```
:paste
case class TempHeader (
recordId: String,
station: String,
month: String,
date: String,
hour: String,
temperature: Double
)

import spark.implicits._

val weatherDF = spark.sparkContext.textFile("1902").
map(
rec => List (
rec.substring(1,26).trim(),
rec.substring(4,10).trim(),
rec.substring(19,21).trim(),
rec.substring(21,23).trim(),
rec.substring(23,25).trim(),
rec.substring(87,92).trim()
)).
map( att => TempHeader( att(0), att(1), att(2), att(3), att(4), (att(5).trim.toDouble)/10 ))
.toDF()

weatherDF.printSchema()

// Exiting paste mode, now interpreting.
```

```
defined class TempHeader

import spark.implicits._

weatherDF: org.apache.spark.sql.DataFrame = [recordId: string, station: string ... 4 more fields]


scala> weatherDF.printSchema()

root

 |-- recordId: string (nullable = true)

 |-- station: string (nullable = true)

 |-- month: string (nullable = true)

 |-- date: string (nullable = true)

 |-- hour: string (nullable = true)

 |-- temperature: double (nullable = true)
```

```
weatherDF.createOrReplaceTempView("temperature")


val query = spark.sql("""SELECT month, max(temperature), min(temperature), avg(temperature) FROM
temperature GROUP BY month ORDER by month""".stripMargin)

query.show()


query: org.apache.spark.sql.DataFrame = [month: string, max(temperature): double ... 2 more fields]


+-----+----------------+----------------+-------------------+

|month|max(temperature)|min(temperature)|   avg(temperature)|

+-----+----------------+----------------+-------------------+

|   01|             3.3|           -31.1|-6.8862007168458765|

|   02|            11.7|           -30.0| -6.758333333333333|

|   03|             4.4|           -32.8|-5.5939068100358424|
|   04|             8.3|           -18.3| -1.482962962962962|
|   05|            21.1|            -7.8|  5.649283154121864|
|   06|            23.9|             1.7| 10.161296296296294|
|   07|            24.4|             5.6| 12.485483870967736|
|   08|            20.6|             5.0| 12.770197486535007|
|   09|            18.3|            -2.8|    8.6987037037037|
|   10|            10.6|           -13.3|  2.615949820788534|
|   11|             9.4|           -25.6|-0.9257407407407409|
|   12|             5.0|           -28.3| -5.389891696750899|
+-----+----------------+----------------+-------------------+
```

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.execution.datasources.hbase._

case class weatherHBRec(
recordId: String,
stnId: String,
OTSMonth: String,
OTSDay: String,
OTSHour: String,
temp: String)

def catalog =
s"""{
|"table":{"namespace":"default","name": "weatherHB"},
|"rowkey":"key",
|"columns":{
|"recordId":{"cf":"rowkey", "col":"key", "type":"string"},
|"stnId":{"cf":"Station", "col":"stationId", "type":"string"},
|"OTSMonth":{"cf":"TimeStamp", "col":"timestampMonth", "type":"string"},
|"OTSDay":{"cf":"TimeStamp", "col":"timestampDay", "type":"string"},
|"OTSHour":{"cf":"TimeStamp", "col":"timestampHour", "type":"string"},
|"temp":{"cf":"Temperature", "col":"temperatureC", "type":"string"}
|}
|}""".stripMargin

val spark: SparkSession =
SparkSession.builder().master("local[*]").appName("SparkByExamples.com").getOrCreate()

import spark.implicits._

val records2df = spark.sparkContext.textFile("1902").
map(
rec => List (
rec.substring(1,26).trim(),
rec.substring(4,10).trim(),
rec.substring(19,21).trim(),
rec.substring(21,23).trim(),
rec.substring(23,25).trim(),
rec.substring(87,92).trim()
)).
map( att => weatherHBRec( att(0), att(1), att(2), att(3), att(4), att(5) )).toDF().limit(10)

records2df.write.options(Map(HBaseTableCatalog.tableCatalog -> catalog, HBaseTableCatalog.newTable
-> "4")).format("org.apache.spark.sql.execution.datasources.hbase").save()
```

For HBase

```
hbase(main):010:0> list
TABLE
CF
TASK5
weatherHB
3 row(s) in 0.0100 seconds

=> ["CF", "TASK5", "weatherHB"]

hbase(main):001:0> scan "weatherHB"
ROW                COLUMN+CELL
 0290290709999190201 column=Station:stationId, timestamp=1590721936887, value=0
 02130             29070
 0290290709999190201 column=Temperature:temperatureC, timestamp=1590721936887,
 02130             value=-0172
 0290290709999190201 column=TimeStamp:timestampDay, timestamp=1590721936887, va
 02130             lue=02
 0290290709999190201 column=TimeStamp:timestampHour, timestamp=1590721936887, v
 02130             alue=13
 0290290709999919020l column=TimeStamp:timestampMonth,
timestamp=1590721936887,
 02130              value=01
 0290290709999919020l column=Station:stationId, timestamp=1590721936887,
value=0
 02200              29070
 0290290709999919020l column=Temperature:temperatureC,
timestamp=1590721936887,
 02200              value=-0178
 0290290709999919020l column=TimeStamp:timestampDay,
timestamp=1590721936887, va
 02200              lue=02
 0290290709999919020l column=TimeStamp:timestampHour,
timestamp=1590721936887, v
 02200              alue=20
 0290290709999919020l column=TimeStamp:timestampMonth,
timestamp=1590721936887,
 02200              value=01
 0290290709999919020l column=Station:stationId, timestamp=1590721936887,
value=0
 03060              29070
 0290290709999919020l column=Temperature:temperatureC,
timestamp=1590721936887,
 03060              value=-0178
 0290290709999919020l column=TimeStamp:timestampDay,
timestamp=1590721936887, va
 03060              lue=03
 0290290709999919020103060   column=TimeStamp:timestampHour, timestamp=1590721936887, v
                             alue=06
 0290290709999919020103060   column=TimeStamp:timestampMonth, timestamp=1590721936887,
                             value=01
 0290290709999919020103200   column=Station:stationId, timestamp=1590721936887, value=0
                             29070
 0290290709999919020103200   column=Temperature:temperatureC, timestamp=1590721936887,
                             value=-0150
 0290290709999919020103200   column=TimeStamp:timestampDay, timestamp=1590721936887, va
                             lue=03
 0290290709999919020103200   column=TimeStamp:timestampHour, timestamp=1590721936887, v
                             alue=20
 0290290709999919020103200   column=TimeStamp:timestampMonth, timestamp=1590721936887,
                             value=01
 0350290709999919020101060   column=Station:stationId, timestamp=1590721936887, value=0
                             29070
 0350290709999919020101060   column=Temperature:temperatureC, timestamp=1590721936887,
                             value=-0094
 0350290709999190201 column=TimeStamp:timestampDay,
timestamp=1590721936887, va

 01060              lue=01

 0350290709999190201 column=TimeStamp:timestampHour,
timestamp=1590721936887, v

 01060              alue=06

 0350290709999190201 column=TimeStamp:timestampMonth,
timestamp=1590721936887,

 01060              value=01

 0350290709999190201 column=Station:stationId, timestamp=1590721936887,
value=0

 01130              29070

 0350290709999190201 column=Temperature:temperatureC,
timestamp=1590721936887,

 01130              value=-0100

 0350290709999190201 column=TimeStamp:timestampDay,
timestamp=1590721936887, va

 01130              lue=01

 0350290709999190201 column=TimeStamp:timestampHour,
timestamp=1590721936887, v

 01130              alue=13

 0350290709999190201 column=TimeStamp:timestampMonth,
timestamp=1590721936887,

 01130              value=01

 0350290709999190201 column=Station:stationId, timestamp=1590721936887,
value=0

 01200              29070
 0350290709999190201 column=Temperature:temperatureC,
timestamp=1590721936887,
 01200              value=-0117
 0350290709999190201 column=TimeStamp:timestampDay,
timestamp=1590721936887, va
 01200              lue=01
 0350290709999190201 column=TimeStamp:timestampHour,
timestamp=1590721936887, v
 01200              alue=20
 0350290709999190201 column=TimeStamp:timestampMonth,
timestamp=1590721936887,
 01200              value=01
 0350290709999190201 column=Station:stationId, timestamp=1590721936887,
value=0
 02060              29070
 0350290709999190201 column=Temperature:temperatureC,
timestamp=1590721936887,
 02060              value=-0161
 0350290709999190201 column=TimeStamp:timestampDay,
timestamp=1590721936887, va
 02060              lue=02
 0350290709999190201 column=TimeStamp:timestampHour,
timestamp=1590721936887, v
 02060              alue=06
 0350290709999190201 column=TimeStamp:timestampMonth,
timestamp=1590721936887,
 02060              value=01
 03502907099999190201 column=Station:stationId, timestamp=1590721936887,
value=0

 03130                29070

 03502907099999190201 column=Temperature:temperatureC,
timestamp=1590721936887,

 03130                value=-0172

 03502907099999190201 column=TimeStamp:timestampDay,
timestamp=1590721936887, va

 03130                lue=03

 03502907099999190201 column=TimeStamp:timestampHour,
timestamp=1590721936887, v

 03130                alue=13

 03502907099999190201 column=TimeStamp:timestampMonth,
timestamp=1590721936887,

 03130                value=01

 03502907099999190201 column=Station:stationId, timestamp=1590721936887,
value=0

 04060                29070

 03502907099999190201 column=Temperature:temperatureC,
timestamp=1590721936887,

 04060                value=-0106

 03502907099999190201 column=TimeStamp:timestampDay,
timestamp=1590721936887, va

 04060                lue=04

 03502907099999190201 column=TimeStamp:timestampHour,
timestamp=1590721936887, v

 04060                alue=06
 03502907099999190201 column=TimeStamp:timestampMonth,
timestamp=1590721936887,

 04060                value=01

10 row(s) in 0.4800 seconds
```