项目四 Hive数据定义



《Hive数据仓库应用教程》

目 录 / CONTENTS

1 任务4-1 HiveQL操作数据库

2 任务4-2 HiveQL操作数据表

3 任务4-3 HiveQL操作分区表

4 任务4-4 HiveQL操作分桶表

5 任务4-5 HiveQL操作视图

6 任务4-6 HiveQL相关操作

任务 4-2 HiveQL操作数据表

4.2.1 任务描述

数据库的基本操作完成之后，需要在数据库中完成数据表的基本操作，比如创建数据表、查看数据表、修改数据表和删除数据表。

接下来，将详细讲解如何在Hive的数据库中完成数据表的相关操作。

1. CREATE TABLE句式分析

1．CREATE TABLE句式分析

（1）CREATE TABLE句式语法

```
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
[(col_name data_type [COMMENT col_comment], ... ]
[COMMENT table_comment]
[PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
[CLUSTERED BY (col_name, col_name, ...) [SORTED BY (col_name [ASC|DESC], ...)] INTO num_buckets BUCKETS]
[ROW FORMAT DELIMITED|SERDE serde_name WITH SERDEPROPERTIES (property_name=property_value,...)]
[STORED AS file_format]
[LOCATION hdfs_path]
[TBLPROPERTIES (property_name=property_value, ...)];
```

1. CREATE TABLE句式分析

1．CREATE TABLE句式分析

（1）CREATE TABLE句式语法

注意事项

- 蓝色字体是建表语法的关键字，用于指定某些功能。
- [ ]中括号的语法表示可选。
- |表示使用的时候，左右语法二选一。
- 建表语句中的语法顺序要和语法树中顺序保持一致。

```
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
[(col_name data_type [COMMENT col_comment], ... ]
[COMMENT table_comment]
[PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
[CLUSTERED BY (col_name, col_name, ...) [SORTED BY (col_name [ASC|DESC], ...)] INTO num_buckets BUCKETS]
[ROW FORMAT DELIMITED|SERDE serde_name WITH SERDEPROPERTIES (property_name=property_value,...)]
[STORED AS file_format]
[LOCATION hdfs_path]
[TBLPROPERTIES (property_name=property_value, ...)];
```

4.2.2 任务准备知识

（2）Serde是什么

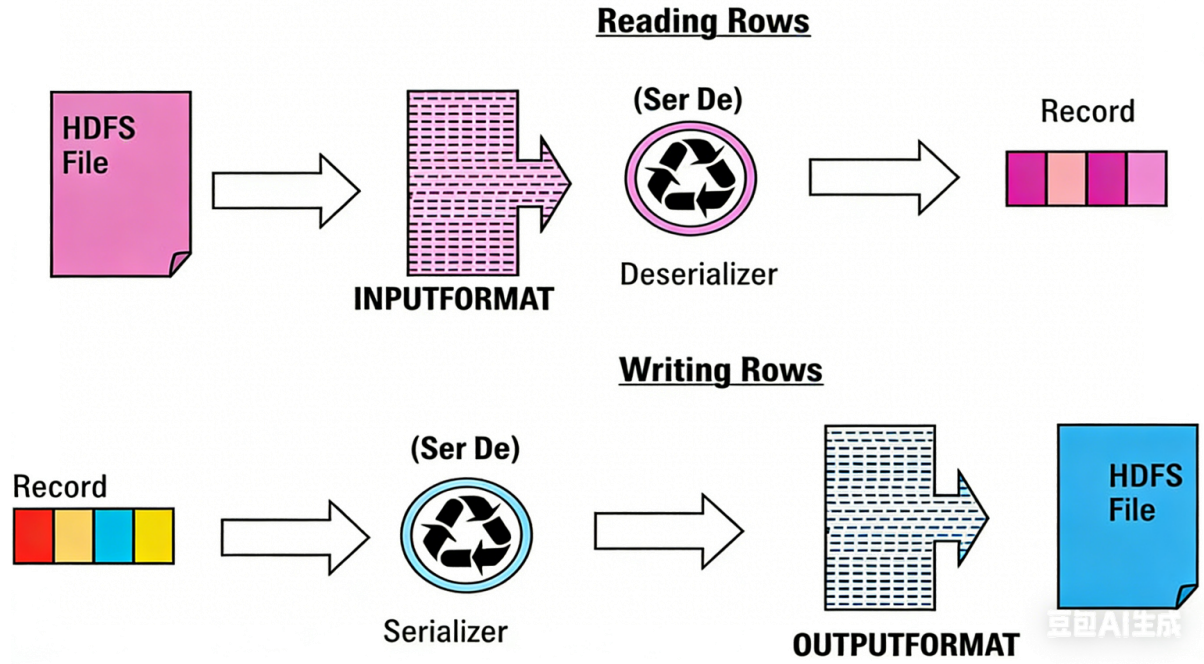
- SerDe是Serializer、Deserializer的简称，目的是用于序列化和反序列化。
- 序列化是对象转化为字节码的过程；而反序列化是字节码转换为对象的过程。
- Hive使用SerDe（包括FileFormat）读取和写入表行对象。需要注意的是，“key”部分在读取时会被忽略，而在写入时key始终是常数。基本上行对象存储在“value”中。

Read: 反序列化

HDFS files --> InputFileFormat --> <key, value> --> Deserializer --> Row object

Write: 序列化

Row object --> Serializer --> <key, value> --> OutputFileFormat --> HDFS files



2．数据表简介

数据表是Hive存储数据的基本单位，Hive数据表主要分为内部表（又叫托管表:Managed Table）和外部表，以内部表和外部表为基础可以创建分区表或分桶表，即内/外部分区表或内/外部分桶表。

（1）内部表(Internal Table)

（2）外部表(External Table)

Hive 内、外部表

完整语法树

```
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
[(col_name data_type [COMMENT col_comment], ... ]
[COMMENT table_comment]
[PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]
[CLUSTERED BY (col_name, col_name, ...) [SORTED BY (col_name [ASC|DESC], ...)] INTO num_buckets BUCKETS]
[ROW FORMAT DELIMITED|SERDE serde_name WITH SERDEPROPERTIES (property_name=property_value,...)]
[STORED AS file_format]
[LOCATION hdfs_path]
[TBLPROPERTIES (property_name=property_value, ...)];
```

Hive 内、外部表

什么是内部表

- **内部表**（Internal table）也称为被Hive拥有和管理的托管表（Managed table）。
- 默认情况下创建的表就是内部表，Hive拥有该表的结构和文件。换句话说，Hive完全管理表（元数据和数据）的生命周期，类似于RDBMS中的表。
- 当您删除内部表时，它会删除数据以及表的元数据。

```
-- Hive 内、外部表
--默认情况下 创建的表就是内部表
create table student(
num int,
name string,
sex string,
age int,
dept string)
row format delimited
fields terminated by ',';
```

```
0: jdbc:hive2://localhost:10000> desc student;
+-----------+------------+----------+
| col_name  | data_type  | comment  |
+-----------+------------+----------+
| num       | int        |          |
| name      | string     |          |
| sex       | string     |          |
| age       | int        |          |
| dept      | string     |          |
+-----------+------------+----------+
```

Hive 内、外部表

什么是内部表

- 可以使用

desc formatted tablename

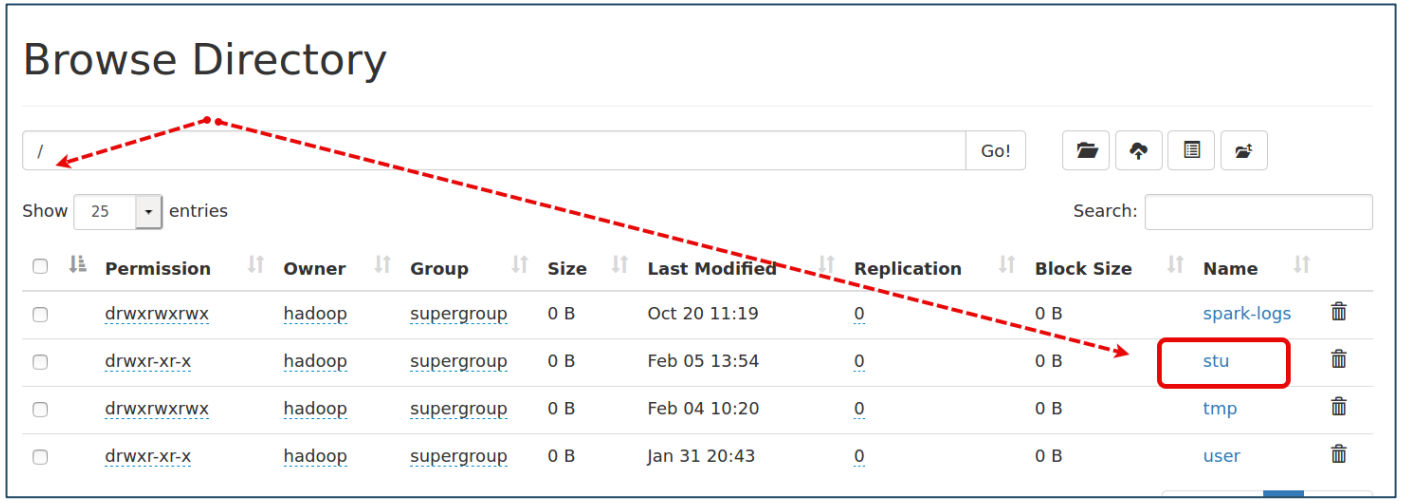
```
| NULL
# Detailed Table Information   | NULL
Database:                      | itcast
OwnerType:                     | USER
Owner:                         | hadoop
CreateTime:                    | Thu Feb 05 13:40:31 KST 2026
LastAccessTime:                | UNKNOWN
Retention:                     | 0
Location:                      | hdfs://hadoop01:9000/user/hive/warehouse/itcast.db/
Table Type:                    | MANAGED TABLE
Table Parameters:              | NULL
                               | COLUMN_STATS_ACCURATE
```

Hive 内、外部表

什么是外部表

- 外部表（External table）中的数据不是Hive拥有或管理的，只管理表元数据的生命周期。
- 要创建一个外部表，需要使用EXTERNAL语法关键字。
- 删除外部表只会删除元数据，而不会删除实际数据。在Hive外部仍然可以访问实际数据。
- 实际场景中，外部表搭配location语法指定数据的路径，可以让数据更安全。

```
--创建外部表 需要关键字 external
--外部表数据存储路径不指定 默认规则和内部表一致
--也可以使用location关键字指定HDFS任意路径
create external table student_ext(
num int,
name string,
sex string,
age int,
dept string)
row format delimited
fields terminated by ','
location '/stu';
```



Hive 内、外部表

内、外部表差异

- 无论内部表还是外部表，Hive都在Hive Metastore中管理表定义、字段类型等元数据信息。
- 删除内部表时，除了会从Metastore中删除表元数据，还会从HDFS中删除其所有数据文件。
- 删除外部表时，只会从Metastore中删除表的元数据，并保持HDFS位置中的实际数据不变。

	内部表、托管表	外部表
创建方式	默认情况下	使用EXTERNAL语法关键字
Hive管理范围	元数据、表数据	元数据
删除表结果	删除元数据，删除HDFS上文件数据	只会删除元数据
操作	支持ARCHIVE，UNARCHIVE，TRUNCATE，MERGE，CONCATENATE	不支持
事务	支持ACID /事务性	不支持
缓存	支持结果缓存	不支持

Hive 内、外部表

如何选择内、外部

- 当需要通过Hive完全管理控制表的整个生命周期时，请使用内部表。
- 当数据来之不易，防止误删，请使用外部表，因为即使删除表，文件也会被保留。

Hive 内、外部表

Location再探究



- 在创建外部表的时候，可以使用location指定存储位置路径，如果不指定会如何？
- 创建内部表的时候，是否可以使用location指定？
- 是否意味着Hive表的数据在HDFS上的位置不是一定要在/user/hive/warehouse下？

Hive 内、外部表

Location关键字的作用

- 在创建外部表的时候，可以使用location指定存储位置路径，如果不指定会如何？
  - 如果不指定location, 外部表的默认路径也是位于/user/hive/warehouse, 由默认参数控制。
- 创建内部表的时候，是否可以使用location指定？
  - 内部表可以使用location指定位置的。
- 是否意味着Hive表的数据在HDFS上的位置不是一定要在/user/hive/warehouse下？
  - 不一定， Hive中表数据存储位置，不管内部表还是外部表，默认都是在/user/hive/warehouse, 当然可以在建表的时候通过location关键字指定存储位置的任意路径。

3．查看数据表

显示当前数据库所有数据表，语法格式如下：

```
SHOW TABLES [LIKE 'identifier_with_wildcards'];
```

查看指定数据表的结构信息，语法格式如下：

```
DESCRIBE|DESC [formatted] table_name;
```

- SHOW TABLES：表示查看当前数据库的所有数据表；
- LIKE 'identifier_with_wildcards'：可选，LIKE子句用于模糊查询；
- DESCRIBE|DESC：表示查询指定数据表的表结构基本信息；
- formatted：可选，表示查询指定数据表的表结构详细信息。

4．修改数据表

修改数据表结构信息使用ALTER TABLE语句实现，该语句只是修改数据表的元数据信息，数据表中的数据不会随之发生变化。下面介绍几种修改数据表结构信息的方法。

（1）重命名数据表是指修改数据表的名称，语法格式如下：

```
ALTER TABLE table_name RENAME TO new_table_name;
```

（2）修改数据表属性的语法格式：

```
ALTER TABLE table_name SET TBLPROPERTIES(property_name = property_value, property_name = property_value, ... );
```

（3）修改数据表列是指修改数据表中列的名称、描述、数据类型或者列的位置，语法如下：

```
ALTER TABLE table_name CHANGE [COLUMN] col_old_name col_new_name column_type [COMMENT col_comment] [FIRST|AFTER column_name];
```

- ALTER TABLE：表示修改数据表结构信息的语句；
- CHANGE [COLUMN]：用于修改数据表列的操作；
- table_name：指定需要修改列的数据表名称；
- col_old_name：指定需要修改的列名；
- col_new_name：指定列名被修改后的新列名；
- column_type：指定列修改后的字段类型；

（4）添加数据表列会在数据表的尾部添加指定列，语法格式如下：

ALTER TABLE table_name ADD COLUMNS (col_name data_type [COMMENT col_comment], ...)

（5）替换数据表列是指替换当前数据表中的所有列，语法格式如下：

ALTER TABLE table_name REPLACE COLUMNS (col_name data_type [COMMENT col_comment], ...)

（6）Hive中强制数据类型转换规则

在Hive中执行修改列的数据类型操作时只能按照强制数据类型转换规则进行修改。

5．删除数据表

（1）删除数据表的语法格式如下：

```
DROP TABLE [IF EXISTS] table_name [PURGE];
```

（2）回收站

1．创建数据表

(1)首先，使用在Hive数据库h_sample中创建内部表managed_table：

```
CREATE  TABLE IF NOT EXISTS
h_sample.managed_table(
staff_id INT COMMENT "This is staffid",
staff_name STRING COMMENT "This is staffname",
salary FLOAT COMMENT "This is staff salary",
hobby ARRAY<STRING> COMMENT "This is staff hobby",
deductions MAP<STRING, FLOAT> COMMENT "This is staff deduction",
address STRUCT<street:STRING,city:STRING> COMMENT "This is staff address")
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '_'
MAP KEYS TERMINATED BY ':'
LINES TERMINATED BY '\n'
STORED AS textfile
TBLPROPERTIES("comment"="This is a managed table");
```

4.2.3 任务实施

(2) 在Hive数据库h_sample中创建外部表external_table:

```
CREATE EXTERNAL TABLE IF NOT EXISTS
h_sample.external_table(
staff_id INT COMMENT "This is staffid",
staff_name STRING COMMENT "This is staffname",
salary FLOAT COMMENT "This is staff salary",
hobby ARRAY<STRING> COMMENT "This is staff hobby",
deductions MAP<STRING, FLOAT> COMMENT "This is staff deduction",
address STRUCT<street:STRING,city:STRING> COMMENT "This is staff address")
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '_'
MAP KEYS TERMINATED BY ':'
LINES TERMINATED BY '\n'
STORED AS textfile
LOCATION '/user/hive_external/external_table/'
TBLPROPERTIES("comment"="This is a external table");
```

2．查看数据表

在虚拟机中使用Hive客户端工具Hhive-bee， 讲解查看数据表实际应用。

（1）切换到数据库h_sample，语句如下：

```
USE h_sample;
```

（2） 查看数据库h_sample的所有数据表，数据库h_sample中包含两个表，语句如下：

```
SHOW TABLES;
```

执行结果如图4-5所示。



4.2.3 任务实施

（3）通过模糊查询查看数据库h_sample中数据表名称首字母为“m”的数据表，语句如下：

```
SHOW TABLES LIKE 'm*';
```

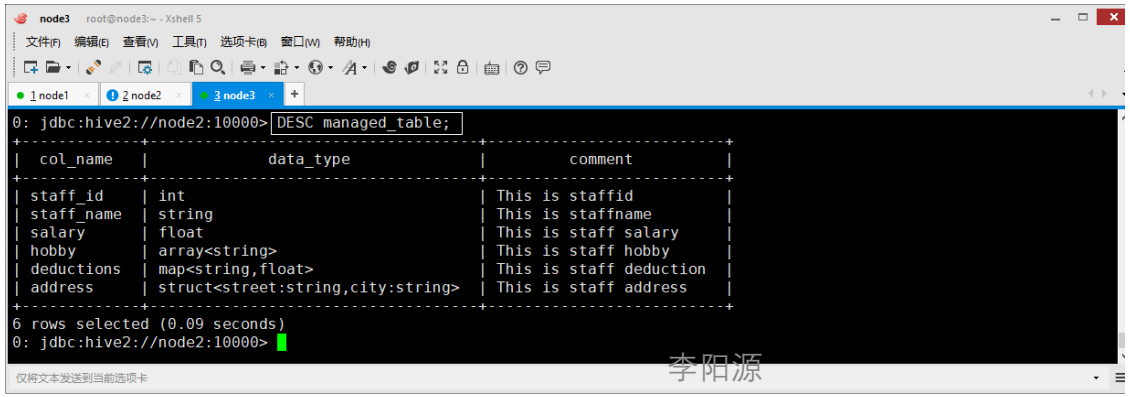
执行结果如图4-6所示：



（4）查看数据库h_sample中内部表managed_table表结构的基本信息，语句如下：

```
DESC managed_table;
```

执行结果如图4-7所示：

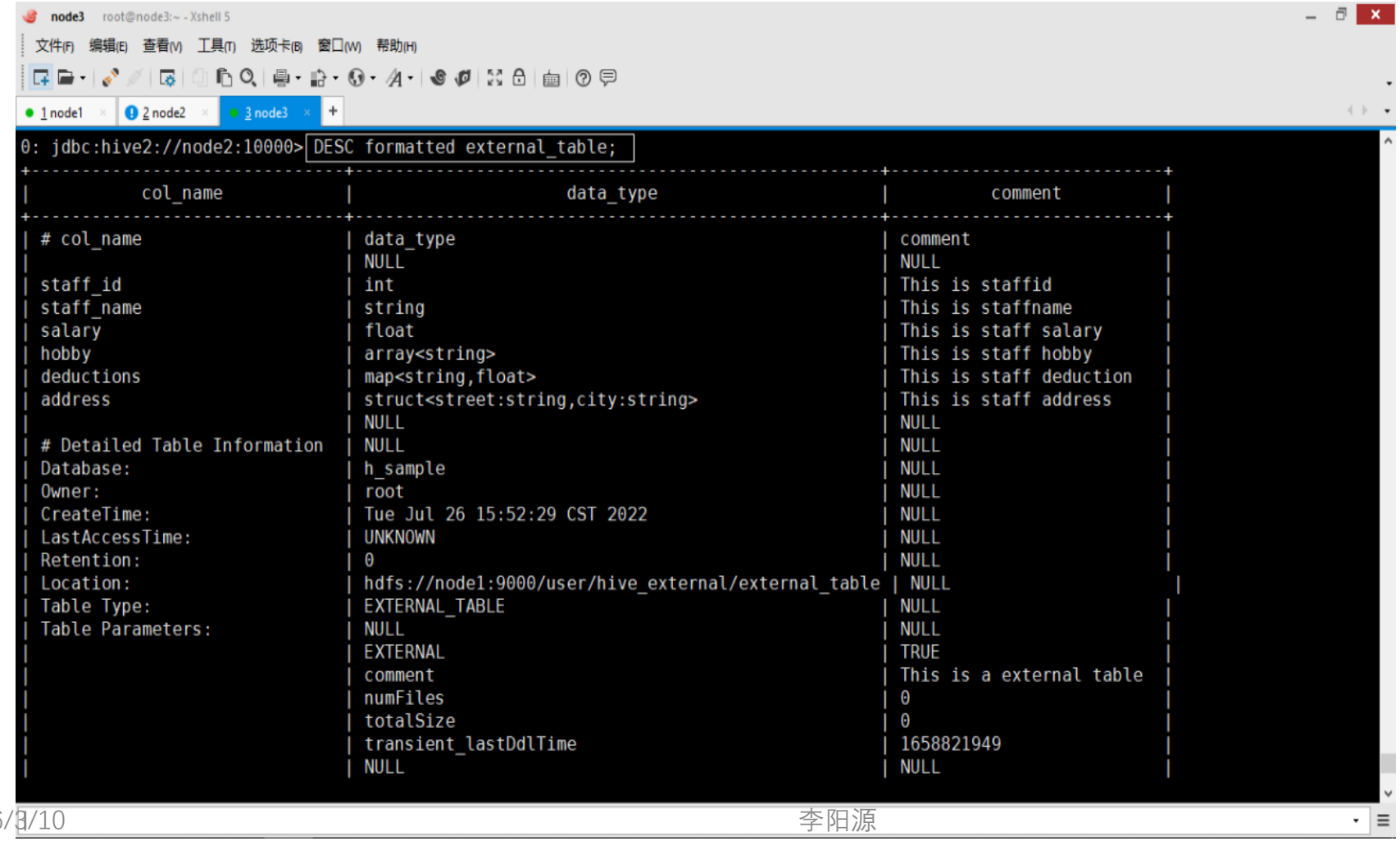


4.2.3 任务实施

（5）查看数据h_sample中外部表external_table表结构的详细信息，语句如下：

DESC formatted external_table;

执行结果如图4-8所示：



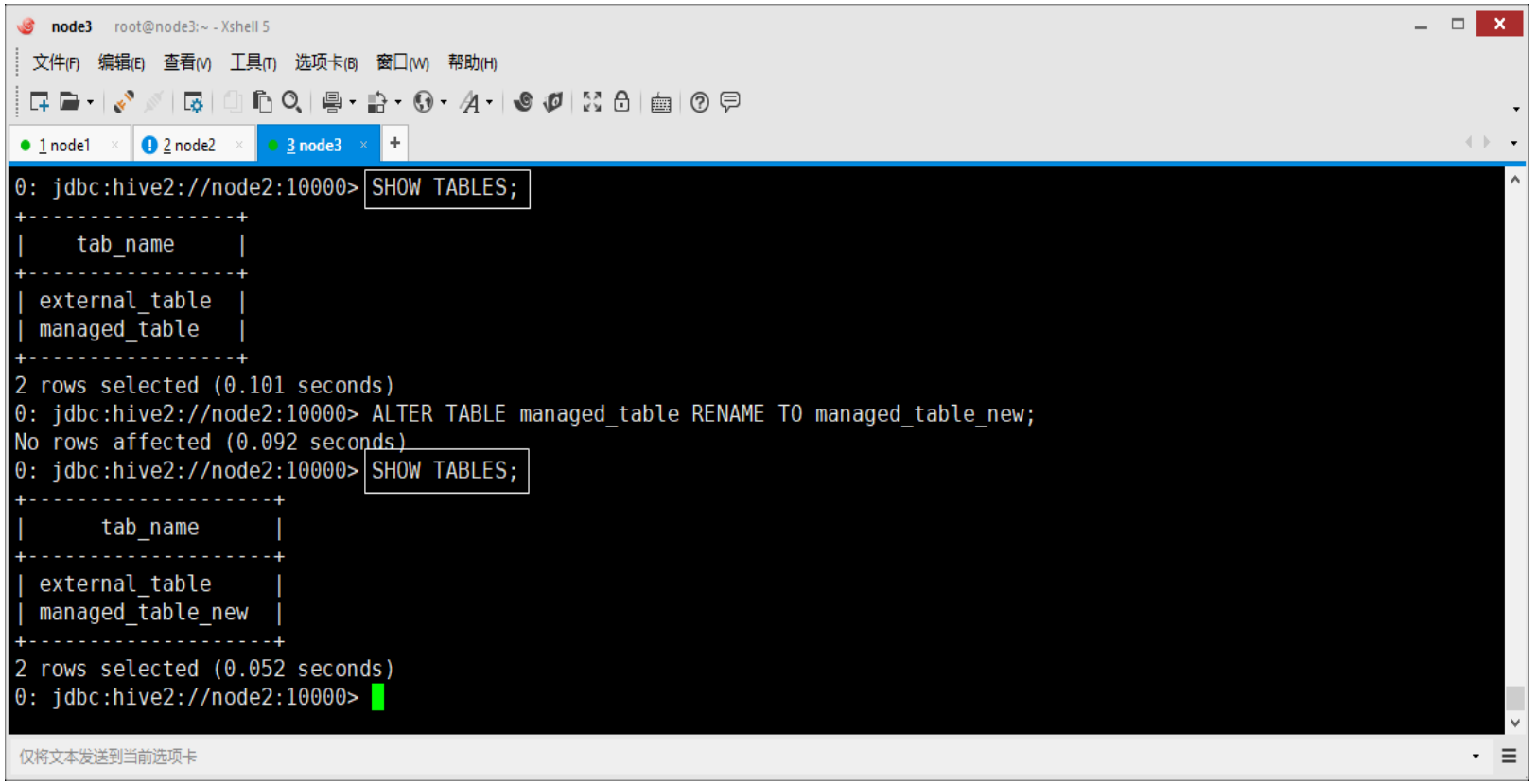
4.2.3 任务实施

3．修改数据表

（1）将数据库h_sample的内部表managed_table重命名为managed_table_new，语句如下：

```
ALTER TABLE managed_table RENAME TO managed_table_new;
```

执行结果如图4-8所示：

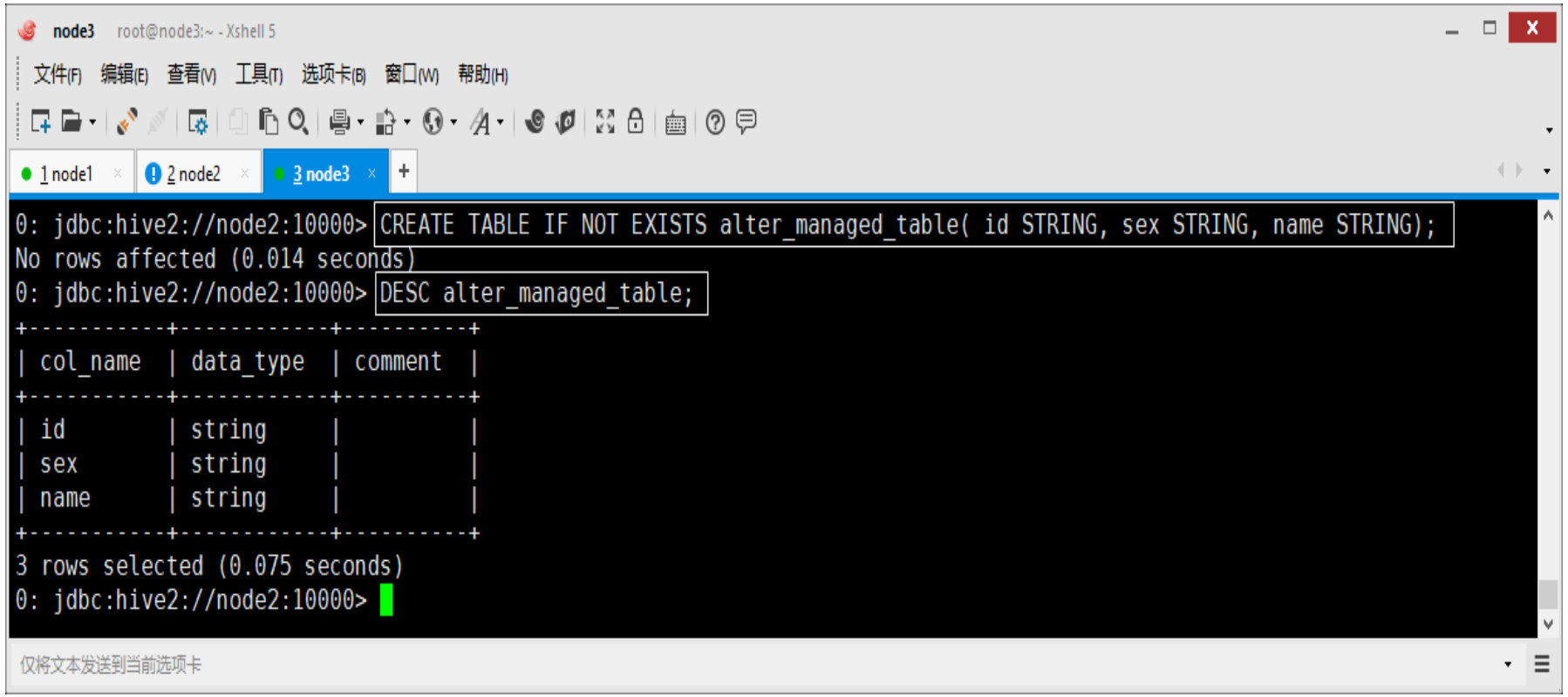


4.2.3 任务实施

（2）在数据库h_sample中创建数据表alter_managed_table，该表中所有列的数据类型一致，语句如下：

CREATE TABLE IF NOT EXISTS alter_managed_table(id STRING, sex STRING, name STRING);

执行结果如图4-10所示。

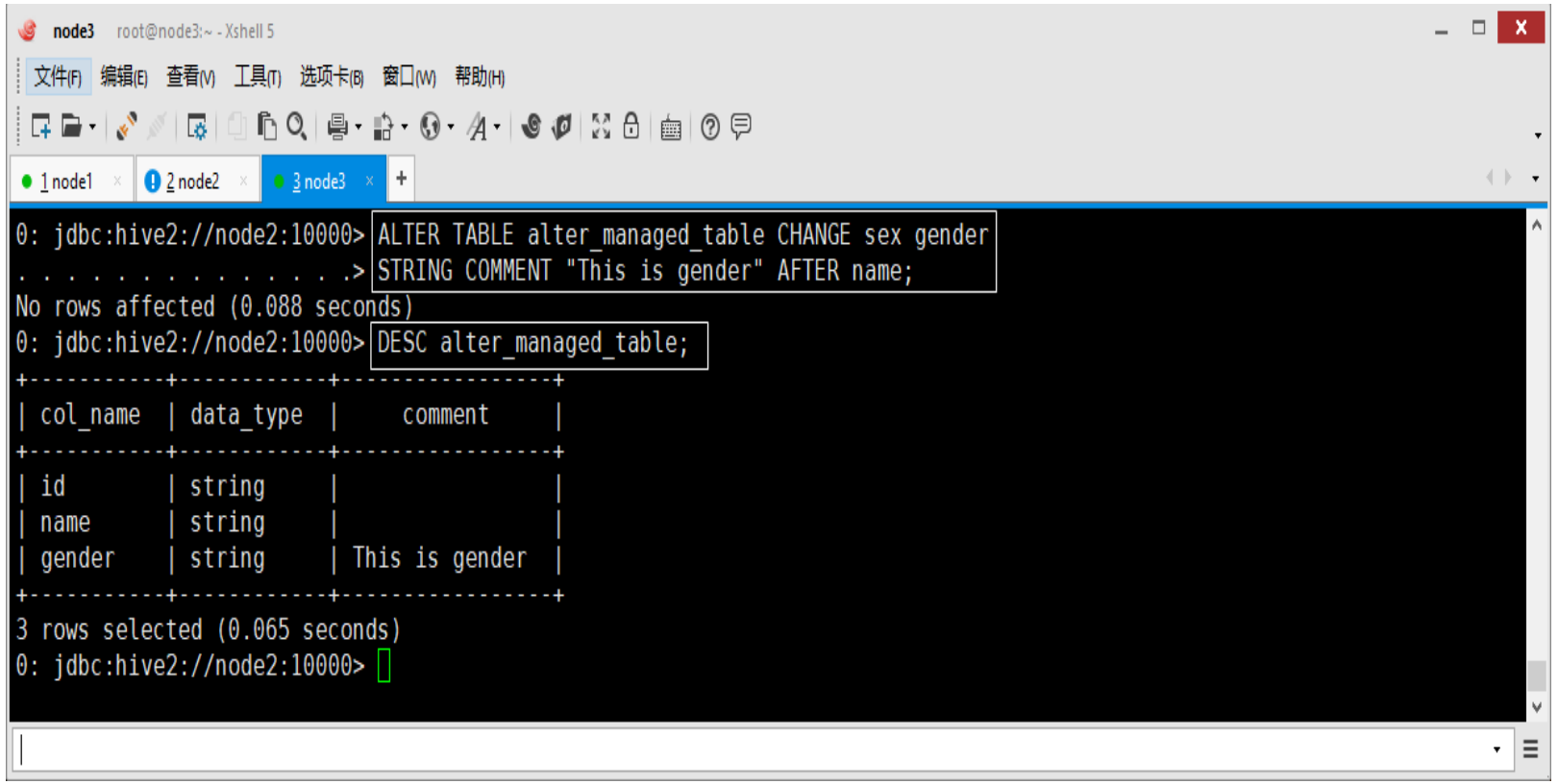


4.2.3 任务实施

（3）修改数据库h_sample中数据表alter_managed_table的列sex，首先重命名列sex为gender，然后移动列gender的位置到列name的后边，最后修改列gender的描述为“This is gender”，语句如下：

```
ALTER TABLE alter_managed_table CHANGE sex gender
STRING COMMENT "This is gender" AFTER name;
```

执行结果如图4-11所示。

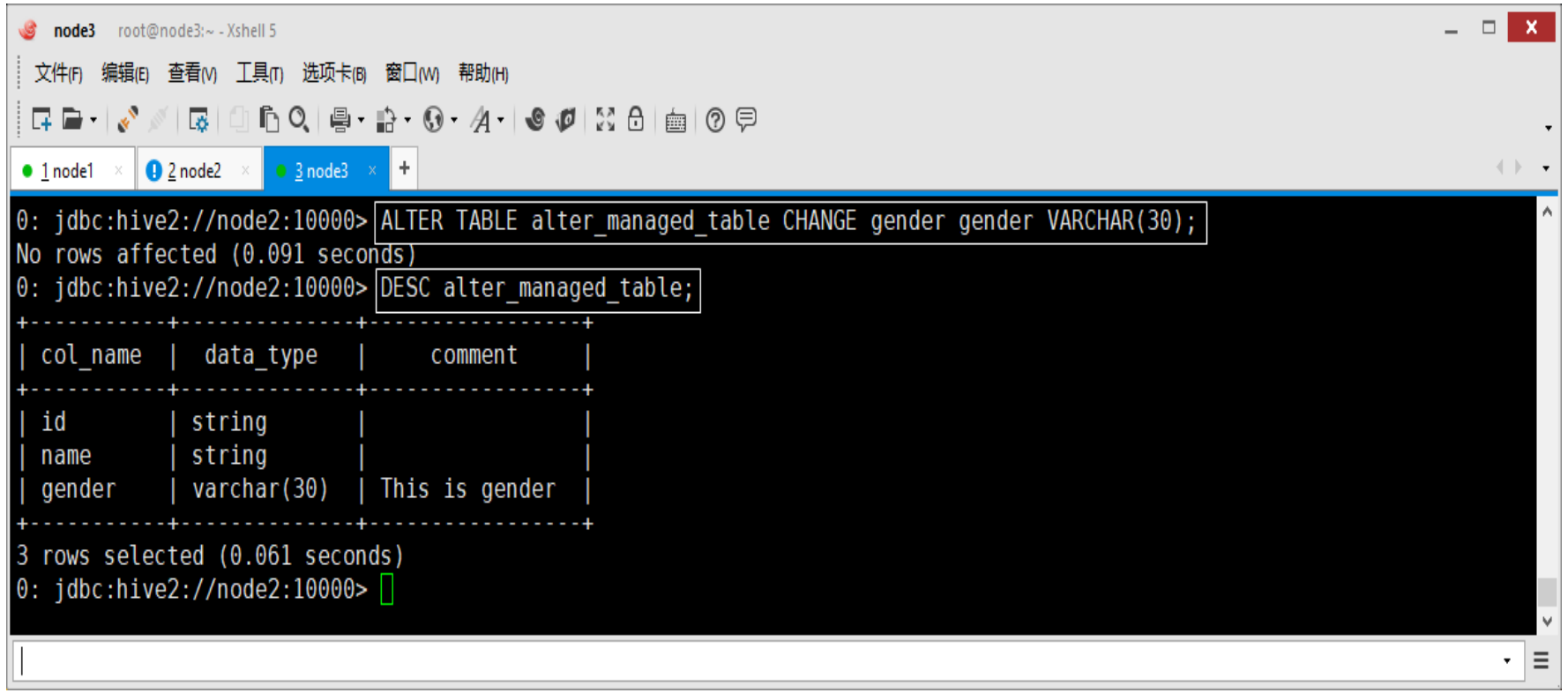


4.2.3 任务实施

（4）修改数据库h_sample中内部表alter_managed_table的列gender，将列的数据类型由STRING 修改为VARCHAR(30)，语句如下：

ALTER TABLE alter_managed_table CHANGE gender gender VARCHAR(30);

执行结果如图4-12所示。

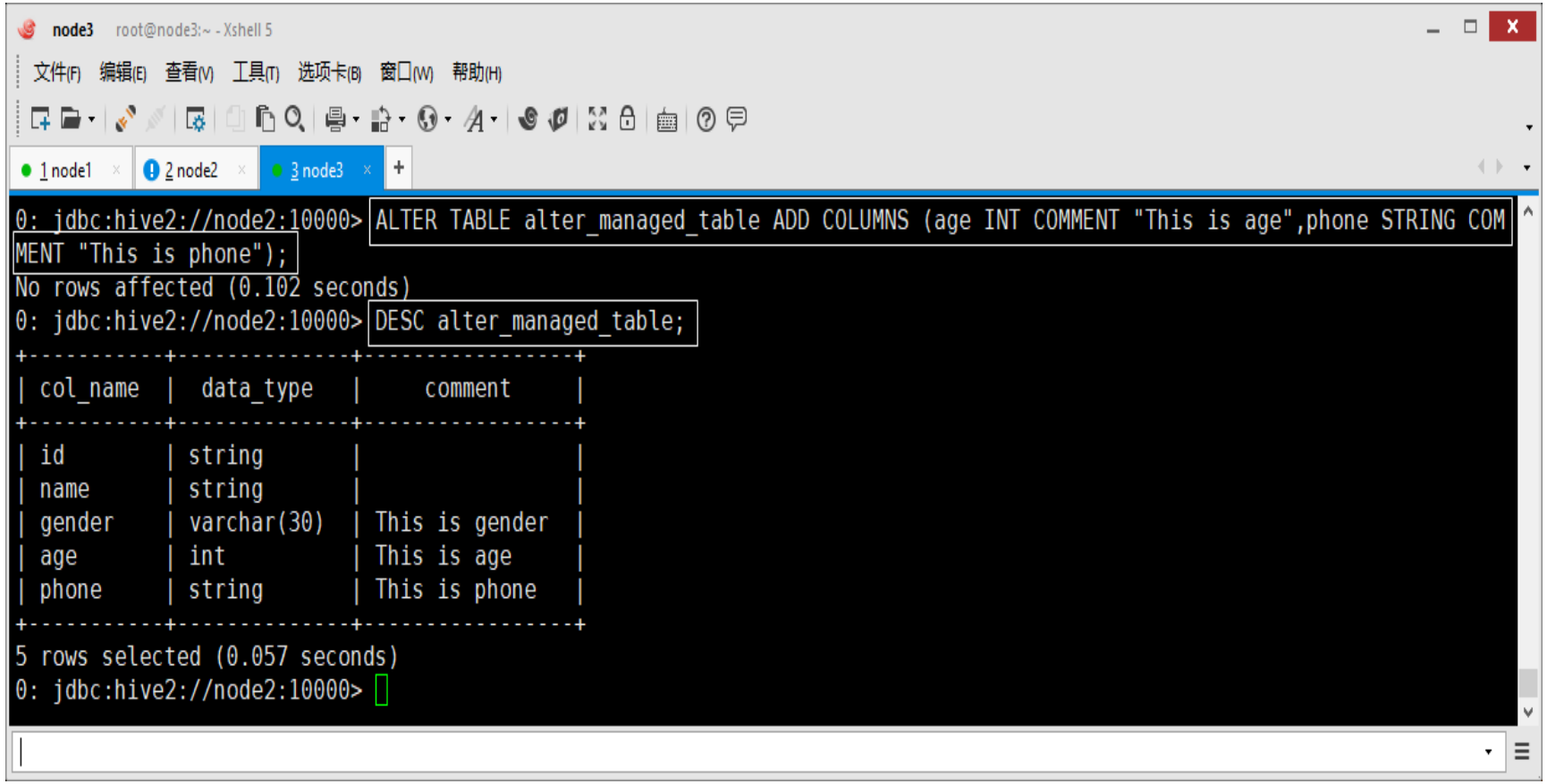


4.2.3 任务实施

（5）在数据表alter_managed_table中添加列，语句如下：

ALTER TABLE alter_managed_table ADD COLUMNS (age INT COMMENT "This is age",phone STRING COMMENT "This is phone");

执行结果如图4-13所示：

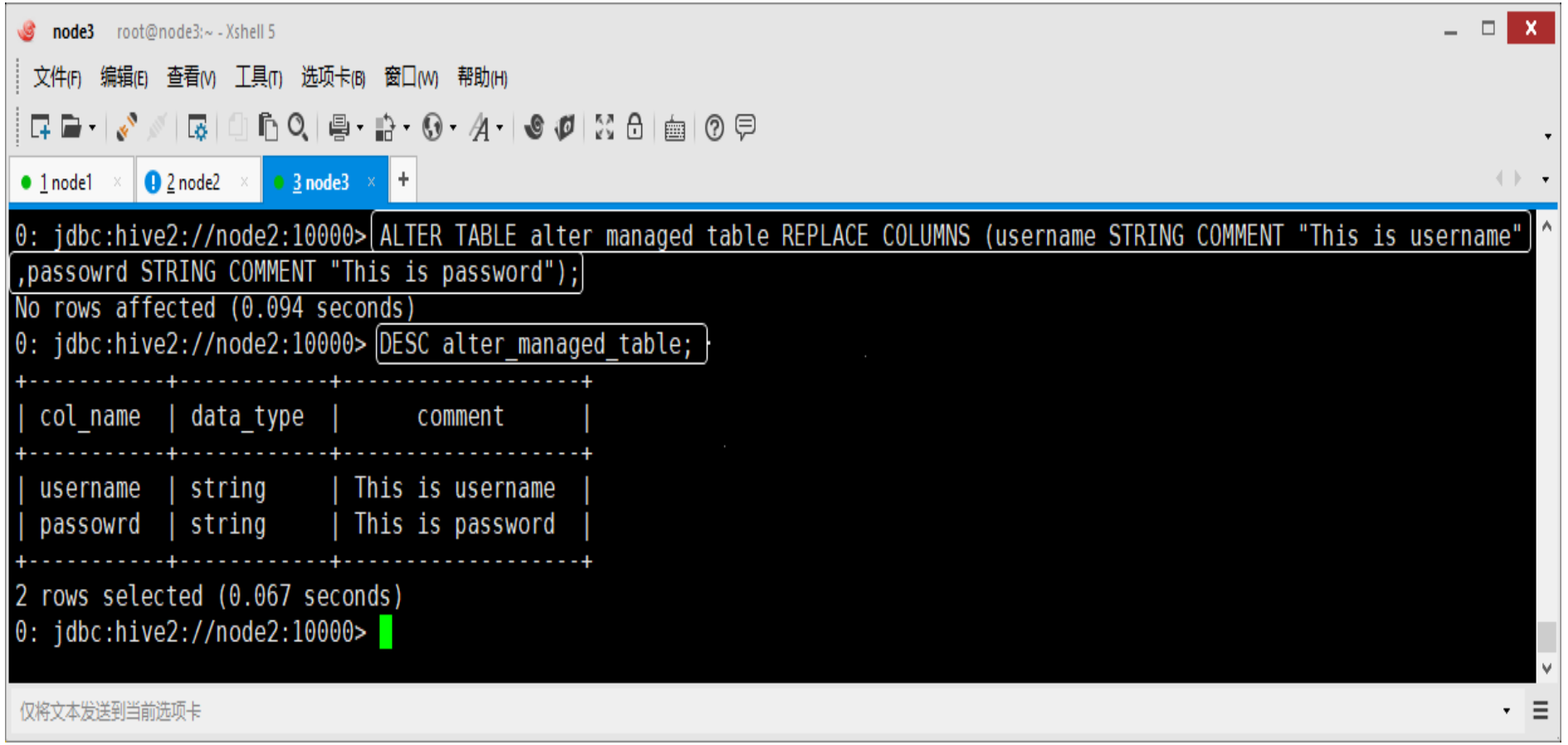


4.2.3 任务实施

（6）替换数据表alter_managed_table的列，语句如下：

ALTER TABLE alter_managed_table REPLACE COLUMNS (username STRING COMMENT "This is username",passowrd STRING COMMENT "This is password");

执行结果如图4-14所示：



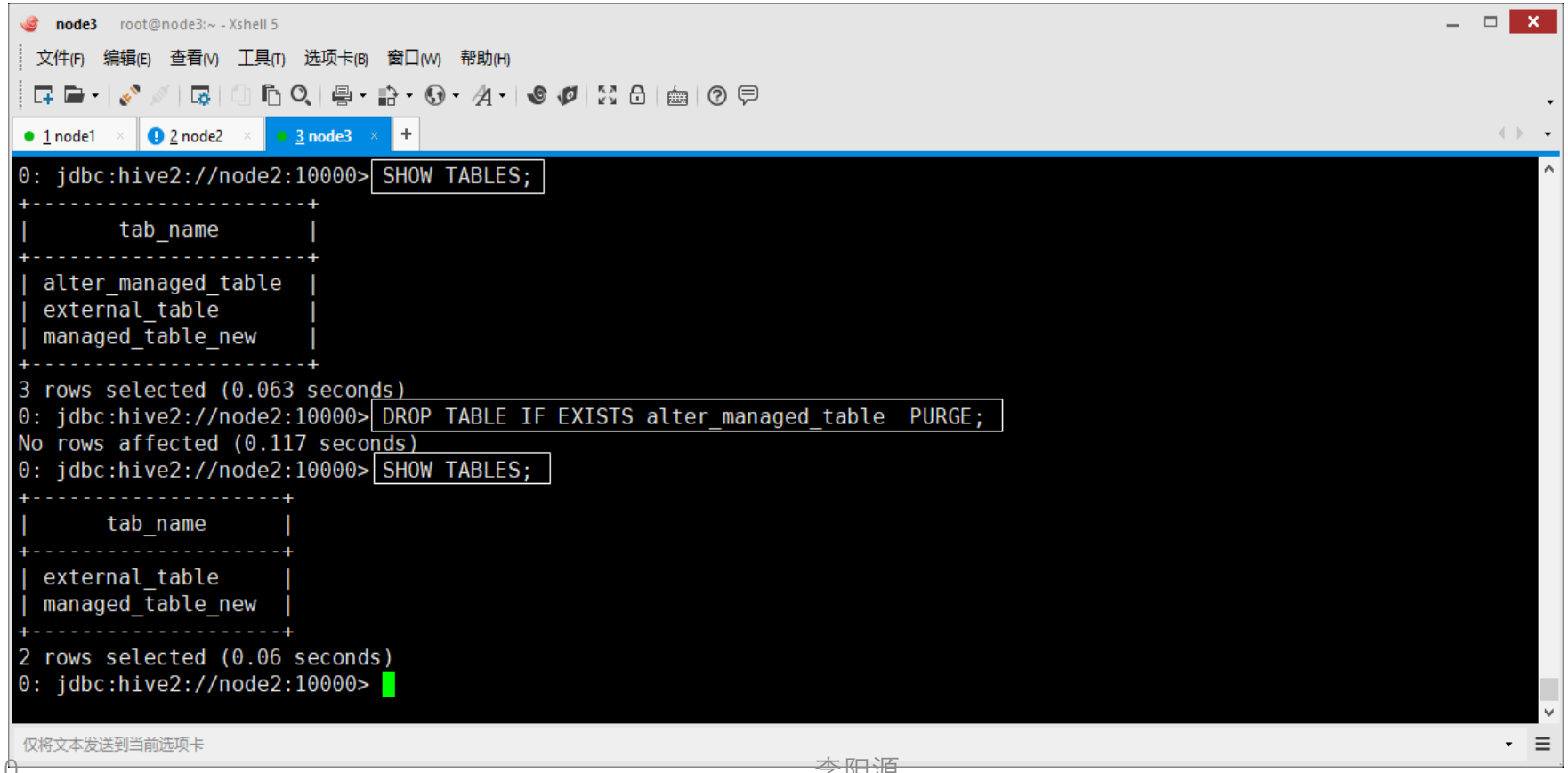
4.2.3 任务实施

4．删除数据表

删除数据库h_sample的数据表alter_managed_table，语句如下：

```
DROP TABLE IF EXISTS alter_managed_table  PURGE;
```

执行结果如图4-15所示：



5．修改数据表

修改数据表结构信息使用ALTER TABLE语句实现，该语句只是修改数据表的元数据信息，数据表中的数据不会随之发生变化。下面介绍几种修改数据表结构信息的方法。

（1）重命名数据表是指修改数据表的名称，语法格式如下：

```
ALTER TABLE table_name RENAME TO new_table_name;
```

（2）修改数据表属性的语法格式：

```
ALTER TABLE table_name SET TBLPROPERTIES(property_name = property_value, property_name = property_value, ... );
```

（3）修改数据表列是指修改数据表中列的名称、描述、数据类型或者列的位置，语法如下：

```
ALTER TABLE table_name CHANGE [COLUMN] col_old_name col_new_name column_type [COMMENT col_comment] [FIRST|AFTER column_name];
```

- ALTER TABLE：表示修改数据表结构信息的语句；
- CHANGE [COLUMN]：用于修改数据表列的操作；
- table_name：指定需要修改列的数据表名称；
- col_old_name：指定需要修改的列名；
- col_new_name：指定列名被修改后的新列名；
- column_type：指定列修改后的字段类型；

感谢您的观看



《Hive数据仓库应用教程》