

2018 年秋季学期“并行与分布式数据库”课程  
实验系统测试方案说明  
(学生版)

Beta 1 – 2018.12.13

中国人民大学信息学院计算机系

# 目 录

1.	测试数据 .....	1
1.1.	全局表 .....	1
1.2.	数据划分 .....	2
1.3.	数据分配 .....	4
2.	测试用例（基本划分） .....	5
2.1.	基本测试 .....	5
2.2.	垂直划分测试 .....	5
2.3.	条件测试 .....	6
2.4.	并操作测试 .....	6
2.5.	连接测试 .....	6
2.6.	剪枝测试 .....	7
2.7.	综合测试（1） .....	7
2.8.	综合测试（2） .....	8
2.9.	综合测试（3） .....	8
3.	测试用例（混合划分） .....	8
3.1.	基本测试 .....	8
3.2.	综合测试（同 2.9） .....	8
4.	测试流程 .....	8
4.1.	初始化数据库 .....	9
4.2.	插入与删除 .....	9
4.3.	导入数据 .....	10
4.4.	查询测试 .....	10
4.5.	P2P 测试（任选） .....	11
附录	.....	12
A.	数据划分定义脚本示例 .....	12

# 1. 测试数据

本章介绍实验系统所需的测试数据。

测试数据涉及 4 个全局表，分成 14 个（只涉及水平和垂直划分，以下简称“基本划分”）或 16 个（混合划分）分片，分配在 4 个逻辑站点上。在上述两种划分中，支持基本划分是必须达到的目标。

本章内容安排如下：第 1.1 节定义 4 个全局表；第 1.2 节介绍相应的数据划分方案，包括基本划分和混合划分；第 1.3 节介绍站点配置和数据分配方案。

## 1.1. 全局表

本次测试所用的全局表共有四个，分别是 Student、Teacher、Course、Exam：

Student (id int key, name char(25), sex char(1), age int, degree int)  
Teacher (id int key, name char(25), title int)  
Course (id int key, name char(80), location char(8), credit\_hour int, teacher\_id int)  
Exam (student\_id int, course\_id int, mark int)

下面分别进行说明：

**Student** (id int key, name char(25), sex char(1), age int, degree int)

“学生”表，共 150,000 条

各列说明：

名称	含义	类型	值
id（主码）	学号	整数	1000001 至 1150000，无重复
name	姓名	字符串	-
sex	性别	字符串	'M': 男，约占 50% 'F': 女，约占 50%
age	年龄	整数	18 至 28 之间均匀分布
degree	学生类型	整数	1: 本科生，约占 50% 2: 硕士生，约占 33.3% 3: 博士生，约占 16.7%

**Teacher** (id int key, name char(25), title int)

“教师”表，共 50,000 条

各列说明：

名称	含义	类型	值
id (主码)	编号	整数	2000001 至 2050000, 无重复
name	姓名	字符串	-
title	教师类型	整数	1: 讲师, 约占 40% 2: 副教授, 约占 40% 3: 正教授, 约占 20%

**Course** (id int key, name char(80), location char(6), credit\_hour int, teacher\_id int)

“课程”表, 共 2357 条

各列说明:

名称	含义	类型	值
id (主码)	课号	整数	3000001 至 3002357, 无重复
name	课程名	字符串	-
location	上课地点	字符串	'CB-1': 教 1, 约占 33.3% 'CB-2': 教 2, 约占 33.3% 'CB-3': 教 3, 约占 33.3%
credit_hour	学分	整数	1: 约占 20% 2: 约占 35% 3: 约占 35% 4: 约占 10%
teacher_id	教师编号	整数	Teacher.id

**Exam** (student\_id int, course\_id int, mark int)

“考试成绩”表, 共 600,000 条

各列说明:

名称	含义	类型	值
student_id	学生学号	整数	Student.id
course_id	课程号	整数	Course.id
mark	考试分数	整数	大致符合均值为 80, 方差为 20 的正态分布, 并且最大值限制为 100。

## 1.2. 数据划分

**Student** (id int key, name char(25), sex char(1), age int, degree int)

水平划分:

分片名	划分条件
<b>Student.1</b>	id<1050000
<b>Student.2</b>	id>=1050000 and id<1100000
<b>Student.3</b>	id>=1100000

**Teacher** (id int key, name char(25), title int)

水平划分：

分片名	划分条件
<b>Teacher.1</b>	id<2010000 and title<>3
<b>Teacher.2</b>	id<2010000 and title=3
<b>Teacher.3</b>	id>=2010000 and title<>3
<b>Teacher.4</b>	id>=2010000 and title=3

**Course** (id int key, name char(80), location char(6), credit\_hour int, teacher\_id int)

方案一，垂直划分：

分片名	划分条件
<b>Course.1</b>	(id, name)
<b>Course.2</b>	(id, location, credit_hour, teacher_id)

方案二，混合划分：

**Step 1.** 垂直划分（同方案一）

分片名	划分条件
<b>Course.1</b>	(id, name)
<b>Course.2</b>	(id, location, credit_hour, teacher_id)

**Step 2.** 将 Course.2 进一步水平划分如下：

分片名	划分条件
<b>Course.2.1</b>	credit_hour=1
<b>Course.2.2</b>	credit_hour>1 and credit_hour<4
<b>Course.2.3</b>	credit_hour=4

**Exam** (student\_id int key, course\_id int key, mark int)

水平划分：

分片名	划分条件
<b>Exam.1</b>	student_id<1070000 and course_id<301200
<b>Exam.2</b>	student_id<1070000 and course_id>=301200

<b>Exam.3</b>	student_id>=1070000 and course_id<301200
<b>Exam.4</b>	student_id>=1070000 and course_id>=301200

## 1.3. 数据分配

### 站点配置

共 4 个站点：Site 1 ~ Site 4，部署在局域网内的 3 台机器上。

### 数据分配

基本划分：

站点名	分片名
<b>Site 1</b>	Student.1
	Teacher.1
	Course.1
	Exam.1
<b>Site 2</b>	Student.2
	Teacher.2
	Course.2
	Exam.2
<b>Site 3</b>	Student.3
	Teacher.3
	Exam.3
<b>Site 4</b>	Teacher.4
	Exam.4

混合划分：

站点名	分片名
<b>Site 1</b>	Student.1
	Teacher.1
	Course.1
	Exam.1
<b>Site 2</b>	Student.2
	Teacher.2
	Course.2.1
	Exam.2
<b>Site 3</b>	Student.3
	Teacher.3
	Course.2.2
	Exam.3
<b>Site 4</b>	Teacher.4
	Course.2.3
	Exam.4

## 2. 测试用例（基本划分）

本章给出对全局表仅进行基本的划分时的各个测试用例、参考执行结果（默认在 Site 1 上执行查询）、参考查询树。测试用例共 9 个，包括 1 个基本测试、5 个分项测试、以及 3 个综合测试。其中前 7 个在检查之前公布，最后 2 个不公布。

检查人员须仔细检查如下项目：

- 结果集是否正确（行数、列数）；
- 剪枝是否正确；
- 对投影和选择操作的优化是否充足（是否下推到了查询树中合适的位置）。

本章给出的优化后的查询树仅供参考。为简洁起见，查询树中：

- S 表示 Student 表，T 表示 Teacher 表，C 表示 Course 表，E 表示 Exam 表；
- S.1、T.2 等表示相应全局表的分片。

### 2.1. 基本测试

查询所有学生的信息

SQL 语句	结果	站点	剪枝
select * from Student	行：150000 列：5	隐藏	隐藏

检查要点：结果集、执行站点

优化后的查询树（请展示）

### 2.2. 垂直划分测试

查询所有课程的名称

SQL 语句	结果	站点	剪枝
select Course.name from Course	行：2357 列：1	隐藏	隐藏

检查要点：结果集、剪枝

优化后的查询树（请展示）

## 2.3. 条件测试

查询所有学分大于 2 且上课地点在教三的课程信息

SQL 语句	结果	站点	剪枝
select *	行: 314	隐藏	隐藏
from Course	列: 5		
where credit_hour>2 and location='CB-3'			

检查要点: 结果集

优化后的查询树 (请展示)

## 2.4. 并操作测试

查询所有考试的课程号和成绩

SQL 语句	结果	站点	剪枝
select course_id, mark	行: 600000	隐藏	隐藏
from Exam	列: 2		

检查要点: 结果集

优化后的查询树 (请展示)

## 2.5. 连接测试

查询由正教授讲授的学分大于 2 的课程 (课程名、学分、教师姓名)

SQL 语句	结果	站点	剪枝
select Course.name, Course.credit_hour, Teacher.name	行: 184	隐藏	隐藏
from Course, Teacher	列: 3		
where Course.teacher_id=Teacher.id and Course.credit_hour>2 and Teacher.title=3			

检查要点: 结果集、执行站点、选择运算的位置、剪枝

优化后的查询树 (请展示)



## 2.6. 剪枝测试

查询各个学生的所有考试成绩

SQL 语句	结果	站点	剪枝
select Student.name, Exam.mark	行: 600000	隐藏	隐藏
from Student, Exam	列: 2		
where Student.id=Exam.student_id			

**检查要点:** 结果集、执行站点、投影运算、剪枝

优化后的查询树（请展示）

## 2.7. 综合测试（1）

查询年龄大于 26 且参加了授课地点不在教三的课程的学生考试的 id、姓名、考试分数、以及相应的课程名称

SQL 语句	结果	站点	剪枝
select Student.id, Student.name, Exam.mark, Course.name	行: 75136	隐藏	隐藏
from Student, Exam, Course	列: 4		
where Student.id=Exam.student_id and			
Exam.course_id=Course.id and			
Student.age>26 and			
Course.location<>'CB-3'			

**检查要点:** 结果集、剪枝

优化后的查询树（请展示）

## 2.8. 综合测试（2）

隐藏

## 2.9. 综合测试（3）

隐藏

# 3. 测试用例（混合划分）

上一章给出的基本划分下的 9 个测试用例同样适用于混合划分。

本章再另外给出 2 个测试用例。

## 3.1. 基本测试

查询所有课程的信息

SQL 语句	结果	站点	剪枝
select * from Course	行：2357 列：5	Site 1	-
		Site 2	
		Site 3	
		Site 4	

检查要点：结果集

## 3.2. 综合测试（同 2.9）

隐藏

# 4. 测试流程

本章给出检查实验时的参考测试流程。整个测试分为 6 部分：

1. 初始化数据库
2. 单条数据的插入、删除测试
3. 导入数据
4. 查询测试
5. P2P 测试（任选项）
6. 自由展示（任选项）

其中，各组可以在自由演示部分尽量充分的展示所做系统的新特性、新功能、或是其它与众不同的设计与实现，测试人员注意仔细记录。

各小组实现的系统必须提供良好的用户界面，要求如下：

- 1) 可以清晰的显示当前的数据字典。各组必须提供如下信息的显示：
  - ◆ 站点信息，
  - ◆ 全局表的信息，
  - ◆ 统计信息，
  - ◆ 分片信息；
- 2) 可以方便的导入数据，并且显示导入结果。各组必须提供如下信息的显示：
  - ◆ 成功或失败，
  - ◆ 导入数据总量，
  - ◆ 所用时间；
- 3) 可以方便的输入 SQL 语句（插入语句、删除语句、查询语句），并且可以对用户的不同输入显示相应的结果。各组必须提供如下信息的显示：
  - ◆ 执行 SQL 语句所涉及的站点，
  - ◆ 插入、删除、查询等命令执行是否成功，
  - ◆ 执行时间
  - ◆ 查询语句的返回结果集，包括行、列计数，
  - ◆ 优化后的查询树、查询执行计划；
- 4) 对系统运行时出现的错误和用户输入的错误给出相应的提示。

## 4.1. 初始化数据库

按第 1 章所述，配置站点并建立各个表（基本划分）。

## 4.2. 插入与删除

共 10 条 SQL 语句（Insert 和 Delete 各 5 条），依次执行之后数据库应该刚好为空。

Insert:

- 1) insert into Student values (1000001, 'xiao ming', 'M', 20, 1)  
涉及站点：Site 1

- 2) insert into Teacher values (2900010, 'Santa Claus', 2)  
涉及站点: Site 3
- 3) insert into Teacher values (2000001, 'St. Nicholas', 1)  
涉及站点: Site 1
- 4) insert into Course values (3900001, 'Defence Against Dark Arts', 'CB-3', 4, 200001)  
涉及站点: Site 1、Site 2  
Site 1 上 Course.1 中插入(3900001, 'Defence Against Dark Arts')  
Site 2 上 Course.2 中插入(3900001, 'CB-3', 4, 200001)
- 5) insert into Exam values (1900001, 3900001, 96)  
涉及站点: Site 4

Delete:

- 1) delete from Teacher where title=1  
涉及站点: Site 1、Site 3  
删除 Site 1 上 Teacher.1 中的(2000001, 'St. Nicholas', 1)
- 2) delete from Teacher where id>=2900010 and title=2,  
涉及站点: Site 3  
删除 Site 3 上 Teacher.3 中的(2900010, 'Santa Claus', 2)
- 3) delete from Student where id=1000001  
涉及站点: Site 1  
删除 Site 1 上 Student.1 中的(1000001, 'xiao ming', 'M', 20, 1)
- 4) delete from Course where location='CB-3'  
涉及站点: Site 1、Site 2  
删除 Site 1 上 Course.1 中的(3900001, 'Defence Against Dark Arts')  
删除 Site 2 上 Course.2 中的(3900001, 'CB-6', 4, 200001)
- 5) delete from Exam  
涉及站点: Site 1、Site 2、Site 3、Site 4  
删除 Site 4 上 Exam.4 中的(1900001, 3900001, 96)

至此数据库已清空。

### 4.3. 导入数据

测试的数据库数据将以 TSV 文件的形式发布，通过导入程序可以把它们装载到分布式数据库中，

### 4.4. 查询测试

#### 基本划分（所有组必测）

依次执行第 2 章给出的所有测试用例。

### 混合划分（提高要求，任选）

依次执行第 3 章给出的所有测试用例。

## **4.5. P2P 测试（任选）**

在第 2 章所述的 9 个 SQL 查询中找一个不访问所有站点的（比如查询 2.2，只需要访问 Site 1），然后分别测试下面 1 和 2：

- 1) 让用户断掉 Site 2，看看系统能否正常工作、显示出结果；
- 2) 让用户断掉 Site 1，看看系统能否给出正确提示，指出系统存在的问题。

# 附录

## A. 数据划分定义脚本示例

### B1. 基本划分

```
define site S1 127.0.0.1:2001
define site S2 127.0.0.1:2002
define site S3 127.0.0.1:2003
define site S4 127.0.0.1:2004
//
create table Student (id int key, name char(25), sex char(1), age int, degree int)
create table Teacher (id int key, name char(25), title int)
create table Course (id int key, name char(80), location char(8), credit_hour int,
    teacher_id int)
create table Exam (student_id int, course_id int, mark int)
//
fragment Student horizontally into id<105000, id>=105000 and id<110000, id>=110000
fragment Teacher horizontally into id<201000 and title<>3, id<201000 and title=3,
    id>=201000 and title<>3, id>=201000 and title=3
fragment Course vertically into (id, name), (id, location, credit_hour, teacher_id)
fragment Exam horizontally into student_id<107000 and course_id<301200,
    student_id<107000 and course_id>=301200, student_id>=107000 and course_id<301200,
    student_id>=107000 and course_id>=301200
//
allocate Student.1 to S1
allocate Student.2 to S2
allocate Student.3 to S3
allocate Teacher.1 to S1
allocate Teacher.2 to S2
allocate Teacher.3 to S3
allocate Teacher.4 to S4
allocate Course.1 to S1
allocate Course.2 to S2
allocate Exam.1 to S1
allocate Exam.2 to S2
allocate Exam.3 to S3
allocate Exam.4 to S4
```

### B2. 混合划分

```
define site S1 127.0.0.1:2001
define site S2 127.0.0.1:2002
define site S3 127.0.0.1:2003
define site S4 127.0.0.1:2004
//
create table Student (id int key, name char(25), sex char(1), age int, degree int)
create table Teacher (id int key, name char(25), title int)
create table Course (id int key, name char(80), location char(8), credit_hour int,
    teacher_id int)
create table Exam (student_id int, course_id int, mark int)
//
fragment Student horizontally into id<105000, id>=105000 and id<110000, id>=110000
fragment Teacher horizontally into id<201000 and title<>3, id<201000 and title=3,
    id>=201000 and title<>3, id>=201000 and title=3
```

```

fragment Course vertically into (id, name), (id, location, credit_hour, teacher_id)
fragment Course.2 horizontally into credit_hour=1, credit_hour>1 and credit_hour<4,
    credit_hour=4
fragment Exam horizontally into student_id<107000 and course_id<301200,
    student_id<107000 and course_id>=301200, student_id>=107000 and course_id<301200,
    student_id>=107000 and course_id>=301200
//
allocate Student.1 to S1
allocate Student.2 to S2
allocate Student.3 to S3
allocate Teacher.1 to S1
allocate Teacher.2 to S2
allocate Teacher.3 to S3
allocate Teacher.4 to S4
allocate Course.1 to S1
allocate Course.2.1 to S2
allocate Course.2.2 to S3
allocate Course.2.3 to S4
allocate Exam.1 to S1
allocate Exam.2 to S2
allocate Exam.3 to S3
allocate Exam.4 to S4

```