

## 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

### 스마트카 상태 정보 데이터 생성 (1/2)

로그 시뮬레이터를 이용해 오늘 날짜의 “스마트카 상태 정보 데이터”를 생성한다. 필자의 경우 로그 시뮬레이터를 작동시킨 날짜가 2020년 3월 22일이었고 100대의 스마트카를 지정했다. 독자들도 파일럿 프로젝트를 수행하는 시점의 날짜를 반드시 입력하고 사용하기 바란다. 우선 로그 시뮬레이터가 설치돼 있는 Server02에 SSH를 통해 접속하고 다음 명령어들을 실행해 본다.

저사양 파일럿 환경: 플럼 서비스를 시작한다.

- 플럼 서비스: CM 홈 → [Flume] → [시작]

## 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

### 스마트카 상태 정보 데이터 생성 (2/2)

```
$ cd /home/pilot-pjt/working
```

```
$ java -cp bigdata.smartcar.loggen-1.0.jar com.wikibook.bigdata.smartcar.loggen.CarLogMain  
20200322 100 &
```

/home/pilot-pjt/working/SmartCar/의 위치에 SmartCarStatusInfo\_20200322(실행일).txt  
파일이 100MB 파일 크기로 생성된 것을 확인한 후, 로그 시뮬레이터를 종료한다.

```
$ cd /home/pilot-pjt/working/SmartCar/
```

```
$ ls -ltrh SmartCarStatusInfo_20200322.txt
```

```
$ ps -ef | grep smartcar.log
```

```
$ kill -9 [pid]
```

## 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

### 스마트카 상태 정보 데이터 적재 및 확인 (1/2)

스마트카 상태 정보 데이터를 플럼의 수집 디렉터리로 옮긴다. 플럼이 수집 작업을 시작한다.

```
$ mv /home/pilot-pjt/working/SmartCar/SmartCarStatusInfo_20200322.txt /home/pilot-pjt/working/  
car-batch-log/
```

플럼이 수집해서 하둡에 적재 완료되기까지 약 5~10여 분의 시간이 걸린다.

## 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

### 스마트카 상태 정보 데이터 적재 및 확인 (2/2)

오늘 날짜의 스마트카 상태 정보가 HDFS에 정상적으로 적재됐는지 확인한다.

```
$ hdfs dfs -ls -R /pilot-pjt/collect/car-batch-log/
```

크기가 각각 65MB, 52MB인 두 개의 파일이 HDFS에 만들어져 있어야 한다. wrk\_date=20200322 디렉터리 아래에 .tmp인 파일이 있으면 아직 플럼에서 적재 중이니 완료될 때까지 기다린다.

## 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

### 스마트카 운전자 운행 로그 생성 및 확인 (1/6)

저사양 파일럿 환경: 아래 서비스들이 정지돼 있다면 시작시킨다.

- 플럼 서비스: CM 홈 → [Flume] → [시작]
- 카프카 서비스: CM 홈 → [Kafka] → [시작]
- 스톰 서비스: Server02에 SSH로 접속해 다음 명령을 실행
  - \$ service storm-nimbus start
  - \$ service storm-supervisor start
  - \$ service storm-ui start
- 레디스 서비스: Server02에 SSH로 접속해 다음 명령을 실행
  - \$ service redis\_6379 start
- HBase 서비스: CM 홈 → [HBase] → [시작]

## 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

### 스마트카 운전자 운행 로그 생성 및 확인 (2/6)

수집/적재 기능이 모두 정상이면 스마트카의 실시간 운행 로그 데이터를 다음 명령으로 생성한다. 아래 명령의 “20200322”는 필자가 실행시킨 시점의 날짜다. 각자의 실행 날짜로 바꿔 설정한다.

저사양 파일럿 환경: 스마트카 대수를 10 이하로 조정한다.

저사양 환경에서 스마트카 100대로 시뮬레이션 하면, 서버에서 병목 현상과 타임아웃 등의 문제가 발생할 수 있다. 이를위해 스마트카 대수 옵션을 10 이하로 설정한다.

```
$ cd /home/pilot-pjt/working
```

```
$ java -cp bigdata.smartcar.loggen-1.0.jar com.wikibook.bigdata.smartcar.loggen.DriverLogMain 20200322 100 &
```

## 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

### 스마트카 운전자 운행 로그 생성 및 확인 (3/6)

실시간 운행 로그 데이터가 정상적으로 생성됐는지 확인한다. /home/pilot-pjt/working/driver-realtime-log/에 SmartCarDriverInfo.log 파일이 생성됐을 것이다. 로그는 24시간을 기준으로 지속적으로 생성된다. tail 명령으로 실시간 로그가 계속 기록되는지 확인해 보자.

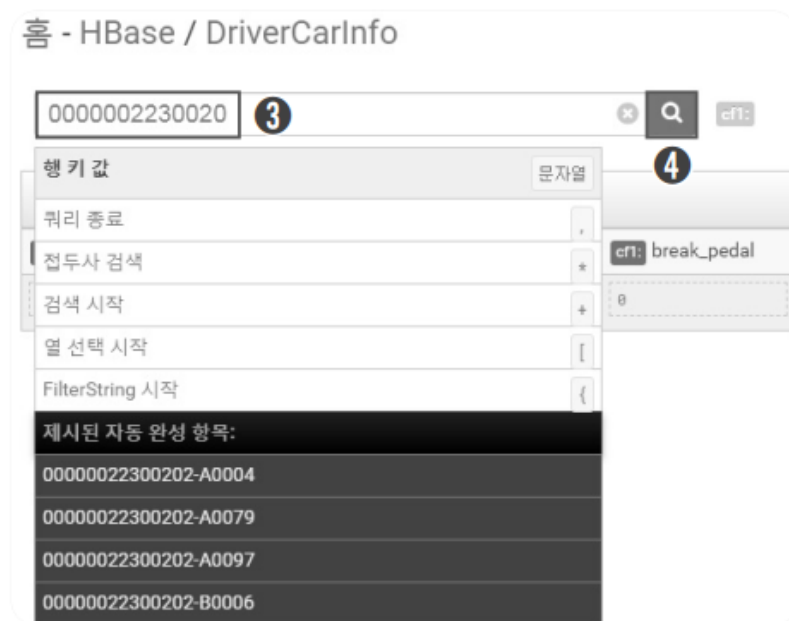
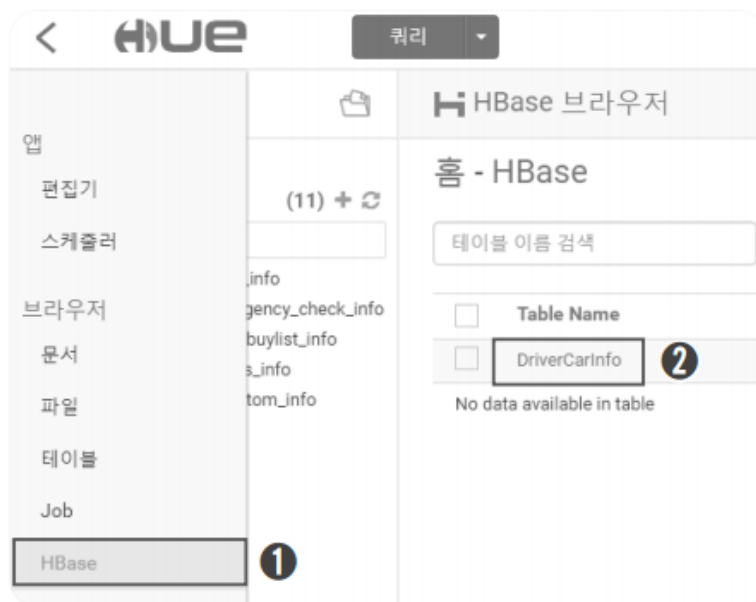
```
$ cd /home/pilot-pjt/working/driver-realtime-log
```

```
$ tail -f SmartCarDriverInfo.log
```

# 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

## 스마트카 운전자 운행 로그 생성 및 확인 (4/6)

휴에 접속해 [좌측 메뉴 펼치기] → [HBase] → [DriverCarInfo] 테이블을 선택해 실행일자(“2020 0322”)의 운행 데이터가 생성됐는지 확인한다. HBase 브라우저 검색에 실행 일자를 역변환한 로우키의 prefix “00000022300202”를 입력하면 등록된 로우키 목록이 자동 완성되어 나타난다. 이 가운데 아무 로우키나 선택하고 뒤에 콤마(“,”)를 입력하고 [검색] 버튼을 클릭하면 해당 로우키 스마트카 운전자의 실시간 운행 정보가 HBase에서 조회된다.





## 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

### 스마트카 운전자 운행 로그 생성 및 확인 (5/6)

레디스 CLI를 실행해 오늘 날짜로 과속한 스마트카 차량 정보를 확인한다. 로그 시뮬레이터 상황에 따라 아직 과속 차량이 발생하지 않았을 수도 있다.

```
$ redis-cli
```

```
$ 127.0.0.1:6379> smembers 20200322
```

과속 차량이 3대 이상 발견되면 스마트카 운전자에 대한 운행 로그 시뮬레이터도 종료시킨다. 앞서 실행했던 스마트카 상태 정보 로그 시뮬레이터도 강제로 종료한다.

```
$ ps -ef | grep smartcar.log
```

```
$ kill -9 [pid] [pid]
```

## 6.6 탐색 파일럿 실행 4단계 - 데이터 준비

### 스마트카 운전자 운행 로그 생성 및 확인 (6/6)

저사양 파일럿 환경: 수집/적재 기능을 정지시킨다.

- 플럼 서비스: CM 홈 → [Flume] → [정지]
- 카프카 서비스: CM 홈 → [Kafka] → [정지]
- 스톰 서비스: Server02에 SSH로 접속한 후 다음 명령을 실행
  - \$ service storm-ui stop
  - \$ service storm-supervisor stop
  - \$ service storm-nimbus stop
- 레디스 서비스: Server02에 SSH로 접속한 후 다음 명령을 실행
  - \$ service redis\_6379 stop
- HBase 서비스: CM 홈 → [HBase] → [정지]

## 6.6 탐색 파일럿 실행 4단계

### 데이터 준비

# 실습