

《数据仓库》练习01

• 说明

- 在D盘创建考生文件夹：命名为“《数仓》+练习01+学号后2位”
- 在IDEA里创建sql文件，名称为 练习01.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01.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_db1 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,并按要求创建考生文件即可得分

二. 创建数据库

- 删除并创建一个名为 exam_db1 的数据库，并将其在HDFS上的位置设置为 /user/hive/exam/exam_db1。

```
-- 2.1 删除 exam_db1 数据库
drop database if exists exam_db1 cascade;

-- 2.2 创建 exam_db1 数据库
create database if not exists exam_db1
    location '/user/hive/exam/exam_db1';

-- 2.3 选择 exam_db1 数据库
use exam_db1;
```

三. 创建数据表

在 `exam_db1` 数据库中创建四个表：

- 3.1 `employee` 表结构：

列	类型	描述
id	int	员工ID
name	string	员工姓名
dept_id	int	部门ID

- 3.2 `department` 表结构：

列	类型	描述
id	int	部门ID
name	string	部门名称
location	string	部门所在地

- 3.3 `bonus` 表结构：

列	类型	描述
emp_id	int	员工ID
amount	double	奖金金额
year	string	年
month	string	月

- 3.4 分区表 `bonus_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
emp_id	int	员工ID
amount	double	奖金金额

```

-- 3.1 创建 employee 数据表
CREATE TABLE if not exists employee (
    id int comment '员工ID',
    name string comment '员工姓名',
    dept_id int comment '部门ID'
)
row format delimited
fields terminated by ',';

-- 3.2 创建 department 数据表
create table if not exists department (
    id int comment '部门ID',
    name string comment '部门名称',
    location string comment '部门所在地'
)
row format delimited
fields terminated by ',';

-- 3.3 创建 bonus 数据表
create table if not exists bonus (
    emp_id int comment '员工id',
    amount double comment '奖金金额',
    year string comment '年',
    month string comment '月'
)
row format delimited
fields terminated by ',';

-- 3.4 创建按 year和 month 分区的 bonus_partitioned 分区表
create table if not exists bonus_partitioned (
    emp_id int comment '员工id',
    amount double comment '奖金金额'
)
partitioned by (year string, month string)
row format delimited
fields terminated by ',';

```

四. 修改数据表

```

-- 4.1 在 employee 数据表中添加一个列名为 `age` 的列，类型为int。
ALTER TABLE employee
    ADD COLUMNS (age int COMMENT '年龄');

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

```

五. 加载数据

```
/* 5.1 将考试数据素材`employee.txt`、`department.txt`、`bonus.txt`上传到master节点上的
`/opt/apps/hive/exam/`下 */
-- 能够正确的接收考试素材文件，并上传到指定路径下即可得分

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

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

-- 5.4.1 把`bonus.txt`从来地上传到HDFS中，路径为 `/user/hive/exam/exam_db1`；
hdfs dfs -put /opt/apps/hive/exam/bonus.txt /user/hive/exam/exam_db1
-- 5.4.2 把路径 `/user/hive/exam/exam_db1`下的`bonus.txt`数据加载到 `bonus` 表中。
LOAD DATA INPATH '/user/hive/exam/exam_db1/bonus.txt' OVERWRITE INTO TABLE bonus;

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

INSERT OVERWRITE TABLE bonus_partitioned PARTITION(year,month)
SELECT emp_id, amount, year, month FROM bonus;
```

六. 查询数据

```
-- 6.1 统计每个部门的员工数量。COUNT(*)计算包括所有列为NULL的行在内的总行数
SELECT dept_id, COUNT(*) AS employee_count
FROM employee
GROUP BY dept_id;

-- 6.2 查询奖金最高的人的姓名和金额。
SELECT e.name, b.amount
FROM bonus b
      JOIN employee e ON b.emp_id = e.id
ORDER BY b.amount DESC
LIMIT 1;

-- 6.3 查询哪些部门位于北京？
SELECT name FROM department WHERE dept_location = '北京';

-- 6.4 将表employee按部门ID分发到3个节点上（reducer），在每个节点内按年龄升序对员工数据排序。
set mapreduce.job.reduces=3;

select * from employee
      distribute by dept_id
      sort by age asc;
```

-- 6.5 在分区表bonus_partitioned中查询2022年2月员工发放的最低奖金金额的信息

```
select *  
from bonus_partitioned  
where year="2022" and month="2"  
order by amount asc  
limit 1;
```