

6.6 탐색 파일럿 실행 4단계 - 주제영역 2

 스마트카 운전자 운행 기록 정보

과제 I

다음 페이지 부터 내용들을 실습해서 주제영역 2

“스마트카 운전자 운행 기록 정보” 마트를 생성 하세요.

주제 영역 2. 스마트카 운전자 운행 기록 정보 - 워크플로 작성 _ 실습

주제 영역 2부터는 많은 부분이 주제 영역 1과 유사한 작업이 반복되므로 지면상 약식으로 진행한다. 이번 장의 워크플로 작성이 어렵게 느껴지는 독자의 경우 주제 영역 1을 다시 한번 꼼꼼히 살펴보기 바란다.

이번 주제 영역 2의 워크플로는 2020년 03월 22일자로 HBase의 테이블에 적재된 “스마트카 운전자의 운행 데이터”를 우지 워크플로를 이용해 하이브의 Managed 영역인 Mart 테이블로 매일 이동시키는 프로세스다. 기억을 되살려 보면 HBase에 적재된 “스마트카 운전자 운행 데이터”는 하이브의 HBase 핸들러라는 것을 이용해 하이브의 테이블(SmartCar_Drive_Info)에 연결해서 하이브의 조회로 확인이 가능했다. 이를 이용해 “스마트카 운전자 운행 데이터”와 “스마트카 마스터 데이터”를 조인해서 좀 더 확장된 스마트카 운전자 운행 데이터를 만든다. 워크플로의 하이브 작업에 사용되는 하이브 QL은 C://예제소스/bigdata2nd-master/CH06/HiveQL/의 경로에서 제공되므로 필요 시 해당 파일을 열어 참고한다.

저사양 파일럿 환경: HBase 서비스를 시작한다.

- HBase 서비스: CM 홈 → [HBase] → [시작]

01. 휴의 좌측 드롭박스 메뉴에서 [문서]를 선택해 [내 문서]에 생성해 놓은 주제 영역 2의 작업 디렉터리로 이동한다.

- 휴 내 문서: /workflow/hive_script/subject2

02. 주제 영역 2에서는 사용할 하이브 스크립트 파일 4개를 작성한다. 먼저 내 문서의 /workflow/hive_script/subject2로 이동해서 우측 상단의 [새 문서] → [Hive 쿼리]를 클릭한다.

03. 스마트카 운전자의 운행 기록을 저장하기 위한 CREATE TABLE 스크립트를 작성하고, 상단의 [저장] 버튼을 클릭해 파일 이름을 “create_table_smartcar_drive_info_2.hql”로 입력한 후 저장한다.

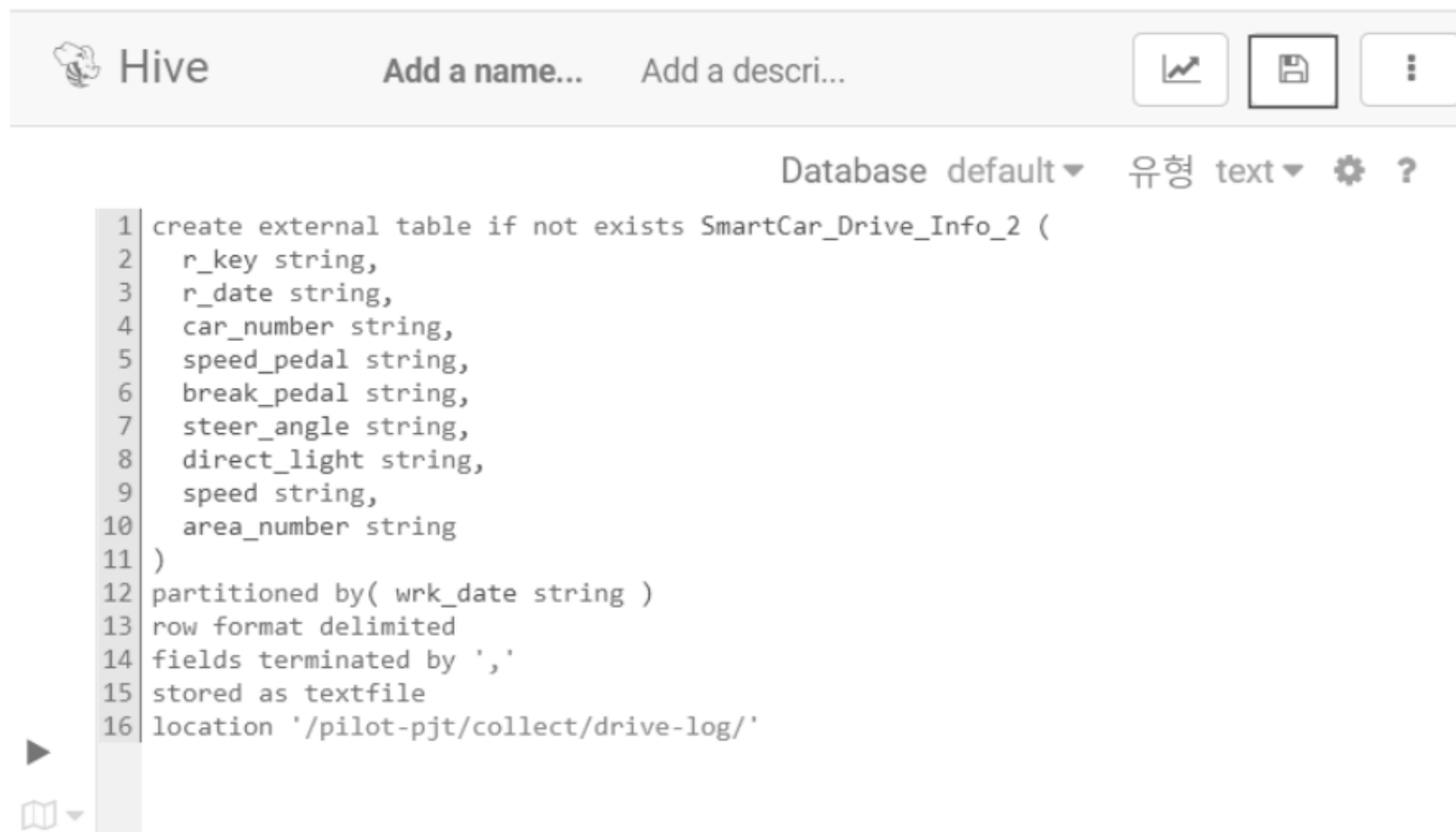


그림 6.97 주제 영역 2의 운행 기록을 관리하기 위한 하이브 테이블 생성 쿼리

그림 6.97의 SmartCar_Drive_Info_2 테이블은 HBase의 테이블에 연결된 SmartCar_Driver_Info 데이터를 하이브 테이블로 재구성하기 위해 생성한 테이블이다.

04. 계속해서 내 문서의 /workflow/hive_script/subject2에 두 번째 하이브 스크립트 파일을 만들어 본다. subject2 디렉터리에서 [새 문서] → [Hive 쿼리]를 클릭한다.

05. 하이브 에디터 창이 활성화되면 Hbase의 테이블에 연결된 SmartCar_Drive_Info 테이블로부터 “2020년 03월 22일”에 발생한 운행 데이터를 조회해서 하이브 테이블인 SmartCar_Drive_Info_2에 등록하기 위한 동적 파티션 설정과 쿼리를 작성한다. 상단의 [저장] 버튼을 클릭하고 파일 이름은 “insert_table_smartcar_drive_info_2.hql”로 입력하고 저장한다.

- set hive.exec.dynamic.partition=true;
- set hive.exec.dynamic.partition.mode=nonstrict;



그림 6.98 주제 영역 2의 운행 데이터 생성 하이브 쿼리

06. 내 문서의 /workflow/hive_script/subject2에 세 번째 하이브 스크립트 파일을 만든다. subject2 디렉터리에서 [새 문서] → [Hive 쿼리]를 클릭한다.

07. 하이브 에디트 창이 활성화되면 하이브의 Managed 영역에 운행 데이터를 저장하기 위한 테이블 생성 스크립트를 작성하고 저장한다. 파일 이름은 "create_table_managed_smartcar_drive_info.hql"로 지정한다. 아래 그림 6.99를 자세히 보면 "스마트카 마스터" 데이터와 "스마트카 운전자의 운행" 데이터가 결합된 테이블로 생성하는 것을 확인할 수 있다.

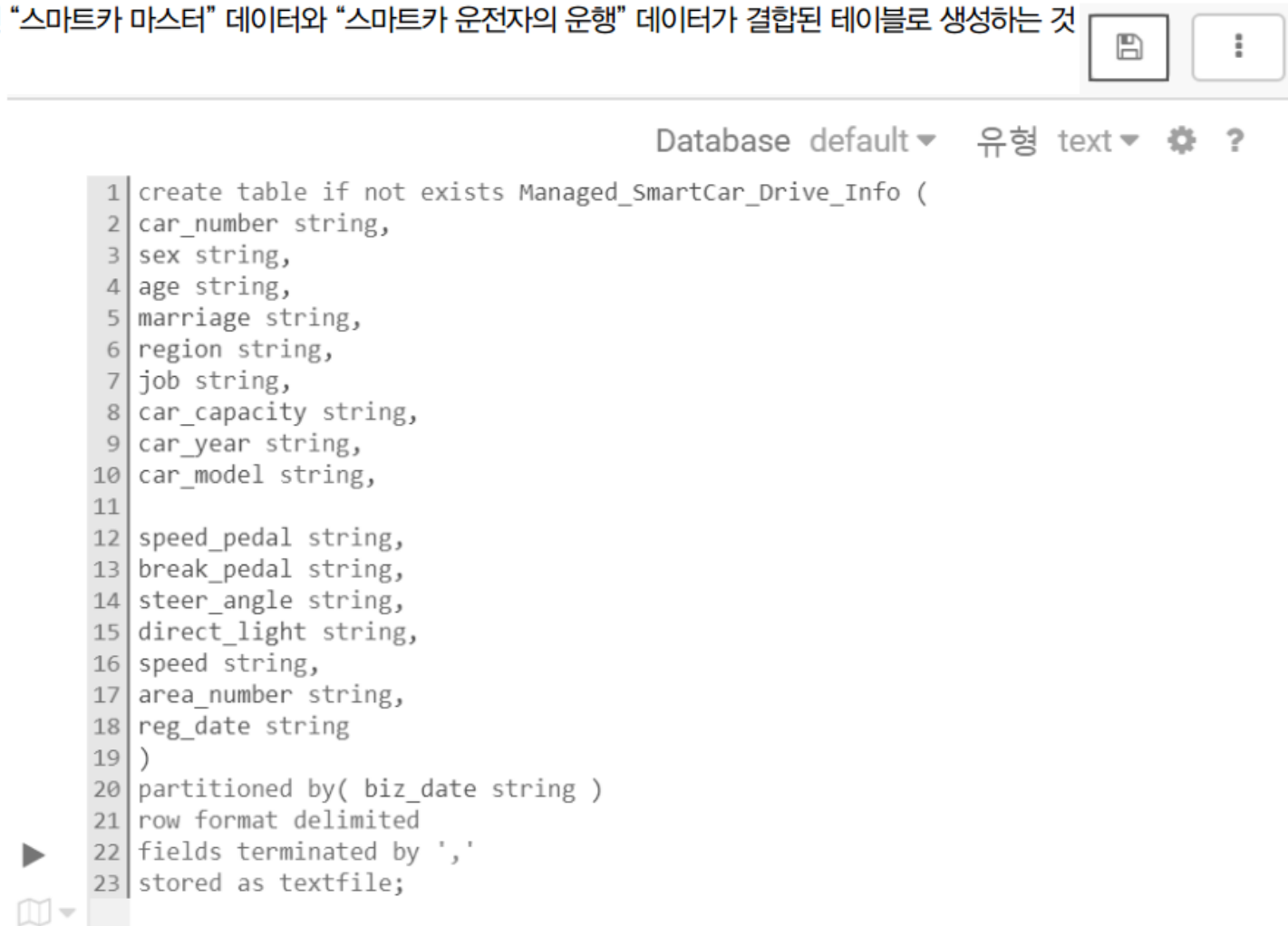



그림 6.99 주제 영역 2의 Managed 테이블 생성 하이브 쿼리

08. 마지막으로 내 문서의 /workflow/hive_script/subject2에 네 번째 하이브 스크립트 파일을 만든다. subject2 디렉터리에서 [새 문서] → [Hive 쿼리]를 클릭한다.

09. 하이브 에디트 창이 활성화되면 “스마트카 운전자 운행” 데이터와 “스마트카 마스터” 데이터를 조인한 후, 삽입하는 하이브 스크립트를 동적 파티션 설정과 함께 작성한다. 파일 이름을 “insert_table_managed_smartcar_drive_info.hql”로 입력해 저장한다.


- set hive.exec.dynamic.partition=true;
- set hive.exec.dynamic.partition.mode=nonstrict;


 Hive



Add a nam...

Add a desc...







Database default ▼ 유형 text ▼ ⚙ ?

```
1 set hive.exec.dynamic.partition=true;
2 set hive.exec.dynamic.partition.mode=nonstrict;
3
4 insert overwrite table Managed SmartCar drive Info partition(biz_date)
5 select
6     t1.car_number,
7     t1.sex,
8     t1.age,
9     t1.marriage,
10    t1.region,
11    t1.job,
12    t1.car_capacity,
13    t1.car_year,
14    t1.car_model,
15    t2.speed_pedal,
16    t2.break_pedal,
17    t2.steer_angle,
18    t2.direct_light ,
19    t2.speed ,
20    t2.area_number ,
21    t2.r_date,
22    substring(t2.r_date, 0, 8) as biz_date
23 from SmartCar_Master_Over18 t1 join SmartCar_Drive_Info_2 t2
24 on t1.car_number = t2.car_number and substring(t2.r_date,0,8) = '${working_day}';
```

그림 6.100 주제 영역 2의 Managed 테이블에 데이터를 생성하는 하이브 쿼리

10. 이제 워크플로를 만든다. 휴 상단 쿼리 콤보박스 메뉴의 [스케줄러] → [Workflow]를 선택해 워크플로를 작성한다.
11. 첫 번째 작업으로 워크플로 작성을 위한 우지 편집기가 나타나면 상단의 작업 툴 박스에서 “Hive 쿼리” 작업을 선택해 워크플로의 첫 번째 작업 노드에 드래그 앤드 드롭한다.
12. 사용할 Hive 스크립트 파일을 선택한다. 앞 단계에서 만든 create_table_smartcar_drive_info_2.hql을 선택한 후 [추가] 버튼을 클릭한다.
13. 두 번째 작업을 위해 워크플로의 작업 툴박스에서 “Hive 쿼리” 작업을 선택해 워크플로의 두 번째 작업 노드에 드래그 앤드 드롭한다.
14. 사용할 Hive 스크립트 파일을 선택한다. 앞 단계에서 만든 insert_table_smartcar_drive_info_2.hql을 선택한 후 [추가] 버튼을 클릭한다.
15. [매개변수+]를 누르고 working_day의 매개변수에 우지의 예약 스케줄러에서 정의할 \${today} 매개변수를 할당한다.
 - working_day=\${today} ※ 즉시 실행시 - 스마트카 시뮬레이션 날짜(교재기준: 20200322)로 설정
ex) working_day=20200322

16. 세 번째 마지막 작업을 위해 워크플로의 작업 툴박스에서 “Hive 쿼리” 작업을 워크플로의 세 번째 작업 노드에 드래그 앤드 드롭한다.
17. 사용할 Hive 스크립트 파일을 선택한다. create_table_managed_smartcar_drive_info.hql을 선택한 후 [추가] 버튼을 클릭한다.
18. 네 번째 작업을 위해 워크플로의 작업 툴박스에서 “Hive 쿼리” 작업을 워크플로의 네 번째 작업 노드에 드래그 앤드 드롭한다.
19. 사용할 Hive 스크립트 파일을 선택한다. 앞 단계에서 만든 insert_table_managed_smartcar_drive_info.hql을 선택한 후 [추가] 버튼을 클릭한다.
20. [매개변수+]를 누르고 working_day의 매개변수에 우지의 예약 스케줄러에서 정의할 \${today} 매개변수를 할당한다.
 - working_day=\${today} ※ 즉시 실행시 - 스마트카 시뮬레이션 날짜(교재기준: 20200322)로 설정
ex) working_day=20200322
21. 워크플로의 이름을 작성한다. 워크플로 상단의 “My Workflow”를 클릭하고 “Subject 2 – Workflow”로 변경한 후 [확인] 버튼을 클릭한다.
22. 워크플로 작성을 완료한다. 우측 상단의 [저장] 버튼을 누른다.
23. 이제 작성한 워크플로를 작동하기 위한 예약 작업을 생성한다. 쿼리 콤보박스 메뉴의 [스케줄러] → [예약]을 선택한다.
24. 먼저 예약 작업 이름을 입력한다. 상단의 [My Schedule]를 클릭하고 “Subject 2 – 예약”으로 입력한다.

25. 예약 작업이 사용할 워크플로를 선택한다. “예정된 Workflow는 무엇입니까?”라는 메시지의 하단에 있는 “Workflow 선택...”을 클릭해 앞서 만든 주제 영역 2의 워크플로인 “Subject 2 – Workflow”를 선택한다.

26. 예약 작업 워크플로를 실행하기 위한 스케줄 값을 입력한다.

- 실행 간격: 매일, 02시
- 시작 일자: 2020년 03월 23일, 00시 00분
- 종료 일자: 2020년 12월 31일, 23시 59분
- 시간대: Asia/Seoul

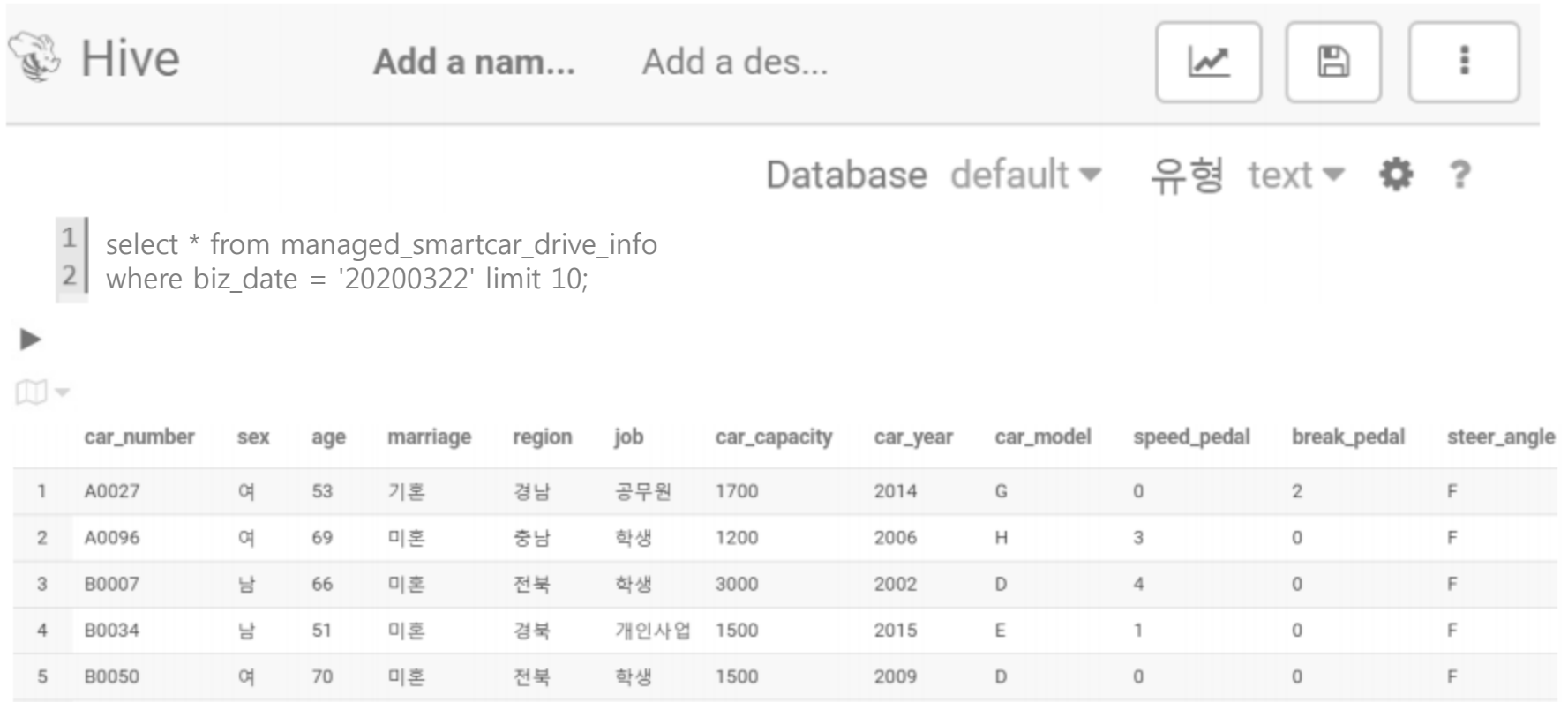
27. 예약 작업에서 사용할 매개변수인 `working_day` 값을 예약 작업의 매개변수로 정의한다.

앞서 워크플로의 하이브 작업에서는 매개변수를 “`working_day=${today}`”로 등록했다. `working_day` 값을 예약 작업의 내장 함수를 통해 설정한다.

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

28. 우지의 예약 작업 설정이 모두 끝났다. [저장] 버튼을 클릭해 작성을 완료한다.
29. 작성이 완료된 예약 작업을 우측 상단의 [제출] 버튼을 클릭해 실행한다. 참고로 이번 주제 영역 2의 워크플로는 처리량이 많은 작업으로 필자의 파일럿 환경에서 약 10여 분 정도 실행됐다.
30. 제출된 예약 작업 상태를 확인해 본다. 좌측 드롭박스 메뉴에서 [Job] → [일정]을 선택한다. 앞서 등록한 “Subject 2 – 예약”이 “Running” 상태로, 매일 새벽 02시가 되면 등록된 워크플로(Subject 2 – Workflow)를 작동시키게 된다. 새벽 2시까지 기다릴 수 없으니 앞서 주제 영역 1에서 설명한 “Workflow 즉시 실행해 보기”를 참고해 곧바로 실행해 본다.

31. “Subject 2 – Workflow”가 정상적으로 작동됐는지 확인한다. 휴의 Hive Editor로 이동해서 그림 6.101과 같이 하 이브 QL을 작성해서 실행한다. “biz_date=20200322” 날짜는 독자들의 파일럿 환경의 실행 날짜와 맞춰야 한다.



The screenshot shows the Hive Editor interface. At the top, there's a header with the Hive logo, a name field "Add a nam...", and a description field "Add a des...". To the right are icons for a chart, a save icon, and a menu icon. Below the header, there's a "Database default" dropdown and a "유형 text" dropdown. The main area contains a SQL query:

```
1 select * from managed_smartcar_drive_info
2 where biz_date = '20200322' limit 10;
```

Below the query is a play button icon. Underneath, there's a table icon and a table of results with 12 columns: car_number, sex, age, marriage, region, job, car_capacity, car_year, car_model, speed_pedal, break_pedal, and steer_angle. The table contains 5 rows of data.

	car_number	sex	age	marriage	region	job	car_capacity	car_year	car_model	speed_pedal	break_pedal	steer_angle
1	A0027	여	53	기혼	경남	공무원	1700	2014	G	0	2	F
2	A0096	여	69	미혼	충남	학생	1200	2006	H	3	0	F
3	B0007	남	66	미혼	전북	학생	3000	2002	D	4	0	F
4	B0034	남	51	미혼	경북	개인사업	1500	2015	E	1	0	F
5	B0050	여	70	미혼	전북	학생	1500	2009	D	0	0	F

그림 6.101 주제 영역 2 워크플로의 실행 결과 확인

주제 영역 2를 정리하자면 앞서 만든 워크플로는 매일 새벽 2시가 되면 HBase에 적재돼 있는 2020년 03월 22일
자의 “스마트카 운전자 운행 데이터”를 모두 하이브의 테이블로 옮기는 작업을 선행하게 된다. 그런 다음 “스마트
카 마스터 데이터”와 조인 작업으로 운전자 기본정보를 추가해서 확장된 “스마트카 운전자 운행기록 정보” 마트 데
이터를 최종적으로 만든다.

저사양 파일럿 환경: HBase 서비스를 정지한다.

- HBase 서비스: CM 홈 → [HBase] → [정지]