



HBase Introduction

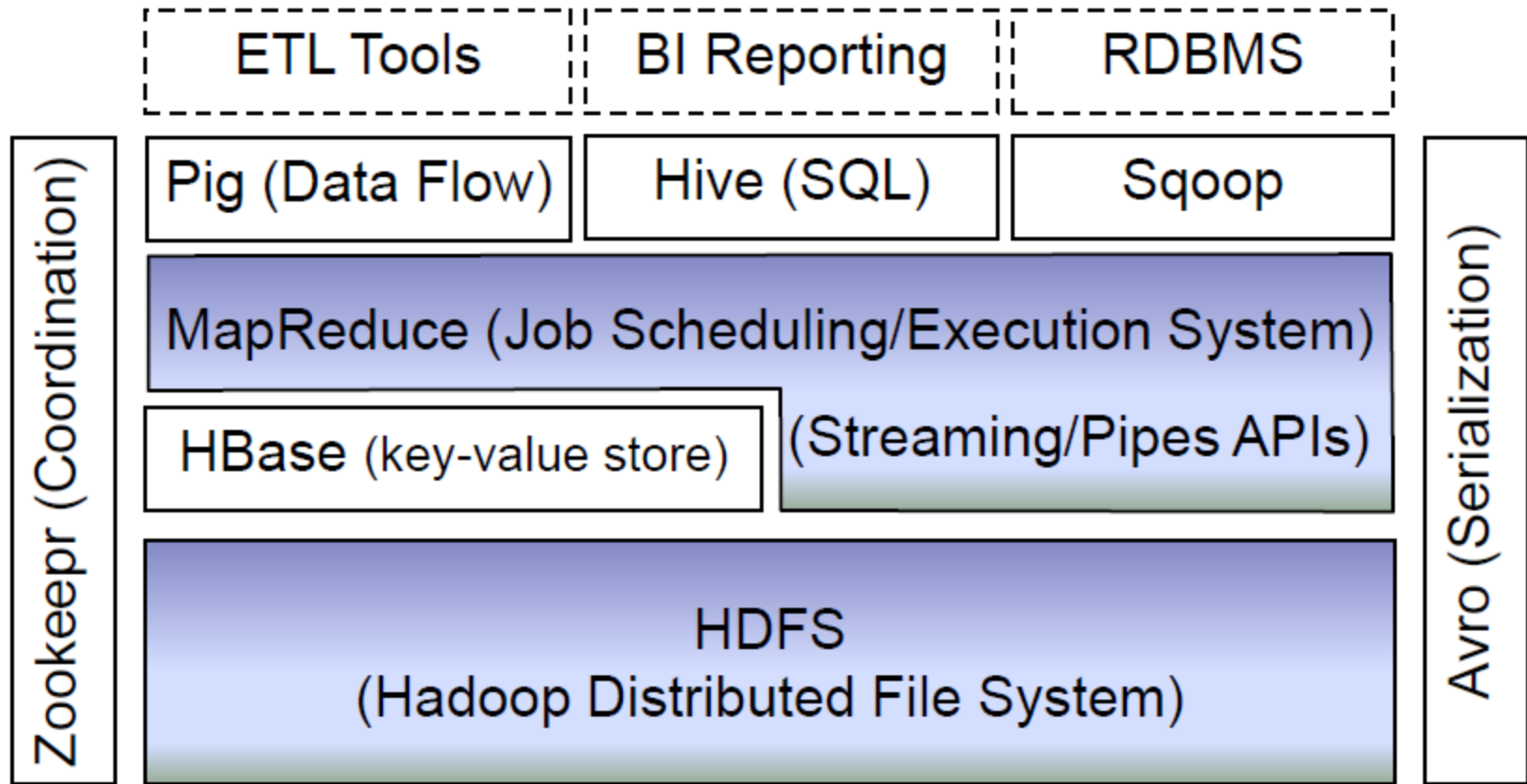
Yang Liu
2011.11.10



- ① 杨柳
 - ① Kaifeng
 - ① Aquarius
 - ① Interests: Sports Travel Games
-

- HBase is a distributed, column-oriented store, modeled after Google's BigTable.
 - HBase is built on top of Hadoop for its MapReduce and distributed file system implementation.
-

Hadoop ecosystem



- 强一致性
 - 同一行数据的读写只在同一regionserver上进行;
- 水平伸缩
 - region的自动分裂以及master的balance;
 - 只用增加datanode机器即可增加容量;
 - 只用增加regionserver机器即可增加读写吞吐量;
- 行事务
 - 同一行的列的写入是原子的;

- 支持范围查询
- 高性能随机写
- 和Hadoop无缝集成
- Hadoop分析后的结果可直接写入Hbase;
- 存放在HBase的数据可直接通过Hadoop来进行分析。

HBase VS RDBMS

HBase	RDBMS
Column oriented	Row oriented (mostly)
Flexible schema, add columns on the fly	Fixed schema
Good with sparse tables	Not optimized for sparse tables
No query language	SQL
Wide tables	Narrow tables
Joins using MR – not optimized	Optimized for joins
Tight integration with MR	Not really...

HBase VS RDBMS

HBase	RDBMS
De-normalize your data	Normalize as you can
Horizontal scalability. Just add hardware	Hard to share and scale
Consistent	Consistent
No transactions	Transactional
Good for semi structured data as well as structured data	Good for structured data

- ⊙ Storing large amounts of data
 - ⊙ Need high write throughput
 - ⊙ Need efficient random access within large data sets
 - ⊙ Need to scale gracefully with data
 - ⊙ For structured and semi-structured data
 - ⊙ Don't need full RDMS capabilities (cross row/cross table transactions, joins, etc.)
-

- ④ Adobe
 - 内部使用 (Structure data)
 - ④ Kalooga
 - 图片搜索引擎 <http://www.kalooga.com/>
 - ④ Meetup
 - 社交网站 <http://www.meetup.com/>
 - ④ Streamy
 - 成功从 MySQL 转移到Hbase <http://www.streamy.com/>
 - ④ Trend Micro
 - 云安全 <http://trendmicro.com/>
 - ④ Yahoo!
 - 储存文件 fingerprint 避免重复 <http://www.yahoo.com/>
-

Data Model

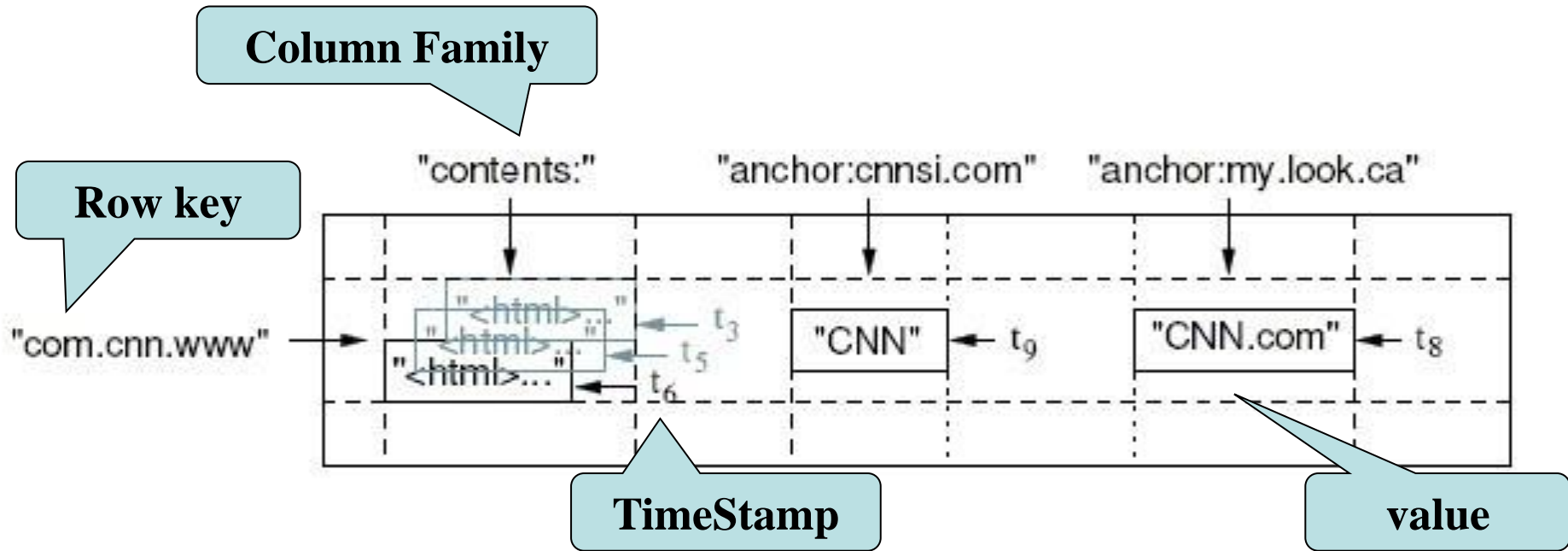




Table & Column Family

Row Key	Timestamp	Column Family	
		URI	Parser
r1	t3	url=http://www.taobao.com	title=天天特价
	t2	host=taobao.com	
	t1		
r2	t5	url=http://www.alibaba.com	content=每天...
	t4	host=alibaba.com	



-ROOT- && .META. Table

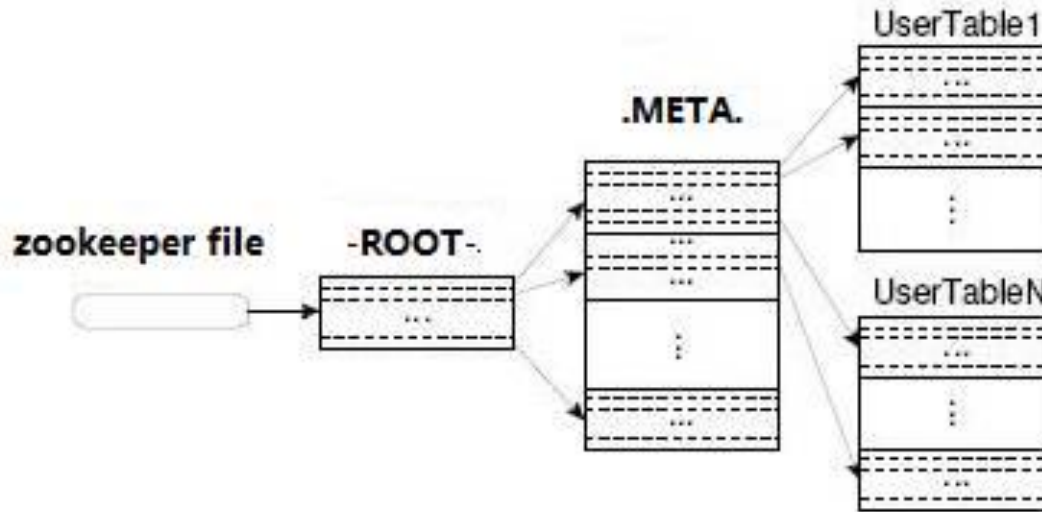


Table & Region

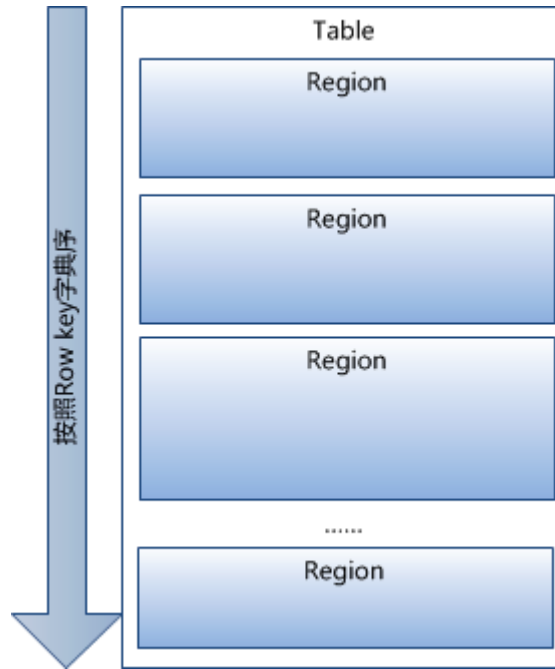


Table & Region

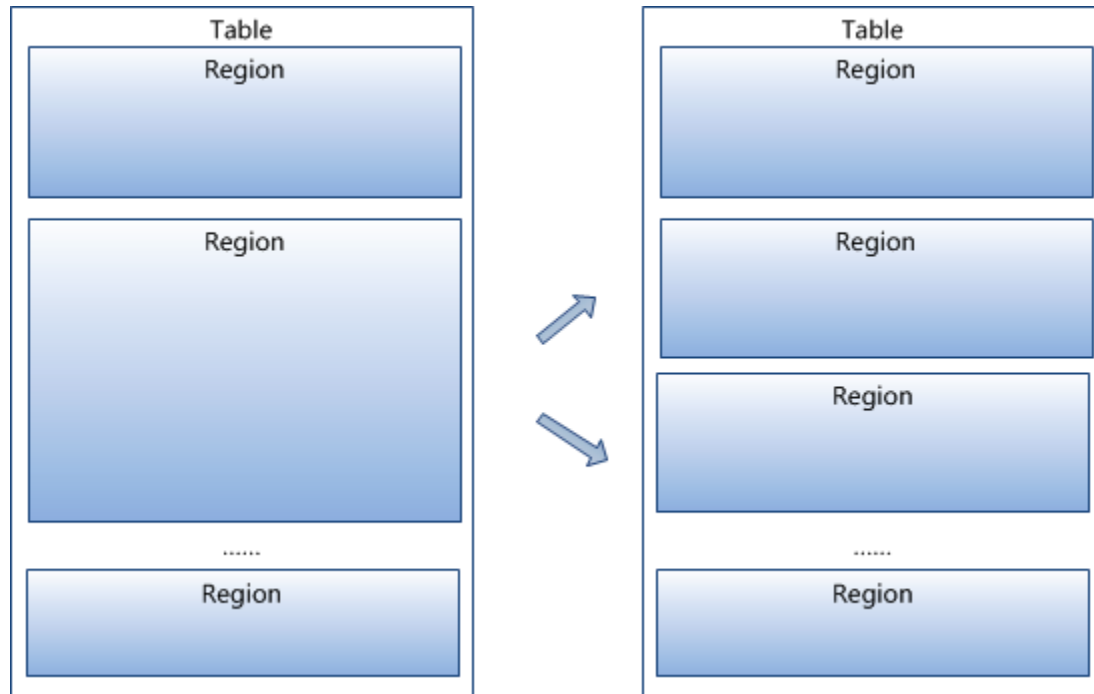
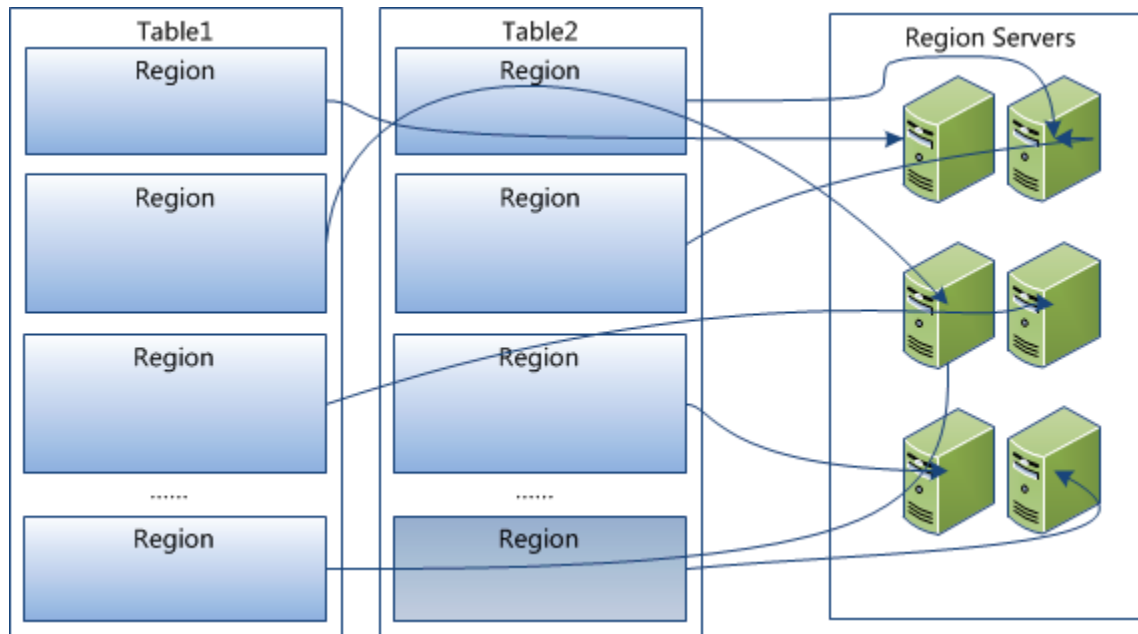
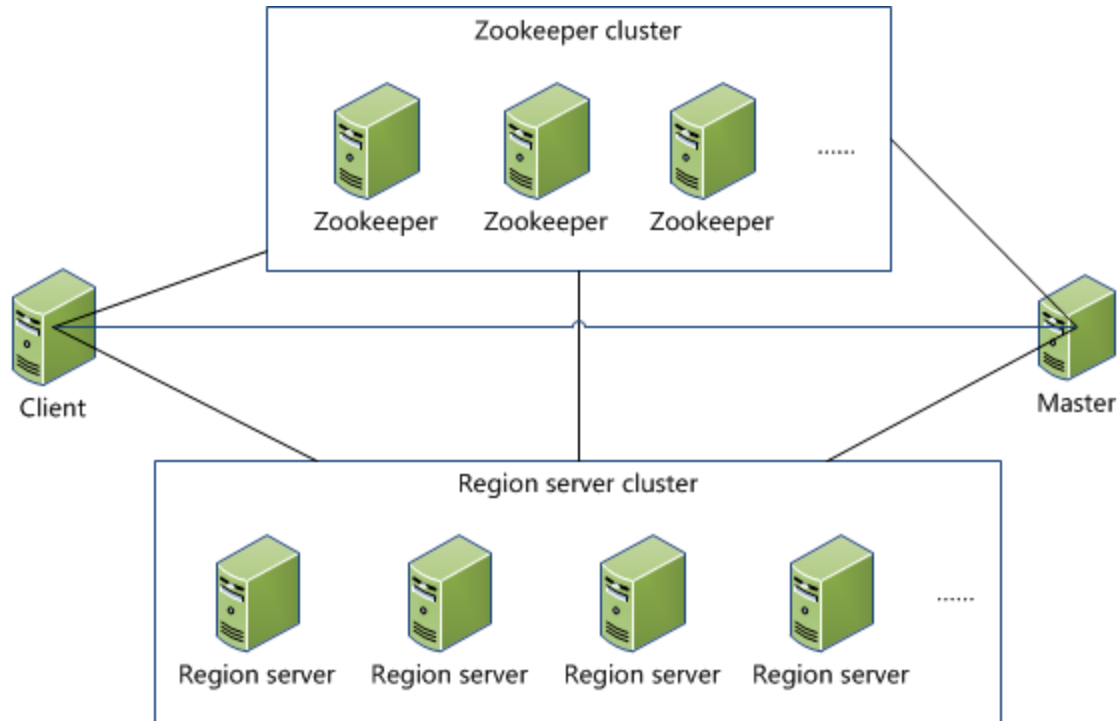


Table & Region



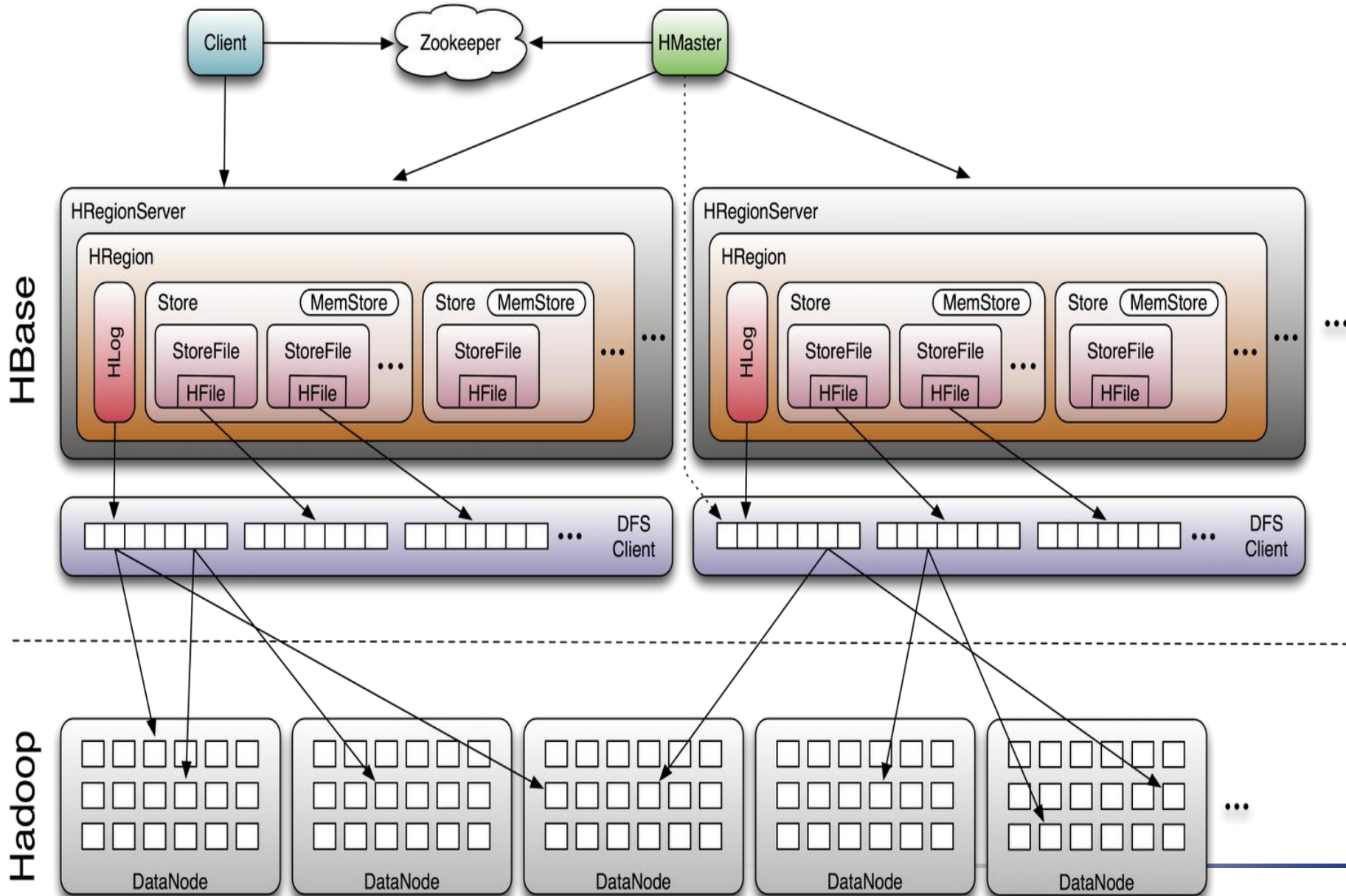


HBase System Architecture

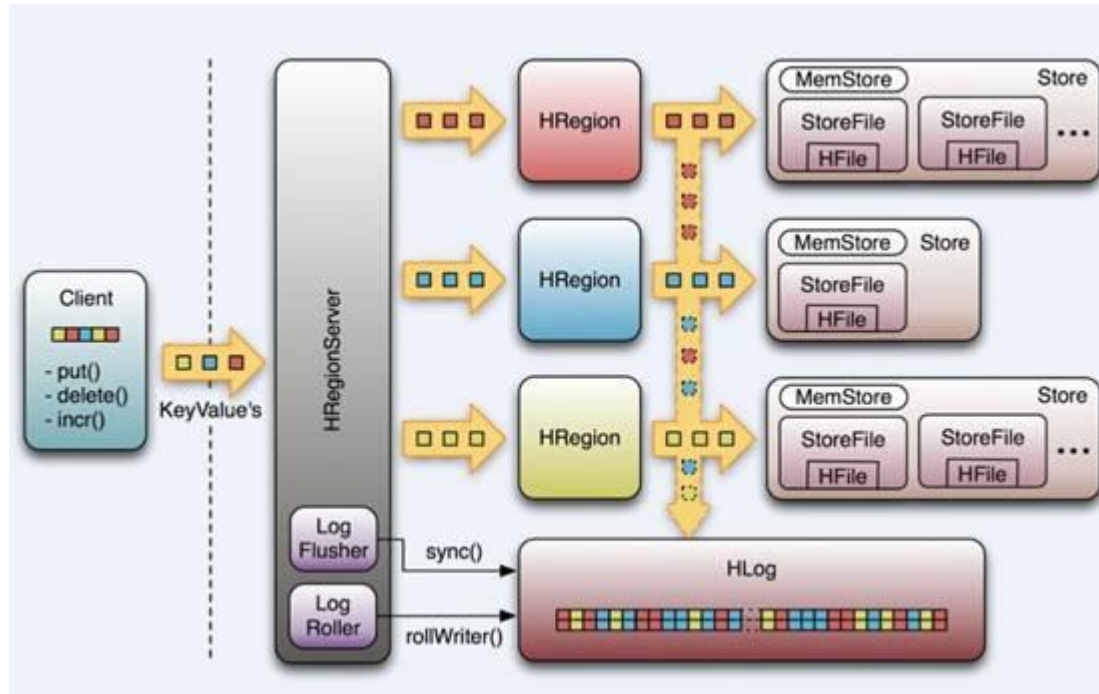




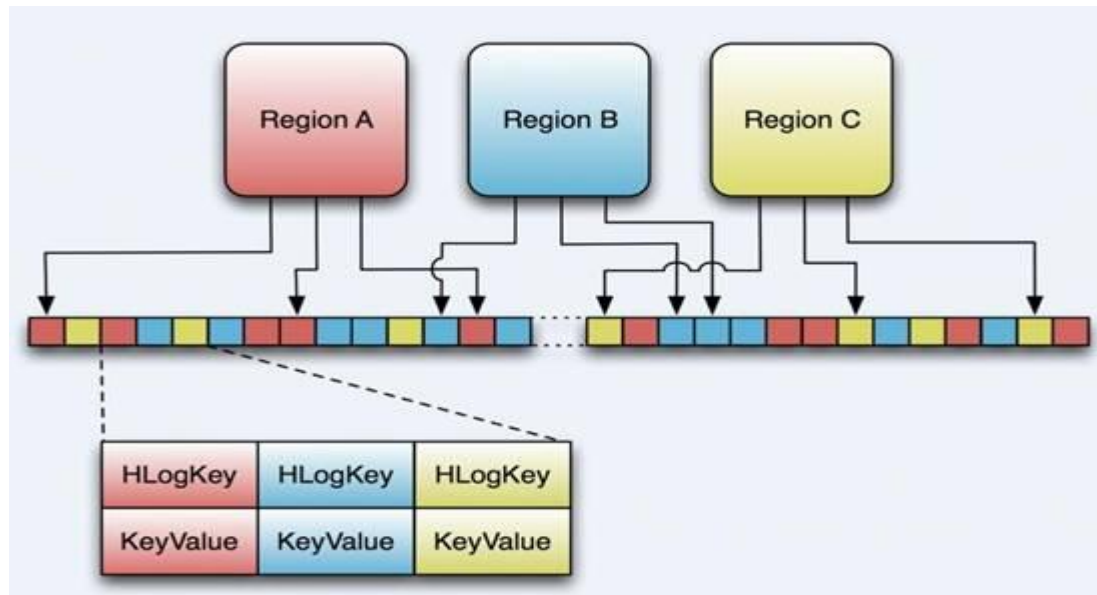
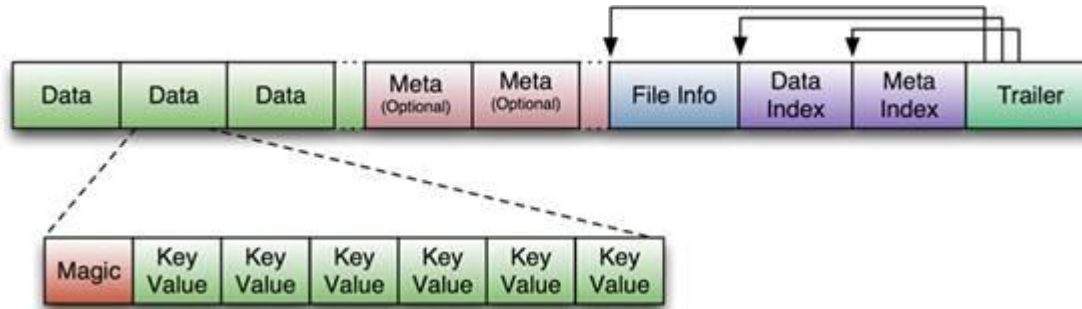
HBase System Architecture



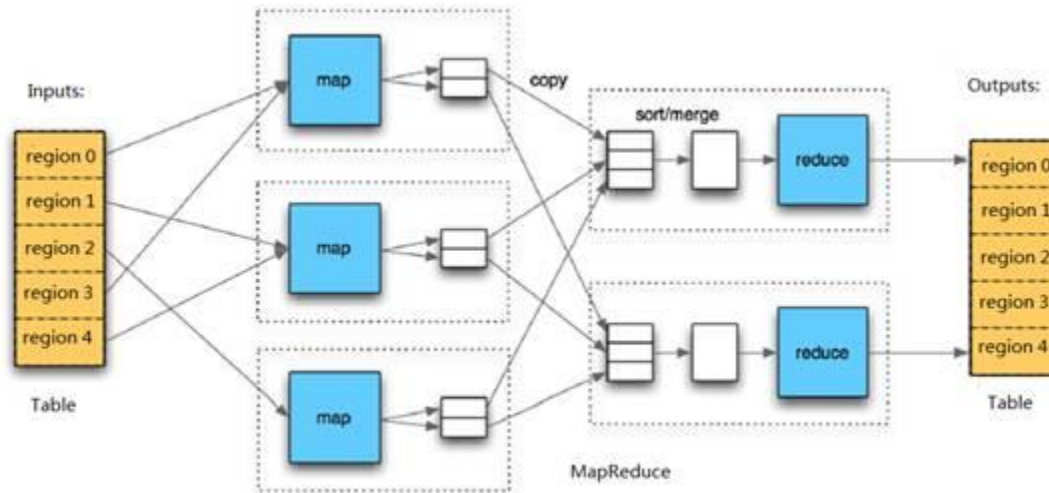
RegionServer



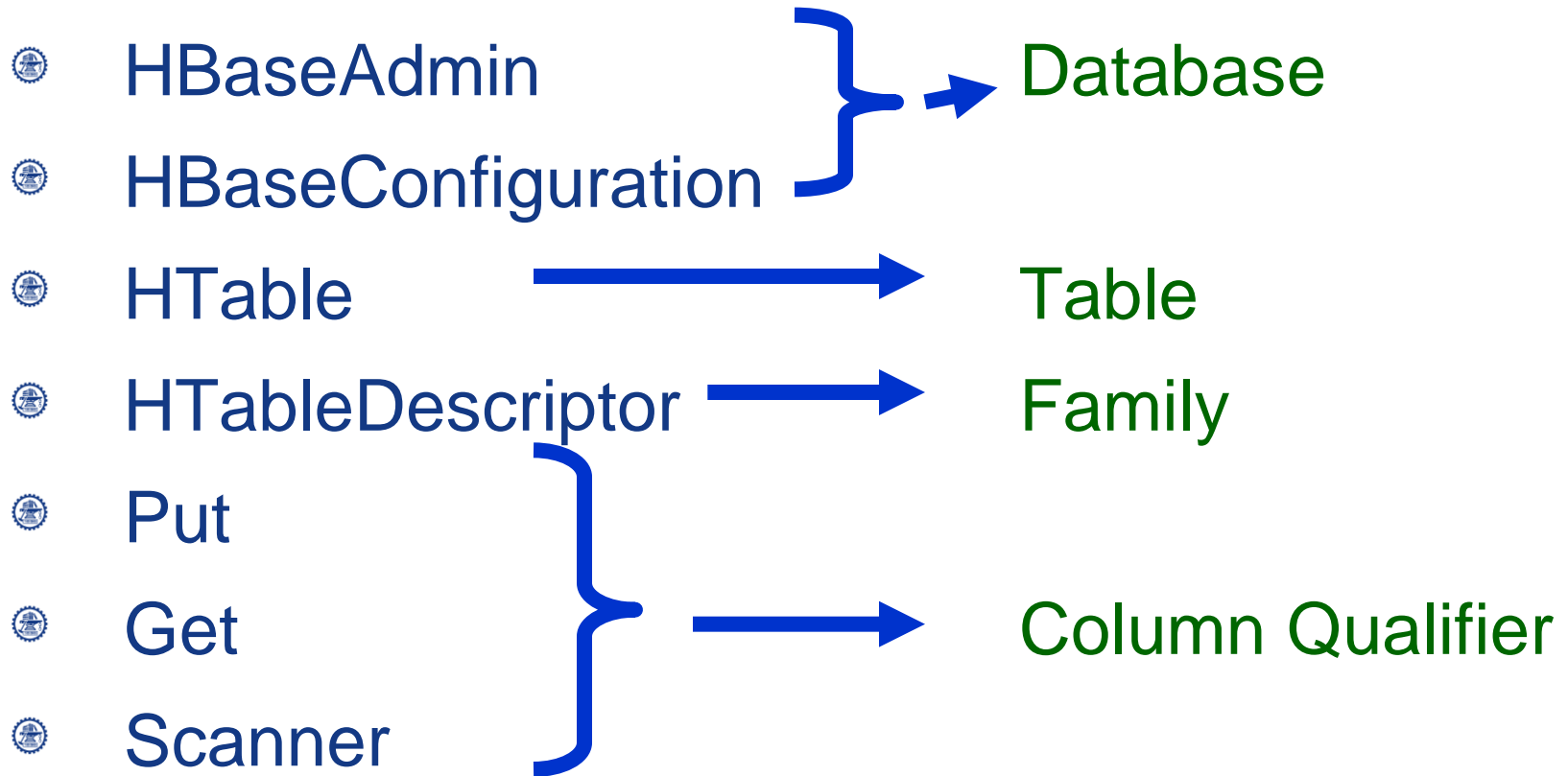
HBase Storage Format



MapReduce on HBase



HBase API



HBaseConfiguration

- ⊕ Adds HBase configuration files to a Configuration
 - = new HBaseConfiguration ()
 - = new HBaseConfiguration (Configuration c)
- ⊕ Inherit from `org.apache.hadoop.conf.Configuration`

```

<property>
  <name> name
</name>
  <value> value
</value>
</property>

```

Return value	function	args
void	addResource	(Path file)
void	clear	()
String	get	(String name)
String	getBoolean	(String name, boolean defaultValue)
void	set	(String name, String value)
void	setBoolean	(String name, boolean value)

HBaseAdmin

- new HBaseAdmin(HBaseConfiguration conf)

- Ex:

```
HBaseAdmin admin = new HBaseAdmin(config);
admin.disableTable ("tablename");
```

Return value	function	args
void	addColumn	(String tableName, HColumnDescriptor column)
	checkHBaseAvailable	(HBaseConfiguration conf)
	createTable	(HTableDescriptor desc)
	deleteTable	(byte[] tableName)
	deleteColumn	(String tableName, String columnName)
	enableTable	(byte[] tableName)
	disableTable	(String tableName)
HTableDescriptor[]	listTables	()
void	modifyTable	(byte[] tableName, HTableDescriptor htd)
boolean	tableExists	(String tableName)

HTableDescriptor

- ⊙ HTableDescriptor contains the name of an HTable, and its column families.
 - = new HTableDescriptor()
 - = new HTableDescriptor(String name)
- ⊙ **Constant-values**
 - org.apache.hadoop.hbase.HTableDescriptor.**TABLE_DESCRIPTOR_VERSION**
- ⊙ **Ex:**

```
HTableDescriptor htd = new HTableDescriptor(tablename);
htd.addFamily ( new HColumnDescriptor ("Family"));
```

Return value	function	args
void	addFamily	(HColumnDescriptor family)
HColumnDescriptor or	removeFamily	(byte[] column)
byte[]	getName	() = Table name
byte[]	getValue	(byte[] key) = value
void	setValue	(String key, String value)

HColumnDescriptor

- ⊙ An HColumnDescriptor contains information about a column family
 - = new HColumnDescriptor(String familyname)
- ⊙ **Constant-values**
 - org.apache.hadoop.hbase.HTableDescriptor.**TABLE_DESCRIPTOR_VERSION**

⊙ **Ex:**

```

HTableDescriptor htd = new HTableDescriptor(tablename);
HColumnDescriptor col = new HColumnDescriptor("content:");
htd.addFamily(col);

```

Return value	function	args
byte[]	getName	() = Family name
byte[]	getValue	(byte[] key) = value
void	setValue	(String key, String value)

HTable

- ⊙ Used to communicate with a single HBase table.
- = new HTable(HBaseConfiguration conf, String tableName)

⊙ Ex:

```
HTable table = new HTable (conf, Bytes.toBytes ( tablename ));
ResultScanner scanner = table.getScanner ( family );
```

Return value	function	args
void	checkAndPut	(byte[] row, byte[] family, byte[] qualifier, byte[] value, Put put)
void	close	()
boolean	exists	(Get get)
Result	get	(Get get)
byte[][]	getEndKeys	()
ResultScanner	getScanner	(byte[] family)
HTableDescriptor	getTableDescriptor	()
byte[]	getTableName	()
static boolean	isTableEnabled	(HBaseConfiguration conf, String tableName)
void	put	(Put put)

- ⊙ Used to perform Put operations for a single row.
 - = new Put(byte[] row)
 - = new Put(byte[] row, RowLock rowLock)
- ⊙ Ex:

```

HTable table = new HTable (conf, Bytes.toBytes ( tablename ));
Put p = new Put ( brow );
p.add (family, qualifier, value);
table.put ( p );

```

Put	add	(byte[] family, byte[] qualifier, byte[] value)
Put	add	(byte[] column, long ts, byte[] value)
byte[]	getRow	()
RowLock	getRowLock	()
long	getTimeStamp	()
boolean	isEmpty	()
Put	setTimeStamp	(long timestamp)

Used to perform Get operations on a single row.

- = new Get (byte[] row)
- = new Get (byte[] row, RowLock rowLock)

Ex:

```
HTable table = new HTable(conf, Bytes.toBytes(tablename));
Get g = new Get(Bytes.toBytes(row));
```

Get	addColumn	(byte[] column)
Get	addColumn	(byte[] family, byte[] qualifier)
Get	addColumnns	(byte[][] columns)
Get	addFamily	(byte[] family)
TimeRange	getTimeRange	()
Get	setTimeRange	(long minStamp, long maxStamp)
Get	setFilter	(Filter filter)

- Single row result of a Get or Scan query.

- = new Result()

- Ex:

```
HTable table = new HTable(conf, Bytes.toBytes(tablename));
Get g = new Get(Bytes.toBytes(row));
Result rowResult = table.get(g);
Bytes[] ret = rowResult.getValue( (family + ":" + column ) );
```

boolean	containsColumn	(byte[] family, byte[] qualifier)
NavigableMap <byte[],byte[]>	getFamilyMap	(byte[] family)
byte[]	getValue	(byte[] column)
byte[]	getValue	(byte[] family, byte[] qualifier)
int	Size	()

Scanner

- All operations are identical to **Get**
 - Rather than specifying a single row, an optional startRow and stopRow may be defined.
- If rows are not specified, the Scanner will iterate over all rows.
 - = new Scan ()
 - = new Scan (byte[] startRow, byte[] stopRow)
 - = new Scan (byte[] startRow, Filter filter)

Get	addColumn	(byte[] column)
Get	addColumn	(byte[] family, byte[] qualifier)
Get	addColumnns	(byte[][] columns)
Get	addFamily	(byte[] family)
TimeRange	getTimeRange	()
Get	setTimeRange	(long minStamp, long maxStamp)
Get	setFilter	(Filter filter)

- Interface for client-side scanning. Go to HTable to obtain instances.
- `HTable.getScanner (Bytes.toBytes(family));`

Ex:

```
ResultScanner scanner = table.getScanner (Bytes.toBytes(family));  
for (Result rowResult : scanner) {  
    Bytes[] str = rowResult.getValue ( family , column );  
}
```

void	close	()
Result	next	()

Example 1: Add Table

<Instruction>

```
create <table name>, {<family>, ....}
```

```
$ hbase shell  
> create 'tablename', 'family1', 'family2',  
        'family3 '  
0 row(s) in 4.0810 seconds  
> List  
tablename  
1 row(s) in 0.0190 seconds
```

Example 1: Add Table

<Code>

```
public static void createHBaseTable ( String tablename, String
    familyname ) throws IOException
{
    HBaseConfiguration config = new HBaseConfiguration();
    HBaseAdmin admin = new HBaseAdmin(config);
    HTableDescriptor htd = new HTableDescriptor( tablename );
    HColumnDescriptor col = new
    HColumnDescriptor( familyname );
    htd.addFamily ( col );
    if( admin.tableExists(tablename))
    {   return () }
    admin.createTable(htd);
}
```

<Instruction>

```
put 'table name' , 'row' , 'column' , 'value' ,  
[ 'time' ]
```

```
> put 'tablename','row1', 'family1:qua1', 'value'  
0 row(s) in 0.0030 seconds
```



Example 2: Put Data into Column

<Code>

```
static public void putData(String tablename, String row, String
family,
    String column, String value) throws IOException {
    HBaseConfiguration config = new HBaseConfiguration();
    HTable table = new HTable(config, tablename);
    byte[] brow = Bytes.toBytes(row);
    byte[] bfamily = Bytes.toBytes(family);
    byte[] bcolumn = Bytes.toBytes(column);
    byte[] bvalue = Bytes.toBytes(value);
    Put p = new Put(brow);
    p.add(bfamily, bcolumn, bvalue);
    table.put(p);
    table.close();
}
```



Example 3: Get Column Value

<Instruction>

get 'table name' , 'row'

```
> get 'tablename', 'row1'
```

```
COLUMN
```

```
CELL
```

```
family1:column1    timestamp=1265169495385, value=value
```

```
1 row(s) in 0.0100 seconds
```



Example 3: Get Column Value

<Code>

```
String getColumn ( String tablename, String row,  
String family, String column ) throws IOException  
{  
HBaseConfiguration conf = new  
HBaseConfiguration();  
HTable table;  
table = new HTable( conf,  
Bytes.toBytes( tablename));  
Get g = new Get(Bytes.toBytes(row));  
Result rowResult = table.get(g);  
return Bytes.toString( rowResult.getValue (   
Bytes.toBytes (family + ":" + column)));  
}
```



Example 4: Scan all Column

<Instruction>

scan 'table name'

```
> scan 'tablename'
```

```
ROW COLUMN+CELL
```

```
row1 column=family1:column1, timestamp=1265169415385, value=value1
```

```
row2 column=family1:column1, timestamp=1263534411333, value=value2
```

```
row3 column=family1:column1, timestamp=1263645465388, value=value3
```

```
row4 column=family1:column1, timestamp=1264654615301, value=value4
```

```
row5 column=family1:column1, timestamp=1265146569567, value=value5
```

```
5 row(s) in 0.0100 seconds
```



Example 4: Scan all Column

<Code>

```
static void ScanColumn(String tablename, String family, String
column) throws IOException {
    HBaseConfiguration conf = new HBaseConfiguration();
    HTable table = new HTable ( conf,
Bytes.toBytes(tablename));
    ResultScanner scanner = table.getScanner(
        Bytes.toBytes(family));

    int i = 1;
    for (Result rowResult : scanner) {
        byte[] by = rowResult.getValue(
            Bytes.toBytes(family),
Bytes.toBytes(column) );
        String str = Bytes.toString ( by );
        System.out.println("row " + i + " is \"" + str + "\"");
        i++;
    }
}
```

}}}



Example 5: Delete Table

<Instruction>

disable 'table name'
drop 'table name'

```
> disable 'tablename'  
0 row(s) in 6.0890 seconds  
> drop 'tablename'  
0 row(s) in 0.0090 seconds  
0 row(s) in 0.0090 seconds  
0 row(s) in 0.0710 seconds
```



Example 5: Delete Table

<Code>

```
static void drop ( String tablename ) throws  
    IOExceptions {  
    HBaseConfiguration conf = new  
    HBaseConfiguration();  
    HBaseAdmin admin = new HBaseAdmin (conf);  
    if (admin.tableExists(tablename))  
    {  
        admin.disableTable(tablename);  
        admin.deleteTable(tablename);  
    }else{  
        System.out.println(" [" + tablename+ "] not found!");  
    }  
}
```



Thank you!

