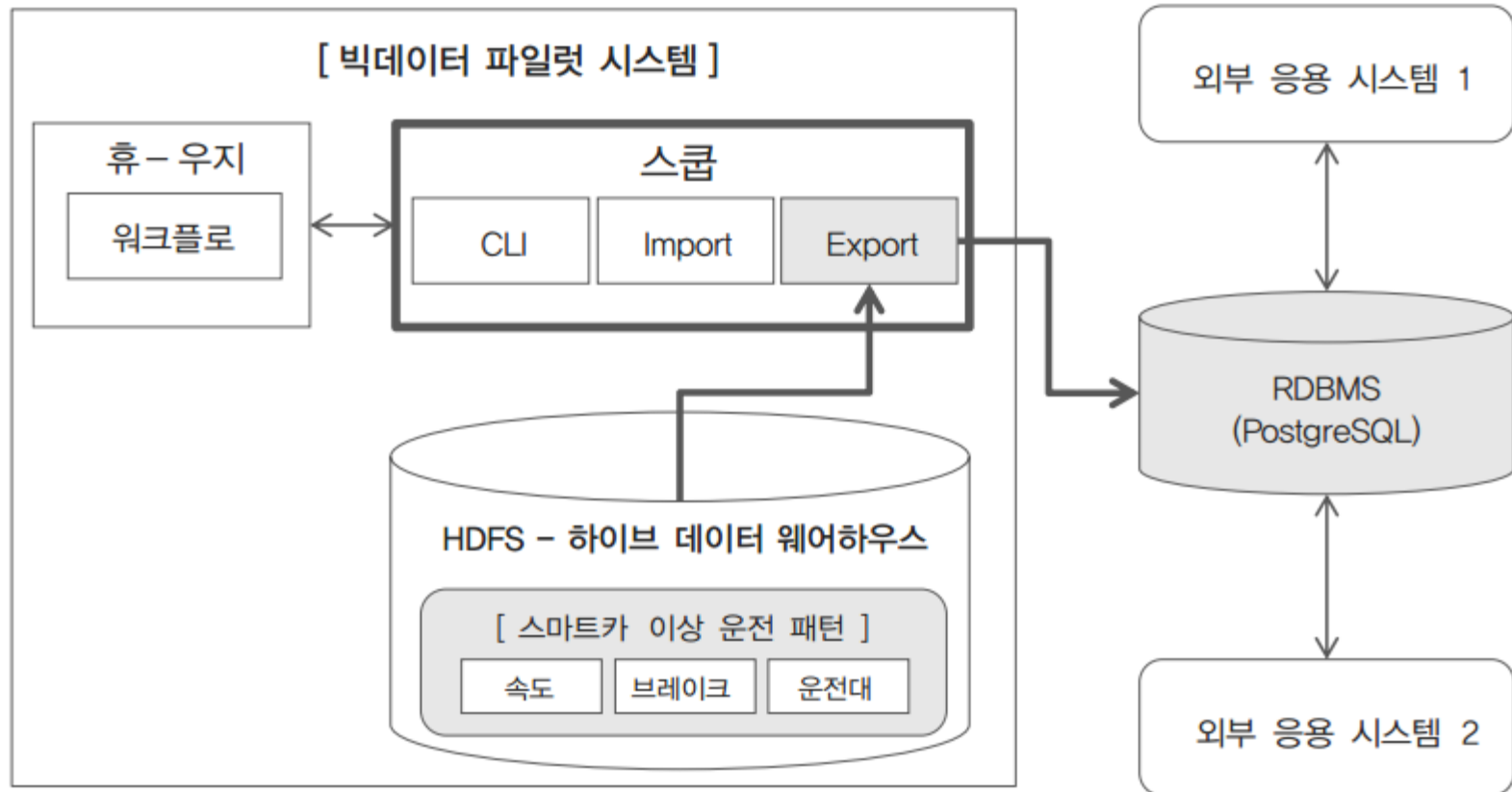


7.8 분석 파일럿 실행 6단계 - 스쿱

분석 결과 외부 제공



☆ 7.8 분석 파일럿 실행 6단계 - 스쿱

☞ 스쿱의 내보내기 기능 - 이상 운전 차량 정보

사용할 DBMS 정보 및 환경설정

- RDBMS 유형: PostgreSQL
- 설치 IP: 192.168.56.101
- 설치 포트: 7432
- 데이터베이스: postgresql
- 접속 계정: cloudera-scm
- 접속 패스워드: acjZt03D1Q

패스워드의 경우 독자들의 파일럿 환경마다 다소 차이가 있다. Server01에 접속해 다음 명령으로 확인한다.

```
$ cat /var/lib/cloudera-scm-server-db/data/generated_password.txt
```

```
[root@server01 ~]# cat /var/lib/cloudera-scm-server-db/data/generated_password.txt  
acjZt03D1Q
```

☆ 7.8 분석 파일럿 실행 6단계 - 스쿱

☞ 스쿱의 내보내기 기능 - 이상 운전 차량 정보

01. PostgreSQL에 콘솔로 접속해서 기본적인 데이터베이스 작업을 진행한다. 먼저 Server01의 SSH를 통해 접속한 후 다음 명령으로 PostgreSQL 데이터베이스에 연결한다.

```
$ psql -U cloudera-scm -p 7432 -h localhost -d postgres
```

패스워드 입력창이 나타나면 그림 7.108에서 확인한 패스워드를 입력하기 바란다. 필자의 경우 "acjZt03D1Q"라는 패스워드가 만들어져 있었다.

```
[root@server01 ~]# psql -U cloudera-scm -p 7432 -h localhost -d postgres
Password for user cloudera-scm:
psql (8.4.20)
Type "help" for help.

postgres=#
```

7.8 분석 파일럿 실행 6단계 - 스쿱

스쿱의 내보내기 기능 - 이상 운전 차량 정보

02. HDFS에 “이상 운전 패턴” 데이터를 PostgreSQL의 테이블로 저장하기 위한 RDBMS 테이블을 생성한다. 테이블명은 smartcar_symptom_info로 하겠다. 다음의 테이블 생성 명령을 실행한다.

```
postgres=# create table smartcar_symptom_info (
    car_number        varchar,
    speed_p_avg        varchar,
    speed_p_symptom    varchar,
    break_p_avg        varchar,
    break_p_symptom    varchar,
    steer_a_cnt        varchar,
    steer_p_symptom    varchar,
    biz_date           varchar
);
```

```
postgres=# create table smartcar_symptom_info (
    car_number        varchar,
    speed_p_avg        varchar,
    speed_p_symptom    varchar,
    break_p_avg        varchar,
    break_p_symptom    varchar,
    steer_a_cnt        varchar,
    steer_p_symptom    varchar,
    biz_date           varchar
);
CREATE TABLE
```

☆ 7.8 분석 파일럿 실행 6단계 - 스쿱

☑ 스쿱의 내보내기 기능 - 이상 운전 차량 정보

SELECT 쿼리를 실행해서 정상적으로 생성됐는지 확인해 본다.

```
postgres=# select * from smartcar_symptom_info;
```

```
postgres=# select * from smartcar_symptom_info;
 car_number | speed_p_avg | speed_p_symptom | break_p_avg | break_p_symptom | steer_a_cnt | steer_p_symptom | biz
_date
-----+-----+-----+-----+-----+-----+-----+-----
(0 rows)
```

☆ 7.8 분석 파일럿 실행 6단계 - 스쿱

☞ 스쿱의 내보내기 기능 - 이상 운전 차량 정보

03. Server01에서 스쿱 명령을 실행하기 위해 새로운 SSH 창을 하나 더 열어 Server01에 접속한다. 먼저 스쿱 명령을 실행하기 위해서는 PostgreSQL JDBC 드라이버를 스쿱의 라이브러리 경로에 복사해 놓아야 한다. 다음 복사 명령을 실행한다.

```
$ cp /opt/cloudera/parcels/CDH/jars/postgresql-*.jar /opt/cloudera/parcels/CDH/lib/sqoop/lib
```

다음으로 스쿱 내보내기 명령을 실행한다. 패스워드 옵션은 독자의 파일럿 환경에서 생성된 값으로 입력해야 한다.

```
$ sqoop export --connect jdbc:postgresql://192.168.56.101:7432/postgres --username
cloudera-scm --password acjZt03D1Q --table smartcar_symptom_info --export-dir /user/hive/
warehouse/managed_smartcar_symptom_info
```

- -username: PostgreSQL 계정
- -password: PostgreSQL 패스워드
- -table: PostgreSQL 데이터베이스 테이블명
- -export-dir: 내보내기 할 HDFS 데이터 경로

```
INFO mapreduce.Job: Running job: job_1476589912276_0004
INFO mapreduce.Job: Job job_1476589912276_0004 running in uber mode : false
INFO mapreduce.Job:  map 0% reduce 0%
INFO mapreduce.Job:  map 25% reduce 0%
INFO mapreduce.Job:  map 100% reduce 0%
INFO mapreduce.Job: Job job_1476589912276_0004 completed successfully
INFO mapreduce.Job: Counters: 30
```

7.8 분석 파일럿 실행 6단계 - 스쿱

스쿱의 내보내기 기능 - 이상 운전 차량 정보

명령이 정상적으로 실행되면 하이브의 Managed_SmartCar_Symptom_Info 테이블의 데이터 전체가 PostgreSQL의 smartcar_symptom_info 테이블에 내보내기됐을 것이다. PostgreSQL 콘솔로 접속해 다음의 SELECT 쿼리를 실행해 보면 결과를 확인할 수 있다.

```
postgres=# select * from smartcar_symptom_info;
```

```
postgres=# select * from smartcar_symptom_info;
 car_number | speed_p_avg | speed_p_symptom | break_p_avg | break_p_symptom | steer_a_cnt | steer_p_symptom | biz_date
-----+-----+-----+-----+-----+-----+-----+-----
A0005      | 1.5097708894878705 | 정상 | 0.3028975741239892 | 정상 | 565 | 정상 | 20160626
A0061      | 1.5590924483576025 | 정상 | 0.2885201490010159 | 정상 | 594 | 정상 | 20160626
A0078      | 1.5018688413183825 | 정상 | 0.3119266055045872 | 정상 | 576 | 정상 | 20160626
A0100      | 1.489788972089857 | 정상 | 0.31994554118447927 | 정상 | 598 | 정상 | 20160626
B0012      | 1.5561930475869052 | 정상 | 0.28552143098211274 | 정상 | 608 | 정상 | 20160626
```

7.8 분석 파일럿 실행 6단계 - 스쿱

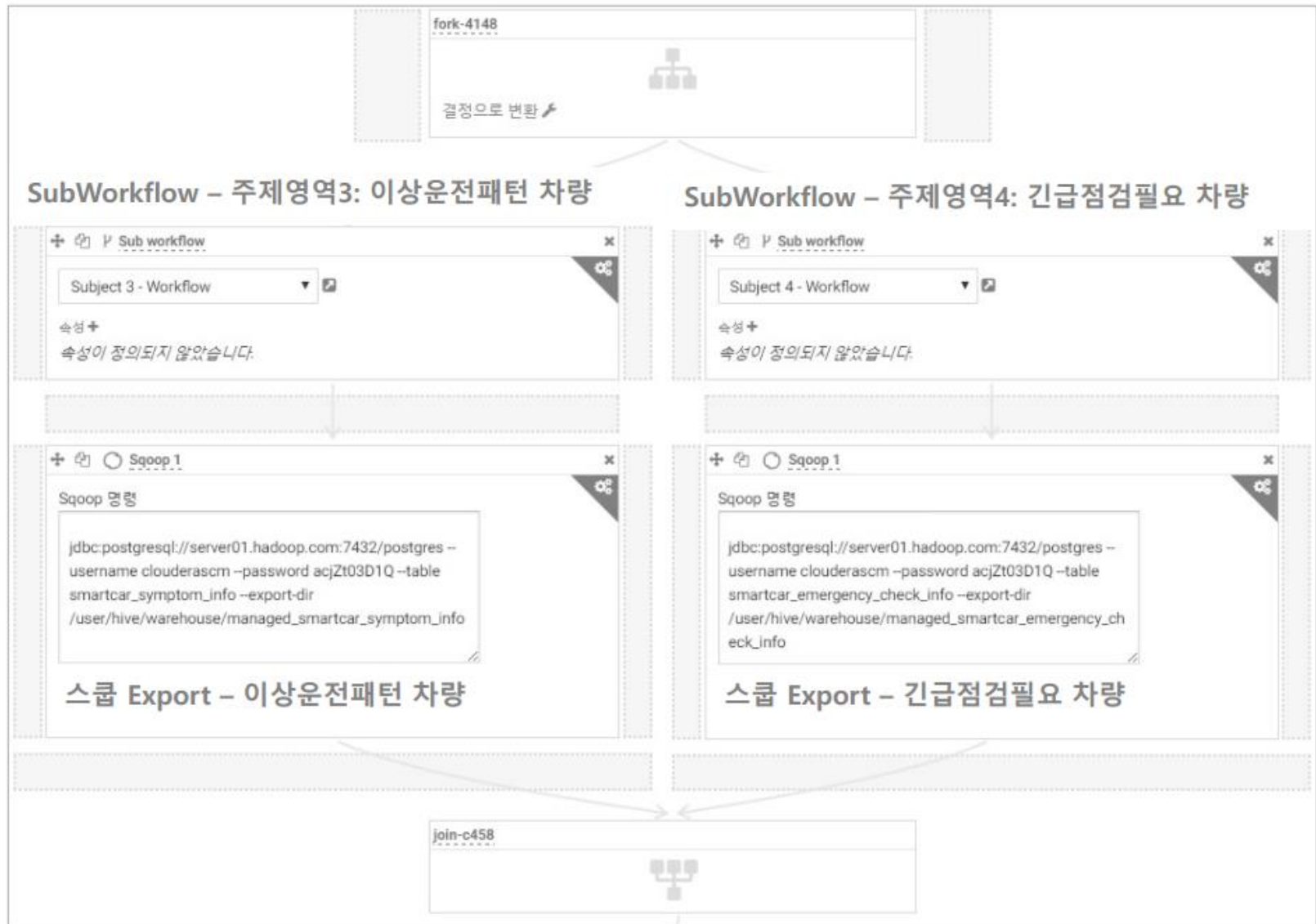


그림 7.114 스쿱작업 워크플로 작성 - Subworkflow 및 Fork 활용

7.8 분석 파일럿 실행 6단계 - 스쿱

Tip _ 스쿱을 활용할 때 주의할 점

스쿱을 활용할 때는 주의해야 할 점이 있다. 일반적으로 RDBMS는 중요 업무 시스템의 온라인 서비스 저장소로 사용되는데, 이때 스쿱의 대규모 Export/Import 맵(Map) 작업들이 RDBMS에 연결되면서 큰 오버헤드를 발생시킬 수 있기 때문이다. 예를 들어, 100대의 하둡 데이터노드가 있고, 스쿱 명령으로 대규모 데이터에 대한 가져오기/내 보내기를 할 경우 최악의 경우에 100대의 서버에서 동시에 Map 작업이 생성되어 RDBMS로 연결된 JDBC 작업이 수행될 수 있기 때문이다. 이로 인해 RDBMS의 자원 점유율이 커지고 RDBMS를 사용하는 중요 온라인 시스템 서비스에 영향을 줄 수 있다.

스쿱의 실행 옵션 중 `--split-by` 또는 `--num-mappers`를 이용하면 문제를 최소화할 수 있으니 참고하기 바란다.

7.8 분석 파일럿 실행 6단계 - 스콧

 분석 결과 외부 제공

실습