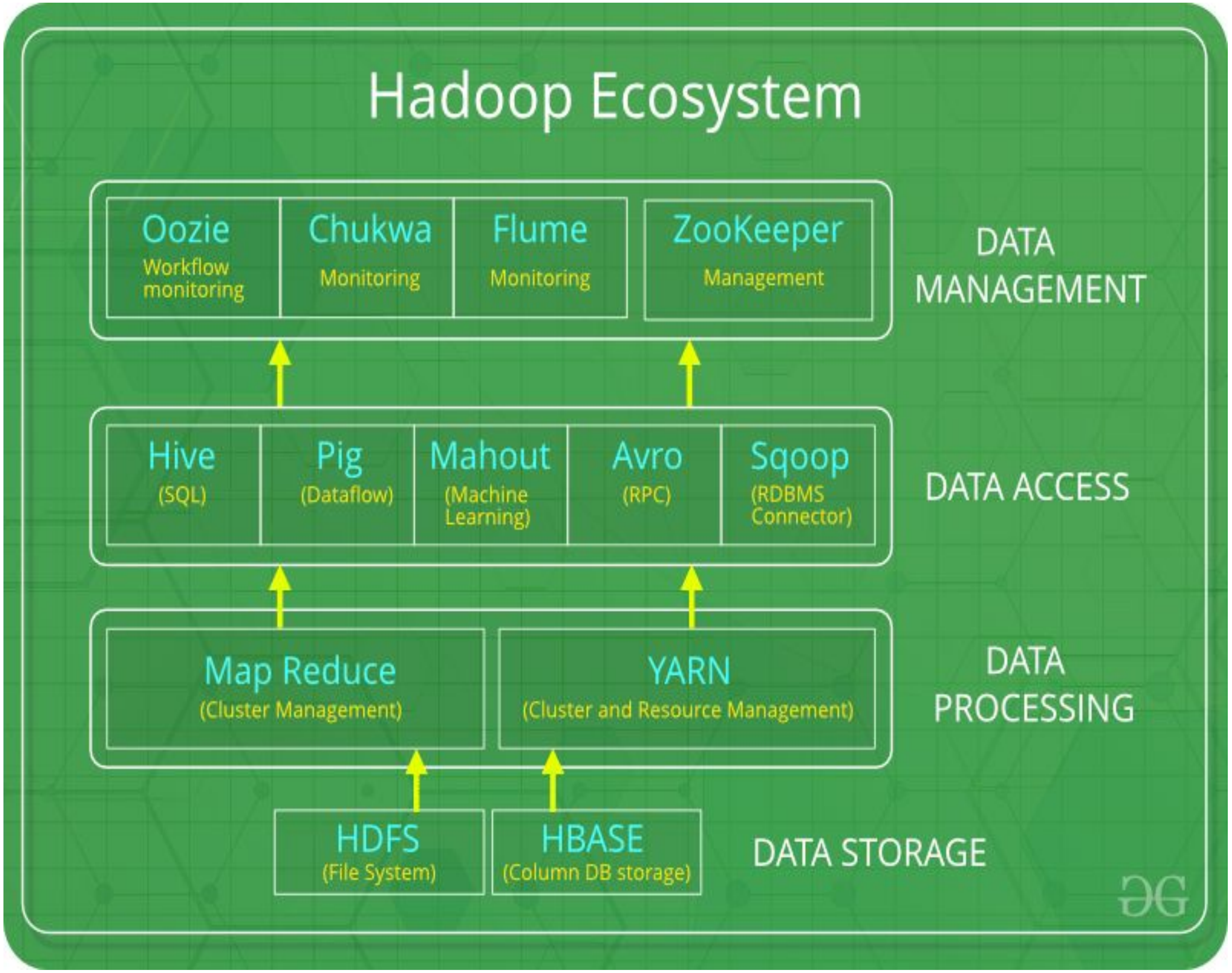
Hadoop Ecosystem

- Apache Hadoop is an open source framework intended to make interaction with **big data** easier.
- Hadoop has made its place in the industries and companies that need to work on large data sets which are sensitive and needs efficient handling.
- Hadoop is a framework that enables processing of large data sets which reside in the form of clusters. Being a framework, Hadoop is made up of several modules that are supported by a large ecosystem of technologies.

- *Hadoop Ecosystem* is a platform or a suite which provides various services to solve the big data problems.
- It includes Apache projects and various commercial tools and solutions.
- There are *four major elements of Hadoop* i.e.
- **HDFS,**
- **MapReduce,**
- **YARN,**
- **Hadoop Common**.


Hadoop Ecosystem
Oozie
Workflow monitoring
Chukwa
Monitoring
Flume
Monitoring
ZooKeeper
Management
DATA MANAGEMENT
Hive
(SQL)
Pig
(Dataflow)
Mahout
(Machine Learning)
Avro
(RPC)
Sqoop
(RDBMS Connector)
DATA ACCESS
Map Reduce
(Cluster Management)
YARN
(Cluster and Resource Management)
DATA PROCESSING
HDFS
(File System)
HBASE
(Column DB storage)
DATA STORAGE

- All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.
- Following are the components that collectively form a Hadoop ecosystem:
- **HDFS:** Hadoop Distributed File System
- **YARN:** Yet Another Resource Negotiator
- **MapReduce:** Programming based Data Processing
- **Spark:** In-Memory data processing
- **PIG, HIVE:** Query based processing of data services
- **HBase:** NoSQL Database
- **Mahout, Spark MLLib:** Machine Learning algorithm libraries
- **Solar, Lucene:** Searching and Indexing
- **Zookeeper:** Managing cluster
- **Oozie:** Job Scheduling

- **YARN:**
- Yet Another Resource Negotiator, as the name implies, YARN is the one who helps to manage the resources across the clusters.
- It performs scheduling and resource allocation for the Hadoop System.
- Consists of three major components i.e.
  - Resource Manager
  - Nodes Manager
  - Application Manager

- Resource manager has the privilege of allocating resources for the applications in a system whereas Node managers work on the allocation of resources such as CPU, memory, bandwidth per machine and later on acknowledges the resource manager.
- Application manager works as an interface between the resource manager and node manager and performs negotiations as per the requirement of the two.

- **PIG:**
- Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL.
- It is a platform for structuring the data flow, processing and analyzing huge data sets.
- Pig does the work of executing commands and in the background, all the activities of MapReduce are taken care of. After the processing, pig stores the result in HDFS.
- Pig Latin language is specially designed for this framework which runs on Pig Runtime. Just the way Java runs on the JVM.
- Pig helps to achieve ease of programming and optimization and hence is a major segment of the Hadoop Ecosystem.

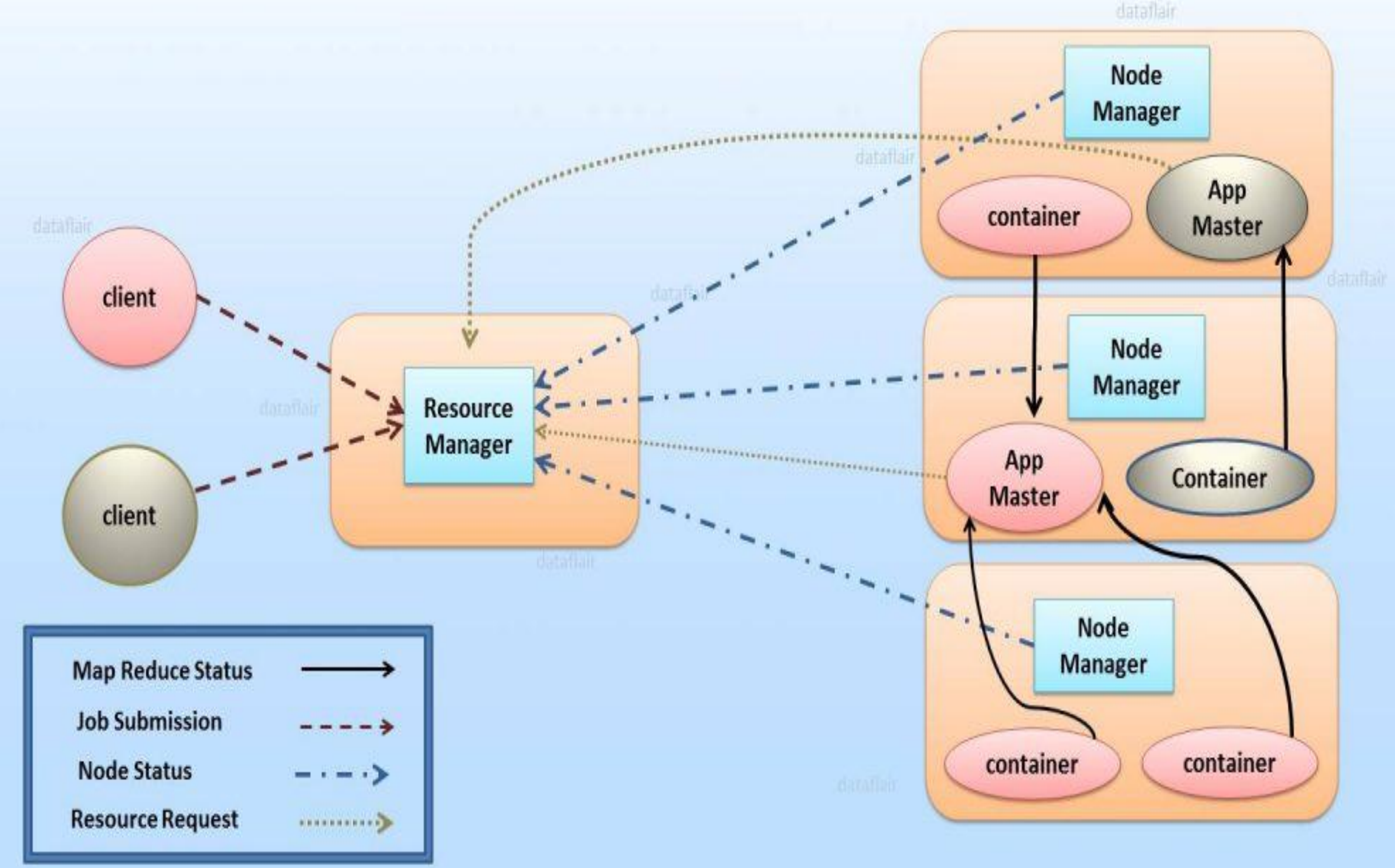
- **HIVE:**
- With the help of SQL methodology and interface, HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query Language).
- It is highly scalable as it allows real-time processing and batch processing both. Also, all the SQL datatypes are supported by Hive thus, making the query processing easier.
- Similar to the Query Processing frameworks, HIVE too comes with two components: *JDBC Drivers* and *HIVE Command Line*.
- JDBC, along with ODBC drivers work on establishing the data storage permissions and connection whereas HIVE Command line helps in the processing of queries.

- **Mahout:**
- Mahout, allows Machine Learnability to a system or application. Machine Learning, as the name suggests helps the system to develop itself based on some patterns, user/environmental interaction on the basis of algorithms.
- It provides various libraries or functionalities such as collaborative filtering, clustering, and classification which are nothing but concepts of Machine learning.
- It allows invoking algorithms as per our need with the help of its own libraries.

- **Apache HBase:**
- It’s a NoSQL database which supports all kinds of data and thus capable of handling anything of Hadoop Database. It provides capabilities of Google’s BigTable, thus able to work on Big Data sets effectively.
- At times where we need to search or retrieve the occurrences of something small in a huge database, the request must be processed within a short quick span of time. At such times, HBase comes handy as it gives us a tolerant way of storing limited data.

- **Apache Yarn** –
  "**Y**et **A**nother **R**esource **N**egotiator" is the resource management layer of **Hadoop**.
- The Yarn was introduced in Hadoop 2.x.
- Yarn allows different data processing engines like graph processing, interactive processing, stream processing as well as batch processing to run and process data store in **HDFS** (Hadoop Distributed File System).
- Apache yarn is also a data operating system for Hadoop 2.x.

Apache Hadoop YARN-Architecture



- Resource Manager (RM)
- It is the master daemon of Yarn. RM manages the global assignments of resources (CPU and memory) among all the applications.
- It arbitrates system resources between competing applications.
- Resource Manager has two Main components
- Scheduler
- Application manager

- a) Scheduler
- The scheduler is responsible for allocating the resources to the running application. The scheduler is pure scheduler it means that it performs no monitoring no tracking for the application and even doesn’t guarantees about restarting failed tasks either due to application failure or hardware failures.
- b) Application Manager
- It manages running Application Masters in the cluster, i.e., it is responsible for starting application masters and for monitoring and restarting them on different nodes in case of failures.

- Node Manager (NM)
- It is the slave daemon of Yarn. NM is responsible for containers monitoring their resource usage and reporting the same to the ResourceManager. Manage the user process on that machine.
- Yarn NodeManager also tracks the health of the node on which it is running.
- The design also allows plugging long-running auxiliary services to the NM; these are application-specific services, specified as part of the configurations and loaded by the NM during startup.
- A shuffle is a typical auxiliary service by the NMs for MapReduce applications on YARN

- Application Master (AM)
- One application master runs per application. It negotiates resources from the resource manager and works with the node manager. It Manages the application life cycle.
  The AM acquires containers from the RM’s Scheduler before contacting the corresponding NMs to start the application’s individual tasks.

- https://data-flair.training/blogs/hadoop-yarn-quiz/

Structuring a MapReduce Job in Hadoop

- A typical MapReduce program in Java consists of three classes: the driver, the mapper, and the reducer.
- The ***driver provides details such as input file locations, the provisions for adding the input*** file to the map task, the names of the mapper and reducer Java classes, and the location of the reduce task output.
- Various job configuration options can also be specified in the driver. For example, the number of reducers can be manually specified in the driver.

- The ***mapper provides the logic to be processed on each data block corresponding to the*** specified input files in the driver code.
- For example, in the word count MapReduce example provided earlier, a map task is instantiated on a worker node where a data block resides.
- Each map task processes a fragment of the text, line by line, parses a line into words, and emits <word, 1> for each word, regardless of how many times word appears in the line of text. The key/value pairs are stored temporarily in the worker node’s memory

- The key/value pairs are processed by the built-in ***shuffle and sort functionality based*** on the number of reducers to be executed.
- In this simple example, there is only one reducer. So, all the intermediate data is passed to it.
- From the various map task outputs, for each unique key, arrays (lists in Java) of the associated values in the key/value pairs are
- constructed.

- Hadoop ensures that the keys are passed to each reducer in sorted order.
- <each,(1,1)> is the first key/value pair processed, followed alphabetically by <For,(1)> and the rest of the key/value pairs until the last key/value pair is passed to the reducer. The ( ) denotes a list of values which, in this case, is just an array of ones.
- In general, each reducer processes the values for each key and emits a key/value pair as defined by the reduce logic. The output is then stored in HDFS like any other file in, say, 64 MB blocks replicated three times across the nodes.

- **Developing and Executing a Hadoop MapReduce Program**
- A common approach to develop a Hadoop MapReduce program is to write Java code using an Interactive Development Environment (IDE) tool such as Eclipse. Compared to a plaintext editor or a command-line interface (CLI), IDE tools offer a better experience to write, compile, test, and debug code.
- A typical MapReduce program consists of three Java files: one each for the driver code, map code, and reduce code. Additional, Java files can be written for the combiner or the custom partitioner, if applicable. The Java code is compiled and stored as a Java Archive (JAR) file.
- This JAR file is then executed against the specified HDFS input files.

- **Hive :**
- Similar to Pig, Apache Hive enables users to process data without explicitly writing MapReduce code.
- One key difference to Pig is that the Hive language, HiveQL (Hive Query Language), resembles Structured Query Language (SQL) rather than a scripting language.
- A Hive table structure consists of rows and columns. The rows typically correspond to some record, transaction, or particular entity (for example, customer) detail.
- The values of the corresponding columns represent the various attributes or characteristics for each row.
- Hadoop and its ecosystem are used to apply some structure to unstructured data. Therefore, if a table structure is an appropriate way to view the restructured data, Hive may be a good tool to use.

- some HiveQL basics

```
$ hive
 hive>
```

- From this environment, a user can define new tables, query them, or summarize their contents.

```
hive> create table customer (
       cust_id bigint,
       first_name string,
       last_name string,
       email_address string)
       row format delimited
       fields terminated by ‘\t’;
```

- The following HiveQL query is executed to count the number of records in the newly created table, customer
- hive> select count(*) from customer;
- When querying large tables, Hive outperforms and scales better than most conventional database queries.
- To load the customer table with the contents of HDFS file, customer.txt, it is only necessary to provide the HDFS directory path to the file.
- hive> load data inpath ‘/user/customer.txt’ into table customer;
- The following query displays three rows from the customer table.

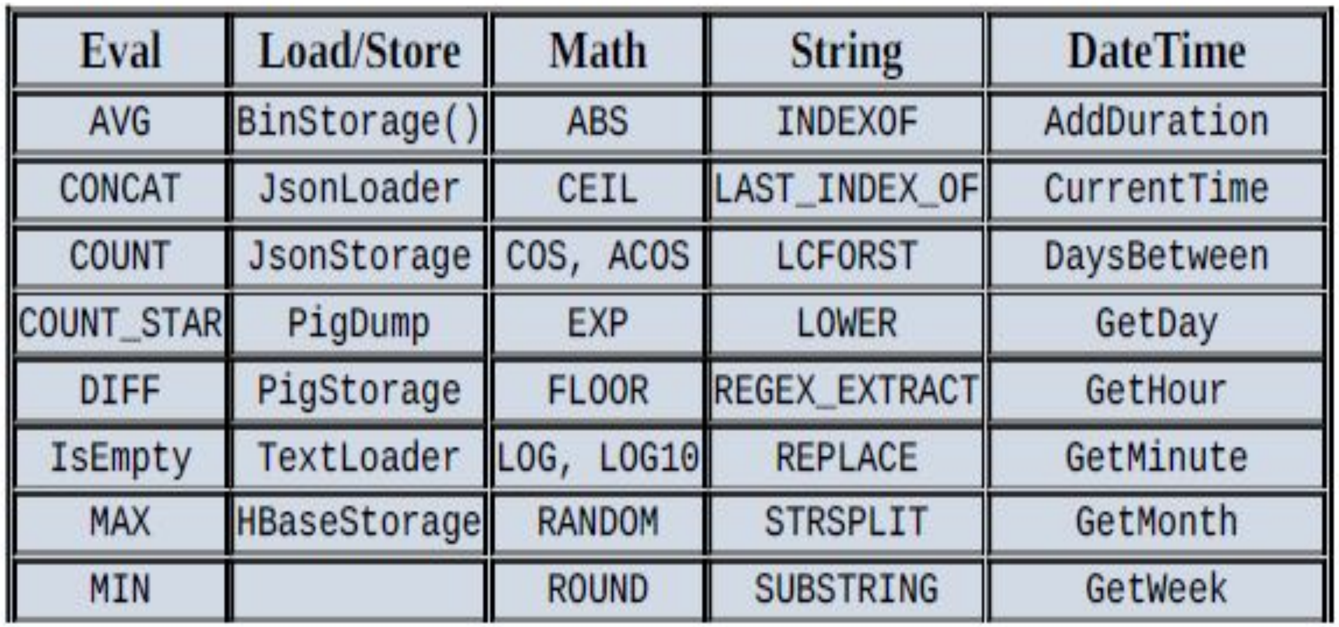
```
hive> select * from customer limit 3;
hive> quit;
```

- Like Hadoop, Pig’s origin began at Yahoo! in 2006. Pig was transferred to the Apache Software Foundation in 2007 and had its first release as an Apache Hadoop subproject in 2008.
- As Pig evolves over time, three main characteristics persist: ease of programming, behind-the-scenes code optimization, and extensibility of capabilities

```
$ pig
grunt> records = LOAD ‘/user/customer.txt’ AS
(cust_id:INT, first_name:CHARARRAY,
last_name:CHARARRAY,
email_address:CHARARRAY);
grunt> filtered_records = FILTER records
BY email_address matches ‘.*@isp.com’;
grunt> STORE filtered_records INTO
‘/user/isp_customers’;
grunt> quit
```

- At the first grunt prompt, a text file is designated by the Pig variable records with four defined fields: cust_id, first_name, last_name, and email_address.
- The variable filtered_records is assigned those records where the email_address ends with @isp.com to extract the customers whose e-mail address is from a particular Internet service provider (ISP).
- Using the STORE command, the filtered records are written to an HDFS folder, isp_customers.

- An additional feature of Pig is that it provides many built-in functions that are easily utilized in Pig code



Eval	Load/Store	Math	String	DateTime
AVG	BinStorage()	ABS	INDEXOF	AddDuration
CONCAT	JsonLoader	CEIL	LAST_INDEX_OF	CurrentTime
COUNT	JsonStorage	COS, ACOS	LCFORST	DaysBetween
COUNT_STAR	PigDump	EXP	LOWER	GetDay
DIFF	PigStorage	FLOOR	REGEX_EXTRACT	GetHour
IsEmpty	TextLoader	LOG, LOG10	REPLACE	GetMinute
MAX	HBaseStorage	RANDOM	STRSPLIT	GetMonth
MIN		ROUND	SUBSTRING	GetWeek

- **Pig :**
- Apache Pig consists of a data flow language, Pig Latin, and an environment to execute the Pig code.
- The main benefit of using Pig is to utilize the power of MapReduce in a distributed system, while simplifying the tasks of developing and executing a MapReduce job.