Hive5 数据操作命令

Hive 中的数据库

- 创建数据库

```
hive> CREATE DATABASE financials;

hive> CREATE DATABASE IF OT EXISTS financials;
```

- 查看数据库

```
hive> CREATE DATABASE;
hive> CREATE DATABASE human_resources;
hive> SHOW DATABASES;

hive> SHOW DATABASE LIKE 'h.*';
```

- 修改默认位置

```
hive> CREATE DATABASE financials
    > LOCATION '/my/preferred/directory';
```

- 增加描述信息

```
hive> CREATE DATABASE financials
    > COMMENT 'Holds all financial tables';
```

- 使用数据库

```
hive> USE financials;
```

- 删除数据库

```
# 先删除表然后删除数据库
hive> DROP DATABASE IF EXISTS financials CASCADE;
#删除数据库
hive> DROP DATABASE IF EXISTS financials;
```

- 修改数据库

```
#数据库的其他元数据信息都是不可以更改的，只有DBPROPERTIES设置键-值对属性值
hive> ALTER DATABASE financials SET DBPROPERTIES('edited-by'='Joe Dba');
```

Hive中的数据表

- 创建表

```
CREATE TABLE IF NOT EXISTS mydb.employees(
    name            STRING COMMENT 'Employee name',
    salary          FLOAT COMMENT 'Employee salary',
    subordinates    ARRAY<STRING> COMMENT 'Name of subordinates',
    deductions      MAP<STRING, FLOAT> COMMENT 'Key are deductions names, values
are percentages',
    address         STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
COMMENT 'Home adress')
COMMENT 'Description of the table'
TBLPROPERTIES('creator'='me', 'created_at'='2020-10-07,...)
LOCATION '/user/hive/warehouse/mydb.db/employees';
```

- 拷贝一个已经存在的表格

```
CREATE TABLE IF NOT EXISTS mydb.employees2
LIKE mydb.employees;
```

- 展示该数据库的表

```
hive> USE mydb;
hive> SHOW TABLES;
employees
```

- 查看表的详细表结构

```
hive> DESCRIBE EXTENDED mydb.employees;
name ...
```

- 查看表中字段的信息

```
hive> DESCRIBE mydb.employees.salary;
salary float Employee salary
```

内部表和外部表

内部表：上面所创建的表都是内部表（也叫管理表），Hive会或多或少控制着数据的生命周期，当删除一个管理表的时候，Hive也会删除这个表的数据。
外部表：因为表是外部的，所以Hive并不认为完全拥有这个数据，所以删除该表的时候不会删除掉该数据。有些HIveQL语法结构并不适用于外部表

- 外部表的建立

```
CREATE EXTERNAL TABLE IF NOT EXISTS stoks(
    exchange        STRING,
    symmbol         STRING,
    ymd             STRING,
    price_open      FLOAT,
    price_high      FLOAT,
    price_low       FLOAT,
    price_close     FLOAT,
    volume          INT,
    price_adj_close FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/data/stocks';
```

- 外部表的复制

```
CREATE EXTERNAL TABLE IF NOT EXISTS mydb.employees3
LIKE mydb.employees
LOCATION '/path/to/data';
```

分区

- 构建分区表

```
CREATE TABLE IF NOT EXISTS mydb.employees(
    name            STRING COMMENT 'Employee name',
    salary          FLOAT COMMENT 'Employee salary',
    subordinates    ARRAY<STRING> COMMENT 'Name of subordinates',
    deductions      MAP<STRING, FLOAT> COMMENT 'Key are deductions names, values
are percentages',
    address         STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
COMMENT 'Home adress')
PARTITIONED BY (country STRING, state STRING);
```

没有经过Hive分析过后得表只有一个employees目录与之对应

```
hdfs://master_server/user/hive/warehouse/mydb.db/employees
```

分区后可以反映分区子目录

```
.../employees/country=CA/state=AB
.../employees/country=CA/state=BC
...
.../employees/country=US/state=AL
```

- 查看表中的所有分区

```
hive> SHOW PARTITIONS employees;
...
Country=CA/state=AB
Country=CA/state=BC
```

- 查看分区过滤

```
hive> SHOW PARTITION employees PARTITION(conutry='US')
hive> SHOW PARTITION employees PARTITION(conutry='US',state='AK')
```

- descibe查看分区

```
hive> DESCRIBE EXTENDED employees;
name    string,
...
adress  strcut<..>
country string,
state   string

...
```

- 外部分区表

```
管理大型生产数据集最常用的情况就是建立外部分区表，这个结合给用户提供了一个可以和其他工具共享数据的方式，也可以优化查询性能。
用户可以自己定义目录结构，对目录结构的使用具有更多的灵活性。
CREATE EXTERNAL TABLE IF NOT EXISTS log_message(
    hms         INT,
    severity    STRING,
    server      STRING,
    process_id  INT,
    message     STRING)
PARTITIONED BY (year INT,month INT, day INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
这样的日志数据按照天进行划分，划分数据尺寸合适，按照这个粒度进行查询速度也很快。
```

```
好处：
- 可以将新数据写入到一个专用的目录中，并与位于其他目录中的数据存在明显的区别。
- 新数据被篡改的风险都降低了，因为新数据的数据子集位于不同目录下
```

- 查看外部表的分区

```
hive> SHOW PARTITIONS log_messages;
...
year = 2011/month=12/day=31
...
```

- 同样的，显示表的详细信息

```
hive> DESCRIBE EXTENDED log_messages;
...
message      string,
year        int,
month       int,
day         int,

Detail Table Information...
...
```

想要知道分区的实际路径的话可以输入以下语句

```
hive> DESCRIBE EXTENDED log_message PARTITION(year=2012, month=1, day=2);
...
location:s3n://ourbucket/logs/2011/01/02,
...
```

删除表

```
DROP TABLE IF EXISTS employees;
```

修改表

> 大多数的表属性可以通过ALTER TABLE 语句来修改，这种操作会修改元数据，但不会修改数据本身。这些语句可用于修改表模式中出现的错误，改变分区路径

- 表重命名

```
ALTER TABLE log_messages RENAME TO logmsgs;
```

- 增加分区

```
ALTER TABLE log_messages ADD IF NOT EXISTS
PARTITION (year = 2020, month = 1, day = 1) LOCATION '/logs/2020/01/01'
...
```

- 移动位置来修改某个分区的路径

```
ALTER TABLE log_message PARTITION (year = 2020, month = 1, day = 1) SET LOCATION
's3n://ourbucket/logs/2020/01/02';
```

- 删除分区

```
ALTER TABLE log_messages DROP IF EXISTS PARTITION (year = 2020, month = 12, day
= 2);
```

对于管理表，使用ALTER TABLE ... ADD PARTITION 语句增加的分区，分区内的数据也是会同时和元数据信息一起被删除的，对于外部表，分区内的数据不会被删除。

- 修改列信息

```
ALTER TABLE log_messages
CHANGE COLUMN hmns hours_minutes_seconds INT
COMMENT 'The hours, minutes, and seconds part of the timestamp'
AFTER severity;
```

- 增加列

```
加到 已有的字段后面
ALTER TABLE log_messages ADD COLUMNS(
    app_name STRING COMMENT 'Application name',
    session_id LONG COMMENT 'The current session id');
```

- 删除或者替换列

```
ALTER TABLE log_messages REPLACE COLUMNS(
    hours_mins_secs INT COMMENT 'hour, minute, seconds from timestamp',
    secerity    STRING COMMENT 'The message severity'
    message     STRING COMMENT 'The rest of the message');
```

- 修改表的属性

```
ALTER TABLE log_messages SET TBLPROPERTIES(
    'notes' = 'The process id is no longer captured; this column is always
NULL');
```

- 修改存储格式

```
ALTER TABLE log_messages
PARTITION (year = 2012, month = 1, day = 1)
SET FILEFORMAT SEQUENCEFILE;
```

- 修改SerDe属性

```
ALTER TABLE table_using_JSON_storage
SET SERDE 'com.example.JSONSerDe'
WITH SERDEPROPERTIES(
    'prop1' = 'value1',
    'prop2' = 'vlaue2');
```

- 对SerDe增加新的属性

```
ALTER TABLE table_using_JSON_storage
SET SERDEPROPERTIES(
    'prop3' = 'value3',
    'prop4' = 'value4'
);
```