What is Apache Hive?

Apache Hive is a distributed and fault-tolerant data warehouse system, which helps in analytics to occur at a massive scale. A data warehouse is a central store of information, which can easily be analyzed to make informed and data driven decisions.
Hive allows the users to read, write and manage petabytes(1000x terabytes) of data using SQL. Hive is built on top of Apache Hadoop, which is an open source framework used to store and process large datasets. Due to this, Hive is closely integrated with Hadoop, and designed to work quickly on petabytes of data.

How does Hive work?

Hive has the ability to query large datasets, utilizing Apache Tez or MapReduce, with a SQL like interface. Hive was basically created by Facebook to allow non-programmers familiar with SQL, so that they can work with petabytes of data, with an SQL like interface called HiveQL. HiveQL offers extensions not in SQL, including multi-table inserts, and creates tables *as select*. A compiler translates the HiveQL statements into a directed acyclic graph of MapReduce, Tez, or Spark jobs, which are submitted to Hadoop for execution.

An example of HiveQL word count program taken from the Wikipedia page of Apache Hive:

```
DROP TABLE IF EXISTS docs;
CREATE TABLE docs (line STRING);
LOAD DATA INPATH 'input_file' OVERWRITE INTO TABLE docs;
CREATE TABLE word_counts AS
SELECT word, count(1) AS count FROM
(SELECT explode(split(line, '\s')) AS word FROM docs) temp
GROUP BY word
ORDER BY word;
```

Hive uses batch processing so that it works quickly across the large distributed database.
Hive stores its database and table metadata in a metastore, which is a database or file backed store that enables easy data abstraction and discovery.
Metastore stores relevant metadata about the Hive table(discussed later in the document) like: Database name, table name, etc. Metastore also includes details about where the table data is located in HDFS, as it is important for the Hive queries to look in the appropriate directories in HDFS.

Components of Hive:

Hive has two components HCatalog and WebHCat.

HCatalog:

It is a table and storage management layer that reads data from the Hive metastore to facilitate integration between Hive, Apache Pig, and MapReduce. So by using the metastore, HCatalog allows Pig and MapReduce to use the same data structures as Hive, so that the metadata does not have to be redefined for each engine.
WebHCat:
Third party integrations can use WebHCat, which is a RESTful API for HCatalog, which provides a service that can be utilized by the user to run Hadoop MapReduce (or YARN), Pig, Hive tasks or function Hive metadata operations with an HTTP interface.

Types of Hive Tables:

There are different categories of Hive tables:

Transactional: Able to perform ACID/CRUD operations
Non-transactional: Not able to perform ACID/CRUD operations
Managed: Stored in Hive-created subfolders in HDFS for example, /usr/hive/warehouse/*.
This means that they are under the hive metastore, which gives you more control i.e. it includes ACID operability and the ability to delete tables.
External: Stored in the HDFS file system, but not within the Hive subfolders. As these tables are not under the Hive metastore, they do not have the ACID operations. Moreover, these tables also require additional settings to be properly deleted.

Some types of the table can be used together:

Managed and transactional tables: When you want more flexibility and control over your table.

External and non-transactional tables: When the tables are used outside of Hive. They can also offer faster table reads and writes, and also run performance tests to make sure.