can’t be shared between many threads without proper synchronization code.
       2. allows one null key and multiple null values. HashMap is an advanced version and improvement on the Hashtable. HashMap was created later.
18. How do I start a thread implementing Runnable?
    1. Create a Runnable implementer and implement run() method.
    2. Invoke start() of Thread instance. Start internally calls run() of the implementer. Invoking start() creates a new Thread which executes the code written in run().
    3. Calling run() directly doesn’t create and start a new Thread, it will run in the same thread. To start a new line of execution, call start() on the thread.
19. Thread methods. (run, start, sleep, wait, notify, notifyAll, join).
    1. getName - Obtain thread’s name
    2. getPriority - Obtain thread’s priority
    3. isAlive - Determine if a thread is still running
    4. join - Wait for a thread to terminate
    5. run - Entry point for the thread
    6. sleep - Suspend a thread for a period of time
    7. start - Start a thread by calling its run method
    8. wait() - It tells the calling thread to give up the lock and go to sleep until some other thread enters the same monitor and calls notify().
    9. notify() - It wakes up one single thread that called wait() on the same object. It should be noted that calling notify() does not actually give up a lock on a resource.
    10. notifyAll() - It wakes up all the threads that called wait() on the same object.
20. Difference between the run and start method?
    1. start(): creates a new thread object
    2. run(): has the code executed on the thread
21. What is a deadlock?
    1. Where two or more threads are blocked from further execution forever, waiting for each other. They need the same locks but obtain them in a different order. The synchronized keyword can cause the executing thread to block while waiting for the lock on a specified object.
22. Which is the fastest File IO available and why?
    1. MappedByteBuffer is the fastest IO in Java. It is used for reading binary files that are not read sequentially- like reading a database index file containing data with offsets to other parts of the file. However, this is a very specific case.
23. Which are the scopes of a variable?
    1. Method
    2. Block
    3. Class
    4. Instance
24. Can you override static methods?
    1. You cannot override static methods in Java. While you can technically have two static methods with the same definition, it is not true run-time polymorphism because it actually uses method hiding.
25. What is shadowing?
    1. Shadowing is the practice of using 2 variables with the same name within scopes that overlap
    2. When this occurs, the variable with the higher-level scope is hidden because the variable with the lower-level scope overrides it
    3. The higher-level variable is then shadowed
26. What are Wrapper classes?
27. Wrapper classes are objects that wrap around or contain a primitive data type.
28. What is Varargs used for?
    1. Varargs is used for when you are not certain about the number of arguments a method can receive
29. What is the difference between protected and default?
    1. Protected provides access to all subclasses regardless of the package and to other code that is in the same package and default provides access to code only within the package.
30. What is the final keyword used for?
    1. For declaring a variable as constant in Java. Also for use in a method where the methods are not to be overridden in any circumstances. Final classes cannot be extended.
31. What is the difference between StringBuilder and StringBuffer?
    1. StringBuilder is used to create a String-like object that is capable of variable length (i.e. what an ArrayList offers over an array). StringBuffer is the thread-safe version of this object.
32. What is synchronization?
    1. The keyword synchronized allows us to use synchronization; that is, its purpose is to only allow one thread at a time into a particular section of code or resources, thus allowing for protection from multiple operations on this resource at once. Used with multithreaded programs to allow them to function correctly.
33. How do you go about starting a thread?
    1. Create a class that implements the Runnable interface.
    2. Implement the method within the interface, called run().
    3. Execute the run() method by passing an instance of this class into a Thread constructor, then call the start() method.
34. What is the difference between a List and a Set?
    1. List
       1. An ordered collection that maintains the elements in insertion order
       2. Allows for duplicate elements
       3. Allows any number of null values
       4. Traversed forward or backward using Listiterator
    2. Set
       1. An unordered collection
       2. Does not allow duplicate elements - will replace instead of append
       3. Allows only one null value
       4. Traversed forward using iterator
35. Some concrete implementations of Set.
    1. Iterable (interface)
       1. Collection (class)
          1. Set (interface)
             1. HashSet

Set with a hash table (unpredictable iteration order).

* + - * 1. LinkedHashSet

Set with a hash table with linked list behavior (predictable iteration order).

* + - * 1. TreeSet

Set with elements ordered by natural ordering using a Comparator.

1. LinkedList vs ArrayList
   1. LinkedList
      1. A doubly linked list
      2. Better performance on add and remove
   2. ArrayList
      1. A resizable array with dynamic size allocation
      2. Elements accessible directly with get and set methods
      3. Better performance on get and set
2. How do you insert elements in a Map?
   1. mapName.put(K, V)
3. Difference between Exception and Error?
   1. Exception
      1. Indicates conditions that an application might want to catch
      2. Can be recoverable if caught
   2. Error
      1. Indicates serious problems that an application cannot reasonably catch
      2. Are generally unrecoverable
4. What is Serialization?
   1. The process of converting the state of an Object into a byte stream. Can be done by implementing the Serializable interface. This is useful for sending it over the network, or for file I/O.
5. What is a Singleton?
   1. A design pattern that denotes that only one instance of the Object can be allowed at any one time, and all references point to this one instance only. Essentially, it only exists in memory once.
   2. Usually implemented by using a private constructor, a static field with its only instance, and a static method for obtaining the instance.
6. What is the IS-A rule?
   1. IS-A is a relationship between abstractions wherein one class is a subclass of another class.
   2. Eg. Class A is a subtype of Class B so, Class A is also Class B
7. Where are variable references stored?
   1. Primitives are stored on the Stack
   2. Objects are stored on the Heap
8. When is an object ready for garbage collection?
   1. Its reference variable is lost or removed from memory (it is an unreachable Object).
9. What is reflection?
   1. Reflection enables Java code to discover information about the fields, methods, and constructors of loaded classes, and to use reflected fields, methods, and constructors to operate on their underlying counterparts within security restrictions
   2. The reflection API accommodates applications that need access to either the public members of a target object (based on its runtime class) or the members declared by a given class
10. What makes the String class special?
    1. Strings have their own dedicated space on the heap called the String pool.
    2. Strings are also immutable.
11. What is the difference between an Exception and a RuntimeException?
    1. Exceptions
       1. Are checked
       2. Must be handled explicitly by the code
       3. Code needs either a try/catch block, or add `throws` clause to a method to indicate that the exception will not be handled in the method that causes the exception
    2. RuntimeExceptions
       1. Are unchecked
       2. Does not need to be explicitly handled
12. Rules of the catch block?
    1. Most specific exception must be handled first, otherwise, the catch block with the more general exception will be called first
    2. If a single catch block handles more than one exception type (using | ), then the catch parameter is implicitly final.
    3. The argument must be of type ExceptionType, which declares the type of an Exception that the handler can handle and must be the name of a class that inherits from the Throwable class.
13. What does the Iterable interface do?
    1. Allows an object to be the target of the “for-each loop” statement. (enhanced for loop)
14. HashSet vs TreeSet
    1. HashSet
       1. Faster than TreeSet on average( O(1) )
       2. Implemented using a hash table
       3. Is not ordered
       4. Allows nulls
       5. Uses equals() method to compare 2 objects in Set and for detecting duplicates
    2. TreeSet
       1. Takes O(Log n) time on average
       2. Implemented using a Self Balancing Binary Search Tree (red-black tree)
       3. Maintains objects in Sorted order defined by either a comparable or comparable methods in Java
          1. Has methods to work with the ordered set like first(), last(), headSet(), tailSet(), etc
       4. Does not allow nulls
       5. Uses compareTo() to compare 2 objects in Set and for detecting duplicates
15. What is a Map?
    1. A Map is an Object that maps keys to values
    2. A Map cannot contain duplicate keys
    3. Each key must map to at most one value
16. Can I sort a Map?
    1. While you may be able to sort the contents of a Map with a helper data structure and put them back into a map, for all intents and purposes, no you cannot sort a Map. It is also an associative data structure which does not rely on order, so it does not make sense to think about sorting.
17. How do I start a Thread extending the Thread class
    1. You start a Thread extending the Thread class by using the start() method
    2. Example: myThread.start()
18. What is Starvation?
    1. Starvation is when a thread is constantly ignored to gain possession of the intrinsic lock in favor of other threads
19. InputStream vs Reader
    1. InputStream
       1. Reads raw, binary data byte by byte
       2. Uses no encoding or translation, good for images and binary streams
    2. Reader
       1. Reads text (character streams)
       2. Use character encoding to decode bytes and return characters to caller
20. When is a class fully synchronized?
    1. When a class has all of its methods and variables synchronized, i.e:
       1. Threads cannot interfere with each other
       2. Threads cannot access the same data at once in such a way that would cause memory inconsistency errors
21. Can an Interface have variables? What are they?
    1. Yes, an interface can have variables.

All variables declared in an interface must be public, static and final.

1. What is an Abstract class?
   1. An abstract class is a class that is declared with the abstract keyword. An abstract class may or may not have all abstract methods. Some of them can be concrete methods. There can be no object of an abstract class. An abstract class can not be directly instantiated with the new operator.
   2. An abstract method is a method that is declared without an implementation. A method defined abstract must always be redefined in the subclass,thus making overriding compulsory OR either make subclass itself abstract.
2. What are default methods in an interface?
   1. Java 8 allows default methods in an interface. This allows methods to have implementation in an interface without affecting the classes that implement the interface. Default methods were introduced so existing interfaces can use lambda expressions without implementing the methods in an implementation class.
3. What's the first line in a constructor?
   1. this() or super() must be the first line in a constructor
      1. super() must be the first method used before any this() so that we can ensure methods called from the parent class are set up correctly/ make sure the parent class is fully formed before the construction of the subclass
      2. super() is added by the compiler if you do not specify it first
4. What is constructor chaining?
   1. Constructor chaining is the process of calling one constructor from another constructor with respect to current object.
   2. Constructor chaining can be done in two ways:
      1. Within the same class: It can be done using this() keyword for constructors in the same class
      2. From base class: by using super() keyword to call constructor from the base class.
5. What is a short circuit operator?
   1. && and ||
   2. A short circuit operator is one that doesn’t necessarily take into account all of its operands.
      1. Example: if (0==2 && 2+2=4) {do something;}
         1. Since 0==2 would return false, the right side operand would not even be evaluated.
6. Where are Strings stored in memory?
   1. In the String Pool, which is located on the heap.
7. What is hashCode() for?
   1. The hashcode of a Java object is a 32 bit int that allows the data to be managed by a hash-based data structure
   2. The hashcode() method returns the integer hashcode value of an object.
8. What's the parent of all exceptions?
   1. Exception is the parent class of all exceptions (all exceptions extend Exception)
   2. Throwable (interface)is the parent of Exception
9. Can I catch an Error? Does it make sense?
   1. You can catch an error, but it does not make sense to do that. Errors are serious issues with the environment in which the Java code is running in, such as StackOverflowError or OutOfMemoryError, so you would not want to catch these things.
10. Is there any case the finally block won't execute?
    1. If the JVM exits while the try or catch block is running
    2. If the thread executing the try or catch block is killed, the finally block may not execute.
11. Array vs ArrayList
    1. Array is a fixed length data structure (can store primitives and objects)
    2. ArrayList is a variable length Collection class (can only store objects)
12. What is a Thread?
    1. A thread is an independent path of execution within a program. Many threads can run concurrently within a program. Every thread in Java is created and controlled by the java.lang.Thread class.
13. What is a Factory?
    1. A type of design pattern in java
    2. In class-based programming, the factory method pattern is a creational pattern that uses factory methods to deal with the problem of creating objects without having to specify the exact class of the object that will be created. This is done by creating objects by calling a factory method—either specified in an interface and implemented by child classes, or implemented in a base class and optionally overridden by derived classes—rather than by calling a constructor.
14. How do I make an object Serializable?
    1. Implement Serializable interface ( a marker interface which tells Java that this object is serializable)

### 

### SQL

1. All SQL Sub Languages.
   1. DQL (Data Query Language)
      1. SELECT
   2. DDL (Data Definition Language)
      1. CREATE
      2. ALTER
      3. DROP
      4. TRUNCATE
      5. RENAME
   3. DML (Data Manipulation Language)
      1. INSERT
      2. UPDATE
      3. DELETE
   4. TCL (Transaction Control Language)
      1. COMMIT
      2. ROLLBACK
      3. SAVEPOINT
   5. DCL (Data Control Language)
      1. GRANT
      2. REVOKE
2. Truncate vs Delete.
   1. TRUNCATE (DDL) de-allocates pages and makes an entry for deallocation of pages in the transaction log; is faster than DELETE and does not fire triggers.
   2. DELETE (DML) deletes records one by one and makes an entry for each and every deletion in the transaction log.
   3. Delete transactions can be rolled back (undone) while truncate cannot be rolled back.
3. DATE vs TIMESTAMP.
   1. DATE is used for values with a date part but no time part. MySQL retrieves and displays DATE values in YYYY-MM-DD format.
      1. There is a DATE TIME that has hour-minute-second
   2. TIMESTAMP is used for values that contain both date and time parts. It takes the '2019-01-16 12:15:00' value you are setting into it and interprets it in the current time zone to compute an internal representation relative to 1/1/1970 00:00:00 UTC. When the column is displayed, it will be converted back for display based on whatever the current time zone is.
4. What is Referential Integrity?
   1. The state of congruence in data between different locations in a database.
   2. Referential integrity refers to the accuracy and consistency of data within a relationship. In relationships, data is linked between two or more tables.

This is achieved by having the foreign key (in the associated table) reference a primary key value (in the primary – or parent – table)

1. WHERE vs HAVING.
   1. WHERE finds individual instances where a particular value matches the condition given.
   2. HAVING assembles groups of data based on the condition given.
2. Scalar Functions.
   1. Scalar functions are those that operate on a one value argument and returns one value based on the given argument. Examples include LEN(), UCASE(), LCASE(), etc.
3. INNER JOIN vs INTERSECT.
   1. INTERSECT does not include duplicates while INNER JOIN does. INNER JOIN will not include nulls, but INTERSECT will.
4. Can I do a subquery in an INSERT statement?
   1. Yes; INSERT INTO table1 SELECT \* FROM table2;
5. What does the 1NF say?
   1. One primary key present and no more than one value per cell. The data is Atomic.
6. Properties of a transaction (ACID)
   1. Atomicity
      1. All or nothing for the transaction; either the entire transaction takes place or none of it does.
   2. Consistent
      1. Data integrity is maintained so that it is consistent before and after the transaction.
   3. Isolated
      1. Multiple transactions can occur simultaneously without interfering or causing inconsistency of the database state. They can occur without conflicts.
   4. Durability
      1. Once the transaction has completed execution, the updates are stored and written to disk, persisting even if some system failure occurs. Updates are permanent and are stored in non-volatile memory spaces.
7. What is Atomicity?
   1. All or nothing for the transaction; either the entire transaction takes place or none of it does.
8. What is a Cursor? What can I use it for?
   1. A control structure that enables traversal over a database. Useful for retrieval, addition and deletion of records.
9. Which are the main interfaces of JDBC?
   1. DriverManager, Connection, Statement, PreparedStatement, ResultSet
10. ALTER vs UPDATE.
    1. ALTER changes the entire schema of a table and is particular to DDL (data definition language) while UPDATE only changes data in the table and belongs to DML (data manipulation language).
11. CLOB vs BLOB.
    1. Character large object (usually text) is an object encoded in characters while a binary large object (could be audio, video, other unstructured data) is one encoded in binary.
12. What is a Primary Key?
    1. A primary key is a candidate key that uniquely identifies each row of a table.
13. GROUP BY vs ORDER BY.
    1. GROUP BY aggregates the records by the specified columns, allowing you to aggregate on non-grouped columns (like SUM, COUNT, AVG).
    2. ORDER BY alters the order in which items are returned.
14. Aggregate Functions.
    1. AVG - average of a set of values
    2. COUNT - counts rows in a specified table or view
    3. MIN - gets the minimum value in a set of values
    4. MAX - gets the maximum value in a set of values
    5. SUM - calculates the sum of values
15. FULL JOIN vs UNION.
    1. FULL JOIN - combine data into new columns; returns all records when there’s a match within either table.
    2. UNION - combine data into new rows; used to combine the result-set of two or more SELECT statements, where each SELECT within the UNION must have the same number of columns and similar data types.
16. Can I do a subquery in the FROM clause of a SELECT statement? Why?
    1. Yes; this is called an inline view. While treated as a view, it is not actually a view.
17. What does the 2NF say?
    1. It is in 1NF.
    2. It does not have any columns that are dependent on anything but the candidate key; that is, everything is dependent entirely on the primary/composite key.
18. What is Isolation?
    1. (Part of ACID) Determines when and how changes made by one transaction become available/visible to other transactions. There are levels of data isolation, where lower levels increases the ability of multiple users to access the same data at the same time, and higher levels reduce the effects of concurrency on users. However, higher data isolation levels use much more resources than lower levels.
19. What are the different Isolation levels?
    1. Read Uncommitted
       1. Transactions can be read even when changes have not been committed from a different transaction.
       2. There is no isolation between transactions, and this leads to dirty reads where uncommitted changes mix together before transactions are fully completed.
    2. Read Committed (Most DBs use as default)
       1. Transactions can read only committed changes, and solves dirty reads. It uses a read/write lock on the row it is reading, therefore preventing other transactions from modifying or deleting this data until it is done with it.
       2. This leads to non-repeatable reads since write locks are kept until the transaction is completed but read locks are released as soon as SELECT operations are performed- therefore reading the same data twice can lead to differing reads.
    3. Repeatable Read
       1. Transactions can only see completely committed changes. It accomplishes this with a ‘read view’, which is created at the start of the transaction, and this read view (consistent snapshot in Oracle terms) is held open for the duration of the transaction.
       2. This leads to phantom reads, where, in the same transaction, identical queries return different rows. The reason for this behavior is that between the first and the second execution of the query, new rows have been inserted into the table by other transactions- with completely committed changes.
    4. Serializable
       1. The Serializable SQL Server isolation level protects from all three consistency anomalies (dirty reads, non-repeatable reads, phantom reads) and this is done by placing range locks in the range of key values that match search conditions for all statements in the transaction. These locks are held until the transaction ends. This ensures that not only dirty reads and not-repeatable reads are prevented, but also phantom reads are excluded.
20. What is a Trigger? When can it execute?
    1. A SQL trigger is a set of SQL statements stored in the database catalog. A SQL trigger is executed or fired whenever an event associated with a table occurs e.g., insert, update or delete. A SQL trigger is a special type of stored procedure. It is special because it is not called directly like a stored procedure.
21. What do I need to connect to a DB with JDBC?
    1. The proper JDBC driver which corresponds with the type of database you want to connect to (such as OJDBC used to connect to Oracle databases).
22. CREATE vs INSERT.
    1. CREATE is part of DDL. It is used to create new tables/views/databases (declare column names and types)
    2. INSERT is part of DML. it is used to add new data into a table after the table has already been created.
23. What does the difference between two DATES return?
    1. You can simply subtract two DATES in sql, if you want the difference between two DATES use the DATEDIFF() function.
24. What is a Foreign Key?
    1. A foreign key is a column in a table that provides a link to another table in the database, and ensures referential integrity of the data.
    2. A foreign key references the primary key of another table.
25. WHERE vs HAVING.
    1. WHERE
       1. Works on a row’s data, but not aggregated data
       2. pre-filter
    2. HAVING
       1. Works on aggregated data
       2. post-filter
26. LEFT JOIN vs MINUS.
    1. LEFT JOIN
       1. Returns all records from the left table and the matched records from the right table
    2. MINUS
       1. Compares the results of 2 queries and returns distinct rows from the first query that aren't output by the second query
27. Can I do a subquery in an UPDATE statement?
    1. Yes, you can use a subquery in an UPDATE statement for the values to put in the columns
28. What does the 3NF say?
    1. Every non primary key column is not reliant on another non primary key column
    2. Everything is totally reliant on the primary key
29. What is Consistency?
30. What is the Serializable Isolation level?
31. Stored Procedure vs UDF.
32. Statement vs PreparedStatement.
33. DROP vs DELETE.
34. VARCHAR vs VARCHAR2.
35. What is a composite Primary Key?
36. GROUP BY vs ORDER BY.
37. What is a SELF JOIN?
38. UNION vs UNION ALL.
39. Can I do a subquery in a DELETE statement?
40. What is Normalization?
41. What is Durability?
42. What is a Phantom read?
43. What is a Dirty read?
44. What is a Sequence?
45. Connection vs DriverManager.
46. INNER JOIN vs OUTER JOIN.
47. What is a theta join?
    1. A theta join is a join where you use an operator other than equals (=) to perform a join
48. What is a natural join?
    1. Compares the columns of the tables and performs an Equi Join, which can be inner or outer
    2. The join column’s name should be identical on both tables
49. All types of joins. You can get asked differences between any of them.
50. What is an index?
51. Clustered vs Unclustered index.
52. What is a view?
53. View vs Materialized view.
54. Unary, Binary and Ternary relations.
    1. The degree of a relationship is the number of entity types that participate in the relationship
    2. Unary
       1. When both participants in the relationship are the same entity
    3. Binary
       1. When two entities participate, it is also the most common relationship degree
    4. Ternary
       1. When three entities participate in the relationship
55. All sections of the SELECT statement.
    1. SELECT <columns> FROM <table> WHERE <condition> GROUP BY <columns> HAVING <condition> ORDER BY <columns>
56. The number of records in a table emp is given by the SQL statement
    1. SELECT COUNT(\*) FROM emp;
57. What term is used to describe records in a database?
    1. ROWS
58. T/F: Each Database can hold only one table.
    1. False; of course a single database can hold multiple tables.
59. T/F: A foreign key cannot reference a column in the table where it is defined.
    1. False; foreign keys can be from the same table that they’re in.
60. For what column types are indexes generated automatically?
    1. UNIQUE
    2. PRIMARY KEYS

### Business Intelligence

1. What is OLTP?
   1. On-Line Transaction Processing. Characterized by large number of short on-line transactions like INSERT, UPDATE, and DELETE. The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second.
   2. Purpose is to run fundamental business tasks.
2. What is OLAP?
   1. On-Line Analytical Processing. Relatively low volume of transactions. Queries are often complex and involve aggregations. This is to help with planning, problem solving, and decision support.
3. What is a Data Warehouse?
   1. A (generally very large) repository of data from one or more sources in which current and historical data is stored. Can be used for analysis (analytical reports) and more.
4. What is the standard structure of tables in a Data Warehouse?
   1. Denormalized (better performance for reading data)
   2. Usually stored in the star schema.
      1. Centralized data repository stored in a fact table
         1. The fact table contains aggregated data to be used for reporting purposes
      2. Splits the fact table into a series of denormalized dimension tables.
         1. The dimension table contains descriptions of the stored data.
   3. Data is usually stored in columnar format rather than row based
5. What is a KPI?
   1. Key Performance Indicator
   2. A measurable value that demonstrates how effectively a company is achieving key business objectives. Big data is good for finding and analyzing these.
6. A Data Warehouse should follow which properties?
   1. Subject-oriented: Can be used to analyze a particular subject area
   2. Integrated: Integrates data from multiple data sources.
   3. Time-variant: Historical data is kept within. As opposed to a transactional system, where only the most recent data is kept.
   4. Non-volatile: Once the data is loaded into it, it will not change (should never be altered).
7. What is ETL?
   1. Extract, Transform, Load
   2. The process of taking data from a source, converting it into an analyzable format, and then storing it into a data warehouse (or similar).
8. What is a Staging Database?
   1. A Staging Database is an intermediate storage area used for data processing during ETL. Data inside is usually transient and wiped between ETL processes.
9. What is a Data Mart?
   1. A subset of a Data Warehouse that is specifically subject-oriented, that is, it contains data from a specific subject (maybe a certain span of years). It is a partitioned part of the whole Data Warehouse that contains only this specific section of data. They’re used to speed up analysis by storing relevant information in a more accessible format.

### Hadoop

1. What is Big Data?
   1. Big Data is a phrase used to mean a massive volume of both structured and unstructured data that is so large it is difficult to process using traditional database and software techniques
2. What kind of data do we have available?
   1. Structured
      1. Numbers, dates strings
      2. Usually stored in a database
      3. Makes up about 20% of the data out there
   2. Unstructured
      1. Photos
      2. Videos
      3. Audio
      4. Email
      5. Mobile data
   3. Semi structured
      1. Json
      2. .xml
      3. .csv
3. Mention a few Big Data problems (Common Hadoopable Problems)
   1. Cybersecurity threat detection
   2. Reduce customer churn
   3. Targeting advertising
   4. Delivering search results
4. Which are the major Big Data Unix distributions?
   1. Cloudera
   2. Hortonworks
   3. MapR
5. What is Hadoop?
   1. Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.
6. Mention a few components that you know of that exist within the Hadoop ecosystem (and their purpose)
   1. HDFS - hadoop distributed file system (Hadoop’s file system that distributes data across a cluster of servers)
   2. YARN -Yet Another Resource Negotiator - allocates resources for processing assigns tasks to nodemanagers which then run those tasks on their associated data nodes
   3. Hive - a SQL-like abstraction of MapReduce which allows you to write sql like queries to quickly do ad hoc analysis of a large data set.
7. What is HDFS?
   1. HDFS stands for Hadoop Distributed File System
   2. It is the primary data storage system for Hadoop applications
   3. It always employs a NameNode and DataNode
   4. It is highly scalable
8. What is YARN?
   1. YARN is the resource management and job scheduling technology for Hadoop
   2. It is responsible for allocating system resources to various applications running on a Hadoop cluster and scheduling tasks to be executed on different cluster nodes
   3. Stands for Yet Another Resource Negotiator
   4. YARN sits between HDFS and the processing engines being used to run applications
   5. Consists of a ResourceManager and a NodeManager
9. Which are the Hadoop 1.x Daemons? (Just mention)
   1. JobTracker
   2. TaskTracker
10. Which are the HDFS Daemons? (Explained)
    1. Name node (master)
       1. holds file system image (and updates image)
       2. Perform checkpointing process
       3. Needs to answer questions by resource manager
       4. Handles all client requests (HDFS commands)
    2. Data node (slave)
       1. Holds the actual data
       2. Performs the processing on the data, when assigned
11. Which are the Hadoop 2.x Daemons (YARN MRv2)? (Explained)
    1. Resource Manager (master)
       1. Assigns MapReduce tasks to data nodes
       2. Usually on the same server as Name Node
    2. Node manager (slave)
       1. Runs on same server as data node
       2. Keeps track of progress of tasks and assigns resources
12. How does Hadoop achieve fault tolerance?
    1. Mainly done by default data replication and also Heartbeat messages
13. How does Hadoop distribute the processing? What does this mean?
    1. Speculative execution
       1. Resource manager will check progress of tasks with node manager
       2. Resource manager may speculate that a task won't be done in time, so he will assign the task to another data node
       3. Whoever finishes first is the result that is used
14. What is the standard replication strategy used in Hadoop?
    1. The default replication count is 3, configured in hdfs-site.xml
15. What is the default block size of files in HDFS? Can you change this?
    1. Default block size is 128 MB. this can be changed in a hdfs-site.xml
16. What is the command to run a jar within a Hadoop cluster?
    1. Hadoop jar <jar-location> <main-class> <arguments>
17. Mention a few HDFS commands
    1. Hdfs dfs -copyFromLocal <local path> <hdfs path>
    2. Hdfs dfs -copyToLocal <hdfs path> <local path>
    3. Hdfs dfs -moveToLocal
    4. Hdfs dfs -moveFromLocal
18. What is Data Locality and how does Hadoop apply it?
    1. Aka rack awareness
       1. Processing strategy to improve velocity, where YARN will always try to run the processing closest to the data. This concept is applied in this fashion (in order of precedence)
          1. Data node that has the block
          2. In a data node in the same rack
          3. In a data node in a nearby rack
19. What is speculative execution?
    1. Resource manager will check progress of tasks with node manager
    2. Resource manager may speculate that a task won't be done in time, so he will assign the task to another data node
    3. Whoever finishes first is the result that is used
20. How do you set up/implement a MapReduce job?
    1. Must create a driver class, a mapper class, and a reducer class (for mapreduce jobs)
    2. Mapper holds map() method.
       1. 4 generics - <Line Number, Value in, key out, value out>
          1. Ex. <LongWritable, Text, Text, IntWritable>
    3. Reduced holds reduce() method
       1. 4 generics - <key in, value in, key out, value out>
          1. Key in and value in depend on mapper
21. What does the generic types in Mapper and Reducer classes mean?
    1. The generic types refer to the type of data that is being passed into the mapper/reducer and also the type of data that will be outputted.
22. How do you test a MapReduce job?
    1. Using MRUnit
       1. Test Mapper individually with mock inputs and outputs
       2. Test Reducer individually with mock inputs and outputs
       3. Test the combination of MapReduce
23. What is a Combiner?
    1. Aka intermediate reducer
    2. Can accept output from mapper and perform some intermediate aggregations and then the output from the combiner can be sent to a reducer to do further aggregations.
24. How does Hadoop IO work?
    1. Group of classes that you have to mandatorily uses with Map Reduce/HDFS
25. Which are some of the Hadoop IO classes?
    1. Text
    2. IntWritable
    3. DoubleWritable
    4. Bunch of writables
26. Which are the different input formats available in Hadoop?
    1. TextInputFormat
    2. FileInputFormat
    3. KeyValueInputFormat
    4. SequenceFileInputFormat
27. What's the default input format in Hadoop?
    1. TextInputFormat
28. What is a map-side join?\*\*\*\*
29. How does sorting work in Hadoop?
    1. Shuffling is the process by which it transfers mappers intermediate output to the reducer. Reducer gets 1 or more keys and associated values on the basis of reducers. The intermediate key – value generated by mapper is sorted automatically by key. InSort phase merging and sorting of map output takes place.
30. What is a Hadoop counter?
    1. Hadoop Counters : Hadoop Counters provides a way to measure the progress or the number of operations that occur within map/reduce job. Counters in Hadoop MapReduce are a useful channel for gathering statistics about the MapReduce job: for quality control or for application-level. They are also useful for problem diagnosis.
31. What is a Partitioner?
    1. The Partitioner in MapReduce controls the partitioning of the key of the intermediate mapper output. By hash function, key (or a subset of the key) is used to derive the partition. A total number of partitions depends on the number of reduce tasks.
32. What is a Combiner?
    1. Refer to question 23
33. What is MR Unit?
    1. A way to test the integrity of your MapReduce code
34. How do you test a MapReduce job with MR Unit?
    1. Refer to 22

### Hive

1. What is Hive?
   1. Sql-like abstraction of MapReduce
2. What kind of Database is Hive?
   1. Data warehouse
3. Does Hive support all CRUD operations? \*\*\*
   1. Hive doesn't support updates (or deletes), but it does support INSERT INTO, so it is possible to add new rows to an existing table. Delete has been recently added in Hive version 0.14 Deletes can only be performed on tables that support ACID Below is the link from Apache
4. What is the Hive MetaStore and where is it located?
   1. Metastore is the central repository of Apache Hive metadata. It stores metadata for Hive tables (like their schema and location) and partitions in a relational database. ... A service that provides metastore access to other Apache Hive services. Disk storage for theHive metadata which is separate from HDFS storage.
5. Where is the Hive Warehouse located?
   1. /user/hive/warehouse within hdfs
6. Which are the different query engines supported in Hive QL?
   1. Oracle
   2. mySQL
   3. derby
7. Which are some downsides of using Derby?
   1. With derby you can only have one connection per machine
8. Which are the Hive data types?
   1. Numbers
      1. Tinyint
      2. Bigint
      3. Int
      4. Double
      5. Float
      6. Decimal
   2. String
      1. Char
      2. Varchar
      3. String
   3. Dates
      1. Date
      2. TIMESTAMP
   4. Special
      1. Boolean
      2. Binary (BLOB)
   5. Complex
      1. Struct(like a class)
      2. Array
      3. map
9. What is the difference between a Managed and an External table?
   1. Managed
      1. Files are moved into data warehouse on LOAD
      2. If you drop the table, data is gone
   2. External
      1. File is moved or stays in specified location
      2. Data is not deleted on DROP
10. How do you create a table with data in Hive?
    1. CREATE [EXTERNAL] TABLE

[PARTITIONED BY column\_name]

[ROW FORMAT DELIMITED]

[FIELDS TERMINATED BY char]

[LINES TERMINATED BY char]

[STORED AS file\_format]

[LOCATION hdfs-path] (for external)

[CLUSTERED BY column] (for bucketing)

1. How do you load data into Hive?
   1. LOAD DATA [LOCAL] INPATH path [OVERWRITE] INTO TABLE tablename
2. What if I have the data locally and want to load the data into the table and copy the data to HDFS? (Copy into HDFS)
3. Which are the different INSERT and LOAD into Hive procedures?
4. What happens if the table already has data in Hive?
   1. If the table already has data, the new data will be appended unless overwrite is specified.
5. How do you specify within the create command that a table is external?
   1. CREATE EXTERNAL TABLE tablename LOCATION hdfs path
6. When loading, what happens if Hive finds a discrepancy between the data and the defined schema?
   1. Hive will insert a null if the data does not match the schema
7. How do you export a SELECT statement into a file?
   1. INSERT OVERWRITE DIRECTORY '/path/to/output/dir' SELECT \* FROM table
8. What are partitions in Hive? How do they improve queries?
   1. Partitions are used to improve query performance
   2. Splits the data by a key column
   3. Sub folders are created within the original table based off the key column, and so queries can be done to a sub table (partitioned table) rather than the entire table to improve speed.
9. How do you specify a partition in the CREATE command and then load data?
   1. In the create statement, include PARTITIONED BY column\_name
   2. INSERT INTO partitioned\_table SELECT \*
10. What is dynamic partitioning in Hive and how do we enable it?
    1. A solution for not having to do multiple insert statements for multiple partitions
    2. SET hive.exec.dynamic.partition = true
    3. SET hive.exec.dynamic.partition.mode=nonstrict;
11. What is bucketing in Hive? How do they improve queries?
    1. Improves query performance by splitting total file into more manageable pieces.
    2. Hive uses specified column to perform hashing algorithm with the specified amount of buckets to decide which bucket.
12. Can you use bucketing without partitioning?
    1. Yes you can use it with or without partitioning
13. How do you specify a bucket in the CREATE command?
    1. CLUSTERED BY <column>
    2. INTO <N> BUCKETS
14. How do you insert into a bucket or partition?
15. How do you enable Bucketing in Hive?
    1. SET hive.enforce.bucketing = true;
16. What does strict mode not allow you to do in Hive?

### Pig

(Surface Level, lowest priority)

1. What is Pig?
   1. Pig is a high level scripting language that is used with Apache Hadoop. Pig enables data workers to write complex data transformations without knowing Java. Pig’s simple SQL-like scripting language is called Pig Latin, and appeals to developers already familiar with scripting languages and SQL.
   2. Pig is complete, so you can do all required data manipulations in Apache Hadoop with Pig. Through the User Defined Functions(UDF) facility in Pig, Pig can invoke code in many languages like JRuby, Jython and Java. You can also embed Pig scripts in other languages. The result is that you can use Pig as a component to build larger and more complex applications that tackle real business problems.
   3. Pig works with data from many sources, including structured and unstructured data, and store the results into the Hadoop Data File System.
   4. Pig scripts are translated into a series of MapReduce jobs that are run on the Apache Hadoop cluster.
2. Which are the main components of Pig?
3. Is Pig case sensitive?
4. Is Pig strongly typed?
5. Which are the Pig data types?
6. What is the syntax for Pig complex data types?
7. Can you actually declare a datatype in Pig?
8. What is the usual execution of commands in Pig?
9. What is the syntax of the load command in Pig?
10. What is PigStorage?
11. Does Pig mandatorily need to run on top of HDFS/MapReduce? If no, which are the different modes?
12. How do you run Pig in a different mode?
13. Mention a few relation operators in Pig
14. How do you foreach through elements in Pig?
15. Can you define your own functions in Pig? If so, which languages are supported?
16. When loading, what happens if Pig finds a discrepancy between the data and the defined schema?
17. What does the history command do in Pig?
18. What do describe, illustrate and explain do in Pig?
19. Pig vs HiveQL vs SQL
20. Can you perform joins in Pig?
21. When performing load and relation operator commands, does data actually get loaded?

### Zookeeper

(Surface Level, lower priority - No Practicals)

1. What is Zookeeper?
   1. ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications. Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable.
   2. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them, which makes them brittle in the presence of change and difficult to manage. Even when done correctly, different implementations of these services lead to management complexity when the applications are deployed.
2. Explain the Zookeeper architecture
3. What is the Zookeeper Data Model?
4. Explain the Zookeeper workflow
5. What are some Zookeeper CLI commands?
6. How do you install Zookeeper and manage its lifecycle?
7. What is a Zookeeper Ensemble?

### Oozie

1. What is Oozie?
   1. Apache Oozie is a workflow scheduler for the Hadoop ecosystem, used to manage and schedule jobs. It supports Hadoop jobs (MapReduce, Pig, Hive, Sqoop) and system specific jobs like shell scripts and Java jobs.
2. Give me some examples of Oozie jobs you could run?
   1. MapReduce, Pig, Hive, Sqoop, Java, Shell Scripts...
3. Which are some different ways that Oozie jobs can run? (Workflow and Bundle jobs)
   1. Workflow - jobs that have a series of sequential actions to be run manually. Run as DAGs.
   2. Coordinator - jobs triggered by time or data availability.
   3. Bundles - the combination of Workflows and Coordinators; Coordinated Workflows.
4. How do you configure Oozie (the Service and a Workflow)?
   1. The Service - oozie-site.xml for customizable properties of oozie
   2. The Workflow - workflow.xml for workflow actions and control flow definitions
5. What is a Control Node in Oozie?
   1. The mechanisms that define the beginning and end of the workflow (start, end, fail). They also allow for forking the execution path, and go from action to action.
6. What is a Decision Node (Action Node) in Oozie?
   1. The mechanisms that trigger the execution of a computation/processing task. These include MapReduce, HDFS, Pig, SSH, HTTP, email, etc...
7. What is an Oozie Coordinator?
   1. Jobs triggered by time or data availability.
8. What is an Oozie properties file?
   1. Job.properties (typically) file that specifies properties for the job; such as the output directories, namenode URLs, class URLs, etc.
9. How do you run an Oozie workflow?
   1. oozie job --oozie <http://localhost:8080/oozie> -config ex/job.properties -run (returns GIVEN\_ID)
   2. oozie job -oozie <http://localhost:8080/oozie> -info GIVEN\_ID
10. Which are the different actions that can happen in an Oozie workflow?
    1. MapReduce
    2. Pig
    3. HDFS - supports move, delete, mkdir, chmod, touchz, chgrp
    4. SSH, HTTP, similar
    5. Java - executes the main method of the specified Java class

### Sqoop

1. What is Sqoop?
   1. Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.
2. How do you import data using Sqoop and why do we do it?
   1. sqoop import <TODO: add the rest>
   2. Sqoop import is used to import an individual table from an RDBMS to HDFS.
      1. Each row from a table is represented as a separate record in HDFS which can be stored as text files (one record per line) or in binary representation such as Avro or SequenceFiles.
3. How do you export data using Sqoop and why do we do it?
   1. sqoop export <TODO: add the rest>
   2. Sqoop export exports a set of files from HDFS back to an RDBMS <TODO: add the rest>
4. Can Sqoop perform reduce jobs?
   1. No, only map-only jobs
5. What is the default amount of mappers run in Sqoop and how do you change it?
   1. The default number of mappers is 4.
   2. You can change this by using the -m parameter
6. What is a Sqoop join?
   1. A Sqoop join is a free-form query. This is where sqoop imports the result set of an arbitrary SQL query
   2. The --query argument is used for free-form queries instead of using --table, --columns, or --where arguments
7. What are some advantages of using a Sqoop job instead of import/export?
   1. A sqoop import/export command can stored as a sqoop job so it can be reused as many times as possible
8. What happens if the output directory of a Sqoop import already exists?
   1. An error will occur, as with all output for MapReduce jobs
9. What is an Incremental append in Sqoop? How to use it through commands?
   1. Import all new records of a table
   2. Sqoop import command requires --incremental-append argument and --last value argument which is the index of the last row that was imported
   3. Instead, store the incremental append as a sqoop job and last value will automatically be stored.
10. What is the Sqoop Metastore? How can you specify the location in a command?
    1. Sqoop metastore is used to store Sqoop job information in a central place. This helps fuel collaboration between Sqoop users and developers. For example, a user can create a job to load some specific data. Then any other user can access from any node in the cluster the same job and just run it again.

### HBase

(Surface Level, lower priority)

1. What is HBase?
   1. A columnar database designed for fault-tolerance and storage of large quantities of sparse data.
2. What is NoSQL?
   1. NoSQL refers to a general class of storage engines that store data in a non-relational format. This is in contrast to traditional RDBMS in which data is stored in tables that have data that relate to each other. NoSQL stands for "Not Only SQL" and isn't meant as a rejection of traditional databases.
3. Which are some of the major NoSQL technologies?
   1. MongoDB
   2. Cassandra
   3. HBase
4. What is the Primary Key structure in a NoSQL database?
5. On the queries Spectrum, what are a few things that need to be taken into consideration?
6. Row vs Columnar Storage (Advantages and Disadvantages)
   1. Row stores have the ability to write data very quickly, whereas a column store is awesome at aggregating large volumes of data for a subset of columns. ... This makes columnar databases a good choice in a query-heavy environment. But you must make sure that the queries you run are really suited to a columnar database.
7. Explain the HBase architecture
   1. The simplest and foundational unit of horizontal scalability in HBase is a Region. A continuous, sorted set of rows that are stored together is referred to as a region (subset of table data). HBase architecture has a single HBase master node (HMaster) and several slaves i.e. region servers.
8. Explain the table structure in HBase
   1. A region contains all the rows between the start key and the end key assigned to that region. HBase tables can be divided into a number of regions in such a way that all the columns of a column family is stored in one region. Each region contains the rows in a sorted order.
9. Which are the different modes in which HBase can be started?
10. Is it mandatory to have Zookeeper running for HBase? Why?
    1. However, to maintain server state in the HBase Cluster,HBase uses ZooKeeper as a distributed coordination service. Basically, which servers are alive and available is maintained by Zookeeper, and also it provides server failure notification.
11. Drop vs Truncate
12. What kind of permissions can be granted or revoked in an HBase table?
13. Alter vs Delete
14. What does the Enable/Disable command do?
15. How do you interact with HBase as a client?
16. List the available DDL and DML CLI commands available in HBase.

### Spark

1. What is Spark?
   1. Data processing technology that can do both real-time and batch processing. Its batch processing runs on top of yarn/HDFS.
2. What are some advantages and disadvantages of using Spark over Hadoop?
   1. Spark improves processing speed against Hadoop by loading in memory.
   2. Transformations in Spark are lazy, they do not compute their results right away. Spark runs more efficiently by waiting for an action that requires a result.
   3. Loading in memory is much more expensive.
3. Which are the different kind of Spark Client libraries?
   1. Spark SQL
   2. Spark ML
   3. Spark Streaming
4. Which are the different ways in which you can communicate with a Spark Master?
   1. Ssh?
   2. Scp?
   3. spark-submit
5. Which are the different Cluster Managers available for use with Spark?
   1. Local[n] - runs local with n amount of threads
   2. Yarn
   3. Mesos
   4. Kubernetes
6. Hadoop vs Spark (Naming)
   1. Name Node = Master Node
   2. Data Node = Worker Node
   3. Resource Manager = Spark Master/Cluster Manager
   4. Node Manager = Executor
   5. Map, Reduce = Task
   6. Input Split = RDD
7. What does RDD stand for? What is it? What are some of its features?
   1. Resilient Distributed DataSet.
   2. They are created by the Executors then used by the Tasks.
   3. Similar to an Input Split in Hadoop.
   4. They have fault tolerance.
8. What is the default size of an RDD in Spark?
9. How does Spark make RDDs fault tolerant?
   1. Lazy execution.
   2. Immutable.
10. RDD Transformation vs Action (Mention a few)
    1. Transformations - create a new DataSet from an existing one.
       1. Filter
       2. Map
       3. Reduce by key
       4. Sort
    2. Actions - return a value to the driver program after running a computation on the DataSet.
       1. Save
       2. Reduce
       3. Collect
11. Which are the most famous programming languages used to develop Spark?
    1. Java
    2. Scala
    3. Python
12. Which are a few considerations that need to be taken before running a Spark application?
    1. Output folder must not exist or you will receive an exception.
13. Which are the different ways in which you can run a Spark application?
    1. Spark-submit --master <master> --class <main class> <jar> <args>
14. What happens if the output folder of a Spark application already exists?
    1. Throws an “output folder already exists” exception.

### EMR

1. What kinds of Services does AWS emphasize on? (IaaS, PaaS)
   1. Amazon Web Services (AWS) offers over 100 cloud computing services such as EC2, RDS, and S3. Most of these services can be used as IaaS, and most companies who use AWS will pick and choose the services they need.
2. Mention different kinds of AWS services
   1. RDS, EMR, S3, Redshift,
3. What is EC2?
   1. Elastic Cloud Compute
   2. Virtual machine in the cloud
4. Which are the different kinds of EC2?
   1. General Purpose – (T2, M4, M3)
   2. Compute Optimized – (C4, C3)
   3. Memory Optimized – (X1, R4, R3)
   4. Accelerated Computing (P2, G3, F1)
   5. Storage Optimized – (I3, D2)
5. What is Virtualization?
   1. In computing, virtualization means to create a virtual version of a device or resource, such as a server, storage device, network or even an operating system where the framework divides the resource into one or more execution environments.
6. Which are the different kinds of Virtualization?
   1. Amazon Machine Images (AWS AMI) offers two types of virtualization: Paravirtual (PV) and Hardware Virtual Machine (HVM). Each solution offers its own advantages.
      1. PV-fast, guest OS requires modification so hypervisor can can export a modified version of the underlying hardware to the VM so allow near-native performance
      2. HVM- VMs running on top of their hypervisors are not aware they are sharing processing time with other clients on the same hardware.
7. Is it possible to reassign a new Key to an EC2 after created? If no, how would you do it?
   1. You cannot reassign a new key to an ec2 after creation, however you can add or remove keys
      1. You can change the key pair that is used to access the default system account of your instance. For example, if a user in your organization requires access to the system user account using a separate key pair, you can add that key pair to your instance. Or, if someone has a copy of the .pem file and you want to prevent them from connecting to your instance (for example, if they've left your organization), you can replace the key pair with a new one.
8. Horizontal Scaling vs Vertical Scaling vs Elasticity
   1. Horizontal scaling means that you scale by adding more machines into your pool of resources
   2. Vertical scaling means that you scale by adding more power (CPU, RAM) to an existing machine.
   3. Elasticity is the ability to grow or shrink infrastructure resources dynamically as needed to adapt to workload changes in an autonomic manner, maximizing the use of resources. This can result in savings in infrastructure costs overall.
9. What is EMR?
   1. Stands for Elastic MapReduce <TODO>
   2. Preconfigured Hadoop cluster in the cloud
10. Which are the different kind of platforms in EMR?
    1. Hadoop Core - Hadoop, Ganglia (monitoring), Hive, Hue, Pig, Mahout, Tez
    2. HBase- HBase, Ganglia, Hadoop, Hive, Hue, Phoenix, Zookeeper
    3. Presto- Presto, Hadoop HDFS, Hive Metastore
    4. Spark- Spark, Hadoop, Ganglia, Zeppelin (streaming)
11. Which are the different launch modes available for EMR? Explain
    1. Cluster - cluster will continue to run until you terminate it
    2. Step execution - you are prompted to add and configure steps to execute and the EMR cluster is terminated automatically once the steps are done.
12. Which are the different kinds of EC2 types available for EMR? Any limitations?
    1. General purpose
    2. Memory optimized
    3. Compute optimized
    4. Storage optimized
    5. GPU optimized
13. What is the command syntax to connect to an EMR cluster?
    1. $ ssh -i ~/Keys/emr-demo.pem hadoop@ec2-3-91-147-76.compute-1.amazonaws.com
14. What is the command syntax to transfer a local file into an EMR cluster?
    1. scp -i ~/Keys/emr-demo.pem target/SparkWordCount.jar hadoop@ec2-3-91-147-76.compute-1.amazonaws.com:SparkWordCount.jar^C
15. Is there any use of having an EMR cluster with no key? \*\*\*\*\*\*\*\*\*
    1. I would say no, because then anyone could ssh into the cluster and get access to the data and other information that could be confidential
    2. Possibly for a step execution emr, because once the steps have finished executing the cluster should terminate automatically (maybe for a low level job).

### Servlets

1. What is the difference between ServletConfig and ServletContext?
   1. ServletContext is available to all servlets, ServletConfig is available only to one servlet.
2. How would you pre-initialize a servlet?
   1. Use the load-on-startup tag
3. What is the method signature of the doPost method?
   1. protected void doPost(HttpServletRequest req, HttpServletResponse resp )
4. What is the difference between the request.getRequestDispatcher forward and response sendRedirect?
   1. Browser is completely unaware that a forward has taken place, so its original URL remains intact
5. What is the difference between HTTP GET and POST methods?
   1. POST requests cannot be bookmarked
6. What file is used to map servlet configuration information? Write its full relative path.
   1. WEB-INF/web.xml (deployment descriptor)
7. The getRequestDispatcher() method is available in which interface?
   1. Available in the RequestDispatcher interface. Used by HttpServletRequest objects
8. True or false… The ServletContext is defined in web.xml as a init-param tag
   1. False, it is contained within its own <context-param> tag
9. Which tags are used to register a servlet in web.xml
   1. Servlet, servlet-name, servlet-class
10. Which tag in web.xml is used to add values to the ServletConfig object?
    1. Init-param
11. Which tag in web.xml is used to add values to the ServletContext object?
    1. context-param
12. True or false, GenericServlet implements Servlet interface?
    1. True
13. How many times are each lifecycle method of a servlet called?
    1. Init once, service once per request,destroy once
14. True or false,The destroy method in the lifecycle of a servlet destroys the current thread for the servlet but not the servlet object
    1. false
15. What is the service() method?
    1. The service() method passes incoming HttpRequest objects to doGet(), doPost() or another service method depending on the HTTP method used to send the request.
16. True or false, the servlet lifecycle model methods are init(), doPost() and destroy()?
    1. False, init(), service(), and destroy() are the methods
17. True or false, you can override the lifecycle methods?
    1. True
18. True or false OutWriter class is used to print on the browser using a servlet?
    1. False
19. What does a Java servlet always extend?
    1. HttpServlet
20. An application server is?
    1. Software that can serve web requests and responses, and can delegate this responsibility to Java applications
21. The method used to retrieve HTML form data from an HttpServletRequest is?
    1. .getParameter()
22. Which best describes forwarding and redirecting?
    1. Forwarding always returns a 300 status
23. What code could be used to forward a request?
    1. getServletContext().getRequestDispatcher(“/hello/”).forward(request,response);
24. What code could be used to redirect a request?
    1. response.sendRedirect("<http://www.google.com>");
25. Unique sessions are tracked by
    1. JSESSIONID cookie
26. Which of the following code is used to get session in servlet?
    1. request.getSession()
27. What is the servlet class/interface hierarchy?
    1. Servlet
    2. GenericServlet
    3. HttpServlet
    4. OurServlet

### 

### Problem Solving

1. Calculate the second maximum value of a given array of numbers.
2. Sort an array with bubble sort algorithm.
3. Return all Fibonacci numbers of a given a number.
4. Return the Factorial of a number, recursively.
5. Determine all prime factors of a given number.
6. Rotate the elements of a given array to the right, n given times.
7. Reverse a String without using StringBuilder or Buffer.
8. Check if a String is a palindrome (reads the same back and forth).
9. Obtain a substring from an array of characters without using the String class.
10. Find the first occurrence of a given word within another String.
11. Count how many times does each character in a String repeat.
12. Reverse the order of elements in a List without using a helper data structure.
13. Transform a decimal number to binary without using Java API methods.
14. Check if a number is palindrome without transforming it to a String.
15. Calculate the max amount of occurrences existent in a List with repeated values.
16. Flip the order of elements in a Stack using stack methods only (pop/push).
17. Calculate the sum of both diagonals existent in a matrix of any size.
18. Find the element in the middle of a LinkedList in a single traverse.
19. Remove duplicates of a given array of numbers without altering its order.
20. Sort a TreeSet of Integers in descending order.

For SQL, practice and remember syntax for all sub languages statements, aggregate/scalar functions, joins, set operators, and heavily review everything about the SELECT statement (GROUP BY, ORDER BY, HAVING, SUB QUERIES, ETC.) Be ready to write Java MapReduce and any particular syntax of the Hadoop ecosystem.

Don’t forget to check the past quizzes to increase the size of your pool of questions, there is a lot more! Go above and beyond, but master what I taught you first.

### To-Do on This Document

* Finish document :> :)
* Add the quiz questions from each of the sections :O
* Look for additional bolded or starred items in physical notes o7
* Add extra questions from those “top interview questions for big-data” or “...java”

Things to know:

* Know method signatures and structures of these systems, including where the NameNode is, what is does, and who it interacts with, etc...

1. All things Java
   1. OOP Pillars
   2. Servlets
   3. MapReduce
      1. Algorithm
      2. Map method & signature
      3. Reduce method & signature
      4. MapReduce method & signature
2. SQL
   1. Languages (DDL, etc)
3. Hadoop
   1. Technologies and features
      1. Blocks
      2. Replication
      3. Data Locality
      4. NameNodes (Secondary and Standby)
      5. High Availability
      6. Speculative Execution
   2. HDFS
      1. Daemons
   3. Hive
      1. Schemas
      2. Table types
      3. HiveQL
      4. Partitioning
      5. Bucketing
      6. Metastore
   4. Sqoop
      1. Drivers
      2. Codegen DAOs
      3. Export / Import command syntax
      4. Jobs
      5. Metastore
   5. Oozie
      1. Configurations
      2. Nodes
      3. Workflow.xml
      4. DAGs
      5. Properties Files
   6. Zookeeper (one liner)
   7. HBase (one liner)
   8. Pig (one liner)
4. AWS
   1. Launch Modes
   2. Instance Types
   3. SSH
   4. SCP
   5. EC2
   6. EMR
5. Spark
   1. Architecture (Clients)
   2. Architecture (Runtime)
   3. RDDs
   4. Transformations
   5. Actions
   6. Commands