

5강

빅데이터 탐색



미사용 서비스 중지

Hbase 구성 요소

Cloudera **Manager** 클러스터 ▼

Cluster 1

! **Kafka**

작업 ▼

상태

인스턴

시작

재시작

롤링 재시작

중지

상태 테스트

Cloudera **Manager** 클러스터 ▼

Cluster 1

● **Flume**

작업 ▼

상태

인스턴

시작

중지

재시작

하이프

Hive 설치

Cloudera Manager

클러스터

호스트

Cluster 1

작업

상태

서비스 추가

Add Hosts

상태

☐ HDFS

☒ Hive

☐ Hue

Select Dependencies

HBase

HDFS

☐

HDFS

☒ HBase

HDFS

Gateway × 1 새로 만들기

server02.hadoop.com

Hive Metastore Server × ...

server02.hadoop.com

WebHCat Server

호스트 선택

HiveServer2 × 1 새로 만들기

server02.hadoop.com

Hive

✓ 건너뛰었습니다. 이후 단계에서 Cloudera Manager가 이 데이터베이스를 생성합니다.

유형

호스트 이름 *

데이터베이스 이름 *

사용자 이름 *

암호 *

PostgreSQL

server01.hadoop.com:7432hive

hive

ADp5DzwznR

테스트 연결

우지 설치

Cloudera Manager 클러스터 ▼ 호스트

Cluster 1 작업 ▼ 상태

서비스 추가

Add Hosts

상태

☐ Kudu

☒ Oozie

☐ S3 Connector

Select Dependencies

<input checked="" type="radio"/> HBase	<input checked="" type="radio"/> HDFS
<input type="radio"/>	HDFS
<input checked="" type="radio"/> HBase	HDFS

Oozie Server × 1 새로 만들기

server02.hadoop.com ▼

Oozie Server

✓ 건너뛰었습니다. 이후 단계에서 Cloudera Manager가 이 데이터베이스를 생성합니다.


현재 server02.hadoop.com에서 실행하도록 할당되었습니다.

유형	호스트 이름 *	데이터베이스 이름 *	사용자 이름 *	암호 *
PostgreSQL ▼	server01.hadoop.com:7432	oozie_oozie_server	oozie_oozie_server	GwvULKc0dv

테스트 연결

우지 설정값 변경

Cluster 1

 Oozie

작업 ▼

상태

인스턴스

구성

명령

차트 라C

memory

Default Launcher Memory

oozie.launcher.default.memory.
mb

Oozie Server Default Group ↶

1

GiB



휴 설치

- ❖ `yum install centos-release-scl`
- ❖ `yum install scl-utils`
- ❖ `yum install python27`
- ❖ `python --version`
- ❖ `curl -k -O https://bootstrap.pypa.io/get-pip.py`
- ❖ `python get-pip.py`
- ❖ `yum install postgresql-devel`
- ❖ `bash -c "source /opt/rh/python27/enable; pip install psycopg2==2.6.2 --ignore-installed"`

휴설치

Cloudera **Manager** 클러스터 ▾ 호스트

! Cluster 1

작업 ▾

상태

서비스 추가

Add Hosts

상태

☐ Hive

☒ Hue

☐ Impala

☒ HBase

☐ HDFS

☐ Hive

Hue Server x ...

server02.hadoop.com...

Load Balancer

호스트 선택 ▾

spark설치

Cloudera Manager 클러스터 ▾ 호스트

Cluster 1 작업 ▾ 상태

서비스 추가

Add Hosts

상태

☐ Solr

☒ Spark

☐ Sqoop 1 Client

HBase	HDFS
<input checked="" type="radio"/> HBase	HDFS
<input type="radio"/>	HDFS

History Server × ...

server02.hadoop.com...

Gateway × ...

server02.hadoop.com...

휴를 이용한 데이터 탐색

환경 설정

- ❖ 클러스터 재시작
- ❖ 수집, 적재 및 모니터링 기능 모두 정지
 - Flume, kafka, Hbase, Cloudera Management

Cloudera Manager 클러스터 ▾ 호스트 ▾ 진단

Cloudera Management **모든 호스트** Add Hosts

모든 호스트 클릭
역할 확장 후
Cloudera management 클릭

IP	역할
192.168.56.101	11 Role(s)
	HDFS Balancer
	! HDFS DataNode
	! HDFS NameNode
	! HDFS
	SecondaryNameNode
	! Cloudera Management Service
	Alert Publisher
	! Cloudera Management Service
	Event Server

Cloudera **Manager** 클러스터 ▾ 호스트 ▾ 진단 ▾

Cloudera Management Service / server01

! Alert Publisher 작업 ▾

상태 구성 프로세스 명령어 하트 카이

Cloudera **Manager** 클러스터 ▾ 호스트 ▾ 진단 ▾ 감사 차트 ▾

상태 테스트 **!** Cloudera Management Service 작업 ▾

상태 인스턴스 구성 명령어 차트 라이브러리

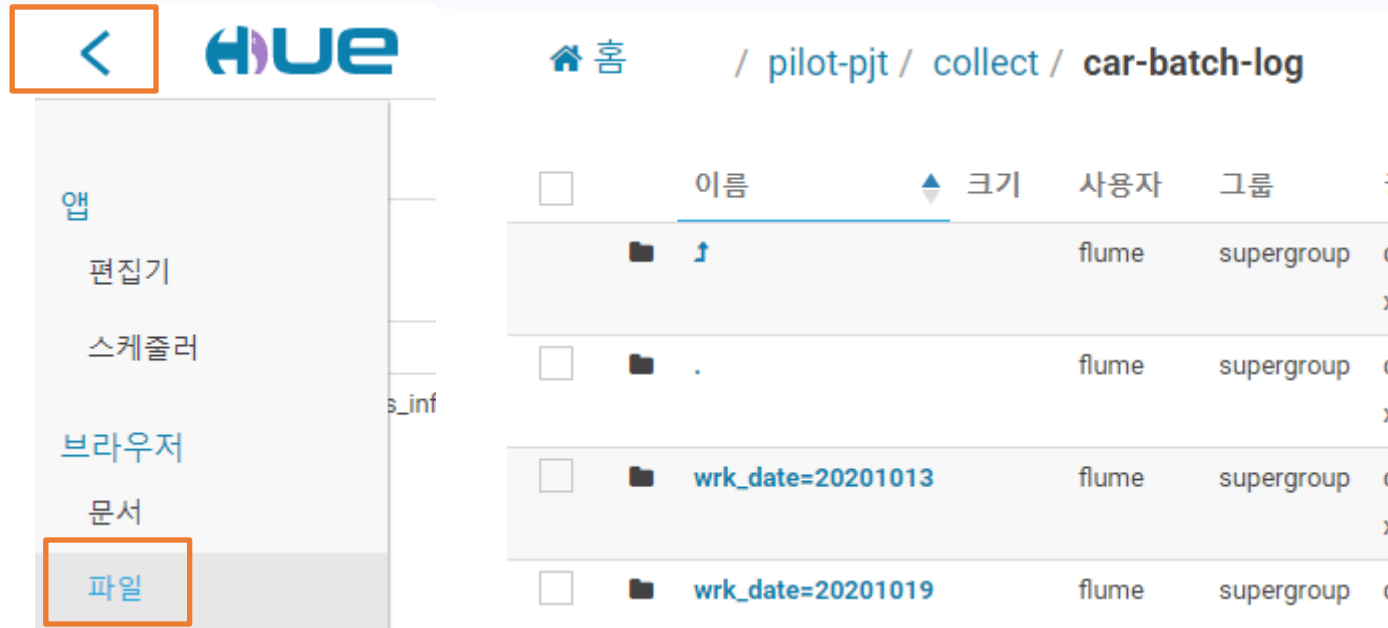
시작

중지

재시작

Cloudera management Service 클릭
중지 클릭

하둡에 저장된 파일 확인



The screenshot shows the HUE web interface. On the left sidebar, the '파일' (File) button is highlighted with an orange box. The main area displays the file browser for the path `/ pilot-pjt / collect / car-batch-log`. The table below shows the contents of this directory.

<input type="checkbox"/>	이름	크기	사용자	그룹
<input type="checkbox"/>	↑		flume	supergroup
<input type="checkbox"/>	.		flume	supergroup
<input type="checkbox"/>	wrk_date=20201013		flume	supergroup
<input type="checkbox"/>	wrk_date=20201019		flume	supergroup

<http://server02.hadoop.com:8888/>

하이브를 이용한 External 데이터 탐색

스마트카 상태 정보 테이블 생성

The screenshot shows the HUE web interface. At the top, there is a navigation bar with the HUE logo and a '쿼리' (Query) dropdown menu. The dropdown menu is open, showing options: '</> 편집기' (Editor), '스케줄러' (Scheduler), 'Impala', 'Hive' (selected), 'Pig', 'Java', and 'Spark'. On the left sidebar, there is a 'default' database selected, and a list of tables including 'managed_smartcar_status_info' and 'smartcar_drive_info'. The main area displays a SQL query for creating a new table:

```
1 create external table if not exists SmartCar_Status_Info (  
2 reg_date string,  
3 car_number string,  
4 tire_fl string,  
5 tire_fr string,  
6 tire_bl string,  
7 tire_br string,  
8 light_fl string,  
9 light_fr string,  
10 ...)
```


분석 데이터 추가

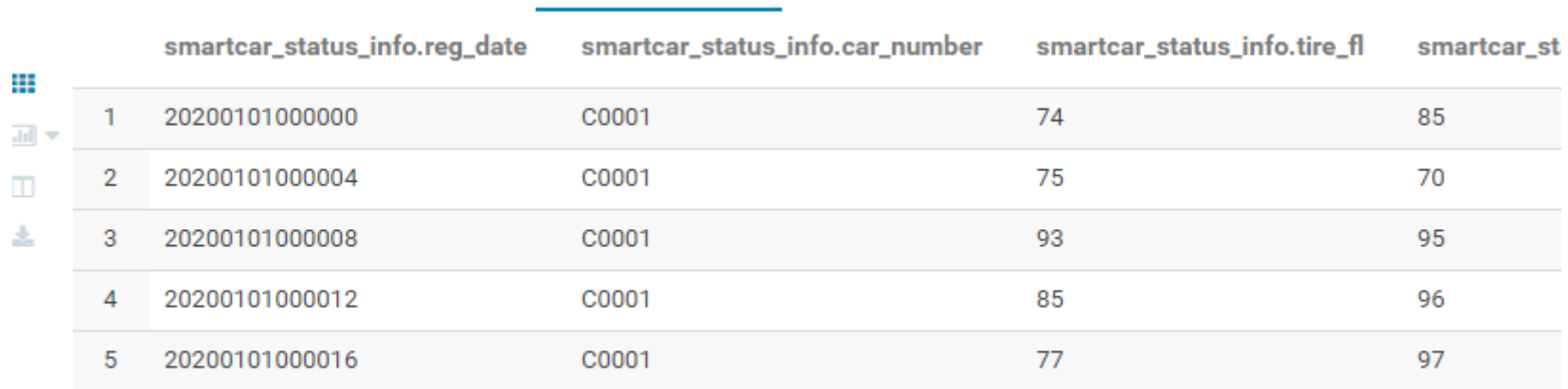
```
ALTER TABLE  
SmartCar_Status_Info ADD  
PARTITION(wrk_date='20201019');
```

🏠 홈 / pilot-pjt / collect / car-batch-log

<input type="checkbox"/>	이름	크기	사용자	그룹
<input type="checkbox"/>	↑		flume	supergroup
<input type="checkbox"/>	.		flume	supergroup
<input type="checkbox"/>	wrk_date=20201013		flume	supergroup
<input type="checkbox"/>	wrk_date=20201019		flume	supergroup

결과 확인

```
select * from SmartCar_Status_Info limit 5;
```



	smartcar_status_info.reg_date	smartcar_status_info.car_number	smartcar_status_info.tire_fl	smartcar_st
1	20200101000000	C0001	74	85
2	20200101000004	C0001	75	70
3	20200101000008	C0001	93	95
4	20200101000012	C0001	85	96
5	20200101000016	C0001	77	97

하둡 데이터 업로드

< HUE

앱

- 편집기
- 스케줄러
- 브라우저
- 문서
- 파일

업로드 새로 만들기

파일

디렉토리

<input type="checkbox"/>	buy-list
<input type="checkbox"/>	car-batch-log
<input type="checkbox"/>	car-master

업로드

홈 / pilot-pjt / collect / car-master

<input type="checkbox"/>	이름
<input type="checkbox"/>	↑
<input type="checkbox"/>	.
<input type="checkbox"/>	CarMaster.txt

홈 / pilot-pjt / collect / buy-list

<input type="checkbox"/>	이름
<input type="checkbox"/>	↑
<input type="checkbox"/>	.
<input type="checkbox"/>	CarItemBuyList_202003.txt

Car_master 테이블 생성

select * from smartcar_master

```
7 job string,  
8 car_capacity string,  
9 car_year string,  
10 car_model string  
11 )  
12 row format delimited  
13 fields terminated by '|'   
14 stored as textfile  
15 location '/pilot-pjt/collect/car-master';|
```

결과 (100+)				
	smartcar_master.car_number	smartcar_master.sex	smartcar_master.age	smartcar_ma
1	A0001	여	32	미혼
2	A0002	남	53	미혼
3	A0003	여	62	기혼
4	A0004	남	31	미혼
5	A0005	남	67	미혼
6	A0006	여	30	미혼
7	A0007	남	61	미혼
8	A0008	여	20	미혼
9	A0009	여	60	미혼

구매내역 테이블 생성

```
1  
2 CREATE EXTERNAL TABLE SmartCar_Item_BuyList (  
3   car_number string,  
4   Item string,  
5   score string,  
6   month string  
7 )  
8 row format delimited  
9 fields terminated by ','
```

select * from smartcar_item_buylist

쿼리 기록

시각화 쿼리

결과 (100+)



	smartcar_item_buylist.car_number	smartcar_item_buylist.item	smartcar_i
1	M0014	Item-018	2
2	G0035	Item-015	3
3	I0090	Item-009	3
4	K0095	Item-018	5
5	Y0042	Item-020	2
6	W0023	Item-030	2
7	Y0036	Item-023	3
8	T0026	Item-028	1

Spark 활용

```
[root@server02 ~]# spark-shell
```

Setting default log level to "WARN".

To adjust logging level use `sc.setLogLevel(newLevel)`. For SparkR, use `setLogLevel(newLevel)`.

```
scala> val smartcar_master_df=spark.sqlContext.sql("select * from  
smartcar_master where age >= 18")
```

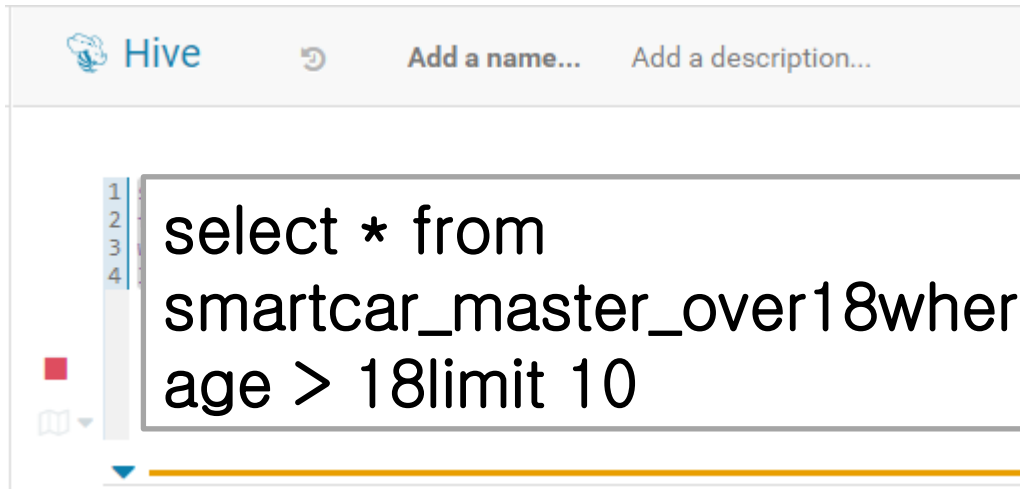
smartcar_master_df: org.apache.spark.sql.DataFrame = [car_number: string, sex: string ... 7 more fields]

```
scala> smartcar_master_df.show()
```

car_number	sex	age	marriage	region	job	car_capacity	car_year	car_model
A0001	여	32	미혼	서울	프리랜서	1000	2009	F
A0002	남	53	미혼	충남	주부	2500	2015	A
A0003	여	62	기혼	대전	회사원	2500	2012	B
A0004	남	31	미혼	광주	공무원	2000	2010	D
A0005	남	67	미혼	대구	공무원	1700	2002	C

hive와의 연동

```
scala> smartcar_master_df.write.saveAsTable("SmartCar_Master_Over18")
```

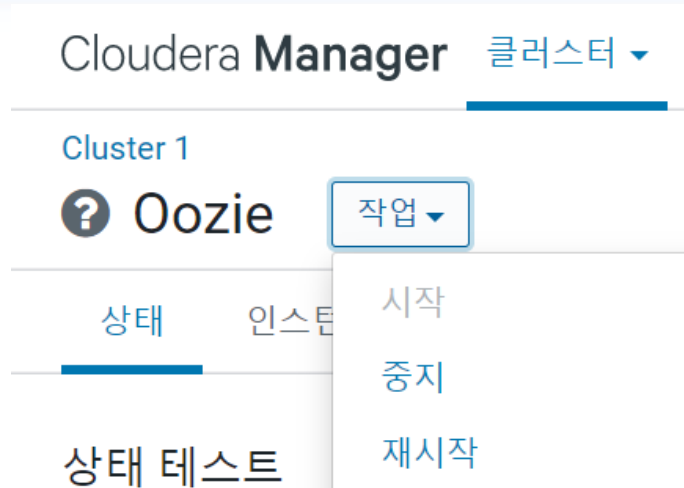


쿼리 기록 저장된 쿼리 결과 (10)

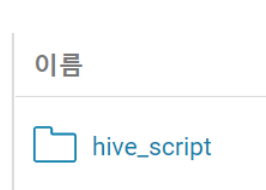
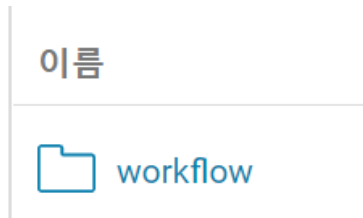
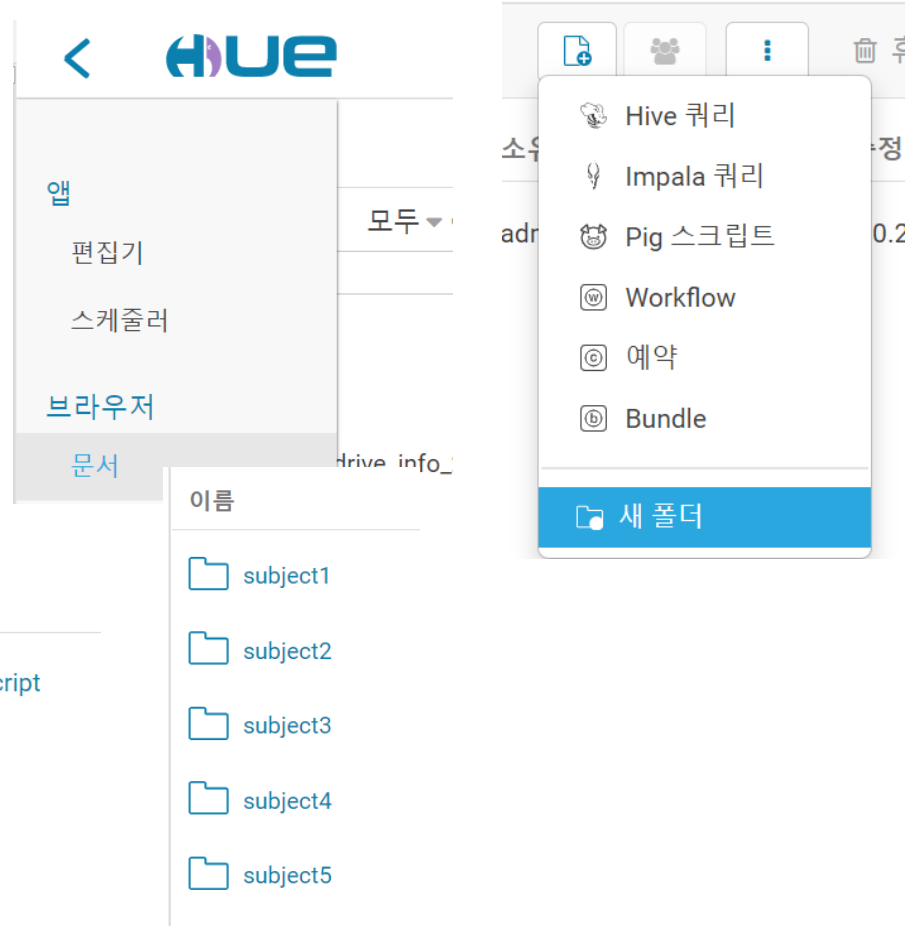
	smartcar_master_over18.car_number	smartcar_master_over18.sex	smartca
1	A0001	여	32
2	A0002	남	53
3	A0003	여	62
4	A0004	남	31
5	A0005	남	67

예약

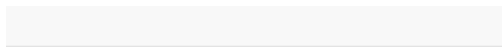
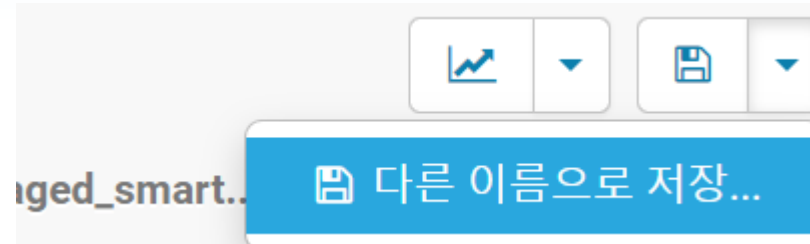
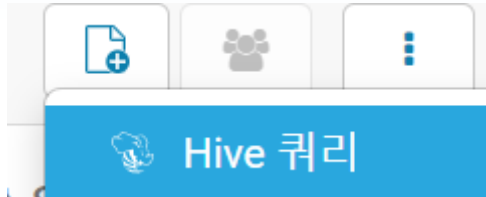
디렉토리 생성



<http://server02.hadoop.com:8888/>



create_table.hql, alter.hql, insert.hql 생성



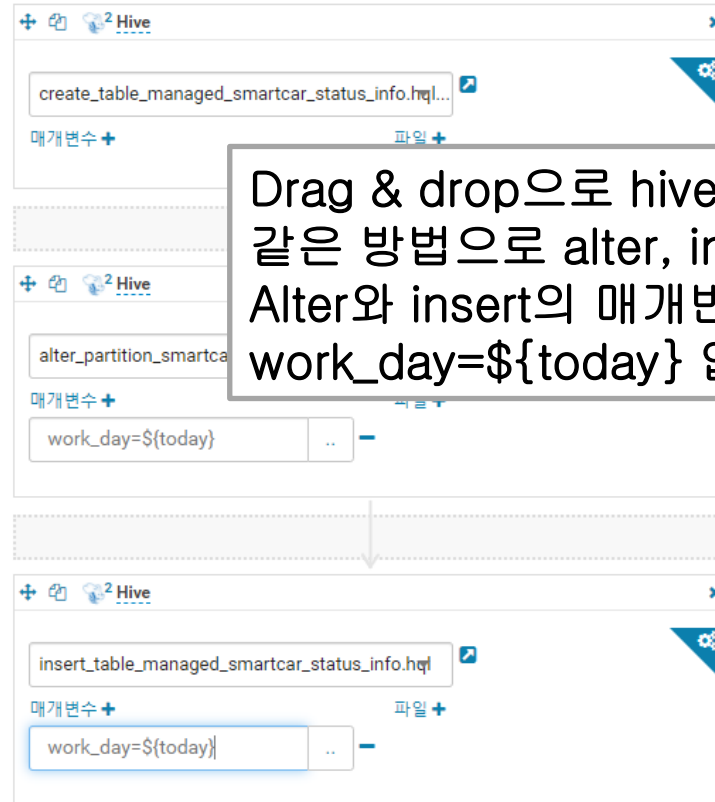
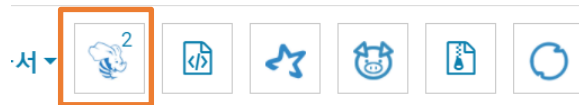
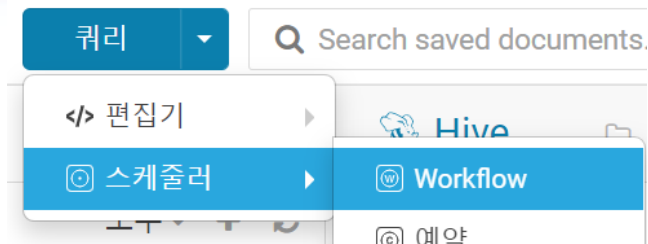
```
1 create table if not
2 car_number string,
3 sex string,
4 age string,
5 marriage string,
6 region string,
7 job string,
8 car_capacity string,
9 car_year string,
```

쿼리를 다른 이름으로 저장...

이름

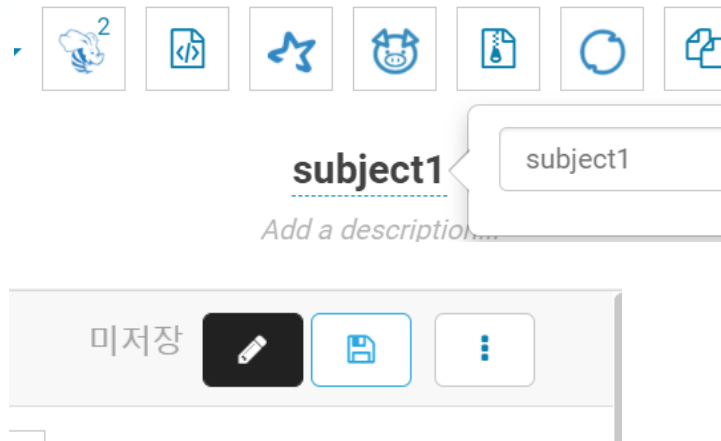
설명

스케줄러 만들기

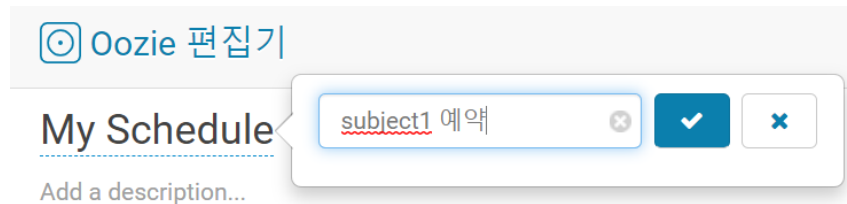
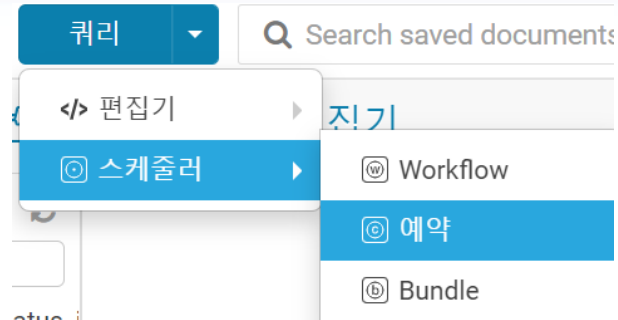


Drag & drop으로 hive 이동
같은 방법으로 alter, insert 추가
Alter와 insert의 매개변수 추가하여
work_day=\${today} 입력

프로젝트 저장 및 예약



위와 같이 subject1으로 저장
예약 진행하고 이름 및 workflow 선택



예정된 Workflow는 무엇입니까?

Workflow 선택...



예약 설정

간격은 얼마입니까?

다음 마다 일 다음에서 1 : 0

[숨기기](#)

☐ 고급 구문

시간대 Asia/Dili

원본 2020-10-27 20:30

-> 2020-11-03 20:30

매일 1시에 예약 진행

아래 내용으로 work_day 설정

```
$(coord:formatTime(coord:dateTzOffset(coord:nominalTime(), "Asia/Seoul"), 'yyyyMMdd'))}
```

매개 변수

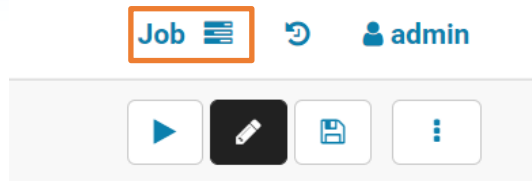
work_day

매개 변수 ▼

\$(coord:formatTime(coord:dateT

[+ 매개 변수 추가](#)

예약확인



오른쪽 상단의 job을 클릭하면 예정정보를 확인할 수 있으며 현재 DB 문제로 실패함

Job Browser

Job 쿼리 Workflow 일정 Bundle SLAs

0000000-201027200907774-oozie-oozi-W

> oozie:launcher:T=hive2:W=subject1 - Workflow:A=hive-1350:ID=0000000-2010272

ID
application_16037
96910539_0001

로그 시도

유형
Oozie Launcher

진행률
100%

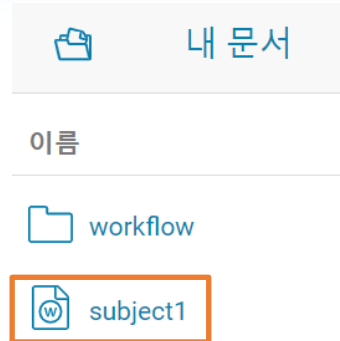
상태
FAILED

시작 시간

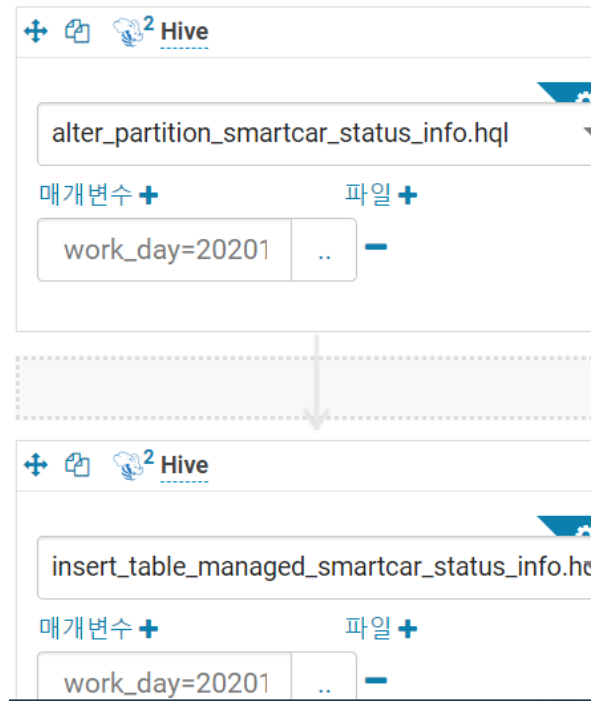
default stdout stderr syslog

Main Class [org.apache.oozie.action.hadoop.Hive2Main], exit code [2

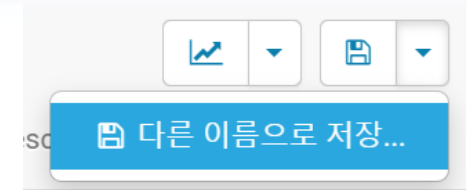
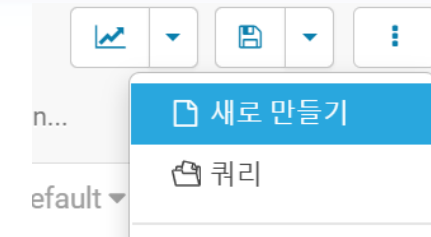
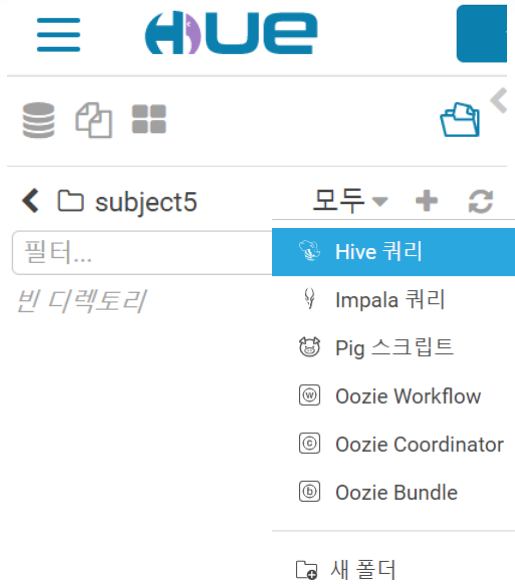
실습을 위한 workflow 직접실행



Alter와 insert의 매개변수를
“work_day=20201019”와 같이 변경하고 저
장 후 실행



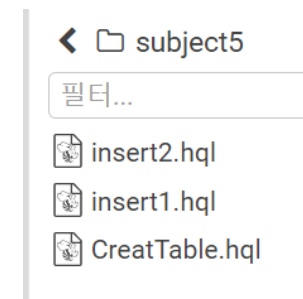




```

2  car_number string,
3  sex string,
4  age string,
5  marriage string,
6  region string,
7  job string,
8  car_capacity string
9  car_year string,
10 car_model string

```



My Workflow

Add a description...

subject2



Hive

CreatTable.hql

매개변수 +

파일 +

Hive

insert1.hql

매개변수 +

파일 +

Hive

insert2.hql

매개변수 +

파일 +