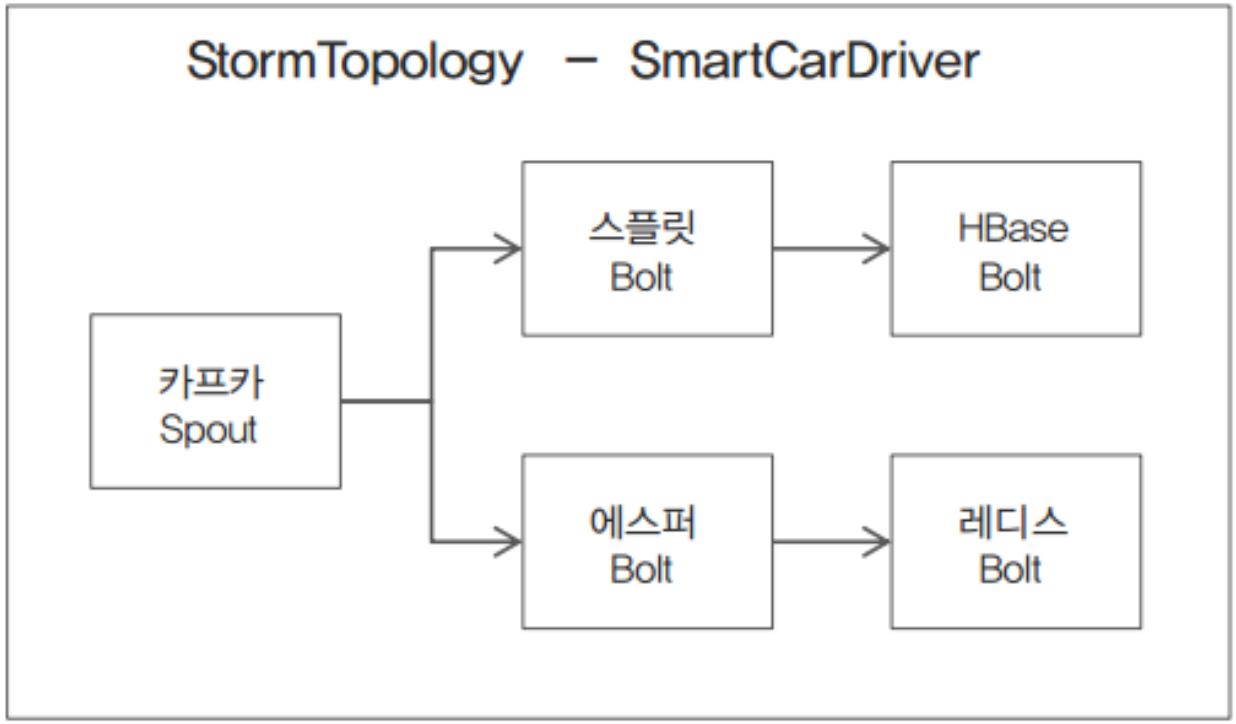
5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

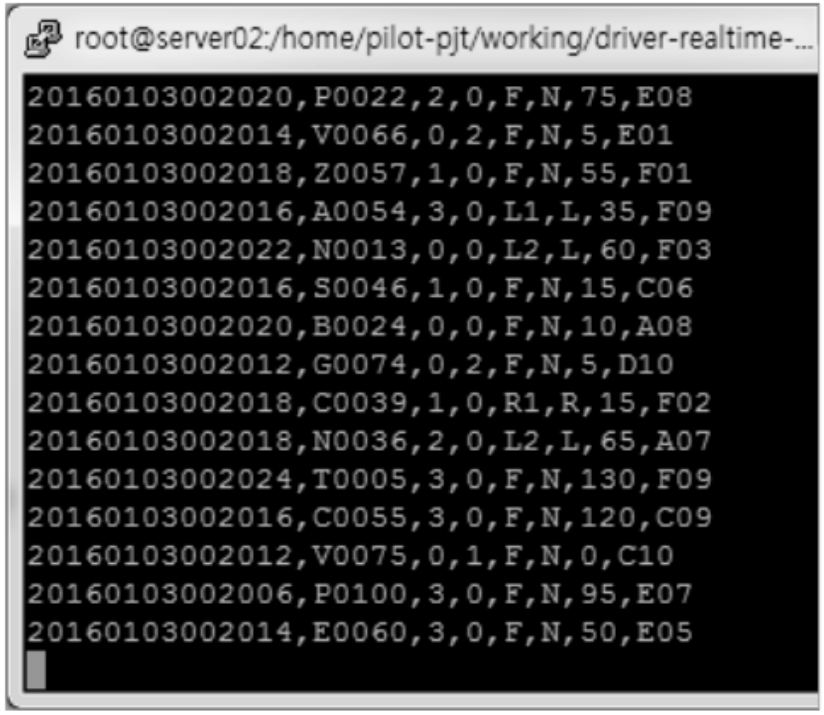


5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

로그 시뮬레이터 작동

```
$ cd /home/pilot-pjt/working
$ java -cp bigdata.smartcar.loggen-1.0.jar com.wikibook.bigdata.smartcar.loggen.DriverLogMain
20160103 10 &
```

```
$ cd /home/pilot-pjt/working/driver-realtime-log
$ tail -f SmartCarDriverInfo.log
```



5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

HBase 적재 데이터 확인 (1/6)

01. HBase 셸의 count 명령으로 실시간으로 적재되고 있는 운전자 정보를 확인해 본다.

```
$ hbase shell
$ hbase(main):001:0> count 'DriverCarInfo'
```

DriverCarInfo 테이블에 적재된 데이터의 로우 수를 1000 단위로 출력한다.

```
hbase(main):006:0> count 'DriverCarInfo'
Current count: 1000, row: 00530020106102-B0002
Current count: 2000, row: 01140030106102-A0001
Current count: 3000, row: 01740030106102-A0001
Current count: 4000, row: 02330030106102-A0001
Current count: 5000, row: 03000030106102-A0001
Current count: 6000, row: 03540030106102-A0001
Current count: 7000, row: 04200020106102-B0002
Current count: 8000, row: 04801020106102-B0002
Current count: 9000, row: 05400030106102-A0001
Current count: 10000, row: 20011030106102-A0001
Current count: 11000, row: 20160103000246Y0007
Current count: 12000, row: 20411020106102-U0008
Current count: 13000, row: 21030020106102-U0008
Current count: 14000, row: 21610030106102-T0007
Current count: 15000, row: 22220020106102-U0008
```

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

HBase 적재 데이터 확인 (2/6)

02. 다음의 Scan 명령으로 DriverCarInfo 테이블에 적재된 칼럼 기반 구조의 데이터를 살펴보자. 그냥 scan 명령을 내리면 모든 데이터가 조회되므로 LIMIT 옵션으로 20개 데이터만 조회하자.

```
$ hbase(main):001:0> scan 'DriverCarInfo', {LIMIT=>20}
```

```
00001030106102-Y0012        column=cf1:area_number, timestamp=1461423214693, value=E06
00001030106102-Y0012        column=cf1:break_pedal, timestamp=1461423214693, value=0
00001030106102-Y0012        column=cf1:car_number, timestamp=1461423214693, value=Y0012
00001030106102-Y0012        column=cf1:date, timestamp=1461423214693, value=20160103010000
00001030106102-Y0012        column=cf1:direct_light, timestamp=1461423214693, value=N
00001030106102-Y0012        column=cf1:speed, timestamp=1461423214693, value=23
00001030106102-Y0012        column=cf1:speed_pedal, timestamp=1461423214693, value=3
00001030106102-Y0012        column=cf1:steer_angle, timestamp=1461423214693, value=F
00001030106102-Z0019        column=cf1:area_number, timestamp=1461423214795, value=D03
00001030106102-Z0019        column=cf1:break_pedal, timestamp=1461423214795, value=0
00001030106102-Z0019        column=cf1:car_number, timestamp=1461423214795, value=Z0019
00001030106102-Z0019        column=cf1:date, timestamp=1461423214795, value=20160103010000
00001030106102-Z0019        column=cf1:direct_light, timestamp=1461423214795, value=N
00001030106102-Z0019        column=cf1:speed, timestamp=1461423214795, value=60
00001030106102-Z0019        column=cf1:speed_pedal, timestamp=1461423214795, value=3
00001030106102-Z0019        column=cf1:steer_angle, timestamp=1461423214795, value=F
00001030106102-Z0020        column=cf1:area_number, timestamp=1461423214785, value=D04
00001030106102-Z0020        column=cf1:break_pedal, timestamp=1461423214785, value=0
00001030106102-Z0020        column=cf1:car_number, timestamp=1461423214785, value=Z0020
00001030106102-Z0020        column=cf1:date, timestamp=1461423214785, value=20160103010000
00001030106102-Z0020        column=cf1:direct_light, timestamp=1461423214785, value=N
00001030106102-Z0020        column=cf1:speed, timestamp=1461423214785, value=25
00001030106102-Z0020        column=cf1:speed_pedal, timestamp=1461423214785, value=1
00001030106102-Z0020        column=cf1:steer_angle, timestamp=1461423214785, value=F
20 row(s) in 1.4290 seconds
```

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

HBase 적재 데이터 확인 (3/6)

표 5.6 HBase의 DriverCarInfo 테이블에 적재된 데이터 모습

	cf1							
RowKey	area _number	break _pedal	car _number	date	direct _light	speed	speed _pedal	steer _angle
00001030106102–Y0012	E06	0	Y0012	20160103 010000	N	23	3	F
00000030106102–Z0019	D03	0	Z0019	20160103 010000	N	60	3	F
00001030106102–Z0020	D04	0	Z0020	20160103 010000	N	25	1	F

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

HBase 적재 데이터 확인 (4/6)

```
$ hbase(main):001:0> scan 'DriverCarInfo', {STARTROW=>'00001030106102-Z0020', LIMIT=>1}
```

```
hbase(main):010:0> scan 'DriverCarInfo', {STARTROW=>'00001030106102-Z0020', LIMIT=>1}
ROW                          COLUMN+CELL
 00001030106102-Z0020        column=cf1:area_number, timestamp=1461423214785, value=D04
 00001030106102-Z0020        column=cf1:break_pedal, timestamp=1461423214785, value=0
 00001030106102-Z0020        column=cf1:car_number, timestamp=1461423214785, value=Z0020
 00001030106102-Z0020        column=cf1:date, timestamp=1461423214785, value=20160103010000
 00001030106102-Z0020        column=cf1:direct_light, timestamp=1461423214785, value=N
 00001030106102-Z0020        column=cf1:speed, timestamp=1461423214785, value=25
 00001030106102-Z0020        column=cf1:speed_pedal, timestamp=1461423214785, value=1
 00001030106102-Z0020        column=cf1:steer_angle, timestamp=1461423214785, value=F
1 row(s) in 0.0720 seconds
```

그림 5.40 HBase 적재 확인 – Scan 명령 2

로우키인 "00001030106102–Z0020"으로 조회된 결과를 보면 아래와 같다.

- car_number: Z0020 → 스마트카 차량 번호가 Z0020인 운전자의
- date: 20160103010000 → 2016년 1월 3일 01시 00분 00초 운행 정보는
- speed: 25 → 시속 25Km/h로 주행
- speed_pedal: 1 → 가속 페달을 1단계 진행
- steer_angle: F → 핸들은 직진 중
- break_pedal: 0 → 브레이크는 밟지 않은 상태
- direct_light: N → 깜박이는 켜지 않은 상태
- area_number: D04 → D04 지역을 운행

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

HBase 적재 데이터 확인 (5/6)

```
$ hbase(main):001:0> scan 'DriverCarInfo', {COLUMNS=>['cf1:car_number','cf1:area_number']
,FILTER=>"RowFilter(=,'regexstring:30106102') AND SingleColumnValueFilter( 'cf1' , 'area_
number' , = , 'regexstring:D04' )"}
```

```
 85631030106102-I0014          column=cf1:car_number, timestamp=1461423769210, value=I0014
 85641030106102-I0014          column=cf1:area_number, timestamp=1461423920515, value=D04
 85641030106102-I0014          column=cf1:car_number, timestamp=1461423920515, value=I0014
 85711030106102-I0014          column=cf1:area_number, timestamp=1461423484409, value=D04
 85711030106102-I0014          column=cf1:car_number, timestamp=1461423484409, value=I0014
 85711030106102-T0010          column=cf1:area_number, timestamp=1461423484367, value=D04
 85711030106102-T0010          column=cf1:car_number, timestamp=1461423484367, value=T0010
 85721030106102-Z0019          column=cf1:area_number, timestamp=1461423635722, value=D04
 85721030106102-Z0019          column=cf1:car_number, timestamp=1461423635722, value=Z0019
 85741030106102-I0014          column=cf1:area_number, timestamp=1461423935662, value=D04
 85741030106102-I0014          column=cf1:car_number, timestamp=1461423935662, value=I0014
 85741030106102-P0006          column=cf1:area_number, timestamp=1461423935528, value=D04
 85741030106102-P0006          column=cf1:car_number, timestamp=1461423935528, value=P0006
 85801030106102-I0014          column=cf1:area_number, timestamp=1461423350670, value=D04
 85801030106102-I0014          column=cf1:car_number, timestamp=1461423350670, value=I0014
 85801030106102-Z0019          column=cf1:area_number, timestamp=1461423350792, value=D04
 85801030106102-Z0019          column=cf1:car_number, timestamp=1461423350792, value=Z0019
 85841030106102-Z0020          column=cf1:area_number, timestamp=1461423950848, value=D04
 85841030106102-Z0020          column=cf1:car_number, timestamp=1461423950848, value=Z0020
 85950030106102-Z0020          column=cf1:area_number, timestamp=1461423214641, value=D04
 85950030106102-Z0020          column=cf1:car_number, timestamp=1461423214641, value=Z0020
798 row(s) in 10.7160 seconds
```

그림 5.41 HBase 적재 확인 – Scan 명령 3

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

HBase 적재 데이터 확인 (6/6)

03. HBase 웹관리자에 접속해서 적재한 데이터가 앞서 실행했던 Pre-Split 명령에 의해 3개의 HRegionServer로 골고루 분산 적재됐는지 확인한다.

- URL: http://server01.hadoop.com:16010/

저사양 파일럿 환경: HBase를 Server02에 설치했으므로 http://server02.hadoop.com:16010/으로 접속한다.

APACHE HBASE

Master server01.hadoop.com

Region Servers

Base Stats | Memory | Requests | Storefiles | Compactions

ServerName	Start time	Requests Per Second	Num. Regions
server01.hadoop.com,60020,1460859560464	Sun Apr 17 11:19:20 KST 2016	36	7
server02.hadoop.com,60020,1460859531411	Sun Apr 17 11:18:51 KST 2016	11	5
server03.hadoop.com,60020,1460859545940	Sun Apr 17 11:19:05 KST 2016	12	6
Total:3		59	18

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

Redis에 적재된 데이터 확인

Server02에 SSH로 접속해 다음 레디스 명령을 실행한다.

```
$ redis-cli
$ 127.0.0.1:6379> smembers 20160103
```

```
127.0.0.1:6379> smembers 20160103
1) "J0027-20160103001326"
2) "N0089-20160103000346"
```

그림 5.43 레디스 적재 확인 – 과속 운행 차량에 대한 정보 및 시간

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

Redis 클라이언트 애플리케이션 작동 (1/2)

먼저 C://예제소스/bigdata2nd-master/CH05/bigdata.smartcar.redis-1.0.jar 파일을 Server02의 /home/pilot-pjt/working 디렉터리에 업로드한다.

- FTP 클라이언트 파일질라 실행
- 업로드 경로: /home/pilot-pjt/working
- C://예제소스/bigdata2nd-master/CH05/bigdata.smartcar.redis-1.0.jar 파일을 Server02의 /home/pilot-pjt/working 경로에 업로드

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

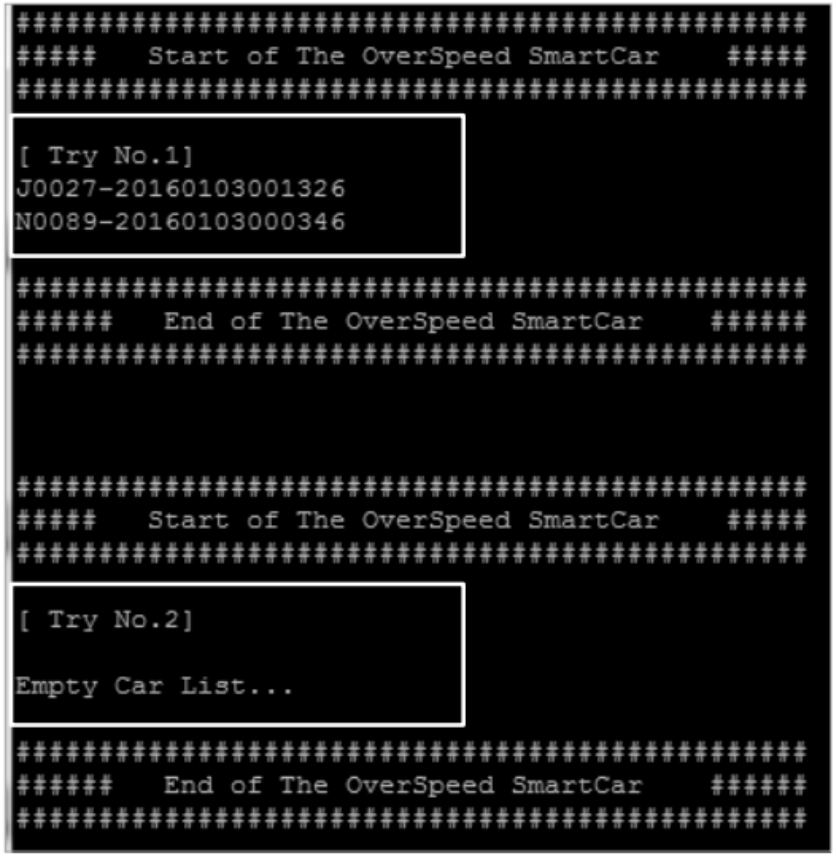
Redis 클라이언트 애플리케이션 작동 (2/2)

PuTTY 콘솔로 Server02에 접속해서 레디스 클라이언트 애플리케이션을 실행한다.

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

$ java -cp bigdata.smartcar.redis-1.0.jar com.wikibook.bigdata.smartcar.redis.OverSpeedCarInfo

20160103
```



5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

실시간 로그 시뮬레이터 중지

레디스에 과속 차량 정보까지 확인됐으면 로그 시뮬레이터를 종료한다.

```
$ ps -ef | grep smartcar.log
$ kill -9 [pid] [pid]
```

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

실습

5.6 실시간 적재 파일럿 실행 4단계 – 적재 테스트

저사양 파일럿 환경: 수집/적재 서비스를 일시 정지시킨다.

다음 6장, 7장의 탐색/분석 단계를 진행할 때 수집/적재 기능이 항상 필요하지는 않다. 파일럿 프로젝트를 원활하게 진행하기 위해 앞으로는 수집/적재 기능들을 일부 정지시켜가며 진행하겠다. 관련 소프트웨어로는 플럼, 카프카, 스톰, 레디스, HBase 등이 있다. 아래 명령으로 관련 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]