

# 《数据仓库》练习02

## 说明

- 在D盘创建考生文件夹：命名为“《数仓》+练习02+学号后2位”
- 在IDEA里创建sql文件，名称为 练习02.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02.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\_db2 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\_db2 的数据库，并将其位置设置为 /user/hive/exam/exam\_db2。

```
-- 2.1 删除 exam_db2 数据库
drop database if exists exam_db2 cascade;

-- 2.2 创建 exam_db2 数据库
create database if not exists exam_db2
    location '/user/hive/exam/exam_db2';

-- 2.3 选择 exam_db2 数据库
use exam_db2;
```

## 三. 创建数据表

在 `exam_db2` 数据库中创建四个表：

- 3.1 `students` 表结构：

列	类型	描述
id	int	学生ID
name	string	学生姓名
class_id	int	班级ID

- 3.2 `class` 表结构：

列	类型	描述
id	int	班级ID
name	string	班级名称
location	string	教室位置

- 3.3 `grades` 表结构：

列	类型	描述
student_id	int	学生ID
score	double	成绩
subject	string	科目
school_year	string	学年
semester	string	学期

- 3.4 分区表 `grades_partitioned`，该表以 `year` 和 `month` 分区列进行分区，分区列为整数类型。

列	类型	描述
student_id	int	员工ID
score	double	成绩
subject	string	科目

-- 3.1 创建 students 数据表

```
CREATE TABLE if not exists students (  
  id int COMMENT '学生ID',  
  name string COMMENT '学生姓名',  
  class_id int COMMENT '班级ID'  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ',';
```

-- 3.2 创建 class 数据表

```
CREATE TABLE if not exists class (  
  id int COMMENT '班级ID',  
  name string COMMENT '班级名称',  
  location string COMMENT '教室位置'  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ',';
```

-- 3.3 创建 grades 数据表

```
CREATE TABLE if not exists grades (  
  student_id int COMMENT '学生ID',  
  score double COMMENT '成绩',  
  subject string COMMENT '科目',  
  school_year string COMMENT '学年',      -- 如 '2024'  
  semester string COMMENT '学期'         -- 如 'spring', 'fall'  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ',';
```

-- 3.4 创建按学年和学期分区的 grades\_partitioned 分区表

```
CREATE TABLE if not exists grades_partitioned (  
  student_id int COMMENT '学生ID',  
  score double COMMENT '成绩',  
  subject string COMMENT '科目'  
)  
PARTITIONED BY (school_year string, semester string)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ',';
```

## 四. 修改数据表

```
-- 4.1 在 students 数据表中添加一个列名为 `phone` 的列，类型为字符串。
ALTER TABLE students
    ADD COLUMNS (phone string COMMENT '学生电话');

-- 4.2 把 class 表中将列 `location` 改名为 `class_location`。
ALTER TABLE class
    CHANGE location class_location string;
```

## 五. 加载数据

```
/* 5.1 将考试数据素材 `students.txt`、`class.txt`、`grades.txt` 上传到master节点上的
`/opt/apps/hive/exam/` 路径下 */

-- 5.2 把 `students.txt` 数据加载到 `students` 表中；
LOAD DATA LOCAL INPATH '/opt/apps/hive/exam/students.txt' OVERWRITE INTO TABLE
students;

-- 5.3 把 `class.txt` 数据加载 `class` 表中；
LOAD DATA LOCAL INPATH '/opt/apps/hive/exam/class.txt' OVERWRITE INTO TABLE
class;

-- 5.4.1 把 `grades.txt` 从本地上传到HDFS中，路径为 `/user/hive/exam/exam_db2/`；
hdfs dfs -put /opt/apps/hive/exam/grades.txt /user/hive/exam/exam_db2/

-- 5.4.2 把路径 `/user/hive/exam/exam_db2/` 下的 `grades.txt` 数据加载到 `grades` 表
中。
LOAD DATA INPATH '/user/hive/exam/exam_db2/grades.txt' OVERWRITE INTO TABLE
grades;

-- 5.5 从表 `grades` 执行动态分区插入数据到 `grades_partitioned` 分区表中
-- 注意：该表是按照 `school_year` 和 `semester` 进行分区。
SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode=nonstrict;

INSERT OVERWRITE TABLE grades_partitioned PARTITION(school_year, semester)
SELECT student_id, score, subject, school_year, semester FROM grades;
```

## 六. 查询数据

```
-- 6.1 统计每个班的学生人数
SELECT class_id, COUNT(*) AS student_count
FROM students
GROUP BY class_id;

-- 6.2 查询成绩最低的前三个人的姓名和成绩
SELECT s.name, g.score
FROM grades g
```

```
        JOIN students s ON g.student_id = s.id
ORDER BY g.score DESC
LIMIT 3;
```

-- 6.3 查询哪些班级位于“二楼”?

```
SELECT name FROM class WHERE class_location = '三楼';
```

-- 6.4 将表grades按student\_id划分到3个节点（reducer）上，并在每个节点内按score字段降序排序。

```
SET mapreduce.job.reduces=3;
```

```
SELECT * FROM grades
      DISTRIBUTE BY student_id
      SORT BY score DESC;
```

-- 6.5 在分区表中查询2023学年Fall学期学生最高成绩信息

```
SELECT *
FROM grades_partitioned
WHERE school_year='2023' AND semester='Fall'
order by score desc
limit 1;
```