CSE545
빅데이터처리 및 실습
(Big Data Processing and Practice)
Practice Part 3-1: Hive

담당교수: 전강욱(컴퓨터공학부)
kw.chon@koreatech.ac.kr

개요

- **실습명**
  - Hive 활용

- **목표**
  - Hive 개발환경 구축
  - Hive 활용 방법 습득

Hive 설치

- **1. Hive 다운로드 및 압축해제**
  - $ wget https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz
  - $ tar xzf apache-hive-3.1.2-bin.tar.gz

- **2. Hive 경로 설정**
  - $ sudo vim /etc/profile
  - 아래 내용을 /etc/profile 에 붙여 넣기
    - export HIVE_HOME=/home/hadoop/Install/apache-hive-3.1.2-bin
    - export PATH:$PATH=$HIVE_HOME/bin
  - $ source /etc/profile

Hive 설치(계속)

- **3. Hive 설정 파일 수정(hive-config.sh)**
  - $ vi $HIVE_HOME/bin/hive-config.sh
  - 아래 내용 추가
    - export HADOOP_HOME=/home/hadoop/Install/hadoop-3.3.1
- **4. Hive 설정 파일 수정(hive-site.xml)**
  - $ cd $HIVE_HOME/conf
  - $ cp hive-default.xml.template hive-site.xml
  - $ vim hive-site.xml

내용 추가

```
<property>
        <name>system:java.io.tmpdir</name>
        <value>/tmp/hive/java</value>
</property>
<property>
        <name>system:user.name</name>
        <value>${user.name}</value>
</property>
```

값 변경

```
<property>
        <name>javax.jdo.option.ConnectionURL</name>
        <value>jdbc:derby:/root/apache-hive3.1.3-
bin/metastore_db;databaseName=metastore_db;create
=true</value>
</property>
```

Hive 설치(계속)

- **5. HDFS에 Hive 관련 디렉토리 생성 및 쓰기 권한 부여**
  - $ hdfs dfs –mkdir /hive
  - $ hdfs dfs –chmod g+w /hive
  - $ hdfs dfs –mkdir –p /user/hive/warehouse
  - $ hdfs dfs –chmod g+w /user/hive/warehouse
- **6. derby Database 시작**
  - $HIVE_HOME/bin/schematool -initSchema –dbType derby
- **7. Hive 실행**
  - $HIVE_HOME/bin/hive

```
hadoop@hadoop-VirtualBox:/usr/local/hive/bin$ ./hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/Install/hadoop-3.3.1/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl
/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 854b0dfd-8339-4919-8950-95b0c0345bd1

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e
. spark, tez) or using Hive 1.X releases.
Hive Session ID = 38be8b9a-52d9-4c52-983e-9145f2f9d09a
hive>
```

Hive 변수와 속성

- **변수 값 지정 및 변경**
  - $ hive --define a=b

```
hive> set a; ↲
a=b
hive> set hivevar:a; ↲
hivevar:a=b
hive> set hivevar:a=c; ↲
hive> set a; ↲
a=c
hive> set hivevar:a; ↲
hivevar:a=c
hive> create table test1 (i int, ${hivevar:a} string); ↲
hive> show tables; ↲
hive> set system:user.name; ↲
system:user.name=root
```

네임스페이스	접근	설명
hivevar	읽기/쓰기	사용자 정의변수(0.8.0 이후)
hiveconf	읽기/쓰기	Hive의 설정속성
system	읽기/쓰기	자바가 정의한 설정 속성
env	읽기	Shell 환경에서 정의한 환경변수

.hiverc 파일 설정

- 사용자가 CLI 환경에서 Hive 실행 시 미리 설정해야 하는 변수들을 설정 가능함
- User의 Home Directory에 생성
  - Home Directory 확인방법
    - $ cd
    - $ pwd
- CLI에 Database 명이 항상 표기되도록 지정하는 예제
  - $ vi ~/.hiverc
  - set hive.cli.print.current.db=true; 추가

```
hive (default)> show databases;
OK
database_name
airlinedb
default
financials
hive_edu
human_resources
Time taken: 0.574 seconds, Fetched: 5 row(s)
hive (default)> use airlinedb;
OK
Time taken: 0.05 seconds
hive (airlinedb)>
```

DB 명 변경 확인

Hive 명령어 요약

- **데이터베이스 생성(CREATE DATABASE)**
  - CREATE (DATABASE|SCHEMA) [IF NOT EXISTS] database_name
    [COMMENT database_comment]
    [LOCATION hdfs_path]
    [WITH DBPROPERTIES (property_name=property_value, ...)];
- **데이터베이스 삭제(DROP DATABASE)**
  - DROP (DATABASE|SCHEMA) [IF EXISTS] database_name [RESTRICT|CASCADE];
- **데이터베이스 변경(ALTER DATABASE)**
  - ALTER (DATABASE|SCHEMA) database_name
    SET DBPROPERTIES (property_name=property_value, ...);
    -- (Note: SCHEMA added in Hive 0.14.0)
  - ALTER (DATABASE|SCHEMA) database_name
    SET OWNER [USER|ROLE] user_or_role;
    -- (Note: Hive 1.2.2 and later; SCHEMA added in Hive 0.14.0)
- **데이터베이스 사용(USE DATABASE)**
  - USE database_name;
  - USE DEFAULT;

Hive 명령어 요약 (계속)

- **hive> SHOW DATABASES;**
  - 이미 존재하는 데이터베이스 확인
- **hive> CREATE DATABASE IF NOT EXISTS financials;**
  - 새로운 데이터베이스 생성
  - 같은 이름의 데이터베이스가 존재하지 않을 경우에만 생성
- **hive> DESCRIBE DATABASE financials;**
  - 생성된 데이터베이스가 저장된 디렉터리의 위치 보여줌
- **hive> ALTER DATABASE financials SET DBPROPERTIES ('edited-by' = 'John DBA')**
  - 키-값 속성을 추가 가능
  - 속성을 삭제하거나 값을 제거할 수는 없음
- **hive> DESCRIBE DATABASE EXTENDED financials;**
  - EXTENDED를 이용하면 데이터베이스 프로퍼티를 볼 수 있음
- **hive> DROP DATABASE IF EXISTS financials;**
  - 데이터베이스를 삭제
  - 테이블이 존재할 경우에는 삭제가 되지 않음
- **hive> DROP DATABASE IF EXISTS financials CASCADE;**
  - 테이블을 포함하는 데이터베이스를 삭제함

Hive 명령어 요약 (계속)

- **Hive의 Background 실행**
  - $ hive –S –f arrivalDelayCnt.hsql > ./result.data &

Script 내용 포함 파일

결과 파일

Background 실행

```
hadoop@hadoop-VirtualBox:~$ hive -f ./arrivalDelayCnt.hsql >> ./result.data
```

명령어 실행

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/Install/hadoop-3.3.1/share/hadoop/common/lib/slf4j-
log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 21bb4bcc-d149-4432-a5c4-918e616e2e86

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-3.1.2.jar!/hiv
e-log4j2.properties Async: true
Hive Session ID = 5c004e08-9477-49f0-8516-2c2e92944862
Query ID = hadoop_20230924144417_8cfb4548-c8b4-47aa-9d11-7b85434429b0
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1695533511537_0002, Tracking URL = http://hadoop-VirtualBox:8088/proxy/applicat
ion_1695533511537_0002/
Kill Command = /home/hadoop/Install/hadoop-3.3.1/bin/mapred job  -kill job_1695533511537_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-09-24 14:44:30,755 Stage-1 map = 0%,  reduce = 0%
2023-09-24 14:44:38,231 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.65 sec
2023-09-24 14:44:44,634 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.93 sec
MapReduce Total cumulative CPU time: 2 seconds 930 msec
Ended Job = job_1695533511537_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.93 sec   HDFS Read: 234074446 HDFS Write: 107
 SUCCESS
```

MR 작업 Log 확인

실습

- **SQL 전문과 캡쳐 화면을 제출**

- **test 데이터베이스를 생성**
  - hive> CREATE DATABASE test;
- **test 데이터베이스가 HDFS 상 디렉토리로 만들어진 것을 확인**
  - hive> !hdfs dfs –ls /user/hive/warehouse/;
- **생성된 데이터베이스를 삭제**
  - hive> DROP DATABASE test;
- **HDFS에 만들어졌던 디렉토리가 삭제된 것을 확인**
  - hive> !hdfs dfs -ls /user/hive/warehouse/;

실습 (계속)

- **SQL 전문과 캡쳐 화면을 제출**

- **Database와 Table을 생성하고 파일을 Load 하시오 (단계별로 캡쳐하여, 제출)**
  - Database명: hive_edu
  - Table명: hive_exec_01
  - Table Data Location: hive_exec_01의 default 위치
  - Column 정보: birth(string), name(string)
  - 데이터: 생년월일, 이름의 CSV 형식으로 각자 작성(20개정도)
  - 데이터구분자: ,
  - 결과확인: SELECT * FROM hive_exec_01;

CSE545
빅데이터처리 및 실습
(Big Data Processing and Practice)
Practice Part 3-2: Hive

담당교수: 전강욱(컴퓨터공학부)
kw.chon@koreatech.ac.kr

실습

- **SQL 전문과 캡쳐 화면을 제출**

- **Dept 테이블 생성**
  - hive> CREATE EXTERNAL TABLE IF NOT EXISTS dept (deptno INT, dname STRING, loc STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION '/dept';

- **Dept 테이블에 Values 입력**
  - hive > INSERT INTO dept VALUES (10, ACCOUNTING, NEW YORK), (20, RESEARCH, DALLAS), (30, SALES, CHICAGO), (40, OPERATIONS, BOSTON);

- **HDFS 경로에 Dept 테이블의 값이 들어간 것 확인(hdfs dfs cat 명령어 활용)**

실습 (계속)

- **SQL 전문과 캡쳐 화면을 제출**

- **Emp 테이블 생성**
  - hive > CREATE EXTERNAL TABLE IF NOT EXISTS emp (empno INT, ename STRING, job STRING, mgr INT, hiredate STRING, sal FLOAT, comm FLOAT, deptno INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION '/emp';
- **Emp 테이블에 아래 Values 입력**

7369	SMITH	CLERK	7902	1980-12-17	800.0	NULL	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.0	300.0	30
7521	WARD	SALESMAN	7698	1981-02-22	1250.0	500.0	30
7566	JONES	MANAGER	7839	1981-04-02	2975.0	NULL	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.0	1400.0	30
7698	BLAKE	MANAGER	7839	1981-05-01	2850.0	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.0	NULL	10
7788	SCOTT	ANALYST	7566	1987-04-19	3000.0	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17	5000.0	NULL	10
7844	TURNER	SALESMAN	7698	1981-09-08	1500.0	0.0	30
7876	ADAMS	CLERK	7788	1987-05-23	1100.0	NULL	20
7900	JAMES	CLERK	7698	1981-12-03	950.0	NULL	30
7902	FORD	ANALYST	7566	1981-12-03	3000.0	NULL	20
7934	MILLER	CLERK	7782	1982-01-23	1300.0	NULL	10

- **HDFS 경로에 Emp 테이블의 값이 들어간 것 확인(hdfs dfs cat 명령어 활용)**

실습 (계속)

- SQL 전문과 캡쳐 화면을 제출

- 부서별 사원의 급여 평균을 분석하는 Query문 작성

Data 설명

	Name	Description
1	Year	1987-2008
2	Month	1-12
3	DayofMonth	1-31
4	DayOfWeek	1-7
5	DepTime	actual departure time (local, hhmm)
6	CRSDepTime	scheduled departure time (local, hhmm)
7	ArrTime	actual arrival time (local, hhmm)
8	CRSArrTime	scheduled arrival time (local, hhmm)
9	UniqueCarrier	unique carrier code
10	FlightNum	flight number
11	TailNum	plane tail number
12	ActualElapsedTime	in minutes
13	CRSElapsedTime	in minutes
14	AirTime	in minutes
15	ArrDelay	arrival delay, in minutes
16	DepDelay	departure delay, in minutes
17	Origin	origin IATA airport code
18	Dest	destination IATA airport code
19	Distance	in miles
20	TaxiIn	in time, in minutes
21	TaxiOut	taxi out time in minutes
22	Cancelled	was the flight cancelled?
23	CancellationCode	reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
24	Diverted	1 = yes, 0 = no
25	CarrierDelay	in minutes
26	WeatherDelay	in minutes
27	NASDelay	in minutes
28	SecurityDelay	in minutes
29	LateAircraftDelay	in minutes

- **Airline 데이터(EL에 업로드 되어 있음)**
- **데이터 출처: https://www.kaggle.com/datasets/wenxingdi/data-expo-2009-airline-on-time-data?select=2008.csv**

실습 (계속)

- **SQL 전문과 캡쳐 화면을 제출**

- **Airline 데이터베이스 및 테이블 생성**
  - 데이터베이스 생성

```
CREATE DATABASE airlinedb LOCATION '/user/hive/airlinedb';
```

  - 테이블 생성

```
CREATE EXTERNAL TABLE IF NOT EXISTS airlinedb.airline (
Year int, Month int, DayofMonth int, DayOfWeek int,
DepTime int, CRSDepTime int, ArrTime int, CRSArrTime int,
UniqueCarrier string, FlightNum int, TailNum string,
ActualElapsedTime int, CRSElapsedTime int, AirTime int,
ArrDelay int, DepDelay int,
Origin string, Dest string, Distance int, TaxiIn int, TaxiOut int,
Cancelled int, CancellationCode string, Diverted string,
CarrierDelay int, WeatherDelay int,
NASDelay int, SecurityDelay int, LateAircraftDelay int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hive/airlinedb/airline';
```

데이터 적재

- **Table에서 Table로 데이터 적재**
  - hive> INSERT OVERWRITE TABLE [*tablename*] [*select_statement*];

- **File Storage에서 Table로 데이터 적재**
  - hive> LOAD DATA [LOCAL] INPATH *'filepath'* INTO TABLE *tablename*;

- **Table에서 File Storage로 데이터 적재**
  - hive> INSERT OVERWRITE [LOCAL] DIRECTORY 'path' [*select_statement*];

실습

- **캡쳐 화면을 제출**

- **EL에 업로드된 데이터를 Airlinedb.Airline에 적재하기 (Load 명령어 활용)**
  - URL: https://el2.koreatech.ac.kr/mod/courseboard/article.php?id=213723&bwid=235242
  - 파일명: 2008.csv
  - "Select * From Airlinedb.Airline Limit 10;"을 이용하여 적재 확인

- **Airlinedb.Airline에서 아래 변수들만 Select 하여 Local 경로에 적재하기 (Insert Overwrite Local Directory 명령어 활용)**
  - Origin string, Dest string, Distance int, TaxiIn int, TaxiOut int, Cancelled int, CancellationCode string, Diverted string, CarrierDelay int, WeatherDelay int, NASDelay int, SecurityDelay int, LateAircraftDelay int

실습 (계속)

- **SQL 전문과 캡쳐 화면을 제출**

- **Airline 데이터를 분석하여 날씨 관계로 지연된 항공편에 대해서 연도 및 월별로 지연 횟수와 지연 시간을 연도, 월별로 출력**
  - Weatherdelay 변수 활용
  - Group By, Order By 활용

CSE545
빅데이터처리 및 실습
(Big Data Processing and Practice)
Practice Part 3-3: Hive

담당교수: 전강욱(컴퓨터공학부)
kw.chon@koreatech.ac.kr

Partition 요약

- **Hive에서의 Partition은 Directory 단위의 관리를 함**
  - Table 생성시 Partition Column을 정할 수 있음
  - 일반 Column과 Partition Column이 겹치면 안됨
  - Partition된 테이블 생성 예제

```
CREATE TABLE sales (
      sales_order_id BIGINT,
      order_amount FLOAT,
      order_date    STRING,
      due_date      STRING,
      customer_id BIGINT )
PARTITIONED BY (country STRING, year INT, month INT, day INT) ;
```

실습

- **Airlinedb.Airline 테이블의 구조 변경하여 Airline_partitioned_by_month 테이블 생성**
  - Month Column을 Partition Column으로 변경

```
CREATE EXTERNAL TABLE IF NOT EXISTS
airlinedb.airline_partitioned_by_month (
Year int, DayofMonth int, DayOfWeek int,
DepTime int, CRSDepTime int, ArrTime int, CRSArrTime int,
UniqueCarrier string, FlightNum int, TailNum string,
ActualElapsedTime int, CRSElapsedTime int, AirTime int,
ArrDelay int, DepDelay int,
Origin string, Dest string, Distance int, TaxiIn int, TaxiOut int,
Cancelled int, CancellationCode string, Diverted string,
CarrierDelay int, WeatherDelay int,
NASDelay int, SecurityDelay int, LateAircraftDelay int)
PARTITIONED BY (Month int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hive/airlinedb/airline_partitioned_by_month';
```

실습 (계속)

- **SQL 전문 및 실행화면 캡쳐하여 제출**
- **Airline_partitioned_by_month 테이블에 데이터 적재(정적 파티션)**
  - Airline에 적재된 데이터 이용하여 적재 예제(1월)

```
INSERT INTO TABLE airline_partitioned_by_month PARTITION(month=1)
SELECT Year, DayofMonth, DayOfWeek, DepTime, CRSDepTime, ArrTime,
CRSArrTime, UniqueCarrier, FlightNum, TailNum, ActualElapsedTime,
CRSElapsedTime, AirTime, ArrDelay, DepDelay, Origin, Dest, Distance,
TaxiIn, TaxiOut, Cancelled, CancellationCode, Diverted, CarrierDelay,
WeatherDelay, NASDelay, SecurityDelay, LateAircraftDelay
FROM airline
WHERE month=1;
```

Static Partition

  - HDFS 경로 확인

```
hadoop@hadoop-VirtualBox:~$ hdfs dfs -ls /user/hive/airlinedb/
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2023-09-24 14:03 /user/hive/airlinedb/airline
drwxr-xr-x   - hadoop supergroup          0 2023-09-24 15:23 /user/hive/airlinedb/airline_partitio
ned_by_month
hadoop@hadoop-VirtualBox:~$ hdfs dfs -ls /user/hive/airlinedb/airline_partitioned_by_month/
Found 1 items
drwxr-xr-x   - hadoop supergroup          0 2023-09-24 15:28 /user/hive/airlinedb/airline_partitio
ned_by_month/month=1
hadoop@hadoop-VirtualBox:~$ hdfs dfs -ls /user/hive/airlinedb/airline_partitioned_by_month/month=1
Found 1 items
-rw-r--r--   1 hadoop supergroup   58154033 2023-09-24 15:28 /user/hive/airlinedb/airline_partitio
ned_by_month/month=1/000000_0
```

실습 (계속)

- **SQL 전문 및 실행화면 캡쳐하여 제출**

- **Airline_partitioned_by_month 테이블에 데이터 적재**
  - Airline에 적재된 데이터 이용하여 2월-12월 Partition도 생성
  - HDFS에서 Partition 확인

실습 (계속)

- **SQL 전문 및 실행화면 캡쳐하여 제출**

- **아래 명령어를 통해 Airline_partitioned_by_month 테이블의 Partition 삭제 후 HDFS 경로 확인 및 조회**

```
ALTER TABLE airlinedb.airline_partitioned_by_month DROP IF
EXISTS PARTITION (month=1) ;
```

실습 (계속)

- SQL 전문 및 실행화면 캡쳐하여 제출

- Airlinedb.airline 테이블의 포맷을 ORC변경하고, hdfs 경로 확인을 통해 파일 크기 차이 확인

CSE545
빅데이터처리 및 실습
(Big Data Processing and Practice)
Practice Part 3-4: Hive

담당교수: 전강욱(컴퓨터공학부)
kw.chon@koreatech.ac.kr

실습 (계속)

- **SQL 전문 및 실행화면 캡쳐하여 제출**
- **EL에 업로드 된 데이터를 Hive에 적재**
  - URL: https://el2.koreatech.ac.kr/mod/courseboard/article.php?id=213723&bid=114994&bwid=235242
  - 파일명: employees.csv.gz, salaries.csv.gz
  - DB명: Company
  - 테이블명: Employees, Salaries

Employees 테이블의 Column 정보
- employee_id INT
- birthday DATE
- first_name STRING
- family_name STRING
- gender CHAR(1)
- work_day DATE

Salary 테이블의 Column 정보
- employee_id INT
- salary INT
- start_date DATE
- end_date DATE

실습 (계속)

- **SQL 전문 및 실행화면 캡쳐하여 제출**

- **Company.Employees 테이블과 Company.Salaries 테이블을 활용하여 아래 질의를 수행**
  - 나이가 어린 10명의 직원의 정보를 찾아라
  - 90년 2월 입사자 10명을 찾아라
  - 성별 평균 연봉을 구하라
    - Join, Group By 이용
  - 연봉 가장 높은 10명의 직원 정보를 추출하여 테이블로 저장하라

감사합니다.

Contact: kw.chon@koreatech.ac.kr