

《数据仓库》练习04

说明

- 在D盘创建考生文件夹：命名为“《数仓》+练习04+学号后2位”
- 在IDEA里创建sql文件，名称为 练习04.sql
- 在IDEA里答题时，每道题前必须加上题号的备注，如‘-- 1.2’。
- 答题结束后，需要提交以下资料
 - 1.需在IDEA中导出 练习04.sql 文件为 HTML 文件，并保存在考生文件夹
 - 2.需对HDFS在web UI 中的 /user/hive/exam/exam_db4 路径进行截图，命名“HDFS.jpg”保存在考生文件夹
 - 3.需对第六题的6道题的查询结果进行截图【截图的范围包括整个窗口和运行结果】，并按题号命名保存在考生文件夹
 - 4.将考生文件夹打包压缩按监考老师的要求进行提交。

一. 启动环境

```
# 1.1
能启动VMware和FinalShell连接虚拟机即可得分

# 1.2 启动hadoop集群
start-dfs.sh
start-yarn.sh

# 1.3 启动Hive服务
nohup hive --service metastore &
nohup hive --service hiveserver2 &

# 1.4 检查进程
jps -m

# 1.5 IDEA能正确连接hive
```

二. 创建数据库

- 删除并创建一个名为 exam_db4 的数据库，并将其位置设置为 /user/hive/exam/exam_db4。

```
-- 2.1 删除 exam_db4 数据库
drop database if exists exam_db4 cascade;

-- 2.2 创建 exam_db4 数据库
create database if not exists exam_db4
    location '/user/hive/exam/exam_db4';

-- 2.3 选择 exam_db4 数据库
use exam_db4;
```

三. 创建数据表

在 `exam_db4` 数据库中创建四个表：

- 3.1 `readers` 表结构：

列	类型	描述
reader_id	int	读者ID
reader_name	string	读者姓名
email	string	邮箱地址

- 3.2 `books` 表结构：

列	类型	描述
book_id	int	图书ID
title	string	图书标题
author	string	作者
category	string	类别

- 3.3 `borrow` 表结构：

列	类型	描述
loan_id	int	借阅记录ID
book_id	int	图书ID
reader_id	int	读者ID
borrow_date	date	借出日期
return_date	date	归还日期

- 3.4 分区表 `loans_partitioned`，该表以 `return_date` 分区列进行分区，分区列为日期类型。

列	类型	描述
borrow_id	int	借阅记录ID
book_id	int	图书ID
reader_id	int	读者ID

-- 3.1 创建 readers 数据表

```
CREATE TABLE if not exists readers (
  reader_id int COMMENT '读者ID',
  reader_name string COMMENT '读者姓名',
  email string COMMENT '邮箱地址'
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

-- 3.2 创建 books 数据表

```
CREATE TABLE if not exists books (
  book_id int COMMENT '图书ID',
  title string COMMENT '图书标题',
  author string COMMENT '作者',
  category string COMMENT '类别'
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

-- 3.3 创建 borrow 数据表

```
CREATE TABLE if not exists borrow (
  borrow_id int COMMENT '借阅记录ID',
  book_id int COMMENT '图书ID',
  reader_id int COMMENT '读者ID',
  borrow_date date COMMENT '借出日期',
  return_date date COMMENT '归还日期'
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

-- 3.4 创建按 return_date 分区的borrow_partitioned分区表

```
CREATE TABLE if not exists borrow_partitioned (
  borrow_id int COMMENT '借阅记录ID',
  book_id int COMMENT '图书ID',
  reader_id int COMMENT '读者ID'
)
PARTITIONED BY (return_date date)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

四. 修改数据表

-- 4.1 在 `readers` 数据表中添加一个列名为 `phone` 的列，类型为字符串。

```
ALTER TABLE readers
    ADD COLUMNS (phone STRING COMMENT '电话号码');
```

-- 4.2 把 `books` 表中将列 `title` 改名为 `book_title`。

```
ALTER TABLE books CHANGE title book_title STRING;
```

五. 加载数据

/* 5.1 将考试数据素材 `readers.txt`、`books.txt`、`borrow.txt` 上传到master节点上的
`/opt/apps/hive/exam/`下 */

-- 5.2 把 `readers.txt` 数据加载到 `readers` 表中；

```
LOAD DATA LOCAL INPATH '/opt/apps/hive/exam/readers.txt' OVERWRITE INTO TABLE
readers;
```

-- 5.3 把 `books.txt` 数据加载 `books` 表中；

```
LOAD DATA LOCAL INPATH '/opt/apps/hive/exam/books.txt' OVERWRITE INTO TABLE
books;
```

-- 5.4.1 把 `borrow.txt` 上传到HDFS中，路径为 `/user/hive/exam/exam_db4`；

```
hdfs dfs -put /opt/apps/hive/exam/borrow.txt /user/hive/exam/exam_db4
```

-- 5.4.2 把路径 `/user/hive/exam/exam_db4` 下的 `borrow.txt` 数据加载到 `borrow` 表中。

```
LOAD DATA INPATH '/user/hive/exam/exam_db4/borrow.txt' OVERWRITE INTO TABLE
borrow;
```

-- 5.5 从表 `borrow` 执行动态分区插入数据到 `borrow_partitioned` 分区表中

-- 注意：该表是按照 `purchase_date` 进行分区。

```
SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode=nonstrict;
```

```
INSERT OVERWRITE TABLE borrow_partitioned PARTITION(return_date)
SELECT borrow_id, book_id, reader_id, return_date
FROM borrow;
```

六. 查询数据

```
/* 5.1 将数据`readers.txt`、`books.txt`、`borrow.txt`上传到master节点上的
`/opt/apps/hive/exam/`下 */

-- 6.1 统计每个类别的图书数量
SELECT category, COUNT(*) AS book_count
FROM books
GROUP BY category;

-- 6.2 查询借出次数最多的图书的书名
SELECT bo.book_title, COUNT(*) AS borrow_count
FROM borrow b
      JOIN books bo ON b.book_id = bo.book_id
GROUP BY bo.book_title
ORDER BY borrow_count DESC
LIMIT 1;

-- 6.3 查询类别为“古典文学”的图书信息
SELECT *
FROM books
WHERE category = '古典文学';

-- 6.4 将表borrow按book_id划分到 3 个节点（reducer）上，并在每个节点内按borrow_date字段降
序排序。
SET mapreduce.job.reduces=3;

SELECT borrow_id, book_id, reader_id, borrow_date
FROM borrow
      DISTRIBUTE BY book_id
      SORT BY borrow_date DESC;

-- 6.5 在分区表中查询2023年1月的借阅记录的总次数
SELECT count(*) as borrow_count
FROM borrow_partitioned
WHERE return_date >= '2023-01-01' AND return_date <= '2023-01-31';

-- 6.6 查看 readers 表的所有信息
SELECT * FROM readers;
```